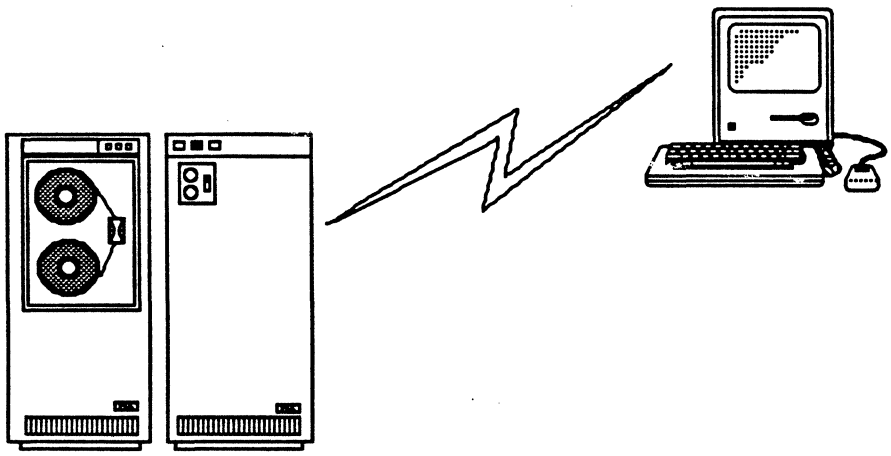
pcLINK™

The Macintosh-Mainframe Solution



Pacer Software, Inc.

La Jolla, California

Westborough, Massachusetts

Contents

1.
Introduction/Getting Started

This document provides information on using pcLINK on the Macintosh and is intended to complement the main pcLINK manual. That manual contains instructions on host pcLINK installation, on using the IBM PC version of pcLINK, and on performing host initiated pcLINK functions.

To get started with pcLINK on the Macintosh you should take the following steps:

A. Make sure your system administrator has installed the host resident portion of pcLINK as outlined in the main pcLINK manual.

B. Check that your Macintosh has been connected to a terminal line as specified in the "Cabling Connection" chapter of this documentation.

C. Familiarize yourself with the following pcLINK files:

The pcLINK program.

Configuration files. These files contain pcLINK communication parameters. Opening a configuration file will open pcLINK with the chosen file. The configuration file also contains the name of the terminal and function files to load. For more information on configuration see the Configuration section of the

Terminal definiton files. These files contain information that pcLINK uses to emulate a specific terminal.

Softkey files. These files contain user definable softkey mappings. For more information on softkey files see the SoftKeys section of the manual.

D. Invoke pcLINK by:

1. Opening pcLINK™ itself. You will be prompted for a configuration file.

2. Opening a configuration file. This will load pcLINK™ with that configuration file.

E. Under the **Config** pull-down menu, edit the System and Communication parameters as appropriate.

Note : This manual assumes knowledge of the information in Macintosh, the owners manual.
Note : **pcLINK will not run on a 128K Macintosh.**

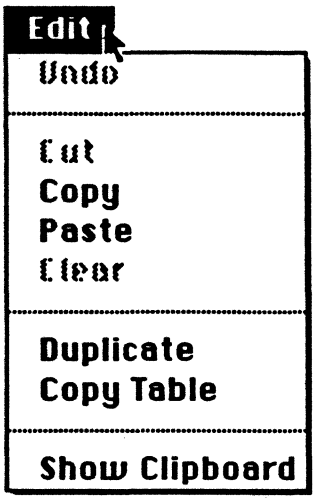
2.
Terminal Emulation

pcLINK is normally in terminal mode, emulating the terminal type that is connected to your host system. In this mode, you enter commands at the Macintosh just as you might enter them at a terminal connected to your host. The type of terminal that is being emulated is determined by the terminal file that has been loaded, and can be set in the System Parameters dialog box under the **Config** menu.

The emulator window can be moved and resized. Double clicking on the drag bar will toggle between full-sized windows and any previous window size and position.

The mouse can be used to move the terminal's text cursor. Clicking the mouse button in the terminal window will cause pcLINK to generate the proper number of cursor keys to send the text cursor to the specified position. Text within the emulator window can be marked by dragging the mouse or by double clicking on a word.

The Macintosh flower key is used as a control key during terminal emulation. Special control key sequences are command-2 to generate a Null (^@), command-6 for a RS (^^), command-'-' for a US (^_) and command-backspace for a Delete.



3.
Editing

pcLINK provides a range of Macintosh editing capabilites. These can be used to edit text appearing in either the emulator window or the display window. See section 4 for information on invoking the file display window.

Copy
Copy the highlighted text to the Clipboard.

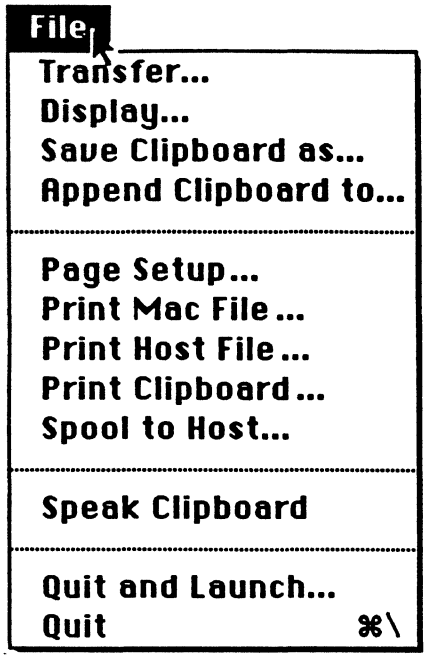
Paste
Send the contents of the Clipboard to the host computer as if it had been input from the keyboard.

Duplicate
Equivalent to a **Copy** followed by a **Paste**.

Copy Table
Copy the highlighted text to the Clipboard in table form.

Show Clipboard/Hide Clipboard
Display or hide a window showing a portion of the current Clipboard contents.

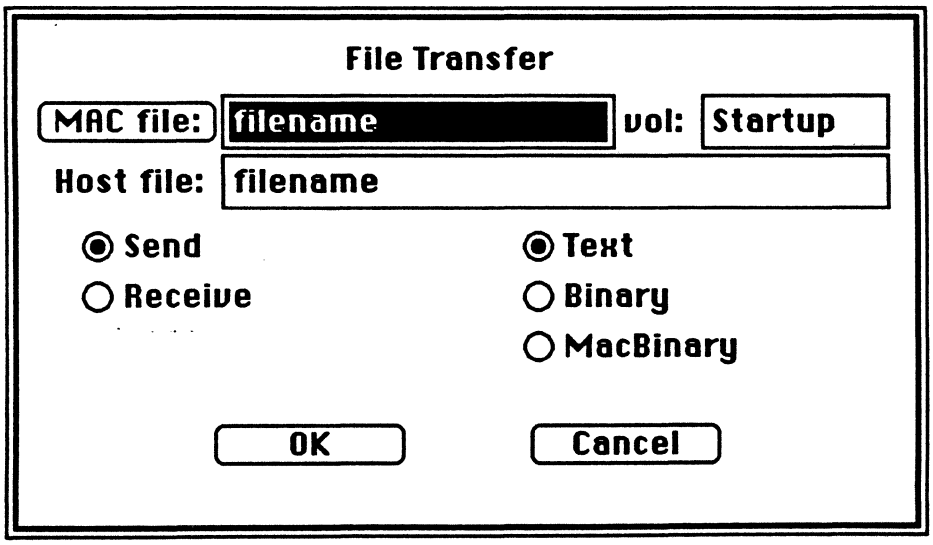
Undo/Cut/Clear
Undo, Cut, and Clear are used by the Macintosh desk accessories.



4.
File Utilities

Transfer

Transfer files from your Macintosh to the host, or from the host to your Macintosh. The files may be either ASCII text files or binary files. In either case, pcLINK performs error detection and recovery, thereby providing virtually error free transmission by finding and correcting errors in the transmitted data. The following dialog box will be displayed:



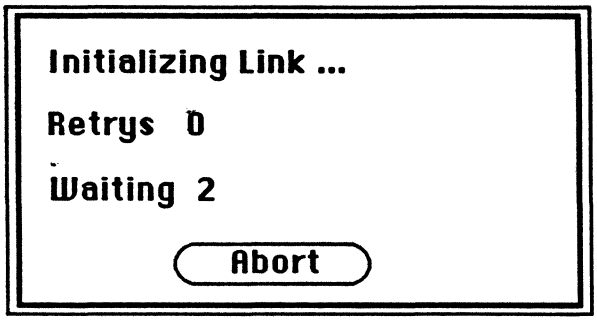
Text: Transfer text files -- file will be converted into the file format expected by the machine to which it is being sent.

Binary: Files are transferred exactly as they are; no file format conversion is performed.

MacBinary: Files sent to the host will be stored in MacBinary format, the data and resource fork concatenated into one file. Files received from the host that are in MacBinary format will be restored to their original Macintosh format. If the host file is not in MacBinary format then a Binary transfer will be performed.

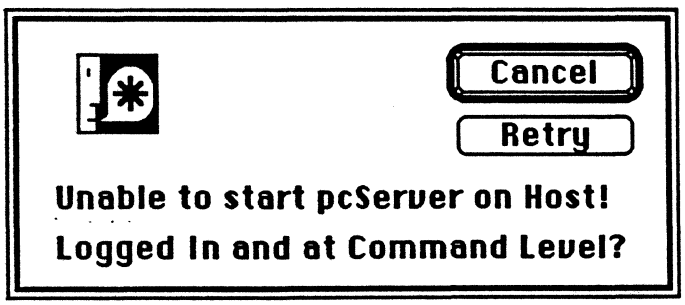
MAC file: A menu from which to choose a MAC file is presented.

OK: Initiates the file transfer after the appropriate choices have been made in the file transfer dialog box. If the destination file name is left blank then it will default to the source file name. The following message will appear on your screen:



pcLINK is trying to invoke the program pcSERVER on the host. This is performed each time you transfer a file when the server is not already running on the host. During link initialization, you may select **Abort** at any time to temrinate initialization.

The link initialization will be unsuccessful if something is wrong with the communications link or if you are not currently logged onto the host. If after two attempts pcLINK cannot initialize the link, an error message will appear:



Once the above error message is displayed on your screen, you may select **Cancel** to return to the terminal emulator and then login to the host. If you are already logged in, contact your system administrator for help.

Once the file has been invoked, pcLINK will ask you to wait while it transfers your file to the host. When transmission is completed, the number of bytes transferred appears on the screen and pcLINK informs you that the file has been sent:

Alerts will warn you in case of any errors.

Display/Close
Open or close a window showing the contents of any MAC text file. pcLINK will prompt you for the name of the file.

