Host Mode

When you select the **Host Mode** choice under the **Special** menu, your Macintosh is transformed into a host system that is capable of sending and receiving files upon command by remote callers. Your keyboard and menu bar are locked out while the Host Mode is active. To return to normal operation, click your mouse button. Red Ryder will graphically invert the Status Bar area (by displaying it as white upon black) when it realizes you've pressed your mouse button.

Before entering the Host Mode, change the Duplex parameter on the **General Status Bar** to ECHO. This causes Red Ryder to echo back all characters received from the remote caller so that they appear properly on the caller's screen. If your remote caller complains that the lines echoed back from Red Ryder write over the top of each other, you should select the **Send linefeed after carriage return** option in the **Terminal Preferences** menu command before entering Host Mode. It doesn't really hurt to go ahead and turn on the **Send linefeed after carriage return** choice before your remote callers complain - the worst that can happen if it's turned on is that they will get double-spaced text.

Host Mode does not answer the phone when it rings, your modem must be an auto-answer type and be configured properly to do this. Nor does Red Ryder have a Host Mode command to disconnect the phone when the caller is finished. This must be done manually by the remote caller. While your modem is busy waiting for calls and connecting, Host Mode is simply waiting for certain commands to come across the line. Anything but these commands are ignored (no error messages are given). This makes Host Mode a "low profile" host that is difficult for phone vandals to mess with. With the addition of a password protection routine, as discussed in the Procedure Examples chapter, Host Mode is impervious to illegal entry.

Let's say you've left Red Ryder in Host Mode and have left for work, where they still use IBM-PC's (perhaps you work for IBM). You call home and connect to your Mac's modem. The first thing to do is type 3 or 4 carriage returns to clear the command line Red is scanning.

Now you're ready to enter in a command, or disconnect the call when you're finished. After any command you enter is completed, type in a carriage return or two before entering the next command. It's also necessary to type a carriage return at the end of each command, and you can backspace to correct errors in typing.

Host Mode does not match communications parameters with the caller. If your parameters are set to 1200-N-8-1 before entering Host Mode, the remote caller's parameters must be set exactly the same or no commands will be recognized.

Most of the below commands have an additional component called **filename**. The correct format for this arguement is:

volume name:file name

The colon (":") separates the volume name and file name. In other words if a host mode command was in the format:

MELTDOWN filename

and you wanted to "MELTDOWN" (whatever that means) a file named "REACTOR" on a disk volume named "NUCLEAR", you would type in:

MELTDOWN NUCLEAR:REACTOR

You don't need the volume name portion if the file is on the same disk as Red Ryder, but I highly recommend you use it to avoid problems. The commands used by Host Mode are fairly simple and straightforward:

SENDA filename - causes Red Ryder to send to the remote caller **filename** using ASCII protocol.

SENDX filename - causes Red Ryder to send to the remote caller **filename** using XMODEM protocol.

SENDK filename - causes Red Ryder to send to the remote caller **filename** using Kermit protocol.

RECA filename - causes Red Ryder to receive from the remote caller **filename** using ASCII protocol.

RECX filename - causes Red Ryder to receive from the remote caller **filename** using XMODEM protocol.

RECK filename - causes Red Ryder to receive from the remote caller **filename** using Kermit protocol.

CLOSE - manually stops a file receive started with **RECA**

DO filename - exits the Host Mode and immediately executes the Procedure file **filename**. Note that you can get back into Host Mode by using the Procedure command **HOST**.

The **SENDA** command does not warn you that it is about to send the file, it immediately does so upon receipt of that command. The **SENDX** and **SENDK** commands both send out "Ready to send..." messages so you know the command was accepted and you can begin the transfer.

The **RECA**, **RECX**, and **RECK** commands all send out a "Ready to receive..." message when the command has been accepted and Host Mode is ready for you to begin. The ***CLOSE*** command sends a "Closed..." message when the file has been closed and the **RECA** ended.

