



LETTER REPORT

AD-A262 401



NUMBER 364

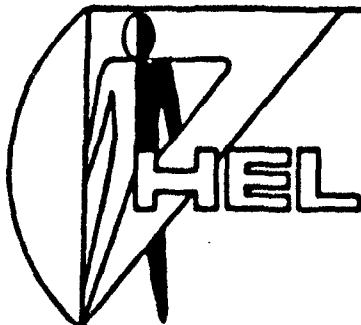
DEVELOPING A HYPERCARD-UNIX INTERFACE FOR ELECTRONIC MAIL TRANSFER

Troy Kelley

June 1992

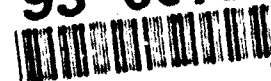
Approved for public release;
distribution unlimited.

DTIC
ELECTE
APR 01 1993
S E D



Field Support Division

93-06720



11501

98 3 31 159
US ARMY HUMAN ENGINEERING LABORATORY

20000920284

Reproduced From
Best Available Copy

DTIC QUALITY INSPECTED 4

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Apple® is a registered trademark of Apple Computer, Inc.

AppleTalk™ is a trademark of Apple Computer, Inc.

Hypercard™ is a trademark of Apple Computer, Inc.

Macintosh™ is a trademark of Apple Computer, Inc.

ACKNOWLEDGMENTS

The external commands stripLastReturn, stripLast, gimelast, CntrlZ, CgRet, xStrip, xgetscreen were written by Troy Kelley © 1991.

The external commands DoList, PopUpMenu, FileLenght, FileName were taken from Developer Stack by Steve Drazga © 1988 AnalytX* Incorporated.

The external commands sendSPortBytes, configureSPort, XModem, setSPortBufferSize, sendSPort, killSPort, closeSPort, breakSPort, recvBytes, SPortConfiguration, SPortVersion, SPortBufferSize, sendSPortDone, recvUpTo, recvChars were taken from the Serial Port XCMDs Stack and the MacTCP ToolKit Stack and were all written by Harry Chesely for Apple Computer Inc.

The external commands FindInField and Password were taken from the Dartmouth XCMDs stack © 1989 by the Trustees of Dartmouth College.

The external command FolderName was taken from the FolderTools stack as was written by Eric Carlson & Anup Murarka.

The external command SelectLine was taken from the SelectLine stack and was written by Phil Beisel for Apple Computer, Inc. © 1989.

Much of the login routine for HEL's Hypercard UNIX E-mail system (MacTCP version) was taken from HyperUnix V.1.7.3 by Greg Anderson. I would like to thank Greg for his help, without which, this stack would not have been possible.

CONTENTS

EXECUTIVE SUMMARY.....	3
PROCEDURES.....	5
Setting up the System.....	5
Design of the System.....	5
Executing the Program.....	6
CONCLUSIONS.....	15
BIBLIOGRAPHY.....	17
APPENDICES	
A. Card Names.....	19
B. Source Code - MactCP Version.....	25
C. Source Code - Modem Version.....	70
FIGURES	
1. The User Preferences Card - MactCP version.....	7
2. The User Preferences Card - Modem version.....	8
3. The Interface card.....	9
4. The File View Card.....	11
5. Mail Card.....	11
6. Themessage card.....	13
7. The Groups Card.....	14

ACKNOWLEDGMENTS

The external commands stripLastReturn, stripLast, gimelast, CntrlZ, CgRet, xStrip, xgetscreen were written by Troy Kelley © 1991.

The external commands DoList, PopUpMenu, FileLenght, FileName were taken from Developer Stack by Steve Drazga © 1988 AnalytX* Incorporated.

The external commands sendSPortBytes, configureSPort, XModem, setSPortBufferSize, sendSPort, killSPort, closeSPort, breakSPort, recvBytes, SPortConfiguration, SPortVersion, SPortBufferSize, sendSPortDone, recvUpTo, recvChars were taken from the Serial Port XCMDs Stack and the MacTCP ToolKit Stack and were all written by Harry Chesely for Apple Computer Inc.

The external commands FindInField and Password were taken from the Dartmouth XCMDs stack © 1989 by the Trustees of Dartmouth College.

The external command FolderName was taken from the FolderTools stack as was written by Eric Carlson & Anup Murarka.

The external command SelectLine was taken from the SelectLine stack and was written by Phil Beisel for Apple Computer, Inc. © 1989.

Much of the login routine for HEL's Hypercard UNIX E-mail system (MacTCP version) was taken from HyperUnix V.1.7.3 by Greg Anderson. I would like to thank Greg for his help, without which, this stack would not have been possible.

CONTENTS

EXECUTIVE SUMMARY.....	3
PROCEDURES.....	5
Setting up the System.....	5
Design of the System.....	5
Executing the Program.....	6
CONCLUSIONS.....	15
BIBLIOGRAPHY.....	17
APPENDICES	
A. Card Names.....	19
B. Source Code - MacTCP Version.....	25
C. Source Code - Modem Version.....	70
FIGURES	
1. The User Preferences Card - MacTCP version.....	7
2. The User Preferences Card - Modem version.....	8
3. The Interface card.....	9
4. The File View Card.....	11
5. Mail Card.....	11
6. Themessage card.....	13
7. The Groups Card.....	14

INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Hypercard-UNIX electronic mail system (HUES) was programmed using Hypercard version 1.2.5 on a Macintosh computer system. HUES was created as a user-friendly way for U.S. Army Human Engineering Laboratory (HEL) employees to transfer electronic mail, as well as files of various types and formats, to anyone connected to the Internet. (Note. The Internet is a collection of many networks that service universities, private industry, and the Department of Defense.) The system is essentially an interface or shell, which receives information from another computer system (in this case, a UNIX system) and displays the information in a format that allows easier user interaction with the UNIX system. The system receives information from another computer by using a group of transmission control protocol/Internet protocol (TCP/IP) code resources developed by Apple Computer, Inc., for use with Hypercard and Macintosh computer systems. This transfer of information allows users to take advantage of the Macintosh's point-and-click user interface to perform desired procedures, instead of having to type UNIX commands.