Save Clipboard
Save the Clipboard contents in a file.

Append Clipboard
Append the Clipboard to a file.

Page Setup
Invokes standard Macintosh page setup for printing dialog box.

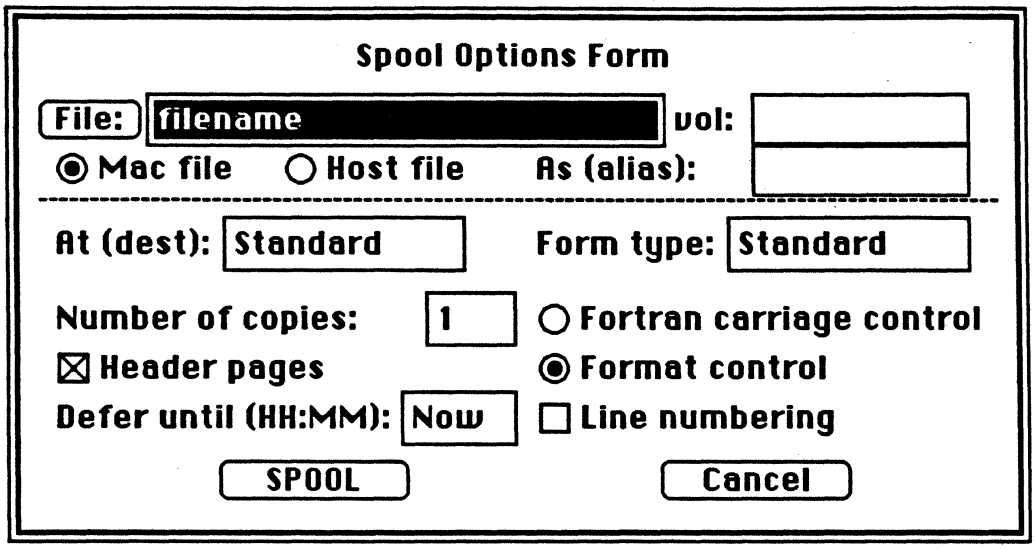
Print Mac File
Print a local Macintosh text file on the Macintosh printer.

Print Host File
Print a host file on your MAC. Only text files may be printed using this function. This function looks and operates like a secure text file transfer, beginning with link initialization.

Print Clipboard
Print the current contents of the clipboard on the Macintosh printer.

Spool to Host

Print a MAC or a host file on a host printer. A wide range of options are available for control of the printout at your host computer. The choices are displayed in the following input form:



With the exception of the Mac or Host file choice, each item on the form corresponds to a host specific spool option. (Each host will have a different spool form). The values showing in the input fields are those normally used.

File: Supplies a list of available MAC files.

Speak Clipboard

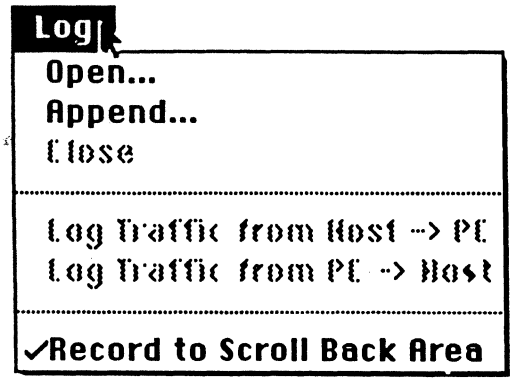
Speak the contents of the clipboard through the Macintalk driver (if available).

Quit and Launch

Quit pcLINK and begin execution of another Macintosh application. pcLINK will present valid application choices.

Quit

Quit pcLINK.



5.
Traffic Logging

pcLINK's Traffic Logging facility allows you to record data transmitted between your Macintosh and host systems. Any data displayed on your screen while in Terminal Mode will also be recorded in a log file on the Macintosh. If no logging is taking place then the words **no logging** will be displayed on the top line of the **Log** menu.

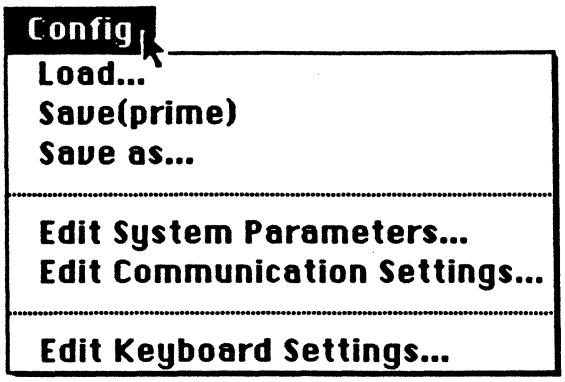
Open
Specify the name of the MAC file to contain the recorded logging information.

Append
Append logging to an existing log file.

Close
Close the current log file.

Log Traffic from Host --> PC
Log Traffic from PC --> Host
Set the mode in which traffic logging takes place by selecting, and thus checking or unchecking, the logging direction entries in the **Log** menu. You may log traffic from host to Macintosh, from Macintosh to host, or in both directions at once. Only data which is transmitted in the selected mode is recorded in the log file.

Record to Scroll Back Area
Toggles the Scroll Back Area on and off. When the Scroll Back Area is active any text that scrolls past the top of the emulator window will be stored in the Scroll Back Area. This text can be viewed by scrolling the vertical scroll bar up. The size of the Scroll Back Area is dependent on available memory.



6.
Configuration

The **Configuration** section of pcLINK allows you to establish the operating characteristics of pcLINK. Communication, emulator, and host information can be stored in configuration files. The name of the current configuration file is displayed in parentheses in the **Config** menu.

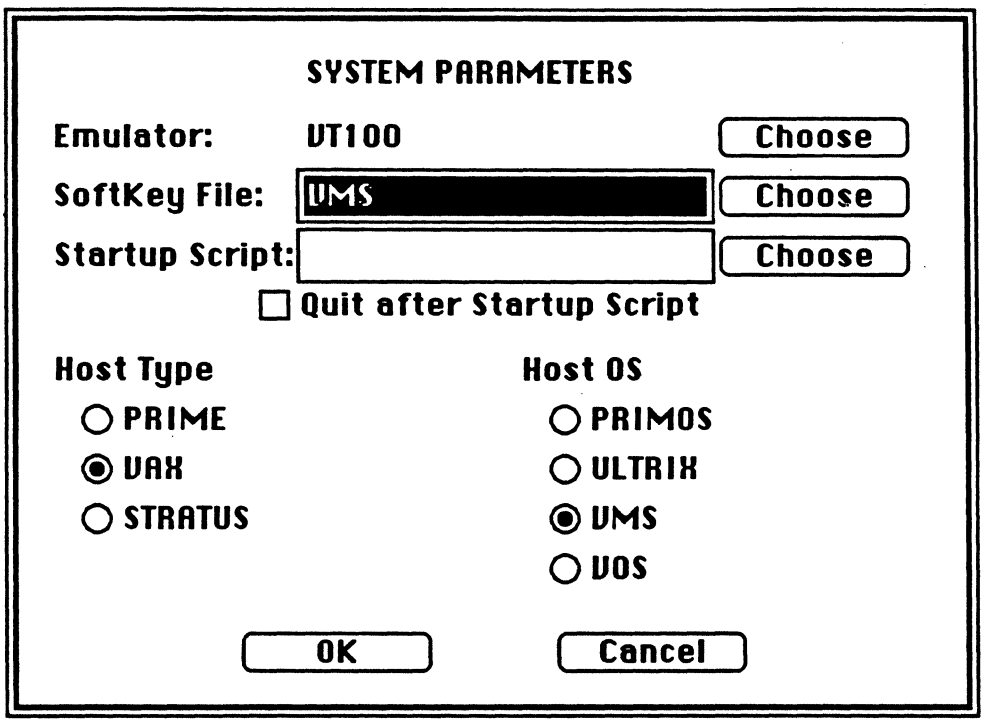
Load
Load a configuration file selected from a list of defined configuration files.

Save
Save the current configuration settings.

Save as
Save the current configuration settings in a file to be designated. pcLINK will request a Macintosh filename.

Edit System Parameters

Produces the following dialog box:



Choose...Emulator: Gives a list of current Emulator files from which to choose.

Choose...SoftKey file: Gives a list of current SoftKey files from which to choose.

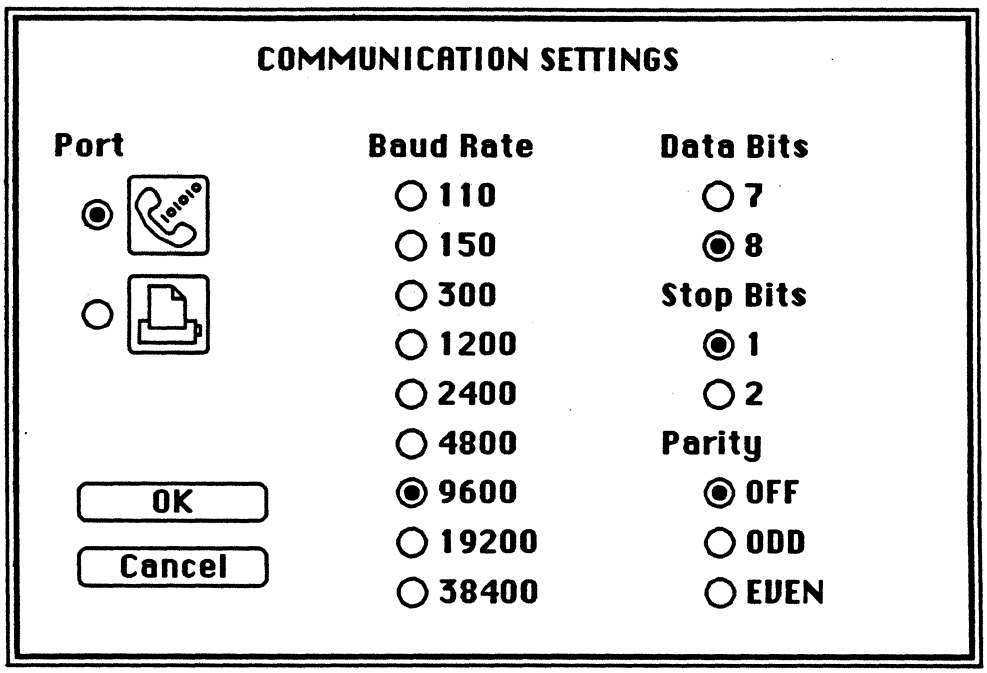
Startup Script: Specifies a Script file that will be invoked if pcLINK is started with this configuration.

Choose...Script: Gives a list of current Script files from which to choose.