Do not use the **RECA** command to send files to Red Ryder that contain a Host Mode command. All kinds of strange things can happen. Instead, use either Kermit or XMODEM to send these files to Red Ryder in Host Mode.

The Host Mode has a special feature that allows you to use it as a electronic mail center. If a **RECA**, **RECX**, or **RECK** command is specified with a **filename** that already exists, the new file received is **appended to the end** of the old file. This is so that Host Mode can be used as a message center, and new messages won't overwrite old ones as they are uploaded. For this reason, write-protect any disks that contain files you don't want destroyed intentionally or by accident.

Here's an example: Bill Smith calls in and commands Host Mode to **RECA MAIL:SCOTT WATSON**. Host Mode responds "Ready to receive..." and Bill types in his message to me. When he's finished, he types in ***CLOSE*** on a new line and Host Mode responds "Closed...". He logs off and Ed Jones calls in next. Ed follows the exact same procedure. When I call in later, I command Host Mode to **SENDA MAIL:SCOTT WATSON** and my messages are sent one after another. When I get back to my machine, I delete the file **SCOTT WATSON** on the volume **MAIL** since I've already read the messages and don't need to keep them around.

There is not a command to list the files available for sending, but you can type in a short file listing those you want remote callers to know about. Just name it something simple like File List, put it on the same disk as Red Ryder, and tell your callers to type the command **SENDA File List** after they log on.

You may wish to use ECHO duplex in the Host Mode. This echos back every character Red Ryder receives so that it is displayed on the callers screen properly if they call using FULL duplex. You must make sure your modem is set up so as not to send any response codes (like "CONNECT", or "NO CARRIER") before entering Host Mode or Red Ryder can get into a fight to the death echoing characters back to the modem (which it in turn echos back to Red Ryder, which Red Ryder in turn...you get the point).

Procedure Files

A Procedure file is simply a set of instructions you want Red Ryder to follow. Sound like a computer program? It is exactly that, and Red Ryder is equipped with a powerful, yet easy to learn and use, computer language of its own with literally dozens of commands. Red Ryder's Procedure language was originally intended to be used for automatic dialing and logon for remote services. It quickly became evident that the possibilities are literally endless, and the Procedure language has grown to several dozen powerful and easy to use commands.

A Procedure file can be as simple as a two-liner that sets the communications parameters and dials a phone number, or a wild bugger that automates an entire session while you sleep. It's up to you how complex you want to get. Start simple, experiment, learn, and have fun!

Just like learning any other computer language, the best way to learn Procedure commands is to browse over the following command descriptions (not trying to memorize them), and then look at some examples, referring back to the command descriptions for a more complete understanding. Learn a few commands at a time, biting off more as you feel comfortable. The examples start easy, and work their way to complex. Although you'll be prepared to write your first procedure after looking at the first example, don't figure on being a master of this language in an hour - it may take several. The beauty of it is that you can make your procedures as simple or complex as suits your needs and ingenuity, so don't overdo it by moving too fast.

You can execute a Procedure file in a number of ways:

• By double-clicking the mouse button (or single-clicking and choosing **Open** from the Finder's **File** menu. If you start up Red Ryder this way, the Procedure file will be immediately executed.

• By naming a Procedure file with the special name "RRJ$" on the same disk as Red Ryder - Red will execute this file immediately upon starting up. If you create one of these, don't start up Red Ryder by double-clicking a Procedure file icon as only the "RRJ$" Procedure will be executed. The RRJ$ file can be used to bypass the title screen, or to set up a default configuration you like instead of using one left over from the last session.

• By using the Host Mode command **DO**. There is also a Procedure file command **DO** for executing one procedure by another.

• Through a macro key. The first character of the macro key should be a backslash ("\") followed by the Procedure filename (see below for the correct format of a filename). Don't get the backslash (which is directly above the RETURN key on your keyboard) confused with the slash ("/").

