COMit Help Index

To choose a topic press Tab to select the underlined topic you want to view and then press Enter. When using the mouse, point to the underlined topic you want to view and then click the mouse button. Use the scroll bars to view topics. To learn how to use Help press F1 or choose Using Help from the Help menu.

Commands

This index lists the commands available on the menu bar.

About COMit Command
Auto Answer Command
Batch Dial Command
Clear Command
Close Command
Colors Command
Connect Local Command
Copy Command
Copy To Command
Dial Command
Directory Command
Disconnect Port Command
Exit Command
Hang Up Command
Hold Command
Modem Command
Open Log Command
Paste Command
Paste From Command
Paths Command
Pause Command
Port Command
Preferences Command
Print Command
Printer Setup Command
Protocol Command
Receive File Command
Record Script Command
Save Configuration Command
Scripts Command
Script Editor Command
Send File Command
Session Command
Stop Command
Terminal Command
Tools Command
Windows Command
ZModem Command

Keyboard

COMit Keys

Script Reference

Script Reference

COMit Commands

To receive help with a particular command select the appropriate option.

About COMit Command
Auto Answer Command
Batch Dial Command
Clear Command
Close Command
Colors Command
Connect Local Command
Copy Command
Copy To Command
Dial Command
Directory Command
Disconnect Port Command
Exit Command
Hang Up Command
Hold Command
Modem Command
Open Log Command
Paste Command
Paste From Command
Paths Command
Pause Command
Port Command
Preferences Command
Print Command
Printer Setup Command
Protocol Command
Receive File Command
Record Script Command
Save Configuration Command
Scripts Command
Script Editor Command
Send File Command
Session Command
Stop Command
Terminal Command
Tools Command
Windows Command
ZModem Command

COMit Keys

Key(s)	Function
Alt+1 - 10	Invokes scripts 1 thru 10.
Alt+F1	Displays the help index.
Alt+F4	Exits COMit and optionally saves changes to the INI file.
Alt+A	Pauses/Continues the recording of commands to a script file.
Alt+B	Invokes the Batch Dialing dialog box.
Alt+C	Closes the open log file.
Alt+D	Connects COMit to the remote computer.
Alt+E	Begins the recording of commands to a script file.
Alt+H	Terminates the current session.
Alt+I	Displays the current directory listing dialog box.
Alt+L	Holds or resumes the logging of current session to a file.
Alt+N	Opens the communications port and allow the direct entering of modem commands.
Alt+O	Echoes the current session to a file.
Alt+P	Prints the contents of the screen buffer.
Alt+R	Receives a file from the remote computer.
Alt+S	Sends a file to the remote computer.
Alt+T	Stops the recording of commands to a script file.
Alt+U	Enter Auto Answer Mode.
Alt+X	Exits COMit and optionally saves changes to the initialization file..
Ctrl+Ins	Copies selected text onto the clipboard.
Shift+Ins	Pastes text from the clipboard.
Delete	Clears the screen buffer.
Home	Scrolls the session window to the top of the scrollback buffer.
Up Arrow	Scrolls the session window back one line.
Down Arrow	Scrolls the session window forward one line.
Left Arrow	Scrolls the session window left one character.
Right Arrow	Scrolls the session window right one character.
Page Up	Scrolls the session window back one screen.
Page Down	Scrolls the session window forward one screen.
End	Scrolls the session window to the bottom of the scrollback buffer.

Script Reference

Scripts are a set of instructions you can create for COMit to follow. Ranging from very simple to extremely complex, scripts are a short-cut which will save you considerable effort over time. These mini-programs can be created for most anything, including the automation of a logon process. Examples of several scripts are provided throughout this chapter.

Script Reference

Script File Name

A script file name can be any valid filename. The default file extension for a script file is MSL. An example of a valid script file name is "MYSCRIPT.MSL".

Contents of a Script File

A MSL script, regardless of size, consists of statements, functions and variables. A statement is a command that specifies the operations which should be performed. Functions manipulate variables, which store values used during the execution of the script. It is suggested that you end every script you create with the END statement.

Costants

A constant is a value which will be interpreted literally by COMit. Constants specify actual information, as opposed to information that is contained within a variable and identified by a name. In the following example, "Hello World" is a string constant.

```
   RPRINT "Hello World"
```

Character (string) constants can contain any alphanumeric characters and special symbols. Character constants are enclosed in double quotes.

Numeric constants identify numeric values and can be used in arithmetic operations. Numeric constants should not be delimited. If delimiters are used, the numbers will be interpreted as a string. In the following example, "9" is a numeric constant.

```
   x = 9
```

Variables

Variables are temporary storage bins for information. The variables and their contents are released when a script is terminated by a syntax error, a runtime error, or when a script reaches an END statement. The name you assign to a variable may contain numerics, letters, or the underscore character ("_"), however the first character must a a letter. The name can be up to 10 characters in length. Variables can be assigned with either a constant or another variable by using the equal sign "=". Several statements can also assign data to a variable. The following examples demonstrate different methods used to assign a value to a variable.

```
   intro$ = "Hello World"            ' Assigned by a constant
   first$ = last$                    ' Assigned by a variable
   INPUT "Enter Password:", pw$      ' Assigned by a statement
```

The variable type is determined by its name. A string variable must end with a dollar sign ($). In the following example, "Hello World" is stored in a string variable named "intro$".

```
intro$ = "Hello World"
```

Strings variables can be concatenated (combined) together using the plus sign (+), as demonstrated in the following example.

```
hello$ = "Hello "
intro$ = hello$ + "World"
```

Numeric expressions can be used in mathematical expressions or whenever a numeric value is needed by a statement. The following examples demonstrate the use of numeric variables.

```
FOR i = 1 to 10
     j = i % 2
     IF j = 1                    ' If so, print "it odd"
        RPRINT "Its odd...^M"
     ELSE                        ' otherwise, print "it even"
        RPRINT "Its even...^M"
     ENDIF
NEXT
```

The following list shows the symbol used and the calculation performed by each supported mathematical operator

Symbol	**Calulation Performed**
+	Addition.
-	Subtraction.
*	Multiplication.
/	Division.
%	Modulus. The modulus operator returns only the remainder of the division operator.
( )	Used to set the order of operations.

This index lists every script command and function currently available.

AUTOANSWER
BEEP
CLEAR
CLOSELOG
CLS
DIAL
END
FOR...NEXT
GOTO
HANGUP
HOLDLOG
IF...THEN...ELSE...ENDIF
INPUTLOWERMPRINT
ONERROR
OPENLOG
REM
RECEIVE
RESUMELOG
RINPUT
RPRINT

SEND
SET BAUDRATE
SET CARRIER
SET COMPORT
SET DATABITS
SET ECHO
SET ERROR
SET FLOWCONTROL
SET INCRLF
SET LOCKBAUDRATE
SET OUTCRLF
SET PARITY
SET PARITYCHECK
SET PHONENUMBER
SET PROTOCOL
SET STOPBITS
SET TERMINAL
SET TEXTWRAP
SET TITLE
SLEEP
STRING
UPPER
VAL
WAITFOR

About COMit Command

Displays copyright notice, the release number and the serial number of your registered copy.   To view this information choose Help, then choose About.   Select OK to remove the dialog box.

Auto Answer Command

To configure COMit to automatically answer the phone choose Session, then choose Auto Answer or select the Auto Answer button from the Tools Bar. The Lights Bar will be updated indicating that COMit will first initialize the modem and then attempt to enter auto answer mode.

If COMit is unable to properly initialize the modem, dialing will fail. After the modem initialization completes, COMit will display a dialog box indicating that COMit has entered auto answer mode.

You may cancel at any time by pressing the Escape key or selecting the Cancel button. Selecting the Background button will minimize COMit and enter background processing.

Notes:
Since proper initialization takes time, it is recommended that you set the Quick Initialization option, located in the Preference dialog box.

COMit does not automatically restore itself on the screen after background processing is complete. To enable restoration check Return From Iconic, located in the Preferences dialog box.

When COMit is not connected to a remote computer the icon is the chain. Once COMit has answered the phone and connected, the icon changes to a modem.

Related topics:
Connect Local Command
Batch Dial Command
Dial Command
Directory Command
Hang Up Command
Preferences Command
Session Command
Windows Command

Batch Dial Command

To build a batch dialing list, perform the following steps:

1. Choose Session, then choose Batch Dial.
2. Select the name of the directory records you wish to use by marking the appropriate entries. When you have finished marking the selections select the Add button.
   To remove a file you have added to the Selected list, mark the entries and select the Delete button.
3. Check the Continuous Loop, Quick Initialization and Override Redial options as desired. If you choose to temporarily override the redial value you have entered in Preferences, you must enter the new redial value.
4. Choose OK to begin the batch dialing process.

The **Continuous Loop** option indicates whether or not COMit will repeat its attempts to connect with every entry selected until they have all been connected to. If you do not select this option COMit will end the batch dialing process after it connects to any of the entries you have selected.