Quit after Startup Script: Will Quit pcLINK after the Startup Script has been run if pcLINK is started with this configuration.

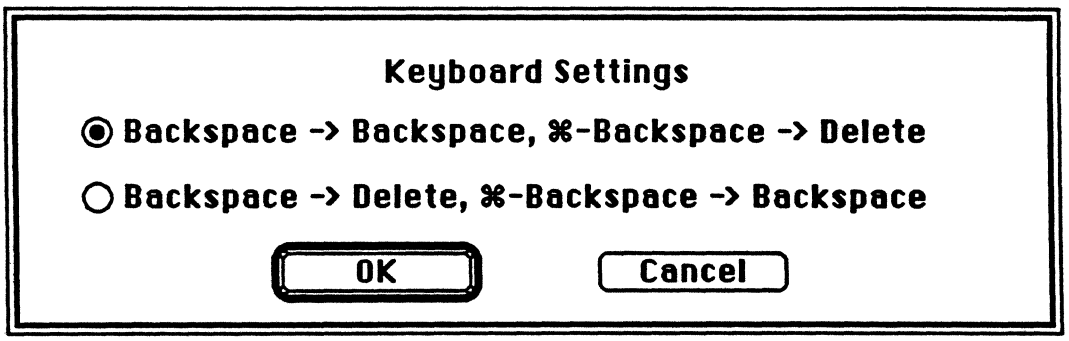
Edit Communication Settings

Produces a dialog box with the current settings:

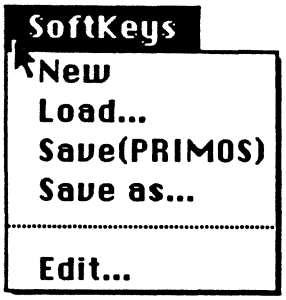


Edit Keyboard Settings...

Produces the following dialog box:



This allows you to define the function of the Backspace and Ctl-Backspace keys.



7.
Soft Keys

Since the Macintosh keyboard does not supply much in the way of keys for emulating terminals but does have a graphical interface, we have supplied user definable function keys that reside above the emulator window. They can be invoked by clicking on them with the mouse or from the keyboard where Option-1 thru Option-0 invokes the bottom row of function keys and Option-Shift-1 thru Option-Shift-0 invokes the top row.

HELP	DIR	MAIL	EDT	WHERE	CTRL Y	PURGE	UP	
TIME	TYPE	PHONE		SETDEF	CTRL C	DELETE	DOWN	LOGOUT

Each function key has associated with it a name and some actions.

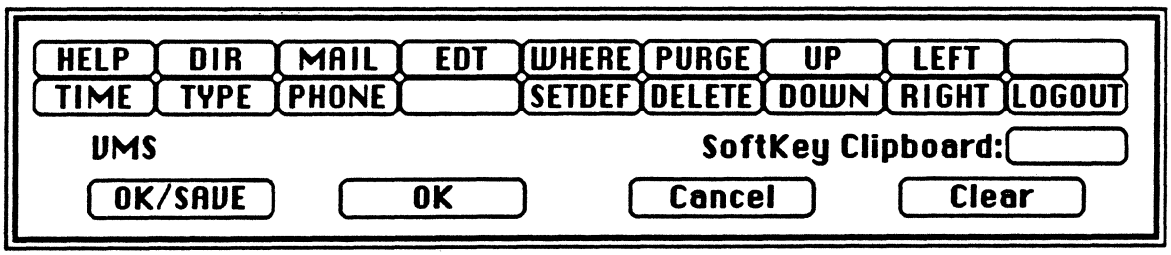
New
Create a new SoftKey file, starting with blank keys.

Load
Load the SoftKey settings from an existing SoftKey file.

Save
Save the current Softkey settings in the active SoftKey file (denoted in parentheses).

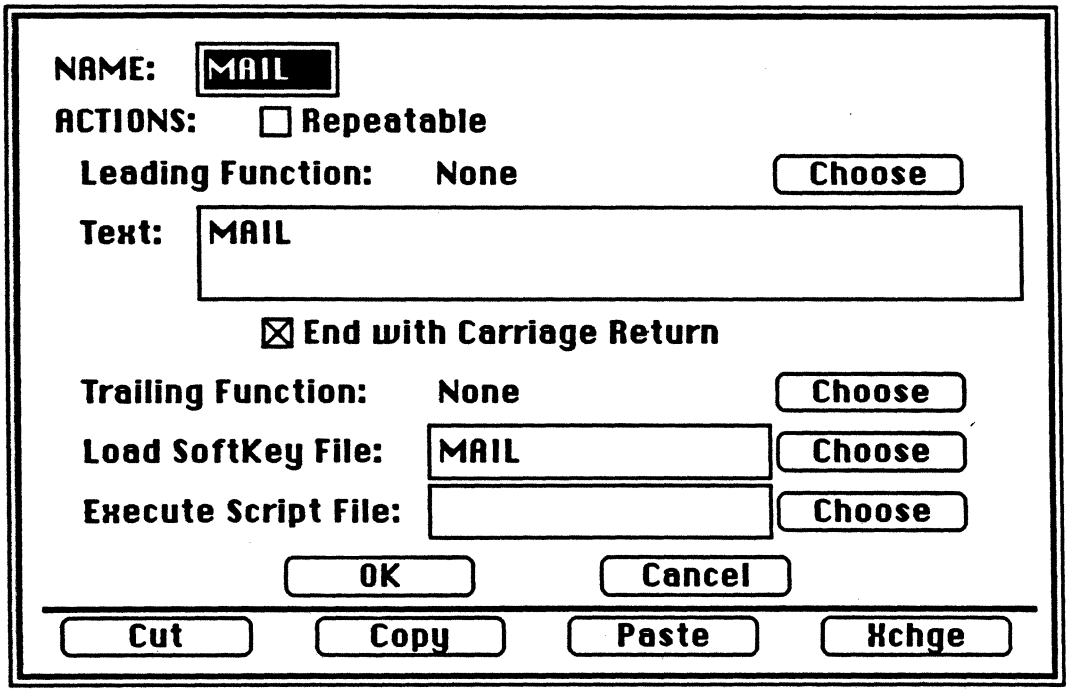
Save as
Save the current SoftKey settings in a file to be designated.

Edit
Edit the SoftKey settings. The SoftKey editing dialog box will appear:



OK/SAVE: Same as selecting **OK**, but the current SoftKey will be saved to the current SoftKey file. The name of the current SoftKey file is displayed above the OK/SAVE button.

Select the SoftKey that you wish to edit and the current settings for that SoftKey will be displayed.



The fields in this dialog box have the following meanings:

NAME: defines the name that will be displayed for this SoftKey.

Repeatable: specifies whether holding down the mouse button on the key will cause it to repeat.

Text: text that will be sent to the host. Control characters can be sent by preceding the character with a ^, as in ^P. Other non-printable characters can be specified in hexadecimal notation by preceding a hexadecimal number with a #, as in #0D. To send a literal ^ you should specify ^^; to send a literal # you should specify ##.

End with Carriage Return: specifies whether a carriage return should be sent after the text.

Load SoftKey File: specifies a SoftKey file to be loaded.

Choose...SoftKey File: gives a list of available SoftKey files.

Execute Script File: specifies a Script file to be executed.

Choose...Script File: gives a list of the available Script files.

Cut: cut the current SoftKey to the SoftKey Clipboard.

Copy: copy the current SoftKey to the SoftKey Clipboard.

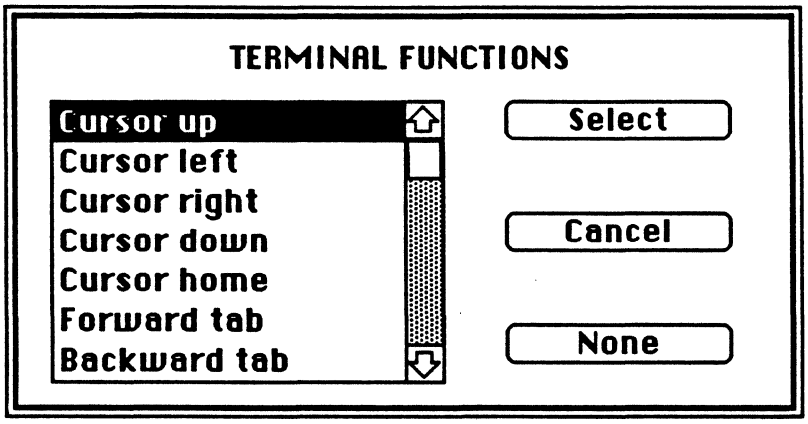
Paste: paste the SoftKey Clipboard to the current SoftKey.

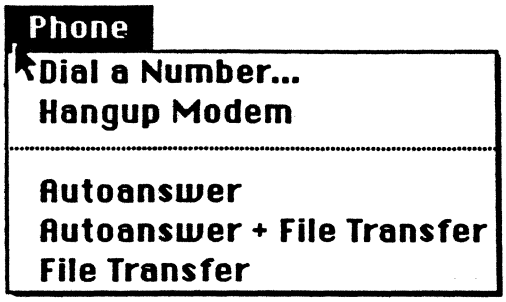
Xchge: exchange the current SoftKey with the SoftKey Clipboard.

Leading Function: specifies a terminal function that will be generated before the text string is sent .

Trailing Function: specifies a terminal function that will be generated after the text string is sent.

Choose...Function: Produces a scrolling list of the terminal functions available for the current emulator. **None** will clear an existing function setting.

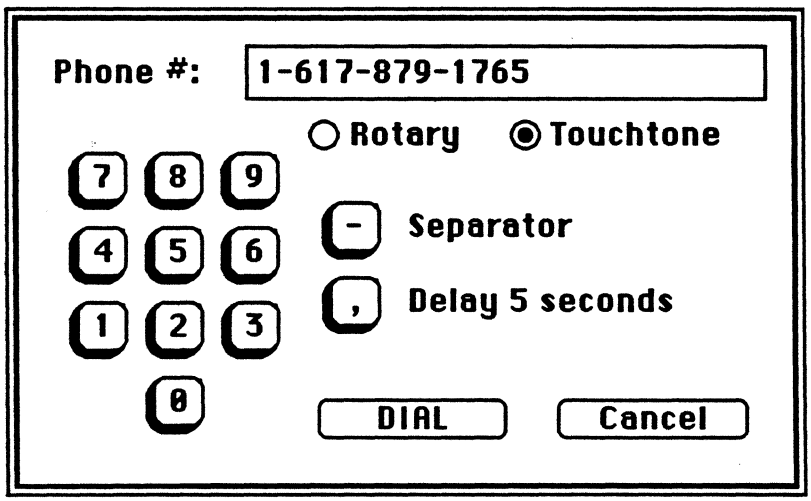




8.
Phone Handling

The **Phone** section of pcLINK supports communication through a Hayes compatible modem.