During the execution of a Procedure, the menu bar and Status Bar are locked out. **Pressing your mouse button will abort the Procedure in progress.** The status bar area will graphically invert (displaying white upon black) when Red realizes you've pressed the mouse button.

Your keyboard, however, is not locked out and you may type freely at any time. This capability was added to Red Ryder so that you don't have to put secret information in a Procedure that you wouldn't want others to view or have access to. During a demonstration of Red Ryder, you might want to have a **PROMPT** command for your Name or user ID question (followed by a **PAUSE** and **TYPE** command), but you don't want to **TYPE** your password in front of others. Just **PROMPT** for something that comes after the Password question, and type it in by hand.

Procedure commands are shown in the Status Bar area as they are executed (unless you use a **QUIET** command to suppress this).

A Procedure file is created with your favorite editor/word processor. If you use MacWrite, make sure you end each line with a carriage return and save the file using the "Text Only" option. I like to use the MockWrite desk accessory for creating short Procedure files while running Red Ryder, and others have recommended the Quick & Dirty Editor from Dreams Of The Phoenix. The "Edit" program from the Macintosh Development System (MDS) is also a good tool for creating Procedure files. Once you have saved your Procedure file, select the **Change TEXT To Procedure** choice under the **Service** menu. This will give your Procedure a distinctive icon (like that found next to the **Initiate Procedure** choice, to keep it from getting mixed in with other TEXT files). If you need to change the Procedure once it has been given the distinctive icon, use the **Change Procedure To TEXT** choice under the **Service** menu to change it back to a TEXT file that can be loaded into your editor/word processor. You can't use either of the **Change** menu choices during a file send or receive.

Changing a Procedure to TEXT and back several times can be a drag during debugging. A neat trick to save steps is to put the Procedure name in a macro key. A Procedure file in a macro key does not have to have the distinctive Procedure icon (it can be any kind of file, including TEXT). Once I have the Procedure working the way I like, I give it the distinctive icon so that it can be chosen from the **Initiate Procedure** command.

All Procedure command lines must be 80 characters long or less. Having a line longer than this can cause serious and unpredictable errors.

It's absolutely essential that you specify the **filename** type arguements correctly. Remember that **filename** consists of the volume's name, a colon, and the file's name. The file "My File" on the volume "My Disk" would be specified "My Disk:My File". Specifying the wrong **filename** can have various effects, depending on the command. The **SEND** commands for example, will give you a "File Not Found" error message. The **MACRO** command will sound two short beeps. The **RUN** command will probably cause a complete system crash! You are warned to be careful.

Enter the Procedure commands **exactly** as they are shown below. Spelling errors, leading, trailing, or unnecessary blanks are all guaranteed to cause problems. Spelling errors usually result in an "Illegal Procedure Command" message box. A **CLOSE** command encountered when an ASCII file receive is not in progress will also yield the "Illegal Procedure Command" message.

If the last command in your Procedure file does not seem to execute, you probably forgot to put a carriage return at the end of it. All Procedure command lines must end with a carriage return. In other words, don't let the automatic wrap-around feature of some word processors and text editors take you to the next line - always press your RETURN key at the end of each Procedure command line as you type them in.

Always put a **PAUSE** command after a **PROMPT** command. Many remote systems need a moment or two to settle down before you deluge it with characters, such as through a **TYPE** command.

If you just can't make a **PROMPT** command work, you're probably doing one of three things:

1) Not prompting for the correct string ("YOU HAVE MAIL", "You Have Mail" and "you have mail" are three <u>different</u> strings because of the upper and lowercase letters).

2) Inserted a leading, trailing, or unneeded space in the string.
3) Never receive the string you are prompting for.

In 99% of the cases where somebody says "my Procedure won't work", the problem has been traced to a faulty PROMPT command. Please, please check these very carefully before calling for help.

OK, let's look at the individual Procedure commands. Remember: first time through don't get overwhelmed, just browse. Then, look over the examples, and come back to the descriptions to see how they're being used, or when you have a need for one in a Procedure you're writing. The commands are presented below arranged in 6 groups according to their function.