The system was designed to be expandable, with two different means of establishing a connection to a host computer (i.e., modem or TCP/IP). The system also allows direct manipulation of UNIX files, (i.e., deleting, viewing) and the option of directly sending UNIX commands.

This report assumes some basic understanding of Hypercard, MactCP, file transfer protocol (FTP), UNIX, and electronic mail. The reader may want to refer to other documentation concerning these subjects for a more complete understanding of the issues involved with this system. Some suggested reading is given in the bibliography.

BLANK

DEVELOPING A HYPERCARD-UNIX INTERFACE FOR ELECTRONIC MAIL TRANSFER

PROCEDURES

Setting up the System

1. The U.S. Army Human Engineering Laboratory (HEL) Hypercard-UNIX E-mail system (HUES) is a Hypercard stack that runs on a Macintosh Plus computer or later machine running system 6.0.x or greater. As of this writing, it has not been tested with system 7.0.

2. The system requires MacTCP, which must be purchased from Apple Computer, Inc., to be properly configured if a transmission control protocol/Internet protocol (TCP/IP) protocol is to be used. If a modem type access is to be used instead of MacTCP, HEL's Hypercard-UNIX E-mail system-modem access can be used instead. Much of the MacTCP documentation is poor, but there is one important thing to remember during the configuration. If an AppleTalk configuration is present, as opposed to an Ethernet type connection, configure MacTCP in the server option (not dynamic or static) and set the zone to the same zone as the gateway or bridge the computer goes through to get to the host computer, not the zone where the computer resides.

Design of the System

HUES is designed to be an interface for transferring mail and files to other users on the Internet system. The stack is customized to the current configuration of the HEL4 UNIX system as of this date. Consequently, the system will have difficulty interfacing with other systems unless changes in the code are made.

The MacTCP version of the system uses the transmission control protocol external command documents (TCP XCMD Docs) and TCP XCMD Example stacks written and distributed by Apple Computer for use with Macintosh computers. The MacTCP version also uses MacTCP, which is a control panel device for Macintosh computers. The modem version uses the Serial Port XCMDs stack written and distributed by Apple Computer to communicate with the host computer. The stack sends information, using these external commands, to the UNIX machine and waits for responses, at which point, it takes the data received and places them in the appropriate data fields.

The stack consists of a main interface screen, which allows users to actually see the files in their current directory on the UNIX host. This main interface screen consists of one background field that receives all the data coming from the UNIX host. As the user progresses to other interface screens, this field is re-sized and adjusted for different procedures throughout the stack.

The file sending and receiving button on the main interface screen allows the user to send and receive files to the host computer. The execution of this procedure opens up a separate file transfer protocol (FTP) connection, while keeping the existing Telenet connection open. The transfer is then made using various external commands to convert the file to or from binary to hexadecimal (BinHex) format, for acceptance by the host computer.

The file reading card contains the single background field and an interface for displaying files on the host computer. Commands are sent by the

user to the host computer to display a file in a specific directory, and the information is then sent back to HUES for viewing. Any actions that the user takes upon the text of the file in the window of HUES has no effect on the actual file on the host system, so essentially, HUES is displaying a copy of the file on the host machine.

The send mail card is simply an area in which users can type their electronic mail messages into a Hypercard field. The data are then formatted with carriage returns using an external command and sent to the host computer as electronic mail. This is done because the Hypercard field in which the user is typing the electronic mail to be sent contains no carriage returns, and if this were to be sent to the host computer, it would appear as one long line, not a paragraph. The host computer is actually idle while the user types an electronic mail message, so a script that executes at regular intervals keeps the user logged into the host computer so the user will not be booted out. The modem version does not include this feature because of differences in the external commands used for the two versions, and the user must click on an Update button to let the host computer know the user is still there but merely typing a message. Once the user clicks the send mail button, HUES begins to send the necessary commands to the host computer to send the electronic mail message. HUES then waits to ensure that the mail has been sent successfully. If an error was encountered, it is displayed along with information relating to any possible problems that may be preventing the message from being delivered.

The read mail card displays each mail message with the opportunity to save or delete each message. The system uses the same commands that would be used to interface with a UNIX machine on the command line level. Messages can be saved into a file format for transferring onto the Macintosh if the user desires. If the user exits this screen, the messages are saved in the mbox, which can be viewed at a later time by selecting the mbox button.

Executing the Program

1. If HUES is used with a MacTCP type connection, when the stack is first opened, a prompt "LOGIN NOW?" will immediately appear. If the login name in the login field is correct, click "OK." This is to accelerate the login process. If the name in the login field is incorrect, click in the "Login Name" field and type the user's login name; then click on the login button. Type the user's password in the dialog box that appears after the login prompt.

If the information entered here is correct, the stack will display the main Hypercard-UNIX interface screen. If an error occurs at this point, problems may exist with the user's MacTCP configuration, the host computer might be down, or the password may have been entered incorrectly.

This card has three buttons and one field (see Figure 1).

- a. The "Login" button executes the login procedure using the name listed in the login field.

- b. The "Show Last Login" button displays information about when the last login was recorded by the host computer using the login name listed.

- c. The "Question" button displays a help field containing information about this stack.