Dial a Number
The following dialog box will be invoked:



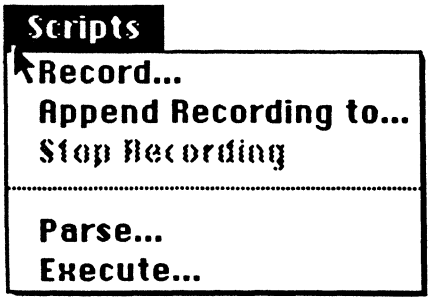
Build the desired number from the keyboard or by clicking the desired numbers on the number pad. If a number has been copied to the clipboard it will appear as the phone number.

Hangup Modem
Hangup the modem.

Autoanswer
Wait and answer an inbound phone call.

Autoanswer and File Transfer
Wait for a pcLINK host initited call and file transfer.

File Transfer
Process host initiated file transfer.



9.
Script Files

With Script files, pcLINK has the ability to record and play back a series of commands.

Record
Record all subsequent actions to the file of your choice.

Append Recording to
Append subsequent actions to the file of your choice.

Stop Recording
Stop recording actions and close the file.

Parse
Parse the Script file of your choice to ensure that it is syntactically correct. This is useful for Script files that were created with an editor.

Execute
Execute the Script file of your choice. Hitting Cntl-'.' while a Script file is executing will terminate execution after the current script has been completed.

These Script files can also be created with an editor and therefore a list of the available commands is given.

comset("port", baud, "parity", databits, stopbits);
- set the communication parameters
  - port = "modem" or "printer"
  - baud = baud rate 110, 150, 300, 1200, 2400, 9600, 19200
  - parity = "no", "odd" or "even"
  - databits = 7 or 8
  - stopbits = 0, 1 or 2

loadconfig("filename");
- load the specified configuration file

portflush();
- flush the async port

realbreak();
- send a real break to the host

send("mode", "macfile", "hostfile");
- send a Mac file to the host
- mode = "text", "binary" or "macbinary"
- macfile = Macintosh file name
- hostfile = host file name

receive("mode", "macfile", "hostfile");
- receive a host file on the Mac
- mode = "text", "binary" or "macbinary"
- macfile = Macintosh file name
- hostfile = host file name

openfilelog("filename");
- open the specified file and record information about subsequent file transfers in it

closefilelog();
- close the current file transfer log file

closlink();
- bring down the server on the host

spool("source", "filename", "destination", "form");
- send the specified file to the hosts spooler
- source = "mac" or "host"
- filename = file name
- destination = name of device
- form = name of form

print("source", "filename");
- print the specified host file on the Macintosh printer
- source = "mac" or "host"
- filename = file name

dial("phone#");
- dial the modem with the specified phone number

hangup();
- hangup the modem

autoanswer();
- wait for the modem to answer

filetransfer();
- respond to host initiated file transfer

aputs("text");
- send the specified text to the host with a carriage return at the end

aputs_nr("text");
- send the specified text to the host without a carriage return

wait(seconds);
- wait the specified number of seconds

waituntil("hh:mm");
- wait until the specified time

emulate("string");
- enter terminal emulation until the host has stopped sending data and the last three characters sent match the specified string. Note: this is best done in record mode.

userinput("prompt text");
- display a dialog box with the specified prompt and send the entered string to the host

openlog("filename");
- open the specified file for loggin, log to pc is the default setting

closelog();
- close the current log file
logtopc();
- toggle the current log to pc state
logtohost();
- toggle the current log to host state

loadsoftkeys("filename");
- load the specified softkey file
softkey(number);
- invoke the specified softkey, 1 - 10 is the top row and 11 - 20 is the bottom

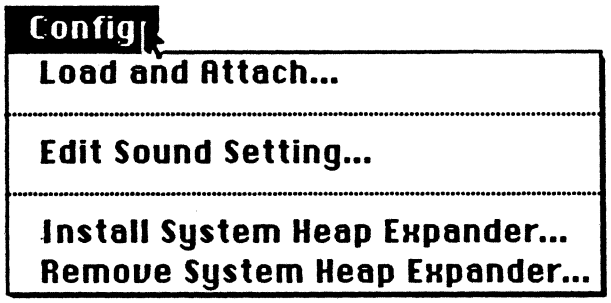
10.
Virtual Disks

A Virtual Disk is a file in the host computer's file system that is treated like a Macintosh Disk by the Macintosh. This functionality is supplied with the MountVDrive™ application on the Macintosh and the MiniMac application on the host.

Refer to the Create command documentation in the MiniMac Section to learn how to create Virtual Disks. A Virtual Disk must be created before it can be accessed from the Macintosh.

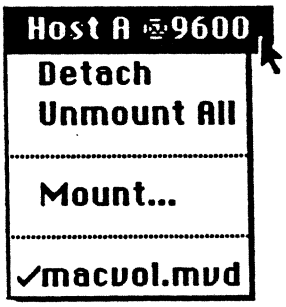
MountVDrive™

Once a Virtual Disk has been created on the Host, the MountVDrive™ application can be used to mount the Disk so that the Macintosh can access it. MountVDrive™ requires that you have logged on to the Host and that you have a Configuration file that specifies the communications parameters required to communicate with the Host. The Configuration file is created by pcLINK™ (see pcLINK Configuration for more information). When this has been accomplished you can open MountVDrive™.



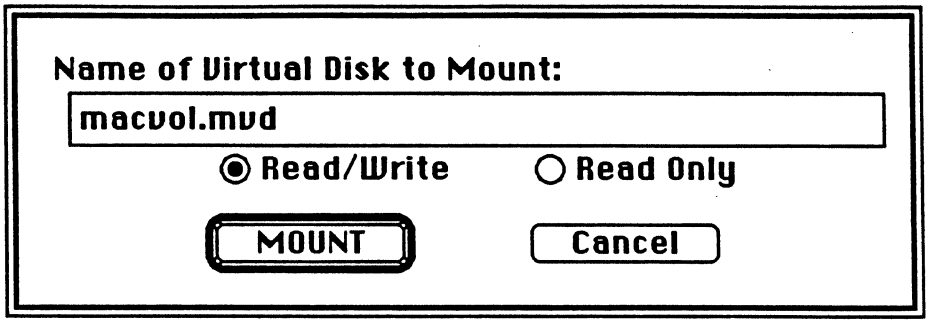
Load and Attach...

Will prompt you for the name of a Configuration file which it will use to set up communications with the Host and then invoke pcServer. This will produce a new menu whose title is the name of the Configuration file.



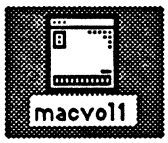
Mount...

Will prompt you with the following dialog box:



You should specify the **full path name** for a Virtual Disk that you created on the host and then select Mount. When a Virtual Disk has been mounted it will be listed in the menu for the specified connection. Selecting a mounted Virtual Disk from the menu will unmount it and remove it from the menu.

The Finder will represent the Virtual Disk with this icon:



Unmount All

will unmount all the mounted Virtual Disks on the specified connection.

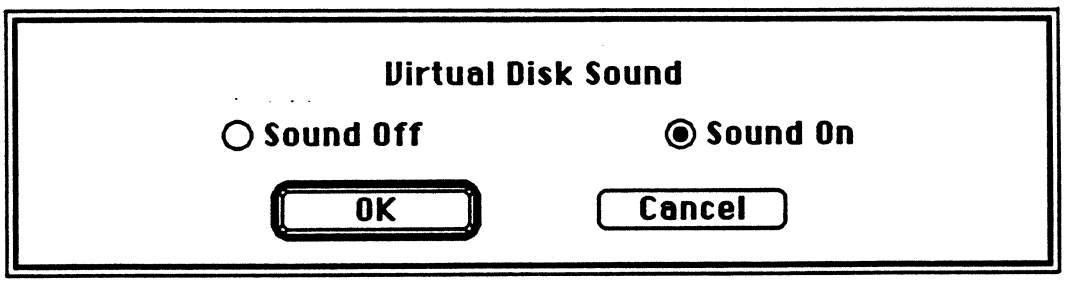
Detach

will unmount all mounted Virtual Disks on the specified connection, bring down the pcServer on the Host, close the communications connection and remove the menu.

You may mount up to four Virtual Disks at once and communicate to hosts on both serial ports. pcLINK™ will unmount all Virtual Disks so that it may perform terminal emulation. It will remount them on exit.

Edit Sound Setting...

Will produce the following dialog box:



With Sound On every access to a Virtual Disk will generate a sound.

The Virtual Disk Driver resides in the System Heap portion of Macintosh memory. The size of the System Heap is set at startup time. It may be that there is not enough System Heap for the Virtual Disk Driver. If this happens, **Install System Heap Expander...** will modify the current system file so that more System Heap will be allocated when the Macintosh is rebooted. You should Quit MountVDrive™ and restart the Macintosh. **We recommend that you do this.**

On a 512K Macintosh with the 64K ROM, some applications can not run with an expanded system heap! **Remove System Heap Expander...** will remove the System Heap Expander from the current System file.

11.
MiniMac - Virtual Disk Host Utility

MiniMac is the host interface to Macintosh virtual disks. MiniMac has the following capabilities:

- Create Macintosh virtual disks.
- Display the contents of Macintosh virtual disks.
- Copy files from one virtual disk to another.
- Import files from the host file system to the virtual disk.
- Export files from the virtual disk to the host file system.

MiniMac serves two main purposes: first, it allows users to create virtual disks (this must be done BEFORE the virtual disk is accessed from the Macintosh using MountVDrive); second, it provides the bridge between the host file system (Prime/Primos, VAX/VMS, VAX/Ultrix, Stratus/VOS) and the Macintosh virtual disk file system.

MiniMac is a host application program and can be run from either a normal host terminal or using pcLINK's terminal emulation capability. MiniMac is invoked by issuing the following host command:

MiniMac

MiniMac is a command driven application with the following command set:

!<Host command>	Gateway to host operating system.
Copy <Mac source> <Mac target>	Inter/intra virtual disk file copy.
CReate <Host file> [<size>]	Create a virtual disk.
DELete <Mac file>	Delete a file from the virtual disk.
Dismount {<Volume name> | -All}	Dismount a volume.
Export <Mac file> <Host file>	Export a file from the virtual disk.
Help	Print this help text.
Import <Host file> <Mac file>	Import a host file to the virtual disk.
List [<Volume name>] [-wizard]	List current volume or directory.
LOck [<Volume name>]	Toggle software lock of current volume.
Mount [<Host file>]	Logically mount a virtual disk.
Quit	Leave MiniMac.
Settings [-<setting> <value>]	Display or set current modes.
Type <Mac file> [-wizard]	Display a text file.
VolumeInfo [-wizard]	Display information about mounted volumes.
<Volume name>:	Set current volume.
<Cr>	Set current volume to next volume.