Unless a Procedure is ended by a command (such as **QUIT**, **RUN**, or **DO**), it will stop normally after executing the physically last Procedure command line or when an error occurs.

Execution/Control

(
Function: The left parenthesis is used to begin a label or comment. A label is used with a **JUMPTO** command. A comment contains any text you like - there is functionally no difference between a label and a comment as neither are directly executed. The right parenthesis is not needed to end the label or comment, but does make it look prettier. To make the Procedure more readable, I usually make my labels one or two words in uppercase letters to tip me off that this is a **JUMPTO** location.
Example:
(This is a comment or label)
(COLLECT MAIL)

JUMPTO label
Function: immediately branches to the command following **label**. All of the characters in **label** are significant, including the "(" character. Upper and lowercase letters must match!
Example:
JUMPTO (HERE)
BELL
(HERE)
BELL

Only one **BELL** instruction would be executed.

END
Function: Stops execution of the Procedure. Useful when used with conditional branching (see the **ALERT** commands below) to stop execution at a place other than the end of a Procedure file
Example:
ALERT1 you have mail/JUMPTO (HERE)
PROMPT command?
BELL
END
(HERE)
RECA MAIL DISK:MAIL FILE

HOST
Function: Immediately stops execution of the Procedure file and enters Host Mode.
Example:
TYPE You are now entering Host Mode^M
HOST

DO filename
Function: Immediately stops execution of the current Procedure file and begins execution of the Procedure file **filename**.
Example
TYPE Now branching to mail routine...^M
DO MYDISK:MAIL PROCEDURE

RUN filename
Function: Immediately exits Red Ryder and begins execution of the Macintosh application **filename**. If this file doesn't exist, you will probably get a system crash - so be careful.
Example:
RUN MYDISK:MacWrite

WAIT hh:mm:ss
Function: Waits until the time specified in **hh:mm:ss** before executing the next procedure command. **hh** is hours, **mm** is minutes, and **ss** is seconds. All must have leading zeros if they are less than 10, and are in 24-hour military format (i.e. 10:32:00 p.m. would be 22:32:00).
Example:
WAIT 01:30:00
TYPE It's now 1:30 in the morning!^M
WAIT 15:30:00
TYPE It's now 3:30 in the afternoon!^M

QUIT
Function: Immediately exits Red Ryder and returns to the Finder.
Example:
(Returning to Finder...)
QUIT

QUIET

Function: Causes no further Procedure commands to appear in the Status Bar area as they execute. Use this to keep passwords or other information secret, or to prevent short procedures from distracting you.
Example:
TYPE No further commands will be displayed^M
QUIET

LOUD
Function: Used after a **QUIET** command to again display Procedure commands in the Status Bar area.
Example:
QUIET
TYPE MYPASSWORD^M
LOUD
TYPE MYNAME^M

PAUSE
Function: Delays for two seconds before executing next Procedure command.
Example:
PROMPT CONNECT
PAUSE
TYPE MYNAME^M
TYPE MYPASSWORD^M

BELL
Function: Causes the Macintosh to emit a short beep sound. Useful to alert you during various stages of a Procedure execution.
Example:
TYPE One ringy-dingy!^M
BELL

PANIC0
PANIC1 filename
PANIC2 filename
PANIC3 filename
PANIC4 filename
PANIC5 filename
PANIC6 filename
PANIC7 filename
PANIC8 filename
PANIC9 filename
Function: These 10 commands are used in conjuntion with the **PROMPT** command. If the string you have specified in the last **PROMPT** command does not come over the line in the specified number of minutes, the Procedure file **filename** is immediately executed. That file typically will contain instructions for disconnecting the modem. The number of minutes to wait is placed directly after the command **PANIC** (no spaces in there!) and is a number from 1 to 9. There is a single space between the number of minutes and **filename**. To disable the **PANIC** timer, use a **PANIC0** command (that's a zero). Once the **PANIC** timer is started, it continues to run until a **PANIC0** is executed or a new **PANIC** time is specified.
Example:
(Wait 5 minutes for the string "FINISHED")
PANIC5 MY DISK:Hang Up Phone
(Or hang up the phone if I don't get it)
PROMPT FINISHED
(I got it! Now disable the PANIC timer)
PANIC0
(That's a zero after the "C")