The **Quick Initialization** option indicates whether or not COMit should bypass the modem initialization process after the first initialization has been completed.

The **Redial** value indicates the number of times COMit will attempt to retry to connect to the remote computer. The range is from 0 to 99.

The Lights Bar will be updated indicating that COMit will first initialize the modem and then attempt to dial and connect to the remote computer.

If COMit is unable to properly initialize the modem, dialing will fail. After the modem initialization completes, COMit will display a dialog box indicating the title of the remote computer it is dialing; its phone number; the number of seconds left until COMit issues a timeout; the remaining number of times COMit will attempt to connect to the remote computer; and the last error that has occurred.

You may cancel at any time by pressing the Escape key or selecting the Cancel button. Selecting the Background button will minimize COMit and enter background processing.

Notes:
Since proper initialization takes time, it is recommended that you set the Quick Initialization option.

COMit does not automatically restore itself on the screen after background processing is complete. To enable restoration check Return From Iconic, located in the Preferences dialog box.

When COMit is not connected to a remote computer the icon is the chain. Once COMit has connected, the icon changes to a modem.

Related topics:
Auto Answer Command
Connect Local Command
Dial Command
Directory Command
Hang Up Command
Preferences Command
Session Command
Windows Command

Clear Command

To clear the contents of the scrollback buffer choose Edit, then choose Clear, or select the Clear button from the Tools Bar.

Related topics:
Copy Command
Copy To Command
Paste Command
Paste From Command
Windows Command

Close Command

To close an open log file choose Session, then choose Close, or select the Close button from the Tools Bar.   The Status Bar will be updated to indicate that COMit has closed the log file.

Related topics:
Open Log Command
Hold Command
Windows Command

Colors Command

To select the foreground and background colors used in the session window, perform the following steps:

1 Choose Settings, then choose Terminal.
2 After the terminal window is displayed, insure that the ANSI Color Attributes option is not checked.
3. Push the Colors button. This will display the current color selections.
4 Select the colors you want for the foreground and background.
5. Choose OK to accept the changes.

To save the window settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
Default colors are only used when the ANSI Color Attributes option is not checked.

Related topic:
Terminal Command

Connect Local Command

To bypass any modem initialization and open the communications port choose Session, then choose Connect Local or select the Connect Local button from the Tools Bar.   Once COMit has opened the communications port you will be able to enter modem commands and control the modem directly.

Related topics:
Auto Answer Command
Batch Dial Command
Dial Command
Directory Command
Hang Up Command
Preferences Command
Session Command
Windows Command

Copy Command

To copy the contents of a marked block to the clipboard, perform the following steps:

1. Scroll thru session buffer to bring the text you want to copy into view.
2. Using the mouse, move to the upper left hand corner of the block and press and hold the left mouse button down.
3. Drag the mouse to the lower right hand corner of the block.   The text that appears within the block will become inverted.
4. Choose Edit, then choose Copy   or select the Copy button from the Tools Bar.

Related topics:
Copy To Command
Clear Command
Paste Command
Paste From Command
Windows Command

Copy To Command

To create a file and copy the contents of a marked block to it, perform the following steps:

1. Scroll thru session buffer to bring the text you want to copy into view.
2. Using the mouse, move to the upper left hand corner of the block and press and hold the left mouse button down.
3. Drag the mouse to the lower right hand corner of the block. The text that appears within the block will become inverted.
4. Choose Edit, then choose Copy To.
5. Type or select the name of the file you want to copy to. If the target directory is not the current directory, either type the directory name in front of the filename, or select the directory from the Directories box.
6. Choose OK to begin copying to the file.

If the file already exists, COMit will display a dialog box giving you the option to cancel the operation or erase the contents of the file. Selecting OK will instruct COMit to destroy the contents of the current file.

Related topics:
Copy Command
Clear Command
Paste Command
Paste From Command
Windows Command

Dial Command

To connect COMit to a remote computer choose Session, then choose Dial or select the Dial button from the Tools Bar. The Lights Bar will be updated indicating that COMit will first initialize the modem and then attempt to dial and connect to the remote computer.

If COMit is unable to properly initialize the modem, dialing will fail. After the modem initialization completes, COMit will display a dialog box indicating the title of the remote computer it is dialing; its phone number; the number of seconds left until COMit issues a timeout; the remaining number of times COMit will attempt to connect to the remote computer; and the last error that has occurred.

You may cancel at any time by pressing the Escape key or selecting the Cancel button. Selecting the Background button will minimize COMit and enter background processing.

Notes:
Since proper initialization takes time, it is recommended that you set the Quick Initialization option, located in the Preference dialog box.

COMit does not automatically restore itself on the screen after background processing is complete. To enable restoration check Return From Iconic, located in the Preferences dialog box.

When COMit is not connected to a remote computer the icon is the chain. Once COMit has connected, the icon changes to a modem.

Related topics:
Auto Answer Command
Batch Dial Command
Connect Local Command
Directory Command
Hang Up Command
Preferences Command
Session Command
Windows Command

Directory Command

To build a listing of directory entries, perform the following steps:

1. Choose File, then choose Directory.

To Add a record to the directory listing:

2. Set the phone number and the file transfer protocol. The file transfer protocol may need to be configured as appropriate for your modem. The port and terminal settings should be changed as necessary by clicking on the current port or terminal settings to display the port or terminal selections.
3. Select the Add button. Enter the name of the entry on title field of the Add Record dialog box.
4. Choose OK to add the entry to your directory listing.

To Change an entry:

2. Select the entry to be changed.
3. Make any necessary changes to the entry, then select the Change button.

To Delete an entry:

2. Select the entry to be deleted.
3. Select the Delete button.

Notes:
The directory file used is the one which you have chosen in the Preferences dialog box.

If you make changes to an entry thru Settings and then choose Directory, you will be prompted with the message "Record has been changed. Update record?" By choosing Yes, the record will be updated in your Directory listing.

If you make changes to the title of an entry thru Settings and then choose Directory, you will be prompted with the message "(Entry) does not exist. Add record to file?" By choosing Yes, the record will be added to your Directory listing.

After selecting an entry, you may select the Dial button to immediately begin the dialing process.

Related topics:
Auto Answer Command
Batch Dial Command
Connect Local Command
Dial Command
Paths Command
Port Command
Preferences Command
Terminal Command

Disconnect Port Command

To close the communications port choose Session, then choose Disconnect Port.

Related topics:
Connect Local Command

Exit Command

To close COMit and optionally save the current settings and window position to the initialization file choose File, then choose Exit or select the Exit button from the Tools Bar.

If the Save Changes option has been set, COMit will save the window coordinates and all user settings to the initialization file.   COMit uses this file to restore itself during startup.

Note:
COMit will automatically disconnect if you are connected to a remote computer.

Hang Up Command

To disconnect from the remote computer choose Session, then choose Hang Up or select the Hang Up button from the Tools Bar.

Note:
If Carrier Detect is enabled, COMit automatically performs a hang up once you have logged off from the remote computer and the carrier is lost .   To enable carrier detect check Carrier Detect , located in the Session dialog box.

Related topics:
Auto Answer Command
Batch Dial Command
Connect Local Command
Dial Command
Exit Command
Modem Command
Windows Command

Hold Command

To temporarily stop/resume the echoing of characters to a log file choose Session, then choose Hold or select the Hold button from the Tools Bar.   The Status Bar will be updated to indicate that COMit is either logging or temporarily holding from logging characters to the file.

Related topics:
Open Log Command
Close Command
Windows Command

Modem Command

To configure the Modem settings including Modem Type, Line Type, Modem Volume, and Dialing Prefix, perform the following steps:

1. Choose Settings, then choose Modem.
2. Select the Modem Type, Line Type and Modem Volume.
3. Enter a Dialing Prefix if necessary.
4. Choose OK to accept these changes.

The **User Initialization String** is not recommended. Selection of the User Initialization field will disable the predefined modem strings. If the User Initialization string is required then it must be entered with the **AT** prefix. The following commands are preset and should not be set: The **Q, E, V** commands are preset and must not be overriden. The program depends on this options preset to internal values. The **S1** command is set to a value of 1 for AutoAnswer and 0 for any other condition. The **S7** is set based on the **Timeout** parameter located in the preferences dialog box and should be set there. The reset commands (**ATZ, AT&F**) are preset in the in the Modem string Database and should not be included in the User Init String.

The Modem Speaker commands are set based on the selection of the modem volume in the Modem Dialog Box. The **M**, and **L** commands will be set appropriately.

The **Dialing Prefix** allows you to insert additional characters in front on the phone number. Codes such as "*70" for call waiting and "9," for business PBX's should be placed here.
To save the modem settings to Comit's initialization file, select the Save Changes checkbox when you exit COMit.