All command arguments are separated by spaces; if a <Mac file> or a <Volume name> needs to contain spaces then enclose the name within quotes (e.g. "My Mac file"). Any <Mac file> may be either just a file name (assumed to be on the current volume, see below) or may be a volume name, file name pair. The following are valid Macintosh file specifications:

AMacFile
volumeName:fileName
"A Mac File with Spaces"
"A Volume Name with Spaces:A Mac File with Spaces"

MiniMac has a concept of mounted volumes (very similar to the disk icons that appear on the right side of the Macintosh desktop); a volume is mounted by creating the volume (with the Create command) or mounting a previously created volume (with the Mount command). Any number of volumes may be mounted at one time (VolumeInfo will list the currently mounted volumes). MiniMac also has a concept of a "current volume" (similar to the selected disk on the Macintosh desktop); the most recently created or mounted volume will initially be the current volume. The current volume may be set by either null carriage returns (this will cycle through all mounted volumes) or by specifying the target volume name as a command (e.g. targetVolume: -- note that the colon is required). The name of the current volume is used as MiniMac's prompt so it is always easy to tell what the current volume is. The volume name is the text that will appear under the disk icon on the Macintosh desktop -- volume names are initially set when the virtual disk is created (similar to initializing the disk from the Macintosh).

The following pages describe, in detail, each of MiniMac's commands (the leading capitalized letters are the minimal abbreviations for each command).

!<Host command> Gateway to host operating system.

The "!" command allows a host operating system command to be executed from within MiniMac. For example, to list the current host directory:

```
!ld       (Primos)
!dir      (VMS)
!ls       (Ultrix/UNIX)
!list     (VOS)
```

Copy <Mac source> <Mac target> Inter/intra virtual disk file copy.

The "Copy" command allows files to be copied within a single virtual disk, or files to be copied from one virtual disk to another. Of course, only currently mounted virtual disks may be accessed with the copy command. If volume names are not specified for either the source or target Macintosh files, then the current volume will be used. For example, the following are valid copy commands:

```
copy    file1  file2
c       "A file name with spaces"  anotherName
copy    vol1:file1  vol2:file2
copy    test.file  vol3:test.file
co      justForFun  "A long volume name:A long file name"
```

CReate <Host file> [<size>] Create a virtual disk.

The "CReate" command must be used to create a new virtual disk; the new virtual disk will be automatically mounted after it is created. The "<Host file>" must be a valid pathname for the host operating system; this will be the name of the disk file that is created within the host file system and will be formatted to be a Macintosh virtual disk. It is STRONGLY suggested that all Macintosh virtual disks be named using a ".mvd" suffix -- this recommendation may be enforced in future versions of MiniMac.

The size parameter is optional, but may be any of the following values:

200K	2.5MB
400K	5.0MB
800K	10MB
1.6MB	

If no size is specified then a 400K Macintosh virtual disk will be created. This size is the formatted capacity of the virtual disk (a 400K disk has the same capacity as a single sided Macintosh micro-floppy). When a virtual disk is created only the required initial disk blocks are written; the disk will expand up to its maximum size as data is moved to the virtual disk (either from the Macintosh or by using MiniMac).

After the create command is entered, the user will be prompted for the name of the virtual disk; this is the volume name, and will be MiniMac's prompt whenever the new disk is the current volume. This is the ONE case (within MiniMac) where quotes should NOT be used to surround Macintosh names that contain spaces -- we wouldn't want to have a totally orthogonal user interface. See also the "Mount" command for further information. The following are valid create commands:

create	newDisk.mvd	
cr	gc>demos>demo.mvd 200K	(Primos)
create	[dgh.demos]demo.mvd 800k	(VMS)
create	>sysAdmin>demos>demo.mvd 1.6mb	(VOS)
create	/usr/gc/demos/demo.mvd 400K	(Ultrix/UNIX)

The following example shows the creation of two Macintosh virtual disks (user input is underlined):

MiniMac
[MiniMac version 1.0 Alpha test]

<none>: create disk1.mvd
  Enter Macintosh volume name [Untitled]: Volume1
  Initializing disk (400K)...
  Disk created.
Volume1: cr disk2.mvd 1.6mb
  Enter Macintosh volume name [Untitled]: Another volume
  Initializing disk (1.6MB)...
  Disk created.
Another volume: quit

DELete <Mac file> — Delete a file from the virtual disk.

The "DELete" command is used to delete files from a virtual disk. This will permanently remove a file from the virtual disk, freeing up its space to hold other files. The following are valid delete commands:

delete	myFile
del	myVolume:myOtherFile

Dismount {<Volume name> | -All} Dismount a volume.

The "Dismount" command will remove a disk from MiniMac's list of currently mounted volumes. If the command is given with no arguments then the current volume will be dismounted; if a volume name is given then the specified volume will be dismounted; if "-all" is given then all currently mounted volumes will be dismounted. The following are valid dismount commands:

```
dismount
dismount        "A volume name:"
dismount        Vol1:
dis             -all
```

Export <Mac file> <Host file> Export a file from the virtual disk.

The "Export" command provides the mechanism to copy files from a Macintosh virtual disk into the host file system. The command takes two file specifications as arguments: first is the Macintosh file that will be moved to the host file system; second is the target host pathname that will be created to hold the exported Macintosh file. The actual operation of the export command changes radically depending on the values of the "exportMode" and "exCRmode" settings (see the "Settings" command for more information). Some examples of valid export commands are as follows:

```
export  file1 hostFile
ex      vol1:file2 text.fil
export  fileName gc>demos>testfile.data          (Primos)
export  fileName [dgh.demos]testfile.dat         (VMS)
export  filename >sysAdmin>demos>testfile        (VOS)
export  filename /usr/gc/demos/testfile.data     (Ultrix/UNIX)
```

Help Print help text.

The "Help" command prints a brief help message explaining the syntax of all MiniMac commands.

Import <Host file> <Mac file> Import a host file to the virtual disk.

The "Import" command provides the mechanism to copy files from the host file system into a Macintosh virtual disk. The command takes two file specifications as arguments: first is the host pathname that will be moved into the virtual disk; second is the virtual disk file that will be created to hold the imported data. The actual operation of the import command changes radically depending on the values of the "importMode" and "imCRmode" settings (see the "Settings" command for more information). Some examples of valid export commands are as follows:

```
import  hostFile file1
im      test.fil vol1:file2
import  gc>demos>testfile.data fileName          (Primos)
import  [dgh.demos]testfile.dat fileName         (VMS)
import  >sysAdmin>demos>testfile fileName        (VOS)
import  /usr/gc/demos/testfile.data fileName     (Ultrix/UNIX)
```

List [<Volume name>] [-wizard] List current volume or directory.

The "list" command will list the contents of a Macintosh virtual disk volume. If no volume name is given then the current volume will be listed; otherwise, the specified volume will be listed. If the "-wizard" flag is present on the command line then additional technical information about each file will be displayed. The following is a valid MiniMac session using the list command:

```
MiniMac
[MiniMac version 1.0 Alpha test]

<none>: mount small.mvd
Small: mount new.mvd
NewDisk: list


config                      799 bytes, 10 Jun 86 16:44:54 Tuesday
Copy of Copy of config      799 bytes, 10 Jun 86 16:40:36 Tuesday
DeskTop                     2357 bytes, 14 Jun 86 16:54:08 Saturday
help.c                      2105 bytes, 11 Jun 86 11:30:46 Wednesday
helpme.c                    2453 bytes, 11 Jun 86 11:32:04 Wednesday
NewPrime                    170 bytes, 10 Jun 86 16:39:50 Tuesday

NewDisk: l Small:


DeskTop                     2353 bytes, 14 Jun 86 16:54:01 Saturday
help                        2105 bytes, 10 Jun 86 12:49:05 Tuesday
helpBin                     1932 bytes, 10 Jun 86 12:50:24 Tuesday
helpme.c                    2453 bytes, 11 Jun 86 11:32:04 Wednesday
PacerNew                    55356 bytes, 10 Jun 86 21:36:34 Tuesday
Parser                      55356 bytes, 10 Jun 86 09:33:53 Tuesday
PRIME                       170 bytes, 07 Jun 86 14:04:10 Saturday
xx                          1931 bytes, 10 Jun 86 21:25:46 Tuesday
xxCR                        1508 bytes, 10 Jun 86 12:54:21 Tuesday


NewDisk: quit
```

LOck [<Volume name>] Toggle software lock of current volume.

The "LOck" command will toggle the software lock of the specified virtual disk; if no volume name is specified then the current volume will be used. If a volume is locked, then a Macintosh that has the volume open (by using MountVDrive) will not be allowed to write to the virtual disk. MiniMac will ignore the setting of the software lock when it is requested to write to the virtual disk.

Mount [<Host file>] Logically mount a virtual disk.

The "Mount" command will logically mount a virtual disk, thus allowing MiniMac to access the volume. The host pathname that was given when creating the virtual disk must be specified with the mount command. All further reference to the virtual disk (within MiniMac) will be done using the volume's name (also specified when the disk was created) rather than the host pathname of the virtual disk. The following are valid mount commands:

```
mount new.mvd
mount gc>demos>demo.mvd                 (Primos)
mount [dgh.demos]demo.mvd               (VMS)
mount >sysAdmin>demos>demo.mvd          (VOS)
mount /usr/gc/demos/demo.mvd            (Ultrix/UNIX)
```

Quit Leave MiniMac.

The "Quit" command is used to exit MiniMac and return to the host operating system. All virtual disks will automatically be dismounted by using the quit command.

Settings [-<setting> <value>] Display or set current modes.

The "Settings" command is used to control the exact operation of the create, import and export commands. There are five modes within MiniMac that can be configured with the settings command. If the settings command is given with no arguments then the current values of these five modes will be displayed. The complete syntax is as follows ('{'s denote required arguments; '['s denote options arguments; '|' means "or"; leading capitalized letters are the minimum abbreviations):

```
Settings {-CreateMode   [Flat | Hierarchical]
          -ImportMode   [Text | Binary | MacBinary]
          -IMCrMode     [AsIs | CrPerParagraph]
          -ExportMode   [Text | Binary | MacBinary]
          -EXCrMode     [AsIs | Generate {<value>}] }
```

The default values are: CreateMode is Flat; ImportMode is Text; ImCRmode is AsIs; ExportMode is Text; ExCRmode is AsIs. Multiple options for the settings command may be specified on a single command line. The operation of each of these modes is defined in the following paragraphs.