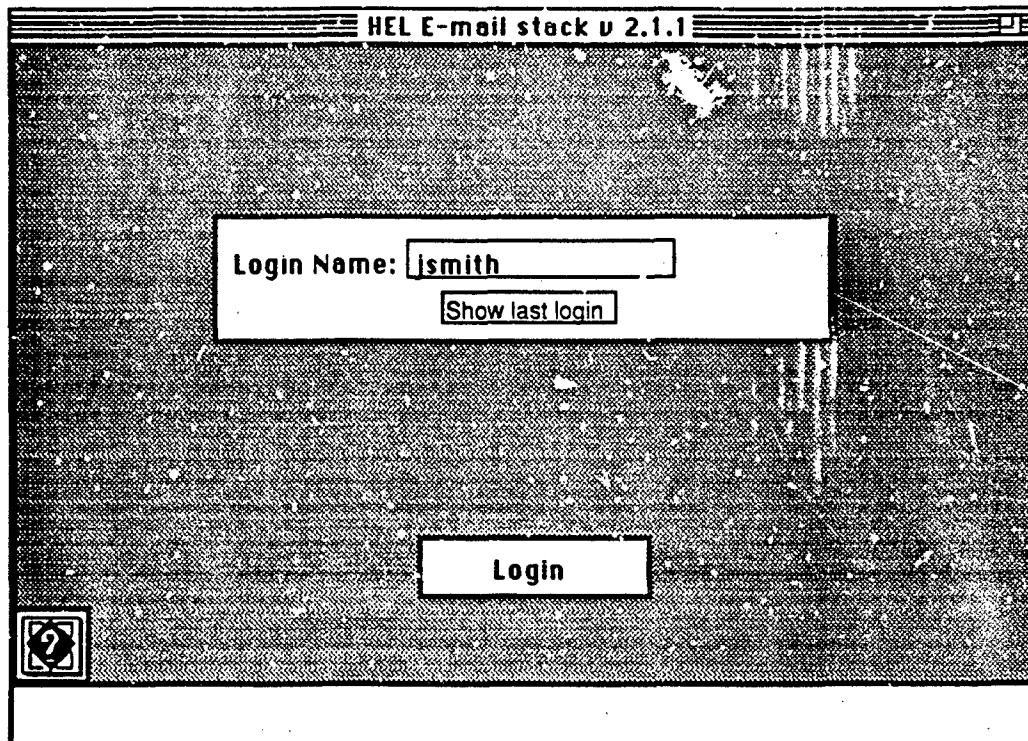


Figure 1. The user preferences card--MacTCP version.

d. The "Login Name" field contains the name that will be sent to the host computer during the login procedure.

2. If HUES is used with a modem type connection, enter the dialing parameters for the modem to establish a connection with the host computer. Simply enter the phone number for the HEL4 computer and the user's login name as shown below. If it is necessary to dial a "9" and the area code "410" first to reach the HEL4 computer, enter that information as well. After the user clicks on the dial button, the modem establishes a connection, and the user can click on the login button to send the login procedure to the host computer. The stack expects the user modem to be attached to the modem port, not the printer port. This card has three fields and five buttons.

Buttons

a. The "Advanced Users Only" button on this card allows the user to change the baud rate and to strip control characters, but the button is already pre-set for the HEL4 computer, so changes should not be necessary.

b. The "Dial" button executes a dial procedure using the information contained on this card. Input the phone number exactly as it would be dialed from a phone; in other words, if a prefix or an extension is needed, include that in the phone number.

c. The "Login" button is not visible in Figure 2. It appears only after the "Dial" button has been selected. Wait until the host computer is fully connected before clicking this button.

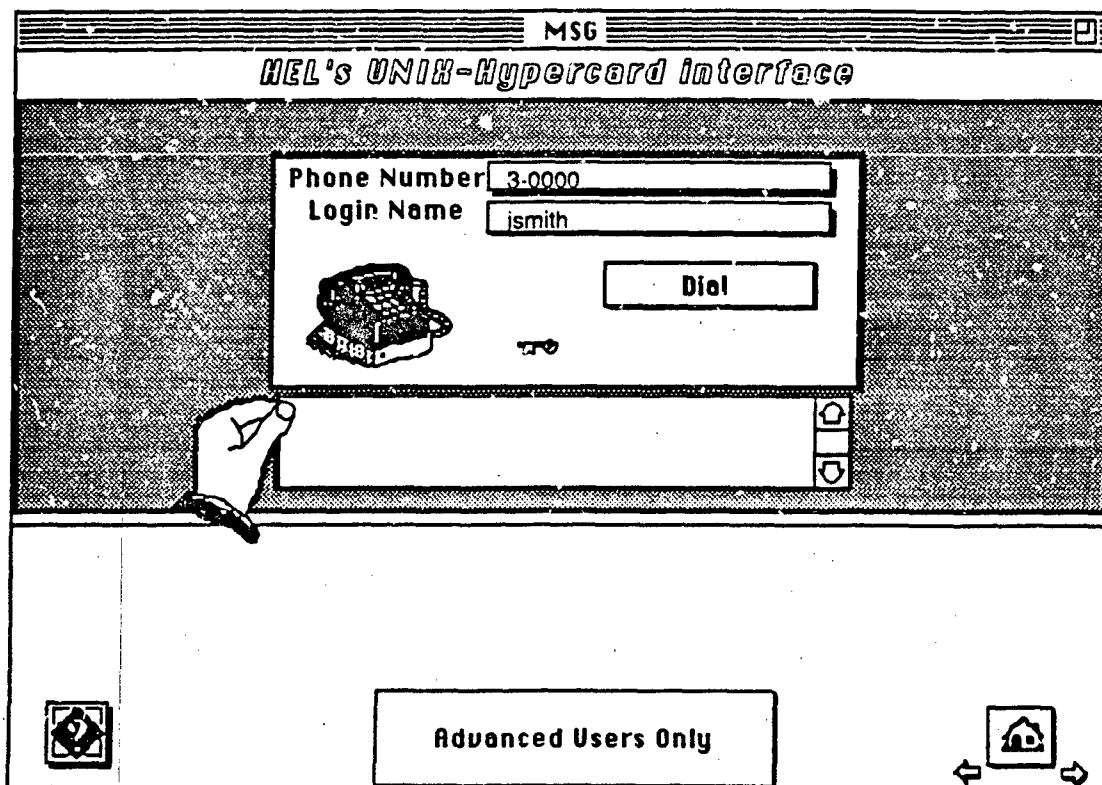


Figure 2. The user preferences card--modem version.