Related topic:
Directory Command

Open Log Command

To open an existing or creates a new log file, perform the following steps:

1. Choose Session, then choose Open or select the Open button from the Tools Bar.
2. Type or select the name of the file you want to log to. If the target directory is not the current directory, either type the directory name in front of the log name or select the directory in the Directories box.
3. Choose OK to open the log file.

If the file already exist COMit will display a dialog box giving you the option of canceling the operation, appending the file, or erasing the contents of the file. Selecting No when asked to truncate the file will instruct COMit to preserve the contents of the selected log file. Selecting Yes will instruct COMit to destroy the contents of the current file. COMit will begin echoing every character received to the selected file. The Status Bar will be updated to indicate that COMit is logging characters to the file.

At any time during logging you may temporarily stop the logging of characters by selecting Hold. To resume logging select Hold again. To close the log file permanently, select Close.

Related topics:
Hold Command
Close Command
Paths Command
Windows Command

Paste Command

To copy the contents of the clipboard to the remote computer choose Edit, then choose Paste or select the Paste button from the Tools Bar. COMit will then begin transmitting the contents of the file to the remote computer. You may cancel the paste command at any time by pressing the Escape key.

Related topics:
Copy Command
Copy To Command
Clear Command
Paste From Command
Windows Command

Paste From Command

To open a file and copy its contents to the remote computer, perform the following steps:

1. Choose Edit, then choose Paste From.
2. Type or select the name of the file you want to paste from.   If the document is not listed in the current directory, either type the directory name in front of the document name, or select the directory in the Directories box.
3. Choose OK to begin pasting the file.

COMit will then begin transmitting the contents of the file to the remote computer.   You may cancel the paste command at any time by pressing the Escape key.

Related topics:
Copy Command
Copy To Command
Clear Command
Paste Command
Windows Command

Paths Command

To set the default file path and directory used by COMit for transferring files, directory files, script files, and log files, perform the following steps:

1. Choose Settings, then choose Paths.
2. Type the complete pathname including the drive letter, such as "C:\MYSUBDIR", for each pathname option.
3. Choose OK to accept changes.

To save the path settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
If necessary COMit will automatically attempt to create a subdirectory when verifying the path entered.

Related topics:
Exit Command
Open Log Command
Preferences Command
Receive File Command
Scripts Command
Send File Command

Pause Command

To temporarily stop/resume the recording of commands to a script file, choose Scripts, then choose Pause or select the Pause button from the Tools Bar.   The Status Bar will be updated to indicate that COMit is either recording or temporarily paused from recording commands to a script file.

Related topics:
Record Script Command
Stop Command
Windows Command

Port Command

To set the serial (COM) port, the ports' baud rate, data bits, stop bits, flow control and parity, perform the following steps:

1. Choose Settings, then choose Port.
2. Select the serial (COM) port that your modem is attached to and its appropriate communications parameters.
3. Choose OK to accept changes.

The **Baud Rate** group specifies how fast information is transferred through the port; **Data Bits** specifies the number of data bits in each packet of information; **Parity** specifies the error-checking method used; **Stop Bits** specifies the number of stop bits in each packet of information; and **Flow Control** specifies the method used to control the transmission, or flow, of data.

The **Lock Baud Rate** option indicates whether COMit should modify the connection speed between the computer and the modem. This option should be enabled when using modems that can connect at baud rates that are not supported by Windows. For example, several US Robotics modems can connect at baud rates of 14,400. To support a modem at this speed, set the baud rate to 19,200 and enable the Lock Baud Rate option.

The **Parity Check** option indicates whether COMit should translate the high order bit on the characters it receives.

The **Carrier Detect** option indicates whether COMit should automatically delete the loss of carrier to automatically perform a hang up.

Note:
It is important that these settings are correct. It is recommended that you refer to the manual of your modem to find its maximum baud rate and that you verify the settings of the remote computer. If these settings are incorrect, a **Receiving Framing** error may occur.

Related topics:
Exit Command
Directory Command
Windows Command

Preferences Command

To set the Redial Count, Timeout value, Pause value, default Directory filename, Delete Partial File option, Prompts option, Elapsed Time Clock option, Return from Iconic option, Quick Initialization option and Warning Beeps option, perform the following steps:

1. Choose Settings, then choose Preferences.
2. Enter the Redial Count, Timeout value and Pause value.
3. Enter the Directory filename or select the Directory button to display a listing of available directory files.
4. Check the Delete Partial File, Prompts, Elapsed Time Clock, Return from Iconic, Quick Initialization and Warning Beeps options.
5. Choose OK to accept changes.

The **Redial** value indicates the number of times COMit will attempt to retry to connect to the remote computer. The range is from 0 to 99.

The **Timeout** value indicates how long COMit will wait to receive a connection. The range is from 15 to 120 seconds.

The **Pause** value indicates how long COMit will wait to before redialing. The range is from 0 to 120 seconds.

The **Directory** file must reside in the default directory path. If the directory file you entered does not exist, COMit will attempt to open it as a new directory file.

The **Delete Partial File** option indicates whether or not COMit should automatically delete files that are a result of a canceled file transfer.

The **Prompts** option indicates whether or not COMit should display additional dialog boxes during file transfers.

The **Elapsed Time Clock** option indicates whether or not COMit should display the amount of time you have been connected to the remote computer.

The **Return from Iconic** option indicates whether or not COMit should restore the window to full screen when it has completed background processing, such as background dialing and background file transfers.

The **Quick Initialization** option indicates whether or not COMit should bypass the modem initialization process after the first initialization has been completed.

The **Warning Beeps** option indicates whether or not COMit should emit audible beeps to indicate connection and completed file transfers.

To save the preferences to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Related topics:
Exit Command
Paths Command
Windows Command

Print Command

To print the current screen or the entire scrollback buffer, perform the following steps:

1. Choose File, then choose Print or select the Print button from the Tools Bar.
2. Select either Current Screen or Entire Buffer.
3. Choose OK to begin printing.

To print a selected block of text, perform the following steps:

1. Scroll thru the session buffer to bring the text you want to copy into view.
2. Using the mouse, move to the upper left hand corner of the block and press and hold the left mouse button down.
3. Drag the mouse to the lower right hand corner of the block. The text that appears within the block will become inverted.
4. Choose File, then choose Print or select the Print button from the Tools Bar. The Selected Text option will be selected by default.
5. Choose OK to begin printing.

Note:
You may cancel the print command at any time by pressing the Escape key.

Related topics:
Copy Command
Printer Setup Command
Windows Command

Printer Setup Command

To select the default printer used by COMit and configure the printer's settings, perform the following steps:

1. Choose File, then choose Printer Setup.
2. Select the printer COMit will use.
3. Push the Setup button if the printer's settings need to be altered.
4. Choose OK to accept changes.

Related topic:
Print Command

Protocol Command

To change the file transfer protocol just before sending and receiving files, select the appropriate protocol, then select OK.   Selecting Cancel will terminate the file transfer.

To save this and other session settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
This dialog box will be displayed only if prompts is enabled.   To enable prompts check Prompts, located in the Preferences dialog box.

Related topics:
Paths Command
Preferences Command
Receive File Command
Send File Command
Session Command

Receive File Command

To receive a file from the remote computer, perform the following steps:

1. Choose Session, then choose Receive or select the Receive button from the Tools Bar.

If you have Prompts enabled, via the Preferences dialog box:

2. Select the protocol you want to use.

If you have selected ASCII, X modem, or Y Modem:

3. Type the name of the file you want to receive. If the target directory is not the current directory, either type the directory name in front of the log name, or select the directory in the Directories box.
4. Choose OK to begin receiving the file.

If you have selected Y modem G, Y modem Batch, or Z modem:

3. Type the complete pathname where the receiving files are to be placed.
4. Choose OK to begin receiving the file.

If COMit is unable to synchronize properly with the remote computer, the file transfer will fail. After the initialization is complete, COMit will display a dialog box indicating the filename of the receiving file, the estimated time remaining to complete the file transfer, the time that has elapsed since the file transfer was initiated, and the last error that has occurred. With Y modem G, Y modem Batch, or Z modem, the total size of the file to be received will be known and the thermometer will be active.

You may cancel receiving the file at any time by pressing the Escape key or selecting the Cancel button. Selecting the Background button will minimize COMit and enter background processing.

Related topics:
Paths Command
Preferences Command
Send File Command
Session Command
Windows Command

Record Script Command

To open a file or creates a new file and logs all commands to it, perform the following steps:

1. Choose Scripts, then choose Record or select the Record button from the Tools Bar.
2. Type or select the name of the file you want to open or create.
3. Choose OK to begin creating the script file.

If the file already exists, COMit will display a dialog box giving you the option to cancel the operation or erase the contents of the file. Selecting OK will instruct COMit to destroy the contents of the current file. COMit will begin to echo commands into the script file. The Status Bar will be updated to indicate that COMit is recording a script.