DELETE filename
Function: This command deletes the file **filename**. No verification if you really want to do that is supplied as is with the **Delete A File** menu command, so be cautious about how you use this Procedure command.
Example:
DELETE DataDisk:OldNews

Communications Parameters/System Configuration

RESET
Function: This resets the elapsed timer to 00:00:00.
Example:
PROMPT CONNECT
RESET

COMM baud-parity-databits-stopbits-duplex
Function: This sets up the communications parameters. The hyphens between the parameters are necessary.
baud - 300, 450, 1200, 2400, 4800, or 9600
parity - N, E, O, K, or S
databits - 7 or 8
stopbits - 1 or 2
duplex - FULL, HALF, or ECHO
Example:
COMM 1200-N-8-1-FULL
COMM 300-E-7-2-HALF

DISPLAY 40
DISPLAY 80
DISPLAY 132
Function: This commands select either the 40 X 12, 80 X 24, or 132 X 24 display modes. The mode selected must be compatible with the emulation being used (i.e. can use 40 X 12 only with TTY emulation) or the command will be ignored.
Example:
(Select VT100 Emulation and 132 X 24 display)
VT100
DISPLAY 132

TTY
Function: Causes Red Ryder to use TTY emulation.
Example:
COMM 300-N-8-1-FULL
TTY

VT52
Function: Causes Red Ryder to use VT52 emulation.
Example:
COMM 300-N-8-1-FULL
VT52

VT100
Function: Causes Red Ryder to use VT100 emulation.
Example:
COMM 300-N-8-1-FULL
VT100

GBAR
Function: When Procedure is finished executing, display the General Status Bar.
Example:
COMM 300-N-8-1-FULL
TTY
GBAR

TBAR
Function: Has nothing to do with sexy underwear. When Procedure is finished executing, display the Tabs Status Bar.
Example:
COMM 300-N-8-1-FULL
TTY
TBAR

MBAR
Function: When Procedure is finished executing, display the Macros Status Bar.
Example:
COMM 300-N-8-1-FULL
TTY
MBAR

VBAR
Function: When Procedure is finished executing, display the VT100 Status Bar.
Example:
COMM 300-N-8-1-FULL
VT100
VBAR

CLEAR
Function: Clears the display screen.
Example:
COMM 300-N-8-1-FULL
DIAL ATDT 555-1212
PROMPT CONNECT
CLEAR

MACRO filename
Function: Loads in the Macro Keys file **filename.** If the Macro Keys file can't be found, you'll hear two short beeps and the Procedure will be aborted.
Examples:
COMM 300-N-8-1-FULL
MACRO MYDISK:CompuServe Macros
DIAL 555-1212

STRIP switch
Function: Tells Red Ryder whether or not to strip control characters out during ASCII file receives, or during TEXT file receives under Kermit. **switch** is either **OFF** or **ON**.
Example:
(Strip out control characters)
STRIP ON
(Don't strip control characters)
STRIP OFF

ESC switch
Function: equivalent to turning on or off the **Enter Key Is ESC Key** choice in the **Keyboard Preferences** dialog box. **switch** is either **OFF** or **ON**.
Example:
(Map the ENTER key as an ESCAPE key)
ESC ON

DEL switch
Function: equivalent to turning on or off the **Backspace Key Is DEL Key** choice in the **Keyboard Preferences** dialog box. **switch** is either **OFF** or **ON**.
Example:
(Make the BACKSPACE key send DEL characters)
DEL ON