Settings -CreateMode [Flat | Hierarchical]

The "-CreateMode" option controls the type of virtual disk that will be created by the create command. The current version of MiniMac only supports the Macintosh flat file system.

Settings -ImportMode [Text | Binary | MacBinary]
Settings -IMCrMode [AsIs | CrPerParagraph]

These two options control the operation of the MiniMac import command. If ImportMode is set to Text then the import command expects to copy text files created on the host into the Macintosh virtual disk. Text files are human readable files such as: letters, memos, data, source code. In text mode, the Macintosh file type will be set to TEXT (most users need not be concerned about this). When the import command is operating in text mode then the value of ImCRmode becomes important. Depending on the type of application that the Macintosh user will use to access the imported file there may be differing requirements for the handling of line termination within the file.

Macintosh editors such as "Qued" and "Edit" are both basically line oriented and expect a line termination at the end of each logical line of the files that they are accessing. Thus, in this case, ImCRmode should be set to AsIs because this is the same way that the host will be storing text files. However, for best results with word processors such as "MacWrite" which does automatic wrapping of text within a paragraph, ImCRmode should be set to CrPerParagraph. When MacWrite first opens one of these imported files, a dialog box will be presented asking whether new lines are "paragraph" or "line" breaks. If the file has been imported in CrPerParagraph mode then "paragraph" should be chosen.

If ImportMode is set to Binary then the MiniMac import command will try to copy practically any random host file into the virtual disk. As with text, only the data fork of the file will be created and the file type will be set to TEXT. However, no data translation will be performed. This feature should be interesting only to the most advanced user.

If ImportMode is set to MacBinary then the MiniMac import command will import previously created MacBinary files from the host file system into the virtual disk. MacBinary is a semi-standard method of representing a complete Macintosh file (data fork, resource fork, menus, fonts, etc.) within the host file system. MacBinary files may be created in a number of different ways: MiniMac creates MacBinary files if ExportMode is set to MacBinary; pcLINK/PRIMElink will create MacBinary files when doing binary Macintosh to host transfer; many other Macintosh micro-to-mainframe communication programs can also create MacBinary files.

Settings -ExportMode [Text | Binary | MacBinary]
Settings -EXCrMode [AsIs | Generate {<value>}]

These two options control the operation of the MiniMac export command. If ExportMode is set to Text then the export command expects to copy text files created by the Macintosh to the host file system. Many Macintosh editors (such as "Qued" and "Edit") will always create text files; other applications (such as "MacWrite") are capable of creating text files with a "save text only" option. When the export command is operating in text mode then the value of ExCRmode becomes important. Depending on the type of application that the Macintosh user used to create the specified file there may be differing requirements for the handling of line termination within the file. If the file has been created with a new line at the end of every logical line of the file then ExCRmode should be set to AsIs. This will work fine if the file was created with an editor like "Qued" or "Edit", or with "MacWrite" if the file has been saved as "text only" using new lines for line breaks rather than paragraph breaks. However, if "MacWrite" is used to create the file and new lines are used as paragraph breaks, then ExCRmore should be set to Generate. This will enable MiniMac to automatically create line breaks (between words) if a line is extending beyond a specified length.

The default value of this length is 60 characters. At the first opportunity beyond the 60th character the line will be broken between words. This value may be reset by specifying a new value following the generate option.

If ExportMode is set to Binary then the MiniMac export command will copy the data fork of any random Macintosh file into the host file system. However, no data translation will be performed. This feature should be interesting only to the most advanced user.

If ExportMode is set to MacBinary then the MiniMac export command will export a complete image (data fork, resource fork, menus, fonts, etc.) of any Macintosh file into the host file system. This representation of a Macintosh file will probably not be meaningful to the host. However, many Macintosh micro-to-mainframe communication programs (including pcLINK/PRIMElink) do understand this format and are capable of transferring it from one machine to another.

Type <Mac file> [-wizard] Display a text file.

The "Type" command will display the contents of a Macintosh TEXT file that resides on a virtual disk. The file will be displayed on the user's terminal, one screen full at a time. If the "-wizard" command line option is specified then the data fork of any type Macintosh file will be displayed in hexadecimal.

VolumeInfo [-wizard] Display information about mounted volumes.

The "VolumeInfo" command will list the names and attributes of all currently mounted virtual disks. If the "-wizard" command line option is specified then additional technical information about each mounted volume will also be displayed. The following session shows usage of the volumeInfo command (user input is underlined):

MiniMac
[MiniMac version 1.0 Alpha test]

<none>: mount small.mvd
Small: mount new.mvd
NewDisk: volumeInfo

NewDisk: Flat file system, Contains 6 files, Unlocked,
Initialized on: 10 Jun 86 16:37:36 Tuesday
Modified on: 14 Jun 86 16:54:09 Saturday
14336 bytes used, 386048 bytes available.

Small: Flat file system, Contains 9 files, Unlocked,
Initialized on: 09 Jun 86 23:46:18 Monday
Modified on: 14 Jun 86 16:54:03 Saturday
132096 bytes used, 268288 bytes available.

NewDisk: quit

<Volume name>: Set current volume.

If a currently mounted volume name is specified as a command then the current volume will be set to the specified volume. Note that the colon is required as the last character of the volume name. The following are valid examples:

VolumeName:
"A volume with spaces:"

<Cr> Set current volume to next volume.

If no command is specified (just a null carriage return) then MiniMac will set the current volume to the next volume in its internal list of mounted volumes. All volumes may be cycled through by typing successive carriage returns.

Appendix A.

Cabling Connection

pcLINK operates over a 3 or 4 wire line (preferably 4), usually with a RS-232 25 pin connector on both sides. The Macintosh uses a 9 pin connector, so a 25-to-9 "patch" cable is needed. The 9 pin side of this patch cable is connected to the "phone" connection on the back of the Macintosh. The 25 pin side then connects to the 25 pin connector from the host computer, or directly to a modem.

The 25-to-9 patch cable should be wired as follows:

Pin Number

25 pin side	9 pin side	
1	1	
2	9	(or 5)
3	5	(or 9)
7	3&8	(3 shorted with 8 on connector)

As shown, two cases are possible: 2 & 3 on the 25 pin side connect to 9 & 5 (respectively) on the 9 pin side, or to 5 & 9. For direct connection to a host computer, the correct orientation depends on the way the host connects on the other side of the line. A little experimentation may be required. For most modems (e.g. Hayes), the 2-5/3-9 case is correct.

Macintosh Installation Notes

The Macintosh version of pcLINK requires no installation. Simply follow the steps outlined below to invoke the software. pcLINK does require 512K of memory to operate properly. Additionally, software on the host system must be properly installed to transfer text or binary files. Documentation on host system installation is available in a separate section.

As always, please don't hesitate to contact the Pacer Software technical support staff in the Westborough, Massachusetts office at (617) 898-3300 if you have any questions.

Step 1: Create backup copies of the pcLINK diskette. Please contact your systems administrator if you have any questions regarding this step.

Step 2: Turn on the power to the Macintosh.

Step 3: Insert the pcLINK diskette into the disk drive.

Step 4: Invoke the pcLINK software by double clicking on the pcLINK icon.

Step 5: Select a host configuration file. The configuration file contains the name of the terminal emulator and softkey definition files to load. Refer to the Configuration section of the pcLINK Macintosh manual for more information on changing terminal emulators.

Step 6: Invoke the "Config" pull-down menu option to edit the system and communication parameters as appropriate.

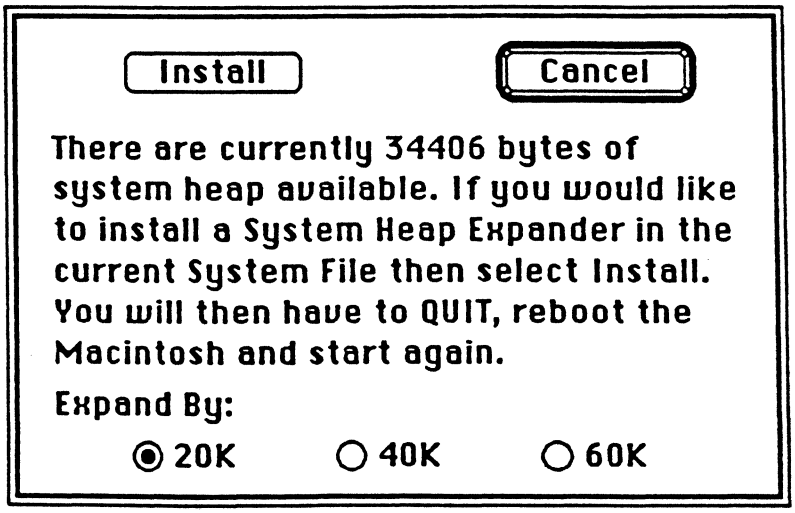
pcLINK immediately enters terminal emulation mode. Commands can be issued at the Macintosh in the same manner as a terminal connected to the host system. Refer to the pcLINK Macintosh manual for detailed information on using pcLINK.

This completes the pcLINK installation on the Macintosh computer!

C.
TCP/IP

pcLINK™ and **MountVDrive™** now support **TCP/IP** over AppleTalk to a Kinetics FastPath, AppleTalk to Ethernet Gateway. This is accomplished through the use of **TCPdriver**, a Macintosh device driver, and modifications to these two applications to interface with this driver.