At any time during recording you may temporarily pause the recording of commands by selecting Pause. To resume recording select Pause again. To close the script file permanently, select Stop.

Related topics:
Paths Command
Pause Command
Stop Command
Windows Command

Save Configuration Command

To save the current settings and window position to the initialization file choose File, then choose Save Configuration from the Tools Bar.

Related topics:
Exit Command

Scripts Command

To set the Script buttons that appear on the Tool Bar, perform the following steps:

1. Enter the Script filename or select the Script button to display a listing of available scripts. Script files must reside in the default script directory.
2. Choose OK to accept changes.

Buttons that are not enabled will appear dimmed.

To save the scripts settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
The Tool Bar is designed to automatically wrap the buttons and expand the height of the Tool Bar to ensure that every button is visible on the screen. As a result, the session window may become reduced in size. If this is undesirable, either expand the size of the application window or reduce the number of buttons that appear on the Tool Bar.

Related topics:
Exit Command
Record Script Command
Paths Command
Windows Command

Script Editor Command

To open a Script Editor Window choose Scripts, then choose Script Editor. The Script Editor allows you to create script files or edit scripts files that were automatically created by the Record Script command.

Note:
Currently this command invokes the Windows notepad.

Related topics:
Record Script Command
Pause Command
Stop Command
Scripts Command
Paths Command

Send File Command

To send a file to the remote computer, perform the following steps:

1. Choose Session, then choose Send or select the Send button from the Tools Bar.

If you have Prompts enabled, via the Preferences dialog box:

2. Select the protocol you want to use.

If you have selected ASCII, X modem, or Y Modem:

3. Type the name of the file you want to send. If the source directory is not the current directory, either type the directory name in front of the log name, or select the directory in the Directories box.
4. Choose OK to begin sending the file.

If you have selected Y modem G, Y modem Batch, or Z modem:

3. Select the name of the file(s) you wish to send by first choosing the directory where the file(s) reside and then marking the appropriate file(s). When you have finished marking the selections from that directory, select the Add button. You may then repeat this process thru the remaining directories. To remove a file you have added to the Selected Files list, mark the file(s) and select the Delete button.
4. Choose OK to begin sending the file(s).

If COMit is unable to synchronize properly with the remote computer, the file transfer will fail. After the initialization is complete, COMit will display a dialog box indicating the filename of the sending file, the estimated time remaining to complete the file transfer, the time that has elapsed since the file transfer was initiated, and the last error that has occurred.

You may cancel sending the file at any time by pressing the Escape key or selecting the Cancel button. Selecting the Background button will minimize COMit and enter background processing.

Related topics:
Paths Command
Preferences Command
Receive File Command
Session Command
Windows Command

Session Command

To set the Session settings including the Session Title, Phone Number, default File Transfer Protocol, and Logon Script, perform the following steps:

1. Choose Settings, then choose Session.
2. Enter the Session Title and Phone Number.
3. Select the default File Transfer Protocol. If the chosen protocol requires additional settings the Configure button will be enabled. The Configure button allows you to choose additional settings for the selected protocol. For example, ZModem allows you to configure send and receive options and enable its auto download feature.
4. Enter the Logon Script filename or select the Logon Script button to display a listing of available scripts.
5. Choose OK to accept these changes.

The maximum number of characters for the **Title** is 25.

The maximum number of characters for the **Phone Number** is also 25. Special dialing codes such as "*70" for call waiting or "9," for business PBXs, should be entered in the dialing prefix option located in the Modem dialog box.

The default **File Transfer Protocol** is the method of file transfer you use most often when transferring files to this remote computer.

The **Logon Script** is the script that is automatically invoked once a connection has been made. This script file must reside in the default scripts directory.

To save the Session settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
None of the text fields allow the tilde character "~", as it is reserved.

Related topics:
Dial Command
Directory Command
Exit Command
Modem Command
Paths Command
Scripts Command
ZModem Command

Stop Command

To stop the recording of a script file choose Scripts, then choose Stop or select the Stop button from the Tools Bar.   The Status Bar will be updated to indicate that COMit has terminated the recording of a script file.

Related topics:
Record Script Command
Pause Command
Windows Command

Terminal Command

To set the Terminal settings including Terminal Type, the default Font Name and Size, the Caret Type, scrollback buffer size and dimensions, outbound Carriage Return expands to Carriage Return plus Line Feeds options, Local Echo, and ANSI Color Attributes, perform the following steps:

1. Choose Settings, then choose Terminal.
2. Select the Terminal Type, Font Name, Font Size and Caret Type.
3. Enter the scrollback Buffer size in lines, the number of Rows per page, and the Column width.
4. Select CR->CR+LF options, Local Echo and ANSI Color Attributes.
5. Choose OK to accept these changes.

The minimum number of lines in the scrollback **Buffer** is 50 and the maximum is 2000. The minimum number of **Rows** per page is 25, the maximum is 50 and the default setting is 25. The minimum number of **Columns** per page is 40, the maximum is 240 and the default setting is 80.

If **ANSI Color Attributes** is set on, COMit will translate the colors the remote computer sends thru escape sequences. Disabling ANSI Color Attributes allows you to set the colors for the session window, thereby overriding the remote color sequences. You can change the default colors by selecting the Colors button, which will display the colors dialog box. For more information on setting colors, see the related topic Colors below.

To save the terminal settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
COMit will display every font that resides in your system. However, the list of available font sizes is a standard one and may not represent the actual font sizes you have in your system. It is designed to accommodate Adobe Type Manager, Bitstreams FaceLift and True Type, which create fonts on the fly. As a result, fonts such as terminal may display the same font for various sizes.

Related topics:
Colors Command
Exit Command
Directory Command

Tools Command

To toggle the display of the various buttons that available on the Tool Bar, perform the following steps:

1. Choose Settings, then choose Tools.
2. Check the buttons that you desire to appear on the Tool Bar.
3. Choose OK to accept changes.

Buttons that are not enabled will appear dimmed.

To save the tool bar settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Note:
The Tool Bar is designed to automatically wrap the buttons and expand the height of the Tool Bar to ensure that every button is visible on the screen. As a result, the session window may become reduced in size. If this is undesirable, either expand the size of the application window or reduce the number of buttons that appear on the Tool Bar.

Related topics:
Exit Command
Windows Command

Windows Command

To toggle the display of the Status Bar, Tool Bar, and Lights Bar, perform the following steps:

1. Choose Settings, then choose Windows.
2. Check the child windows that you desire to be visible.
3. Choose OK to accept changes.

The **Tools Bar** contains a set of push buttons designed to provide mouse accelerators for various commands. The push buttons that appear are definable via the Tools command.

The **Function Key Bar** contains a set of push buttons that represent the function keys on your keyboard. These keys may be programmed via the Function Keys command.

The **Lights Bar** is a set of indicators about the active port. **IM** indicates modem initialization; **HS** indicates high speed; **AA** indicates auto answer mode; **CD** indicates carrier detected; **OH** indicates off hook; **RD** indicates receiving data; **SD** indicates sending data; **TR** indicates terminal ready; and **MR** indicates modem ready.

The **Session Window** displays all incoming characters and, optionally, all outgoing characters. This window is scrollable, retrieving up to 2,000 lines of previously displayed text. The choice of typeface, font size and fore- and background colors may be customized via the Terminal command.

To save the window display settings to COMit's initialization file, select the Save Changes checkbox when you exit COMit.

Related topics:
Exit Command
Tools Command

ZModem Command

To set the Send Options, the Receive Override Options, the CRC size, the Resume Transfer option, and the Auto Download option, perform the following steps:

1. Choose Settings, then choose Session.
2. Select ZModem as the default file transfer protocol.
3. Select the Configure button to display the ZModem dialog box.
4. Select the Send Options, the Receive Override Options, the CRC size.
5. Check the Resume Transfer option, and the Auto Download option.
6. Choose OK to accept changes.

Related topics:
Paths Command
Preferences Command
Receive File Command
Session Command
Windows Command

Script Reference

Scripts are a set of instructions you can create for COMit to follow. Ranging from very simple to extremely complex, scripts are a short-cut which will save you considerable effort over time. These mini-programs can be created for most anything, including the automation of a logon process. Examples of several scripts are provided throughout this chapter.

Script Reference

Script File Name

A script file name can be any valid filename. The default file extension for a script file is MSL. An example of a valid script file name is "MYSCRIPT.MSL".

Contents of a Script File

A MSL script, regardless of size, consists of statements, functions and variables. A statement is a command that specifies the operations which should be performed. Functions manipulate variables, which store values used during the execution of the script. It is suggested that you end every script you create with the END statement.

Costants

A constant is a value which will be interpreted literally by COMit. Constants specify actual information, as opposed to information that is contained within a variable and identified by a name. In the following example, "Hello World" is a string constant.

```
RPRINT "Hello World"
```

Character (string) constants can contain any alphanumeric characters and special symbols. Character constants are enclosed in double quotes.