- d. The "Home" button goes back to the user's home stack.
- e. The "Question" button displays a help field.

Fields

a. The "Connect" field, which looks as though it is being held by a hand, displays information about the connection process. When a connection is made, the word "connect" will appear in this field; this signals the user to click on the "login" button.

b. The "Phone number" field contains the phone number that will be sent to the modem during the connection procedure.

c. The "Login name" field contains the name that will be sent to the host computer during the login sequence.

Both the MacTCP version and the modem version next will display the interface card if a connection is made (see Figure 3). There are some minor differences between the two versions' interface cards. The MacTCP version displays a "MacTCP connection to UNIX host established" in a field in

the top right-hand corner if a connection has been made. The modem version displays a receiving data field in the top left and corner of the card. This field displays the amount of information it is receiving from the host computer. The modem version tells only if the stack is receiving data from the UNIX host, not whether a connection has been made.

3. If a connection has been established with the host computer, this screen will appear with a list of the files contained within the directory in the field on the left of the screen. At this point, users can send and read their mail, send and receive files from the host computer, change and list directories, read and print UNIX files, send any type of UNIX command, and of course, log out of the UNIX system. This card contains ten buttons and one field.

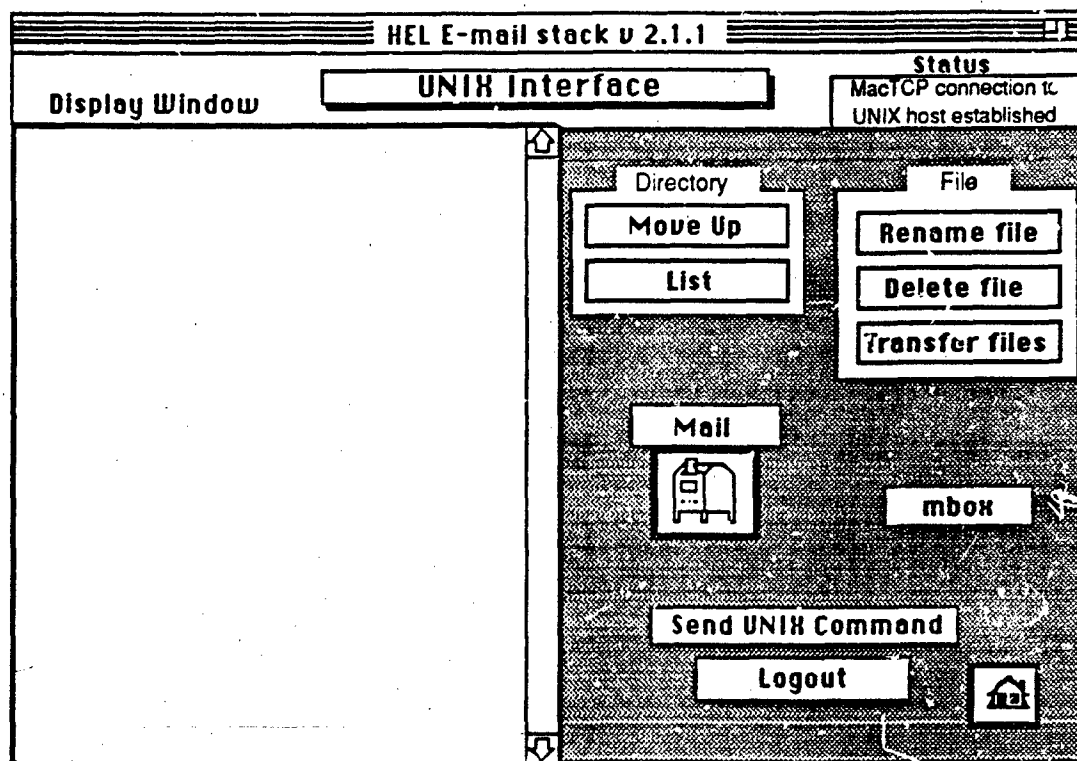


Figure 3. The interface card.

Buttons

a. The "Move Up" button allows the user to change directories on the host computer by going to the directory above the current directory.

b. The "List" button lists the files contained within the current directory to the field at the left of the screen.

c. The "Rename File" button allows the user to rename any file within the current directory.

d. The "Delete File" button allows users to delete any file that they have permission to delete within the current directory.

e. The "Transfer Files" button allows the user to send or receive any file to or from the host computer.

f. The "Mail" button gives the user the option of either sending or checking electronic mail.

g. The "mbox" button allows the user to check the mail that has been saved to the mbox file on the host computer.

h. The "Send UNIX Command" button presents the user with a dialog box to send a specific command to the host computer. The results from the command are then displayed in the field on the left of the card.

i. The "Logout" button will disconnect the user from the host computer. After users have disconnected from the host computer, they are then free to go to the home stack.

j. The "Home" button takes the user to the home card.

Fields

a. The "Display Window" is a field containing the files that reside in the current directory on the host computer. If the user wishes to display a file on the host computer, simply clicking on the file name in this field will display the file in a field on another screen (see Figure 4).