The **TCPdriver** file should be in the same folder as the Pacer Applications or in the System folder. When opened it will be loaded into the Macintosh's System Heap. A Macintosh, as it is normally configured, allocates only 48K of memory for the system heap which it then fills with its own device drivers, system patches and global variables. This leaves very little room for large communication device drivers (the TCPdriver and Virtual Disk drivers each require about 20K of system heap). To solve this problem, the MountVDrive™ application allows the user to modify the system file, used to boot the Macintosh, so that it will allocate more memory to the System Heap. Selecting **Install System Heap Expander...** will bring up this dialog box:



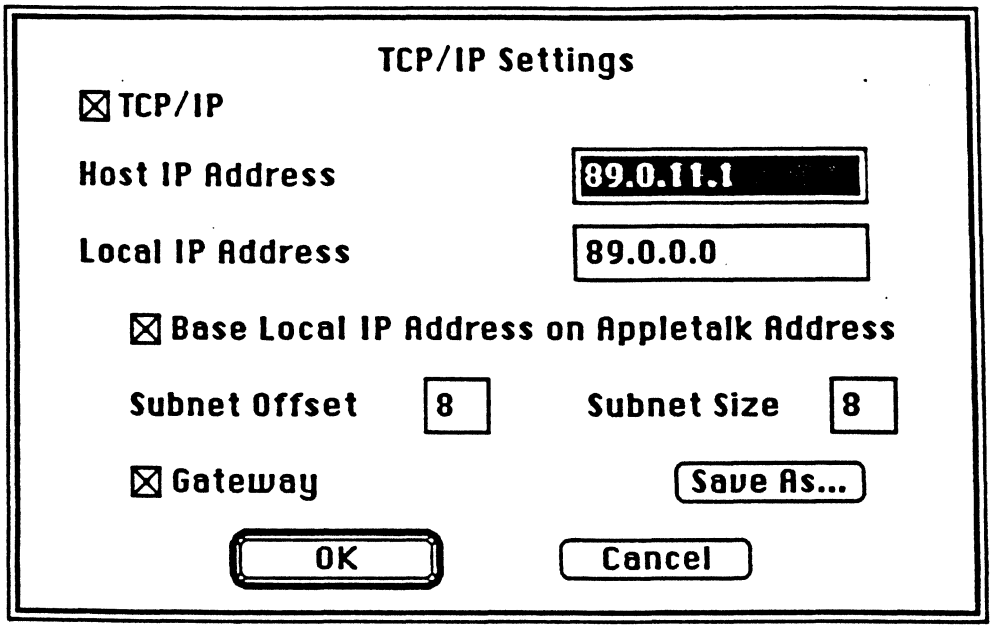
If you are using both the Virtual Disk and TCP driver, you should expand the system heap by at least 40K, and 60K would be even better. Remember, with a megabyte of memory this is not an expensive operation.

The Kinetics FastPath must be connected to your AppleTalk network and the Host computers Ethernet. The FastPath must then be downloaded with code to process the Appletalk to Ethernet packet conversion. A description of this process is given in the documentation that comes with the FastPath.

The TCPdriver is accessed from pcLINK™ by selecting the TCP/IP checkbox from the Communication Settings dialog box:

This checkbox will only be available if TCPdriver can be found by pcLINK™.

This will result in the TCP/IP Settings dialog box being displayed:



An IP address is a 32 bit value that is represented by four 8 bit values separated by decimal points. To connect to a Host with TCP/IP, select the appropriate IP addresses.

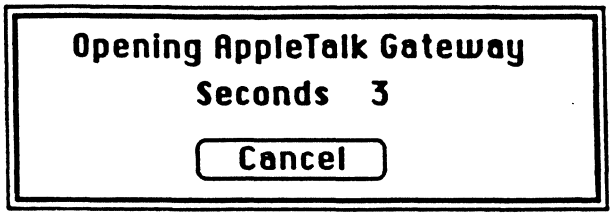
TCP/IP - Return to the Communication Settings dialog box.

Base Local IP Address on Appletalk Address - have the Macintosh ensure that your local IP address is unique within the local Appletalk network. If you select this you must also define the subnet offset (first bit of the subnet address) and the subnet size (length of the subnet address in bits).

Gateway - look for a Kinetics FastPath AppleTalk Gateway before attempting connection to host. With a FastPath the connections will fail until the Gateway is found.With an Ether SC direct ethernet connection this should not be set.

Save As... - save the current TCP settings to a configuration file without attempting to connect to the specified address. This is useful when creating a configuration file for some one else.

When you have specified all the required addresses, select **OK**. This will open the Appletalk driver and, if you selected Gateway, wait for the Kinetics Gateway to Open.



Opening the Gateway may take up to a minute the first time that the Kinetics FastPath is accessed after the Macintosh has been turned on. Any subsequent opening of a TCP connection will not display this dialog box. (Note : the DDP/IP download code for the FastPath seems to open the Gateway immediately.)

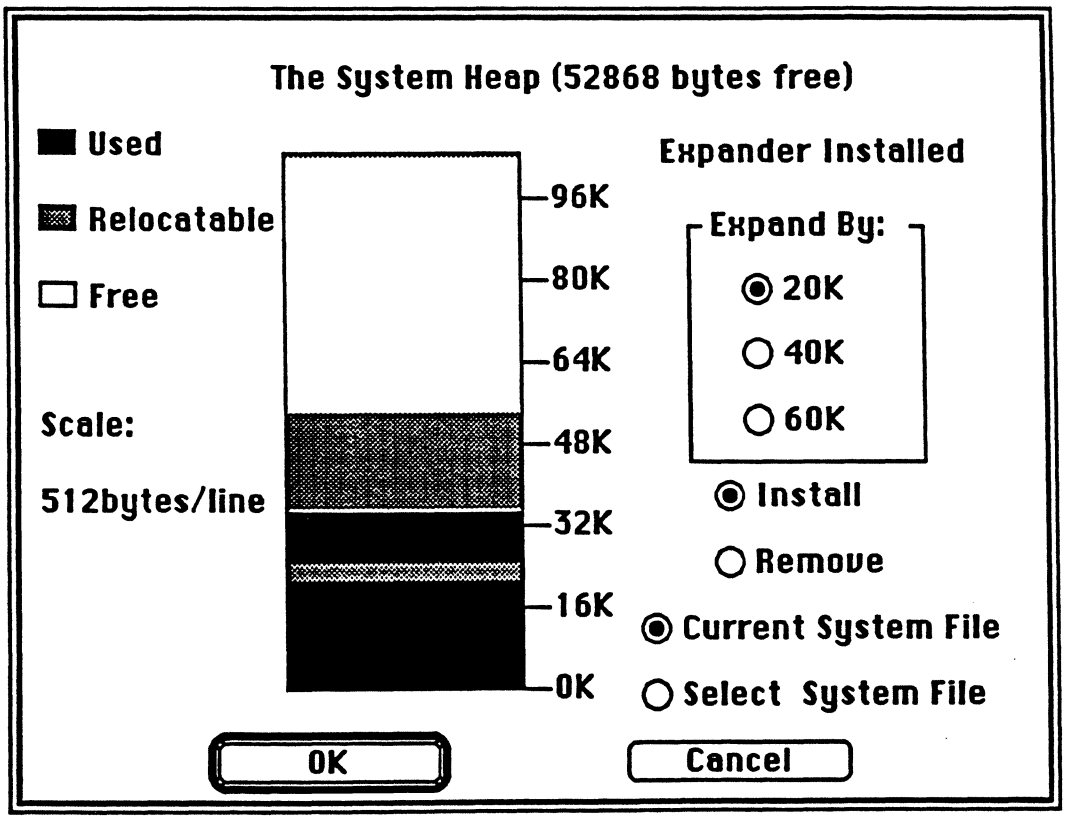
Once the Gateway is open, the TCPdriver will attempt to open a TCP connection with the specified Host. If a problem occurs, an Alert message will be displayed. If all goes well, the Host will prompt you to log in.

Once you have connected successfully you should save your TCP/IP settings in a Configuration File to avoid having to type them in again and in order to allow MountVDrive™ to use them for a Virtual Disk connection.

E.
HeapExpander™

The HeapExpander™ is a utility program designed to allocate more memory for the system heap. pcLINK™ and MountVDrive™ load Virtual Disk, TCP/IP and PNP drivers into the Macintosh's System Heap. A Macintosh, as it is normally configured, allocates only 48K of memory for the system heap which it then fills with its own device drivers, system patches and global variables. This leaves very little room for large communication device drivers. To solve this problem, you must expand the System Heap. The HeapExpander™ installs a Heap Expander into the System file. The Heap Expander is invoked at startup and expands the system heap by a predetermined amount.

After launching HeapExpander™ the following dialog box will be displayed:



Several pieces of information are presented here. The amount of System Heap available is displayed as well as a visual representation of the System Heap. If a Heap

Expander is already installed in the current System file then the message Expander Installed is displayed.

To install the Heap Expander, select Install and the amount that you wish to expand the System Heap by. It can be 20, 40 or 60 kilobytes. If you will be running the Virtual Disks, TCP/IP or PNP drivers then you should select 20K. If you are running any combination of these drivers, i.e. Virtual Disks and TCP/IP, then you should expand by more. If you haven't allocated enough System Heap you will be notified by the various applications. On a Macintosh Plus with one megabyte of memory, 60 kilobytes is inconsequential and is the preferred choice.

To remove the Heap Expander, select Remove.

Specify whether you wish to install the Heap Expander in the current System file or in a System file of your choice. If you choose to select a System file, you will be prompted to choose one when you select OK.

Once you have set your preferences select OK or the return key. The Heap Expander will be installed or removed from the current System file or one of your choice and you will be told that, to experience the results of these changes, you should Shut Down (reboot the Mac) when you return to the desktop.

Appendix B
SUPPORTED ESCAPE CODE SEQUENCES

Supported Escape Code Sequences

There are a number of escape sequences recognized by all of the terminal emulators that provide a level of integration between host applications and the PC. As examples, host applications may cause the emulator to go off and run DOS programs, to write application specified information to the terminal's status line, to execute a pcLINK command file, and more.

A user within ALL-IN-ONE could have a menu pick to run pcLINK's file transfer so that the user would never have to see the DCL prompt. For actual sample command files, refer to the file:

```
SYS$SYSROOT:[PCLINK]README.HLP
```

on the VAX.

Here is a list of the actual sequences recognized by the emulators and their meanings. Wherever filenames are given in '<', the '<' and '>' are not to be included in the escape sequence.

Supported Escape Code Sequences-Generic

Escape Code Sequence	Meaning
<ESC><ESC>T	Go to the File Transfer Menu.
<ESC><ESC>S	Go to the Host Spool Menu.
<ESC><ESC>P	Go to the PC Spool Menu.
<ESC><ESC>C<command_file><ESC>\ file.	Run the specified pcLINK command
<ESC><ESC>Fdm<host_file>#<pc_file><ESC>\	Do the specified file transfer, where the literal "F" stands for file transfer; d is the direction, either an "R" for receive or "S" for send (the arguments are from the PC's perspective); m is the mode, either a "T" for text or a "B" for binary; "host_file" is the host filename; and "pc_file" is the PC filename.
<ESC><ESC>B	Exit pcLINK and return to DOS.
<ESC><ESC>+<pc_file><ESC>\	Send the host output to the specified PC file and to the screen.
<ESC><ESC>-<pc_file><ESC>\	Send the host output ONLY to the specified PC file.
<ESC><ESC>R	Reset the host output to just the screen.
<ESC><ESC>E	Leave the emulator and return to pcLINK main menu.
<ESC><ESC>L<text><ESC>\	Send the supplied text to the terminal's status line.
<ESC><ESC>X	Clear the current terminal's status line.
<ESC><ESC>D<dos_program><ESC>\	Run the given DOS program immediately.
<ESC><ESC>W<dos_program><ESC>\	Run the given DOS program, first prompting the user to any changes he wants to make to the program's arguments.
<ESC><ESC>V	Escape to a DOS shell (with the emulator suspended).