Numeric constants identify numeric values and can be used in arithmetic operations. Numeric constants should not be delimited. If delimiters are used, the numbers will be interpreted as a string. In the following example, "9" is a numeric constant.

```
x = 9
```

Variables

Variables are temporary storage bins for information. The variables and their contents are released when a script is terminated by a syntax error, a runtime error, or when a script reaches an END statement. The name you assign to a variable may contain numerics, letters, or the underscore character ("_"), however the first character must a a letter. The name can be up to 10 characters in length. Variables can be assigned with either a constant or another variable by using the equal sign "=". Several statements can also assign data to a variable. The following examples demonstrate different methods used to assign a value to a variable.

```
intro$ = "Hello World"            ' Assigned by a constant
first$ = last$                    ' Assigned by a variable
INPUT "Enter Password:", pw$      ' Assigned by a statement
```

The variable type is determined by its name. A string variable must end with a dollar sign ($). In the following example, "Hello World" is stored in a string variable named "intro$".

```
intro$ = "Hello World"
```

Strings variables can be concatenated (combined) together using the plus sign (+), as demonstrated in the following example.

```
hello$ = "Hello "
intro$ = hello$ + "World"
```

Numeric expressions can be used in mathematical expressions or whenever a numeric value is needed by a statement. The following examples demonstrate the use of numeric variables.

```
FOR i = 1 to 10
     j = i % 2
     IF j = 1                     ' If so, print "it odd"
        RPRINT "Its odd...^M"
     ELSE                         ' otherwise, print "it even"
        RPRINT "Its even...^M"
     ENDIF
NEXT
```

The following list shows the symbol used and the calculation performed by each supported mathematical operator

Symbol	**Calulation Performed**
+	Addition.
-	Subtraction.
*	Multiplication.
/	Division.
%	Modulus. The modulus operator returns only the remainder of the division operator.
( )	Used to set the order of operations.

This index lists every script command and function currently available.

AUTOANSWER
BEEP
CLEAR
CLOSELOG
CLS
DIAL
END
FOR...NEXT
GOTO
HANGUP
HOLDLOG
IF...THEN...ELSE...ENDIF
INPUTLOWERMPRINT
ONERROR
OPENLOG
REM
RECEIVE
RESUMELOG
RINPUT

RPRINT
SEND
SET BAUDRATE
SET CARRIER
SET COMPORT
SET DATABITS
SET ECHO
SET ERROR
SET FLOWCONTROL
SET INCRLF
SET LOCKBAUDRATE
SET OUTCRLF
SET PARITY
SET PARITYCHECK
SET PHONENUMBER
SET PROTOCOL
SET STOPBITS
SET TERMINAL
SET TEXTWRAP
SET TITLE
SLEEP
STRING
UPPER
VAL
WAITFOR

AUTOANSWER Statement

Action: Sets the modem to auto answer mode.

Syntax: **AUTOANSWER**

Remarks: The **AUTOANSWER** statement sets the modem to auto answer mode.

Example: The following example uses **AUTOANSWER** to place the modem into auto answer mode:

```
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PROTOCOL YMODEMB
SET TITLE "Simple Host Mode"

AUTOANSWER                       ' Set AA mode
RPRINT "Welcome to my PC^M"
SLEEP 5
RPRINT "Send me the name of the file you want^M"
RINPUT "Enter filename:", file$
RPRINT "^M^M"                    ' Send 2 CRs
RPRINT "When your ready to receive the file "
RPRINT "using YMODEMB, send me 'OK'^M"
WAITFOR "OK"
RPRINT "Sending file " + file$ + " using YMODEMB^M"
SEND file$
RPRINT "Bye"
HANGUP
END
```

BEEP Statement

Action: Makes a sound through the speaker.

Syntax: **BEEP**

Remarks: The **BEEP** statement makes a sound through the speaker.

Example: The following example uses **BEEP** to indicate a successful log on to a remote computer:

```
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
CLS                                    ' Clear the screen
BEEP                             ' Log on complete
END
```

CLEAR Statement

Action: Clears the display and the contents of the scrollback buffer.

Syntax: **CLEAR**

Remarks: The **CLEAR** statement clears the display and the contents of the scrollback buffer.

Example: The following example uses **CLEAR** to remove the contents of the scrollback buffer:

```
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                   ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                    ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"               ' Send Password
CLEAR                                   ' Clears the buffer
BEEP                              ' Log on complete
END
```

CLOSELOG Statement

Action: Closes the open log file.

Syntax: **CLOSELOG**

Remarks: The **CLOSELOG** statement closes the open log file.

Example: The following example uses **CLOSELOG** to close the log file:

```
LOG$ = "LOGFILE.LOG"             ' Log File Name
OPENLOG LOG$                           ' Open a log File
HOLDLOG
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
RESUMELOG

REM Keep the script running in background until the BBS
REM ends the session and sends "Goodbye".

WAITFOR "Goodbye"
CLOSELOG
MPRINT "Log File Closed."
END
```

CLS Statement

Action: Clears the display.

Syntax: **CLS**

Remarks: The **CLS** statement clears the display and saves its contents to the scrollback buffer.

Example: The following example uses **CLS** to indicate the successful log on to a remote computer:

```
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
CLS                                    ' Clears the screen
BEEP                             ' Log on complete
END
```

DIAL Statement

Action: Starts the dialing connect process.

Syntax: **DIAL** [*phonenumber$*]

Remarks: The optional argument *phonenumber$* is a string expression that specifies the phone number COMit will use when performing a dial command. If this argument is not supplied, the **DIAL** statement will use the default phone number.

Example: The following example uses **DIAL** to connect to a remote computer:

```
ONERROR DialFailed
DIAL "9,123-4567"
SET ERROR OFF
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
MPRINT "Connection Complete!"
END

DialFailed:
MPRINT "Did not connect"
END
```

END Statement

Action: Stops a script.

Syntax: **END**

Remarks: The **END** statement stops script execution, closes all files and returns control to the user. You can place an **END** statement anywhere in the script to terminate execution.

Example: The following script uses the **END** statement to end script execution:

```
ONERROR DialFailed
DIAL "9,123-4567"
SET ERROR OFF
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
MPRINT "Connection Complete!"
END

DialFailed:
MPRINT "Did not connect"
END
```

FOR...NEXT Statement

Action: Repeats a group of instructions a specified number of times.

Syntax: **FOR** *counter=start* **TO** *end*
[*statementblock*]
[**EXIT FOR**]
[*statementblock*]
NEXT

Remarks: The FOR statement utilizes the following arguments:

Argument	Description
counter	A numeric variable used as a loop counter.
start	The initial value of the counter.
end	The final value of the loop counter.

A **FOR...NEXT** loop executes only if *start* is less than *end*. Within the **FOR...NEXT** loop, the program lines following the **FOR** statement are executed until the **NEXT** statement is encountered. Then the counter is incremented and compared by the final value, *end*.

Avoid changing the value of counter within the loop. Changing the value of counter is poor programming practice and can make the program difficult to read and debug.

The **EXIT FOR** statement provides a convenient method to exit a **FOR...NEXT** loop.

Example: The following example uses a **FOR...NEXT** loop in a script:

```
FIRST$ = "John"                   ' First Name
LAST$  = "Doe"                    ' Last Name
WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"
WAITFOR "Password-> "
INPUT "Enter Password:", PW$      ' Get Password
RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
SLEEP 10
MPRINT "Finishing Up..."

FOR I = 1 TO 5
    WAITFOR "Continue? "          ' Wait for 5 Continues
    RPRINT "^M"                         ' and send each a CR
NEXT

CLS
MPRINT "All Done..."
END
```

GOTO Statement

Action: Branches unconditionally to the specified label.

Syntax: **GOTO** *label*

Remarks: The **GOTO** statement provides a way to branch unconditionally to another label.

Example: The following example uses the **GOTO** statement:

```
FIRST$ = "John"                  ' First Name
LAST$  = "Doe"                   ' Last Name
WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"
WAITFOR "Password-> "
INPUT "Enter Password:", PW$     ' Get Password
RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
SLEEP 10
MPRINT "Finishing Up..."
X = 1

Again:
IF X = 5 THEN
    GOTO Done
ENDIF
WAITFOR "Continue? "             ' Wait for 5 Continues
MPRINT "^M"                            ' and send each a CR
X = X + 1
GOTO Again

Done:
CLS
MPRINT "All Done..."
END
```

HANGUP Statement

Action: Disconnects the modem and ends an active session.

Syntax: **HANGUP**

Remarks: The **HANGUP** statement disconnects the modem and ends an active session.