ANSWERBACK string
Function: sets the VT100 answerback string to **string** in the same manner as the **VT100 Modes** dialog box. **string** must be 30 characters or less
and may contain control characters specified in the same manner as in a Macro Key string.
Example:
(Set up my answerback string with my secret access code)
ANSWERBACK AB123^M

LF switch
Function: Tells Red Ryder whether or not to send linefeeds after carriage returns. **switch** is either **OFF** or **ON**.
Example:
(Send LF's after CR's)
LF ON
(Don't send LF's after CR's)
LF OFF

SLOW switch
Function: Tells Red Ryder whether or not to use slow timeouts for XMODEM file transfers. **switch** is either **OFF** or **ON**.
Example:
(Use slow timeouts for XMODEM - 10 seconds)
SLOW ON
(Don't use slow timeouts for XMODEM - 5 seconds)
SLOW OFF

MODEM
PRINTER
Function: These two commands tell Red Ryder whether to use the modem port or the printer port for communicating with the modem.
Example:
(Change to the modem port)
MODEM
(Change to the printer port)
PRINTER

CONTROL1 char
CONTROL2 char
CONTROL3 char
Function: These three commands allow you to change the control characters sent by the three control character buttons in the General Status Bar. The buttons are numbered 1, 2, and 3 from left to right. No space is between the **CONTROL** and button number, but there is a single space between the button number and **char**. **char** should be a letter from A to Z.
Example:
(Set up the General Status Bar to have Control-A, Control-B)
(and Control-C characters from left to right)
CONTROL1 A
CONTROL2 B
CONTROL3 C

File Transfers

RECA filename
RECX filename
RECK filename
Function: These three commands work like their Host Mode counterparts, telling Red Ryder to receive the file **filename** using ASCII, XMODEM, or Kermit protocols. To prevent an unattended Procedure from staying on the phone all night if something goes wrong during a file transfer, the XMODEM and Kermit receives will abort after 15 cumulative errors.
Example:
RECX MYDISK:MYFILE

SENDA filename
SENDX filename
SENDK filename
Function: These three commands work like their Host Mode counterparts, telling Red Ryder to send the file **filename** using ASCII, XMODEM, or Kermit protocols. To prevent an unattended Procedure from staying on the phone all night if something goes wrong during a file transfer, the XMODEM and Kermit sends will abort after 15 cumulative errors.
Example:
SENDX MYDISK:MYFILE

CLOSE
Function: Manually ends an ASCII file receive. If you execute this command when a file not being received, you will get an "Illegal Procedure Command" error message.
Example:
RECA MYDISK:Messages
PROMPT No more messages
CLOSE

Dialing and Redialing

DIAL dialstring
Function: This command works just like the **Dial Or Redial A Number** choice under the **Service** menu. You can embed control characters in **dialstring** by preceding them with a caret, but you don't need to put a ^M at the end (the carriage return is supplied automatically). **dialstring** must be 70 characters long or less. Many people will point out that this and the **TYPE** command are similar. They are with one exception: a number dialed using the **DIAL** command rather than the **TYPE** command can be redialed at a later time by choosing the **Dial Or Redial A Number** choice under the **Service** menu.
Example:
COMM 300-N-8-1-FULL
(Make sure the modem is ready)
TYPE AT^M
TYPE AT^M
TYPE AT^M
PAUSE
DIAL ATDT 555-1212

REDIAL dialing string
Function: This command works just like the **Dial** command, except it continuously redials a busy number until a connection is established. When a connection is established, execution of the Procedure continues. Note that aborting the redial will also cancel the Procedure.
Example:
COMM 300-N-8-1-FULL
(Make sure the modem is ready)
TYPE AT^M
TYPE AT^M
REDIAL ATDT 555-1212
(If I get to this instruction, a connection has been made!)
(So... never use a PROMPT after a REDIAL!)
TYPE MYNAME^M