The following table of supported escape code sequences applies to PRIME hosts only.

Status Key

Blank indicates that the function is currently supported.

W indicates that the function will be supported in the future.

N means that the function is not supported.

Supported Escape Code Sequences - PST100/PT200

Status	Sequence	Function
N	ESC $ 0	Set G0 ASCII (SG0A)
N	ESC $ 1	Set G1 ASCII (SG1A)
N	ESC $ 2	Set G0 Alternate (SG0E)
N	ESC $ 3	Set G1 Alternate (SG1E)
	ESC $ 4	Start DSC Data (SDD)
	ESC $ 5	End DSC Data (EDD)
W	ESC $ 6	Enter Graphics Operation
	ESC $ A	Cursor Relative Home (CRH)
	ESC $ B	Cursor Absolute Home (CAH)
W	ESC $ C	Field Entry Check (FEC)
	ESC $ E	Blank Screen (BSCN)
	ESC $ F	Soft Keyboard Lock (SKL)
	ESC $ G	Soft Keyboard Unlock (SKU)
	ESC $ H	Escape Key Disable (EKD)
	ESC $ I	Escape Key Enable (EKE)
	ESC $ J	Reset Modified Tags (RMT)
	ESC $ K	Clear (reset) Selected Area (CSA)
	ESC $ L	Start Logical Attributes (SLA)
	ESC $ M	End Logical Attributes (ELA)
	ESC $ N	Insert Cursor (INC)
	ESC $ O	Save Cursor & Attributes (SCA)
	ESC $ P	Unblank Screen (UBS)
	ESC $ Q	Restore Cursor & Attributes (RCA)
	ESC $ R	Reset Inhibit Cursor (RIC)
	ESC $ S	Set Inhibit Cursor (SIC)
	ESC $ T	System Line Reset (SLR)
	ESC $ U	System Line Display (SLD)
	ESC $ V	System Line Set (SLS)
W	ESC $ W	Scroll Inhibit Reset (SIR)
W	ESC $ X	Scroll Inhibit Set (SIS)
N	ESC $ Z	Display Revision (DIS)
	ESC $ a	Page Up (PU)
	ESC $ b	Page Down (PD)
W	ESC $ c	Host Notification of Format Modification
	ESC $ d	DUP Function
	ESC $ e	Cursor Select Functionality
W	ESC $ f	Cursor Select

	ESC 0 <x> <y>	Compressed Cursor Position (CCP)
	ESC 1 <x> <y> <l> <e>	Compressed Logical Attributes (CLAT)
	ESC 2 <x> <y> <v> <e>	Compressed Visual Attributes (CVAT)
	ESC 3 <l>	Compressed Logical Area (CLAR)
	ESC 4 <v>	Compressed Visual Area (CVAR)
	ESC 5	Send Block Data (SBD)
	ESC 6	Dump Block Data (DBD)
	ESC 8 <a>	Set Attribute (SA)
	ESC :	Display Error Message (DEM)
	ESC ;	Read Cursor Character (RCC)
N	ESC < data ESC \	Load Keyboard Tables (LKT)
	ESC =	Set Page Dump (SPD)
	ESC ?	Clear Screen (CS)
	ESC D	Index (IND)
	ESC E	Next Line (NEL)
	ESC F	Start of Selected Area (SSA)
	ESC G	End of Selected Area (ESA)
	ESC H	Horizontal Tabulation Set (HTS)
	ESC M	Reverse Index (RI)
N	ESC O f	Single Shift Three (SS3)
	ESC P	Device Control String (DCS)
	ESC S	Set Transmit State (STS)
	ESC V	Start of Protected Area (SPA)
	ESC W	End of Protected Area (EPA)
	ESC [	Control Sequence Introducer (CSI)
	ESC [4 h	Insertion-Replacement Mode Set
	ESC [4 l	Insertion-Replacement Mode Reset
	ESC [6 h	Erasure Mode Set
	ESC [6 l	Erasure Mode Reset
	ESC [12 h	Send-Receive Mode Set
	ESC [12 l	Send-Receive Mode Reset
	ESC [17 h	Selected Area Transfer Mode Set
	ESC [17 l	Selected Area Transfer Mode Reset
	ESC [20 h	Line Feed New Line Mode Set
	ESC [20 l	Line Feed New Line Mode Reset
	ESC [> 1 h	Auto Line Feed Mode Set
	ESC [> 1 l	Auto Line Feed Mode Reset
	ESC [> 2 h	Block Mode Set
	ESC [> 2 h	Character/Block Mode Set
	ESC [> 2 l	Character/Block Mode Reset
	ESC [> 3 h	Logical Attribute Mode Set
	ESC [> 3 l	Logical Attribute Mode Reset
	ESC [> 4 h	Page/Line Mode Set
	ESC [> 4 l	Page/Line Mode Reset
N	ESC [> 5 h	Hard/Soft Scroll Mode Set
N	ESC [> 5 l	Hard/Soft Scroll Mode Reset

	ESC [> 6 h	Unprotected/Modified Mode Set
	ESC [> 6 l	Unprotected/Modified Mode Reset
	ESC [> 7 h	Null/Space Mode Set
	ESC [> 7 l	Null/Space Mode Reset
	ESC [> 8 h	Screen Wrap Mode Set
	ESC [> 8 l	Screen Wrap Mode Reset
	ESC [> 9 h	Line Truncate Mode Set
	ESC [> 9 l	Line Truncate Mode Reset
W	ESC [> 10 h	Numeric/PF Keypad Mode Set
W	ESC [> 10 l	Numeric Keypad Mode Reset
	ESC [> 11 h	One/Two Page Boundary Mode Set
	ESC [> 11 l	One/Two Page Boundary Mode Reset
	ESC [> 12 h	Visual Attribute Lock Mode Set
	ESC [> 12 l	Visual Attribute Lock Mode Reset
	ESC [> 13 h	Local Cursor Action Mode Set
	ESC [> 13 l	Local Cursor Action Mode Reset
N	ESC [> 14 h	Selective Data Trap Mode Set
N	ESC [> 14 l	Selective Data Trap Mode Reset
N	ESC [> 15 h	Transparent Data Mode Set
N	ESC [> 15 l	Transparent Data Mode Reset
W	ESC [> 16 h	Host Notification Mode Set
W	ESC [> 16 l	Host Notification Mode Reset
	ESC [> 17 h	Sends Tabs Mode Set
	ESC [> 17 l	Send Tabs Mode Reset
	ESC [> 18 h	Function Termination Mode Reset
	ESC [> 18 l	Function Termination Mode Set
	ESC [> 19 h	Soft Lock Option Mode Set
	ESC [> 19 l	Soft Lock Option Mode Reset
	ESC [> 20 h	DSC Mode Set
	ESC [> 20 l	DSC Mode Reset
	ESC [> 21 h	E2 Mode Set
	ESC [> 21 l	E2 Mode Reset
N	ESC [> 22 h	Dead Keys Enable Mode Set
N	ESC [> 22 l	Dead Keys Enable Mode Reset
	ESC [Pn @	Insert Character (ICH)
	ESC [Pn A	Cursor Up (CUU)
	ESC [Pn B	Cursor Down (CUD)
	ESC [Pn C	Cursor Forward (CUF)
	ESC [Pn D	Cursor Backward (CUB)
	ESC [Pn E	Cursor Next Line (CN)
	ESC [Pn F	Cursor Preceding Line (CPL)
	ESC [Pn L	Cursor Horizontal Absolute (CHA)
	ESC [Pn ; Pn H	Cursor Position (CUP)
	ESC [Pn I	Cursor Horizontal Tabulation (CHT)
	ESC [Ps J	Erase in Display (ED)

	ESC [Ps K	Erase in Line (EL)
	ESC [Pn L	Insert Line (IL)
	ESC [Pn M	Delete Line (DL)
N	ESC [Pn N	Set Display Size (SDS)
	ESC [Ps O	Erase in Area (EA)
	ESC [Pn P	Delete Character (DCH)
	ESC [Ps Q	Select Editing Extent Mode (SEM)
	ESC [Pn ; Pn R	Cursor Position Report (CPR)
	ESC [Pn S	Scroll Up (SU)
	ESC [Pn T	Scroll Down (SD)
	ESC [Pn U	Next Page (NP)
	ESC [Pn V	Preceding Page (PP)
	ESC [Ps W	Cursor Tabulation Control (CTC)
	ESC [Pn X	Erase Character (ECH)
	ESC [Pn Y	Set Row Number (SRN)
	ESC [Pn Z	Cursor Backward Tabulation (CBT)
N	ESC [Ps a	Set Language (SL)
	ESC [Pn b	Repeat (REP)
	ESC [Pn c	Device Attributes (DA)
	ESC [Pn d	Vertical Position Absolute (VPA)
N	ESC [Pn ; Pn ; Pn e	Unlock Columns (LC)
	ESC [Pn ; Pn f	Horizontal and Vertical Position (HVP)
N	ESC [Pn ; Pn ; Pn g	Lock Columns (LC)
	ESC [Ps h	Set Mode (SM)
	ESC [Ps i	Media Copy (MC)
	ESC [Pn j	Scroll Back (SB)
	ESC [Pn k	Scroll Forward (SF)
	ESC [Ps l	Reset Mode (RM)
	ESC [Ps n	Device Status Report (DSR)
	ESC [Ps m	Select Graphic Rendition (SGR)
	ESC [Ps o	Define Area Qualification (DAQ)
	ESC [Ps p	Change Visual Attribute of Area (CVA)
	ESC [Pn q	Change Visual Attribute of Char. (CVC)
	ESC [Ps r	Change Visual Attribute of Display (CVD)
	ESC [Pn ; Pn s	Move Memory Pointer (MMP)
	ESC [Ps t	Change Visual Attribute of Line (CVL)
	ESC [Pn ; Pn u	Lock Lines (LL)
	ESC [Ps v	Define Logical Attributes (DLA)
	ESC [Pn ; Pn w	Repeat to Address (RPA)
	ESC [Pn ; Pn x	Erase Unprotected to Address (EUA)
	ESC [Pn ; Pn y	Unlock Lines (UL)
	ESC [Ps z	Program Tab (PT)
W	ESC [Ps {	Select Monochrome Color (color option)
	ESC \	String Terminator (ST)
W	ESC]	Operating System Command (OSC)
W	ESC _	Application Program Command (APC)
	ESC `	Disable Manual Input (DMI)
	ESC b	Enable Manual Input (EMI)
	ESC c	Reset to Initial State (RIS)