Example: The following example uses **HANGUP** to end an active session:

```
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PROTOCOL YMODEMB
SET TITLE "Simple Host Mode"

AUTOANSWER                       ' Sets AA mode
RPRINT "Welcome to my PC^M"
SLEEP 5
RPRINT "Send me the name of the file you want^M"
RINPUT "Enter filename:", file$
RPRINT "^M^M"                    ' Sends 2 CRs
RPRINT "When you are ready to receive the file "
RPRINT "using YMODEMB, send me 'OK'^M"
WAITFOR "OK"
RPRINT "Sending file " + file$ + " using YMODEMB^M"
SEND file$
RPRINT "Bye"
HANGUP
END
```

HOLDLOG Statement

Action: Pauses the logging of text to a file.

Syntax: **HOLDLOG**

Remarks: The **HOLDLOG** statement pauses the logging of text to a file.

Example: The following example uses **HOLDLOG** to pause the logging of text to a file:

```
LOG$ = "LOGFILE.LOG"                ' Log File Name
OPENLOG LOG$                              ' Open a Log File
HOLDLOG
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                     ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                      ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"                 ' Send Password
RESUMELOG

REM Keep the script running in background until the BBS
REM ends the session and sends "Goodbye".

WAITFOR "Goodbye"
CLOSELOG
MPRINT "Log File Closed."
END
```

IF...THEN...ELSE...ENDIF Statement

Action: Allows conditional execution, or branching, based on the evaluation of a Boolean (true or false) expression.

Syntax1: **IF** *booleanexpression* **THEN** *statement* [**ELSE** *statement*]

Syntax2: **IF** *booleanexpression* **THEN**
    [*statementblock*]
[**ELSE**]
    [*statementblock*]
ENDIF

Remarks: The argument boolean expression is any expression that evaluates to true (no zero) or false (zero). If the boolean expression is true the statements following the **THEN** are executed. If the boolean expression is false and an **ELSE** statement exist, the statements that follow the **ELSE** statement are executed. The program continues execution following the line after the **ENDIF** statement.

The single-line form of the statement is best used for short, straightforward tests where only one action is taken. Otherwise it is recommended that the block form of the statement be used. Programs that use the block form of the statement are usually easier to read, maintain and debug.

Example: The following example uses the **IF...THEN...ELSE...ENDIF** statement:

```
FIRST$ = "John"                  ' First Name
LAST$  = "Doe"                   ' Last Name
WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"
WAITFOR "Password-> "
INPUT "Enter Password:", PW$     ' Get Password
RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
SLEEP 10
MPRINT "Finishing Up..."
X = 1

Again:
IF X = 5 THEN
    GOTO Done
ENDIF
WAITFOR "Continue? "             ' Wait for 5 Continues
MPRINT "^M"                            ' and send each a CR
X = X + 1
GOTO Again

Done:
CLS
MPRINT "All Done..."
```

```
      END
```

INPUT Statement

Action: Displays a dialog box with a given prompt to allow the user to enter an expression from the keyboard and place the results in a specified memory variable.

Syntax: **INPUT** *stringexpression$*, *stringID$* [**USING SUPPRESS**]

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression. It is the prompt message that appears before the input area. The *stringID$* is the name of the string variable where the result of the evaluation is placed. The optional argument **SUPPRESS** allows the inputted information to be hidden, replaced only by asterisks (*).

Example: The following example uses the **INPUT** statement to accept dynamic input:

```
FIRST$ = "John"                  ' First Name
LAST$  = "Doe"                   ' Last Name
INPUT "Enter Password:", PW$ USING SUPRESS

WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"
WAITFOR "Password-> "
RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
CLS
END
```

LOWER Function

Action: Returns a string with all letters in upper case.

Syntax: **LOWER**(*stringexpression$*)

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression. Both the LOWER and UPPER functions are useful when making string comparisons.

Example: The following example uses the **LOWER** function to convert a string to lower case:

```
FIRST$ = "John"                  ' First Name
LAST$  = "Doe"                   ' Last Name
INPUT "Enter Password:", PW$ USING SUPRESS
PW$  = LOWER(PW$)

WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"
WAITFOR "Password-> "
RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
CLS
END
```

MPRINT Statement

Action: Prints text in the Status Bar.

Syntax: **MPRINT** *strnigexpression$*

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression.

Example: The following example uses the **MPRINT** statement to print text in the Status Bar:

```
LOG$ = "LOGFILE.LOG"             ' Log File Name
OPENLOG LOG$                           ' Open a log File
HOLDLOG
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
MPRINT "Log File Opened."
RESUMELOG

REM Keep the script running in background until the BBS
REM ends the session and sends "Goodbye".

WAITFOR "Goodbye"
CLOSELOG
MPRINT "Log File Closed."
END
```

ONERROR Statement

Action: Branches unconditionally to the specified label whenever a runtime error occurs.

Syntax: **ONERROR** *label*

Remarks: If the SET ERROR statement   is set to OFF, the ONERROR statement will be ignored.

Example: The following example uses the ONERROR statement to end a script due to a runtime error:

```
ONERROR DialFailed
DIAL "9,123-4567"
SET ERROR OFF
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                    ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                     ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"                ' Send Password
MPRINT "Connection Complete!"
END

DialFailed:
MPRINT "Did not connect"
END
```

OPENLOG Statement

Action: Opens a log file.

Syntax: **OPENLOG** *filename$*

Remarks: The argument *filename$* can be any string variable, string constant, or string expression. If the *filename$* is the file name without a path, the opened log command will prefix it with the path specified in the log path setting.   Any opened log files will be closed.

Example: The following example uses **OPENLOG** to open a log file:

```
   LOG$ = "LOGFILE.LOG"             ' Log File Name
   OPENLOG LOG$                           ' Open a log File
   HOLDLOG
   WAITFOR "Enter FIRST name: "
   RPRINT "John^M"                  ' Send First Name
   WAITFOR "Enter LAST name: "
   RPRINT "Doe^M"                   ' Send Last Name
   WAITFOR "Enter Password: "
   RPRINT "PASSWORD^M"              ' Send Password
   MPRINT "Log File Opened."
   RESUMELOG

   REM Keep the script running in background until the BBS
   REM ends the session and sends "Goodbye".

   WAITFOR "Goodbye"
   CLOSELOG
   MPRINT "Log File Closed."
   END
```

REM Statement

Action: Allows explanatory remarks to be inserted in a script.

Syntax1: **REM** *remark*

Syntax2: **'** *remark*

Remarks: Statements that exist after the **REM** statement are not executed. A remark is terminated by a carriage return.

Example: The following is an example of the use of the REM statement to make a script more readable:

```
   LOG$ = "LOGFILE.LOG"             ' Log File Name
   OPENLOG LOG$                           ' Open a log File
   HOLDLOG
   WAITFOR "Enter FIRST name: "
   RPRINT "John^M"                  ' Send First Name
   WAITFOR "Enter LAST name: "
   RPRINT "Doe^M"                   ' Send Last Name
   WAITFOR "Enter Password: "
   RPRINT "PASSWORD^M"              ' Send Password
   MPRINT "Log File Opened."
   RESUMELOG

   REM Keep the script running in background until the BBS
   REM ends the session and sends "Goodbye".

   WAITFOR "Goodbye"
   CLOSELOG
   MPRINT "Log File Closed."
   END
```

RECEIVE Statement

Action: Receives a file from the remote computer.

Syntax: **RECEIVE** [*filename$*] [**USING** *protocol*]

Remarks: The argument *filename$* is optional when the default file transfer protocol is one which sends the name of the file to be received. YMODEMB, YMODEMG and ZMODEM protocols are of this type. The argument *filename$* can be any string variable, string constant, or string expression. If the *filename$* is the file name without a path, the receive command will prefix it with the path specified in the receive path setting.

The optional argument *protocol* can be ASCII, XMODEM, XMODEMCRC, XMODEM1K, YMODEM, YMODEMB, YMODEMG and ZMODEM. This argument will modify the default file transfer protocol. The **RECEIVE** statement is not case sensitive.

Example: The following example uses the **RECEIVE** statement to receive a file from the remote computer:

```
SET OUTCRLF OFF                    ' Sets Outbound CR->LF
SET INCRLF OFF                     ' Sets Inbound CR->LF
SET ECHO OFF                             ' Sets Local Echo
SET TEXTWRAP ON                    ' Sets Text Wrap
SET TERMINAL ANSI
SET PROTOCOL YMODEMB
SET TITLE "Simple Host Mode"

AUTOANSWER                         ' Sets AA mode
RPRINT "Welcome to my PC^M"
SLEEP 5
RPRINT "Send me the name of the file I need.^M"
RINPUT "Enter filename:", file$
RPRINT "^M^M"                      ' Sends 2 CRs
RPRINT "When your ready to send the file "
RPRINT "using YMODEMB, send me 'OK'^M"
WAITFOR "OK"
RPRINT "Receiving file " + file$ + " using YMODEMB^M"
RECEIVE file$
RPRINT "Bye"
HANGUP
END
```

RESUMELOG Statement

Action: Resumes the logging of characters to a file.

Syntax: **RESUMELOG**

Remarks: The **RESUMELOG** statement resumes the logging of characters to a file.