The file view card displays text from the host computer in its center field. The file name is listed in the field at the very top of the card. This card has three buttons and two fields.

Buttons

a. The "UNIX interface" button will take the user to the main interface screen.

b. The "Show More" button will display more text than is currently being displayed in the window. This is because the UNIX machine receives the "more" command from this stack to display the text file.

c. The "print" button will print the field.

d. The "file name" button will display the file currently being viewed on the host computer.

Fields

a. The "Window" field displays the text file currently being viewed on the host computer.

The mail card (see Figure 5) allows users to send electronic mail. This card contains four buttons and four cards. HEL's Hypercard-UNIX E-mail system-modem version may contain an "Update" button (not shown in Figure 4). While a person is typing an E-mail message, the host computer is essentially idle; therefore, after a certain amount of time, the host computer may boot

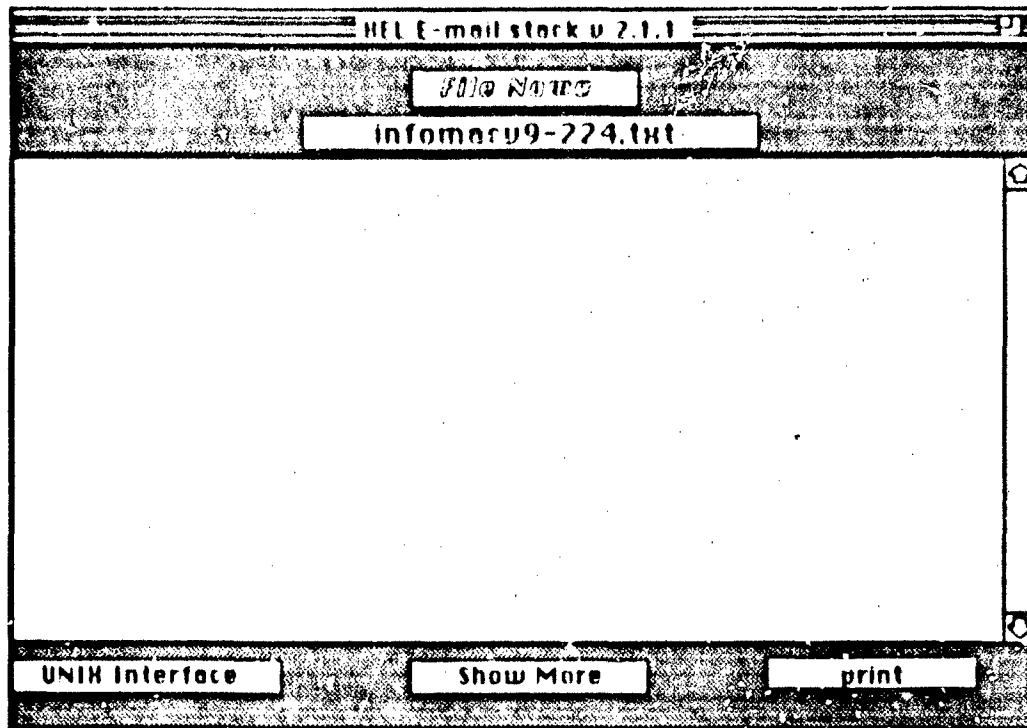


Figure 4. The file view card.

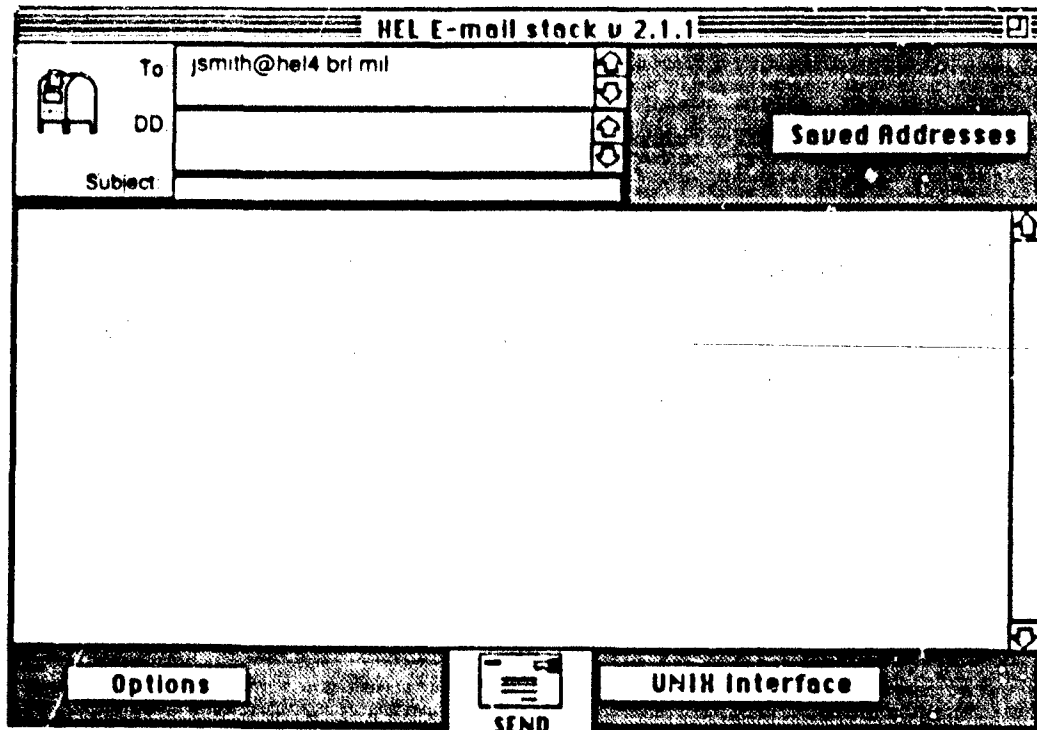


Figure 5. Mail card.

the user out. This update button sends a command to the host computer letting the computer know the user is still connected and not idle. A warning field for the modem version displays this information and instructs the user whenever this card is opened. This is only for the modem version of the stack, not for the MacTCP version.