Printer Output

WRITE filename
Function: This command does exactly the same thing as the **Print TEXT File** choice under the **Local** menu. The Imagewriter must be connected and selected. If it is connected but turned off, or not connected at all, this command is not executed. If it is connected, turned on, but not selected, it will wait for you to push the printer's "Select" button. This command is ignored if you are doing an ASCII file transfer.
Example:
WRITE MYDISK:Messages

SCREENCOPY
Function: This command does exactly the same thing as the "Dump Screen To Printer" button in the General Status Bar. See the **WRITE** command above for cautions about printer hookup and select-state. No printing is done if you are doing a file transfer.
Example:
PROMPT Logging off at:
SCREENCOPY
(Capture the amount of time I was on-line for my records)

ECHO ON
ECHO OFF
Function: These commands turn on and off the echoing of all incoming data to the Imagewriter printer.
Example:
(Send all incoming data to the printer)
ECHO ON

Dealing With Incoming And Outgoing Characters And Signals

TYPE string
Function: This sends **string** over the modem as if you had typed it on your keyboard. Control characters can be embedded in **string** in exactly the same manner as they are in Macro Keys. A carriage return is not automatically sent at the end of **string**, you must tell it to do so explicitly with the ^M sequence. If you need to send a linefeed after a carriage return, use the sequence ^M^J.
Example:
(Send a Control-C)
TYPE ^C
(Send the string "HELLO" followed by a carriage return)
TYPE HELLO^M
(Send the string "HELLO" with no carriage return at the end)
TYPE HELLO

PROMPT string
Function: This waits for the exact string of characters **string** to come over the modem before executing the next Procedure command. Upper and lowercase characters are different and must match. You may not embed control characters in **string**, to do this use the **PROMPT ^char** command below. **string** may be up to 20 characters long. It is recommended that you use as few characters as possible to make **string** unique.
Example:
PROMPT NECT
(Trap the last four characters of the modem's CONNECT)
(message before continuing)
TYPE Hello and welcome to Scott's Macintosh!^M
TYPE Now going to Host Mode^M
HOST

PROMPT ^char
Function: Waits for the control character **char** to come over the modem before executing the next Procedure command. **char** should be an uppercase letter from A to Z. There is a single space between **PROMPT** and the caret, but no space between the caret and **char**.
Example:
PROMPT ^C
(Wait for a Control-C character)
TYPE Connected!^M

SHORT BREAK
Function: sends a short (233 millisecond) modem BREAK signal if the modem is equipped to support it.
Example:
(Send a short BREAK signal now)
SHORT BREAK

LONG BREAK
Function: sends a long (3 1/2 - 4 second) modem BREAK signal if the modem is equipped to support it.
Example:
(Send a long BREAK signal now)
LONG BREAK

ALERT1 string/command
ALERT2 string/command
ALERT3 string/command
Function: These three commands let you look for more than one **string** to come over the modem, and do different commands (typically **JUMPTO** commands) depending on the string received. **string** must be less than 20 characters and upper and lowercase letters are different and must match. These commands do not hold up execution of the Procedure while waiting for the string to come over the modem. Control characters cannot be embedded in **string**. **command** is a full and valid Procedure command to be executed immediately when that **string** is received. To disable an **ALERT** command, simply include the proper **ALERT** number command with no **string/command** supplied. Anytime an **ALERT** or **PROMPT** **string** is received, all active **ALERT** and **PROMPT** commands are disabled. There is a single space between the **ALERT** number and **string**, but no spaces between **string**, the slash character, and **command**. Don't confuse the slash character, which is directly to the left of the righthand SHIFT key on your keyboard, with the backslash character ("\").
Example:
(Wait for either "you have mail" or the string)
("Command?" and branch to appropriate routine)
ALERT1 you have mail/JUMPTO (GET MAIL)
PROMPT Command?
(I got the "Command?" string so I JUMPTO somewhere else)
JUMPTO (DO COMMAND)
(GET MAIL)
(If I get here, I got the "you have mail" string and need to)
(handle that)