Example: The following example uses the **RESUMELOG** statement to resume the logging of characters to a file:

```
LOG$ = "LOGFILE.LOG"              ' Log File Name
SET OUTCRLF OFF                   ' Sets Outbound CR->LF
SET INCRLF OFF                    ' Sets Inbound CR->LF
HOLDLOG
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                   ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                    ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"               ' Send Password
MPRINT "Log File Opened."
RESUMELOG

REM Keep the script running in background until the BBS
REM ends the session and sends "Goodbye".

WAITFOR "Goodbye"
CLOSELOG
MPRINT "Log File Closed."
END
```

RINPUT Statement

Action: Displays a given prompt to allow the remote user to enter an expression from the keyboard and place the results in a specified variable.

Syntax: **RINPUT** *stringexpression$*, *stringID$* [**USING SUPPRESS**]

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression. It is the prompt message that appears before the input area. The *stringID$* is the name of the string variable where the result of the evaluation is placed. The optional argument **SUPPRESS** allows for the inputted information to be hidden, replaced only by asterisks (*).

Example: The following example uses the **INPUT** statement to accept dynamic input:

```
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PROTOCOL ZMODEM
SET TITLE "Simple Host Mode"

AUTOANSWER                       ' Sets AA mode
RPRINT "Welcome to my PC^M"
SLEEP 5
RPRINT "Send me the name of the file you want^M"
RINPUT "Enter filename:", file$
RPRINT "^M^M"                    ' Sends 2 CRs
RPRINT "When your ready to receive the file "
RPRINT "using ZMODEM, send me 'OK'^M"
WAITFOR "OK"
RPRINT "Sending file " + file$ + " using ZMODEM^M"
SEND file$
RPRINT "Bye"
HANGUP
END
```

RPRINT Statement

Action: Sends text to the remote computer

Syntax: **RPRINT** *stringexpression$*

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression.

Example: The following example uses the **RPRINT** statement to send text to the remote computer:

```
   WAITFOR "Enter FIRST name: "
   RPRINT "John^M"                  ' Send First Name
   WAITFOR "Enter LAST name: "
   RPRINT "Doe^M"                   ' Send Last Name
   WAITFOR "Enter Password: "
   RPRINT "PASSWORD^M"              ' Send Password
   CLS                                    ' Clears the screen
   BEEP                             ' Log on complete
   END
```

SEND Statement

Action: Sends a file to the remote computer.

Syntax: **SEND** *filename$* [**USING** *protocol*]

Remarks: The argument *filename$* can be any string variable, string constant, or string expression. If the *filename$* is the file name without a path, the send command will prefix it with the path specified in the send path setting.

The optional argument *protocol* can be ASCII, XMODEM, XMODEMCRC, XMODEM1K, YMODEM, YMODEMB, YMODEMG and ZMODEM. This argument will modify the default file transfer protocol. The **SEND** statement is not case sensitive.

Example: The following example uses the **SEND** statement to send a file to the remote computer:

```
SET OUTCRLF OFF                   ' Sets Outbound CR->LF
SET INCRLF OFF                    ' Sets Inbound CR->LF
SET ECHO OFF                            ' Sets Local Echo
SET TEXTWRAP ON                   ' Sets Text Wrap
SET TERMINAL ANSI
SET PROTOCOL YMODEMB
SET TITLE "Simple Host Mode"

AUTOANSWER                        ' Sets AA mode
RPRINT "Welcome to my PC^M"
SLEEP 5
RPRINT "Send me the name of the file you want^M"
RINPUT "Enter filename:", file$
RPRINT "^M^M"                     ' Sends 2 CRs
RPRINT "When your ready to receive the file "
RPRINT "using YMODEMB, send me 'OK'^M"
WAITFOR "OK"
RPRINT "Sending file " + file$ + " using YMODEMB^M"
SEND file$
RPRINT "Bye"
HANGUP
END
```

SET BAUDRATE Statement

Action: Sets the Baud Rate option.

Syntax: **SET BAUDRATE** *baudrate*

Remarks: The argument *baudrate* can be 300, 600, 1200, 2400, 4800, 9600 or 19200.

Example: The following example uses the **SET BAUDRATE** statement to change the baud rate:

```
CLS
SET COMPORT COM1
SET BAUDRATE 2400
SET PARITY EVEN
SET DATABITS 7
SET STOPBITS 1
SET FLOWCONTROL NONE

SET CARRIERDETECT ON             ' Sets Carrier Detect
SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
SET PARITYCHECK OFF              ' Sets Parity Check

SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET CARRIERDETECT Statement

Action: Sets the Carrier Detect state.

Syntax: **SET CARRIERDETECT** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET CARRIERDETECT** statement to change the carrier detect state:

```
   CLS
   SET COMPORT COM1
   SET BAUDRATE 2400
   SET PARITY EVEN
   SET DATABITS 7
   SET STOPBITS 1
   SET FLOWCONTROL NONE

   SET CARRIERDETECT ON             ' Sets Carrier Detect
   SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
   SET PARITYCHECK OFF              ' Sets Parity Check

   SET OUTCRLF OFF                  ' Sets Outbound CR->LF
   SET INCRLF OFF                   ' Sets Inbound CR->LF
   SET ECHO OFF                           ' Sets Local Echo
   SET TEXTWRAP ON                  ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   DIAL
   END
```

SET COMPORT Statement

Action: Sets the Ports option.

Syntax: **SET COMPORT** *comport*

Remarks: The argument *comport* can be COM1, COM2, COM3 or COM4.

Example: The following example uses the **SET COMPORT** statement to change the communications port:

```
   CLS
   SET COMPORT COM1
   SET BAUDRATE 2400
   SET PARITY EVEN
   SET DATABITS 7
   SET STOPBITS 1
   SET FLOWCONTROL NONE

   SET CARRIERDETECT ON             ' Sets Carrier Detect
   SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
   SET PARITYCHECK OFF              ' Sets Parity Check

   SET OUTCRLF OFF                  ' Sets Outbound CR->LF
   SET INCRLF OFF                   ' Sets Inbound CR->LF
   SET ECHO OFF                           ' Sets Local Echo
   SET TEXTWRAP ON                  ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   DIAL
   END
```

SET DATABITS Statement

Action: Sets the Data Bits option.

Syntax: **SET DATABITS** *databits*

Remarks: The argument *databits* can be 6, 7 or 8.

Example: This following example uses the **SET DATABITS** statement to change the data bits:

```
CLS
SET COMPORT COM1
SET BAUDRATE 2400
SET PARITY EVEN
SET DATABITS 7
SET STOPBITS 1
SET FLOWCONTROL NONE

SET CARRIERDETECT ON             ' Sets Carrier Detect
SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
SET PARITYCHECK OFF              ' Sets Parity Check

SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET ECHO Statement

Action: Sets the Local Echo state.

Syntax: **SET ECHO** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET ECHO** statement to change the local echo state:

```
CLS
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET ERROR Statement

Action: Sets the Error state.

Syntax: **SET ERROR** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET ERROR** statement to change the local echo state:

```
CLS
SET ERROR OFF                    ' Sets On Error Flag
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"

FOR i = 1 to 10
    DIAL
NEXT
END
```

SET FLOWCONTROL Statement

Action: Sets the Flow Control option.

Syntax: **SET FLOWCONTROL** *flowcontrol*

Remarks: The argument *flowcontrol* can be NONE, RTS, CTS, XON. The **SET FLOWCONTROL** statement is not case sensitive.

Example: The following example uses the **SET FLOWCONTROL** statement to change the flow control:

```
   CLS
   SET COMPORT COM1
   SET BAUDRATE 2400
   SET PARITY EVEN
   SET DATABITS 7
   SET STOPBITS 1
   SET FLOWCONTROL NONE

   SET CARRIERDETECT ON             ' Sets Carrier Detect
   SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
   SET PARITYCHECK OFF              ' Sets Parity Check

   SET OUTCRLF OFF                  ' Sets Outbound CR->LF
   SET INCRLF OFF                   ' Sets Inbound CR->LF
   SET ECHO OFF                           ' Sets Local Echo
   SET TEXTWRAP ON                  ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   DIAL
   END
```

SET INCRLF Statement

Action: Sets the Inbound CR->CRLF state.

Syntax: **SET INCRLF** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET INCRLF** statement to change the inbound carriage expands to carriage return plus line feed state:

```
   CLS
   SET OUTCRLF OFF                  ' Sets Outbound CR->LF
   SET INCRLF OFF                   ' Sets Inbound CR->LF
   SET ECHO OFF                           ' Sets Local Echo
   SET TEXTWRAP ON                  ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   END
```

SET LOCKBAUDRATE Statement

Action: Sets the Lock Baud Rate state.

Syntax: **SET LOCKBAUDRATE** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET LOCKBAUDRATE** statement to change the lock baud rate state:

```
CLS
SET COMPORT COM1
SET BAUDRATE 2400
SET PARITY EVEN
SET DATABITS 7
SET STOPBITS 1
SET FLOWCONTROL NONE

SET CARRIERDETECT ON             ' Sets Carrier Detect
SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
SET PARITYCHECK OFF              ' Sets Parity Check

SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET OUTCRLF Statement

Action: Sets the Outbound CR->CRLF state.