Buttons

a. The "Options" button is a pop-up menu containing six options for the electronic mail user.

1. The "Groups" selection in the pop-up menu allows the user to select from a list of pre-defined groups of people to whom a message can be sent.

2. The "Edit Groups" selection in the pop-up menu allows users to define their own list of individuals to whom messages can be sent.

3. The "Include File" selection in the pop-up menu allows the user to include any file on the host computer as part of an electronic mail message. Files that reside on the Macintosh, which the user needs to send as part of a message, must first be transferred to the host computer by using the transfer file button on the interface screen.

4. The "Move Up Directory" selection in the pop-up menu allows the user to change directories to locate a file to send.

5. The "Clear field" selection in the pop-up menu clears the previous message from the card so that a new message may be entered.

b. The "Send" button sends the message to the address(es) in the "to:" field.

c. The "UNIX Interface" button takes the user to the "UNIX interface" card.

d. The "Saved Addresses" button allows users to choose from a list of previously saved addresses.

Fields

a. The "cc:" field allows carbon copies to be mailed to other users.

b. The "to:" field contains the address of the person who will receive the electronic mail.

c. The "Subject" field contains the subject header for the message.

The card in Figure 6 displays electronic mail in the main field, which has been received by the host computer. This card contains eight buttons and two fields.

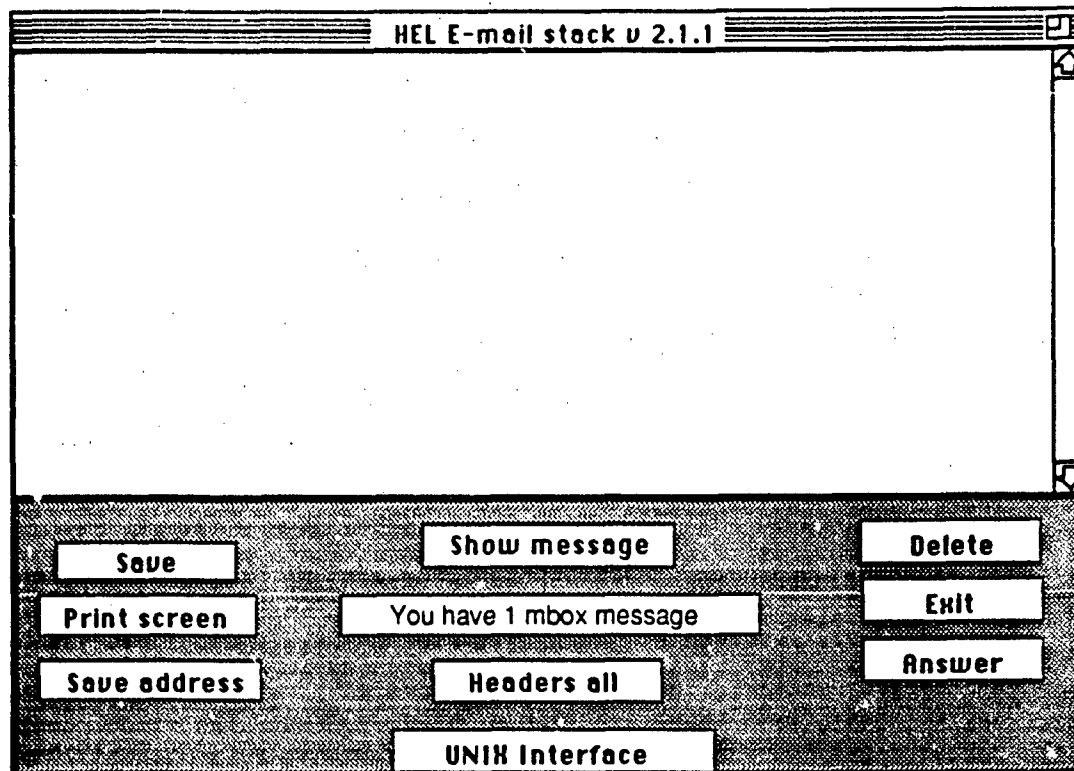


Figure 6. Themessage card.

Buttons

- a. The "Save" button saves the current message to a file in the current directory on the host computer.
- b. The "Print Screen" button prints the contents of the display field.
- c. The "Save Address" button saves the address of the currently displayed message to enable the user to access this information from the send mail screen.
- d. The "Show message" button is a pop-up menu which lists all the message numbers that have been received and are ready for viewing.
- e. The "Headers all" button displays the headers of all the messages.
- f. The "UNIX Interface" button allows the user to return to the UNIX interface screen.
- g. The "Delete" button deletes messages that are currently being viewed.
- h. The "Exit" button exits the current screen and saves the messages to the mbox file.

i. The answer button allows the user to answer the currently displayed message.

Fields

a. The "msgname" field in the middle of the screen displays the number of messages received.

b. The "screen" field displays the message text.

The Groups card (see Figure 7) allows the user to define groups of people to whom messages can be sent from the send mail card. The information must be entered exactly as the directions state at the top of the card. This card has two buttons and one field. It must be the group name followed by a comma, then open parenthesis, address of an individual, close parenthesis.

Buttons

a. The "Return" button takes the user back to the mail card.

b. The "UNIX Interface" takes the user back to the UNIX Interface card.

Fields

a. The "Group Name" screen allows users to define their groups for sending electronic mail.