Syntax: **SET OUTCRLF** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET OUTCRLF** statement to change the outbound carriage expands to carriage return plus line feed state:

```
CLS
SET OUTCRLF OFF                   ' Sets Outbound CR->LF
SET INCRLF OFF                    ' Sets Inbound CR->LF
SET ECHO OFF                            ' Sets Local Echo
SET TEXTWRAP ON                   ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
END
```

SET PARITY Statement

Action: Sets the Parity.

Syntax: **SET PARITY** *parity*

Remarks: The argument *parity* can be NONE, ODD, EVEN, MARK, or SPACE. The **SET PARITY** statement is not case sensitive. The parity setting must match the setting of the remote computer. Improper parity settings will result in frame errors.

Example: The following example uses the **SET PARITY** statement to change the parity:

```
   CLS
   SET COMPORT COM1
   SET BAUDRATE 2400
   SET PARITY EVEN
   SET DATABITS 7
   SET STOPBITS 1
   SET FLOWCONTROL NONE

   SET CARRIERDETECT ON              ' Sets Carrier Detect
   SET LOCKBAUDRATE OFF              ' Sets Lock Baud Rate
   SET PARITYCHECK OFF               ' Sets Parity Check

   SET OUTCRLF OFF                   ' Sets Outbound CR->LF
   SET INCRLF OFF                    ' Sets Inbound CR->LF
   SET ECHO OFF                            ' Sets Local Echo
   SET TEXTWRAP ON                   ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   DIAL
   END
```

SET PARITYCHECK Statement

Action: Sets the Parity Check state.

Syntax: **SET PARITYCHECK** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET PARITYCHECK** statement to change the parity check state:

```
CLS
SET COMPORT COM1
SET BAUDRATE 2400
SET PARITY EVEN
SET DATABITS 7
SET STOPBITS 1
SET FLOWCONTROL NONE

SET CARRIERDETECT ON             ' Sets Carrier Detect
SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
SET PARITYCHECK OFF              ' Sets Parity Check

SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET PHONENUMBER Statement

Action: Sets the Phone Number to the specified string.

Syntax: **SET PHONENUMBER** *phonenumber$*

Remarks: The argument *phonenumber$* can be any string variable, string constant, or string expression that passes validation. Valid strings cannot contain the alpha characters, the tilde "~" character or exceed 25 characters in length.

Example: The following example uses the **SET PHONENUMBER** statement to change the phone number used when dialing:

```
CLS
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET PROTOCOL Statement

Action: Sets the default File Transfer Protocol.

Syntax: **SET PROTOCOL** *protocol*

Remarks: The argument *protocol* can be ASCII, XMODEM, XMODEMCRC, XMODEM1K, YMODEM, YMODEMB, YMODEMG and ZMODEM. The **SET PROTCOL** statement is not case sensitive.

Example: The following example uses the **SET PROTCOL** statement to change the default file transfer protocol:

```
CLS
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET STOPBITS Statement

Action: Sets the Stop Bits option.

Syntax: **SET DATABITS** *stopbits*

Remarks: The argument *stopbits* can be 1 or 2.

Example: The following example uses the **SET STOPBITS** statement to change the stop bits:

```
   CLS
   SET COMPORT COM1
   SET BAUDRATE 2400
   SET PARITY EVEN
   SET DATABITS 7
   SET STOPBITS 1
   SET FLOWCONTROL NONE

   SET CARRIERDETECT ON             ' Sets Carrier Detect
   SET LOCKBAUDRATE OFF             ' Sets Lock Baud Rate
   SET PARITYCHECK OFF              ' Sets Parity Check

   SET OUTCRLF OFF                  ' Sets Outbound CR->LF
   SET INCRLF OFF                   ' Sets Inbound CR->LF
   SET ECHO OFF                           ' Sets Local Echo
   SET TEXTWRAP ON                  ' Sets Text Wrap
   SET TERMINAL ANSI
   SET PHONENUMBER "9,123-4567"
   SET PROTOCOL ZMODEM
   SET TITLE "Bill's Computer"
   DIAL
   END
```

SET TERMINAL Statement

Action: Sets the Terminal Type.

Syntax: **SET TERMINAL** *terminaltype*

Remarks: The argument *terminaltype* can be ANSI, TTY and VT100. The **SET TERMINAL** statement is not case sensitive.

Example: The following example uses the **SET TERMINAL** statement to change the terminal type:

```
CLS
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET TEXTWRAP Statement

Action: Sets the Text Wrap state.

Syntax: **SET TEXTWRAP** *state*

Remarks: The argument *state* can be either ON or OFF.

Example: The following example uses the **SET TEXTWRAP** statement to change the text wrap state:

```
CLS
SET OUTCRLF OFF                  ' Sets Outbound CR->LF
SET INCRLF OFF                   ' Sets Inbound CR->LF
SET ECHO OFF                           ' Sets Local Echo
SET TEXTWRAP ON                  ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL
END
```

SET TITLE Statement

Action: Sets the optional Title to the specified string.

Syntax: **SET TITLE** *title$*

Remarks: The argument *title$* can be any string variable, string constant, or string expression that passes validation.   Valid strings cannot contain the tilde "~" character or exceed 25 characters in length.

Example: The following example uses the **SET TITLE** statement to change the caption bar:

```
CLS
SET OUTCRLF OFF                 ' Sets Outbound CR->LF
SET INCRLF OFF                  ' Sets Inbound CR->LF
SET ECHO OFF                          ' Sets Local Echo
SET TEXTWRAP ON                 ' Sets Text Wrap
SET TERMINAL ANSI
SET PHONENUMBER "9,123-4567"
SET PROTOCOL ZMODEM
SET TITLE "Bill's Computer"
DIAL                            ' Connects
END
```

SLEEP Statement

Action: Suspends the execution of the running script for a specified period of time.

Syntax: **SLEEP** *seconds*

Remarks: The argument *seconds* determines the number of seconds to suspend the script.

Example: The following example uses the **SLEEP** statement to suspend execution for ten seconds.

```
CLS
RPRINT "Taking a ten second nap..."
SLEEP 10                         ' Pause 10 seconds.
RPRINT "Awake!"
END
```

STRING Function

Action: Returns the string equivalent to a numeric value.

Syntax: **STRING**(*numericexpression*)

Remarks: The argument *numericexpression* can be any numeric variable, numeric constant, or numeric expression.

Example: The following example uses the **STRING** function to convert a numeric value to a string:

```
' Prints the value of i

FOR i = 1 TO 10
    output$ = STRING(i)
    RPRINT output$ + "^M"
NEXT

END
```

UPPER Function

Action: Returns a string with all letters in upper case.

Syntax: **UPPER**(*stringexpression$*)

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression. Both the LOWER and UPPER functions are useful when making string comparisons.

Example: The following example uses the **UPPER** function to convert a string to upper case:

```
FIRST$ = "John"                   ' First Name
LAST$  = "Doe"                    ' Last Name
INPUT "Enter Password:", PW$ USING SUPRESS

WAITFOR "First Name-> "
RPRINT FIRST$
RPRINT "^M"
WAITFOR "Last Name-> "
RPRINT LAST$
RPRINT "^M"

WAITFOR "Password-> "
INPUT "Enter Password:", PW$ USING SUPRESS
PW$  = UPPER(PW$)

RPRINT PW$
RPRINT "^M"
MPRINT "Log-on Complete!"
CLS
END
```

VAL Function

Action: Returns the numeric value for a string expression.

Syntax: **VAL**(*stringexpression$*)

Remarks: The argument *stringexpression$* can be any string variable, string constant, or string expression.

Example: The following example uses the **VAL** function to convert a string to a numeric value :

```
INPUT "Enter a number:", value$

val = VAL(value$)

val = val + 100

RPRINT STRING(VAL)

END
```

WAITFOR Statement

Action: Suspends the execution of the running script until a specified string is received from the remote computer.

Syntax: **WAITFOR** *stringexpreession$*

Remarks: The argument *stringexpreession$* can be any string variable, string constant, or string expression.

Example: The following example is a log on script that uses the **WAITFOR** statement:

```
WAITFOR "Enter FIRST name: "
RPRINT "John^M"                  ' Send First Name
WAITFOR "Enter LAST name: "
RPRINT "Doe^M"                   ' Send Last Name
WAITFOR "Enter Password: "
RPRINT "PASSWORD^M"              ' Send Password
CLS                                    ' Clear the screen
MPRINT "Log-on Complete!"
BEEP                             ' Log on complete
END
```