The screenshot shows a window titled "HEL E-mail stack v 2.1.1". Inside the window, there is a text box with the following text: "Directions: Enter the name of the group followed by a comma and then the address of the group." To the right of this text box is a button with a circular arrow icon. Below the text box is a large text area containing the following text: "Great people,(jsmith@hel4.brl.mil,djones,wryan)", "Marketing,(jsmith@hel4.brl.mil,djones,tbrown)", and "Sales,(rthompson,jdoe,plucas)". To the right of this text area is a vertical scrollbar. At the bottom of the window is a button labeled "UNIX Interface".

Figure 7. The groups card.

CONCLUSIONS

The intent of HUES was to merge the powerful electronic mail capabilities of the UNIX system with the easy-to-use, point-and-click environment of the Macintosh. However, mixing different system architectures and operating systems cannot easily be accomplished without making certain sacrifices. In the case of HUES, the operator gains a user-friendly environment and increased productivity but sacrifices the speed and power of the UNIX operating system. To some extent, the system's reliability is also compromised because of increased system complexity. These costs must be weighed against increased productivity and decreased training times.

The TCP/IP external commands used during the development of this system can also be used to communicate with other operating systems besides the UNIX environment. Thus, future applications could focus on whether other platforms might benefit from an integration with a Macintosh graphically oriented operating system.

HEL's Hypercard-UNIX E-mail system is a successful Hypercard program and an efficient tool for transferring electronic mail across the Internet.

BIBLIOGRAPHY

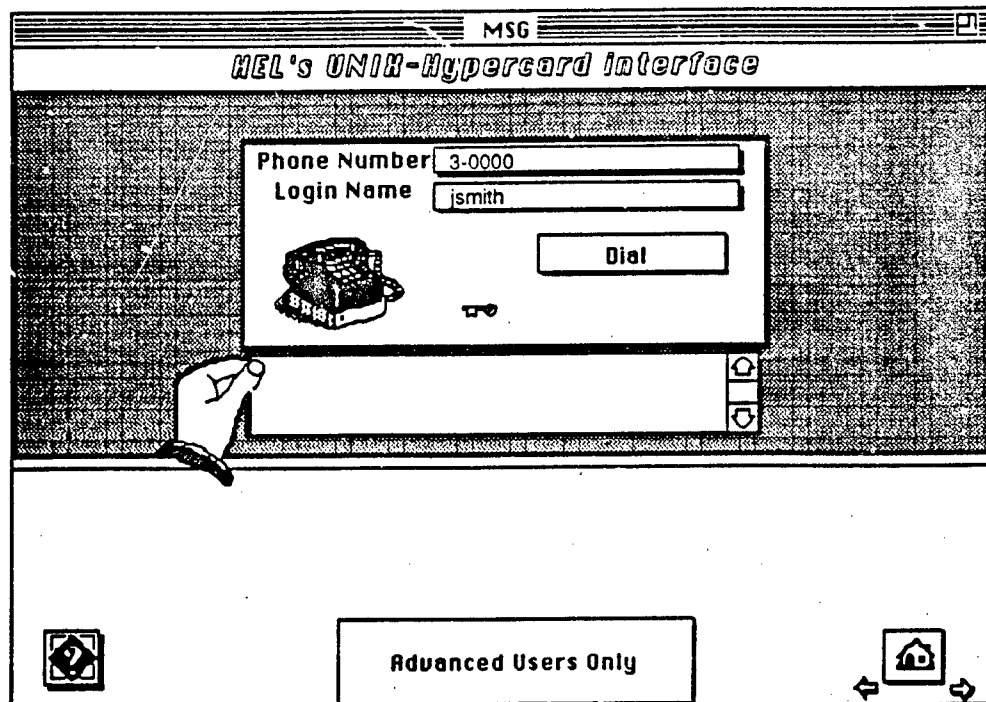
Goodman, D. (1987). The complete hypercard handbook. New York: Bantam Books.

Madron, T. (1988). local area networks next generation. New York: Wiley Inc.

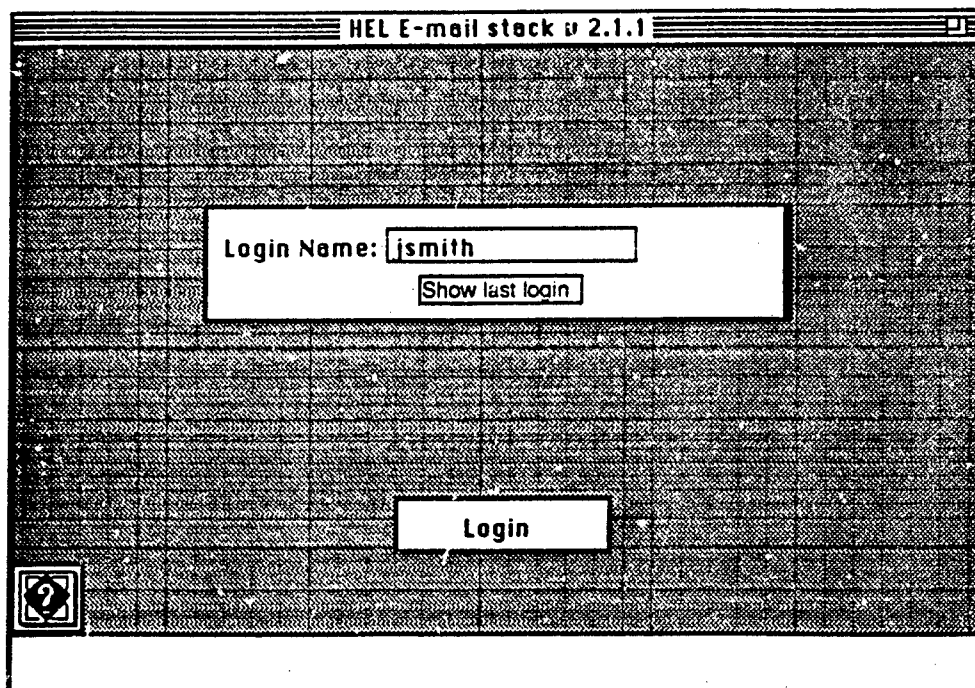
Apple Programmers Developer Association (1990). MacTCP 1.0 release notes available about the developer compact disk, Volume III.

APPENDIX A

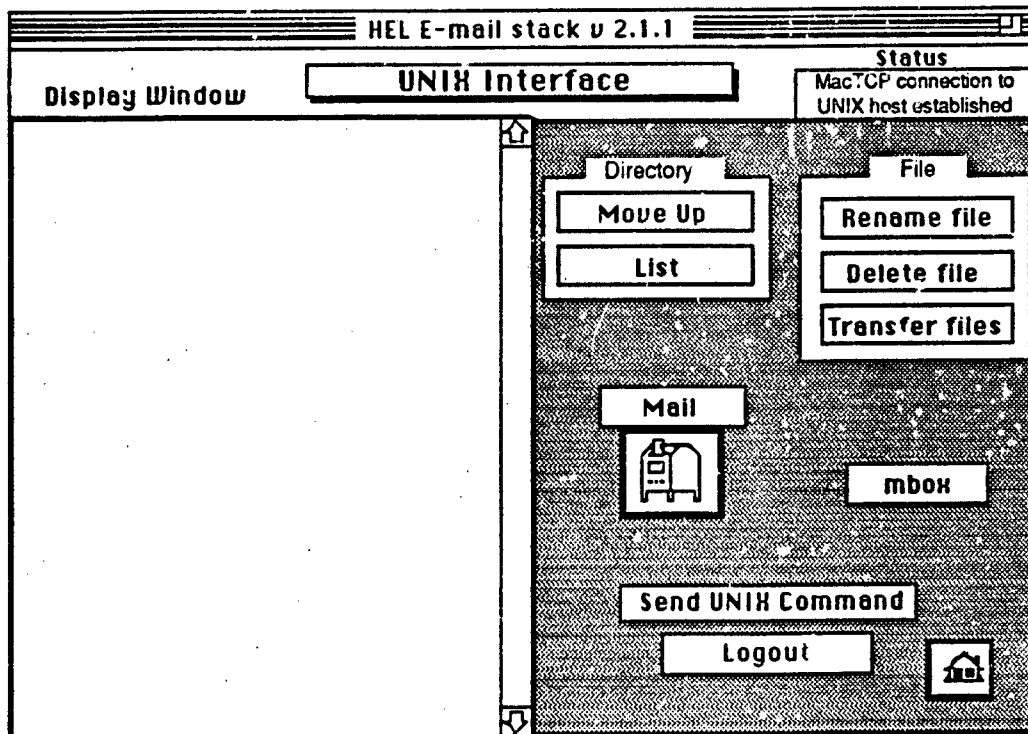
HEL'S HYPERCARD-UNIX E-MAIL SYSTEM (HUES)
CARD NAMES



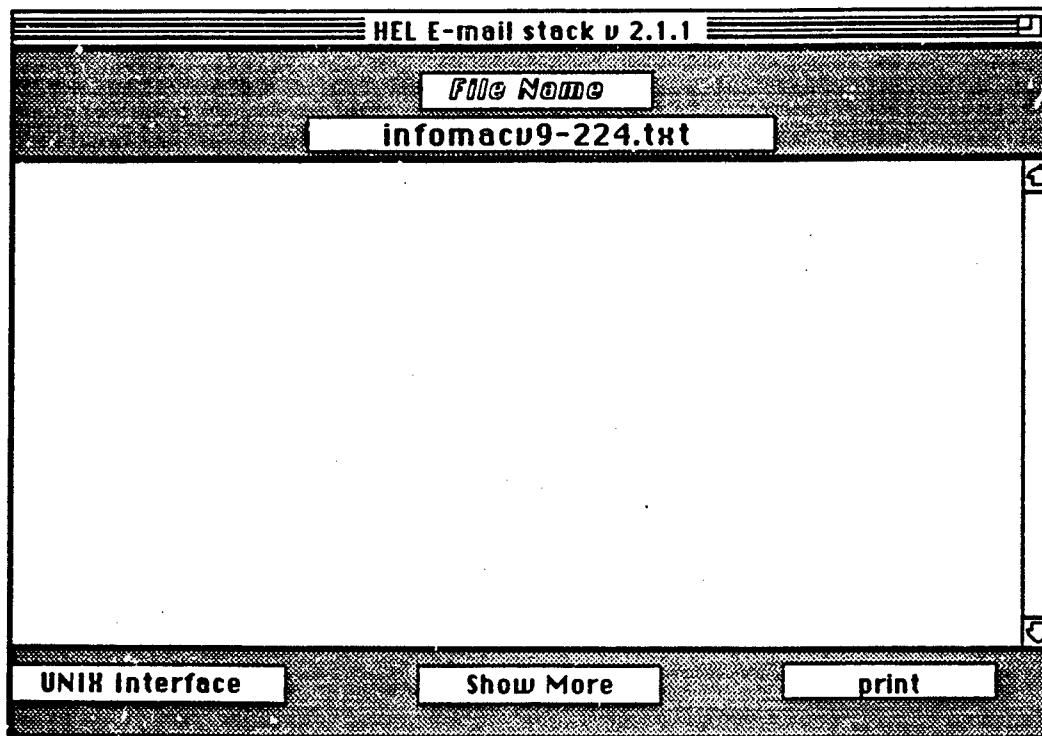
Card name: The user preferences card--modem version.



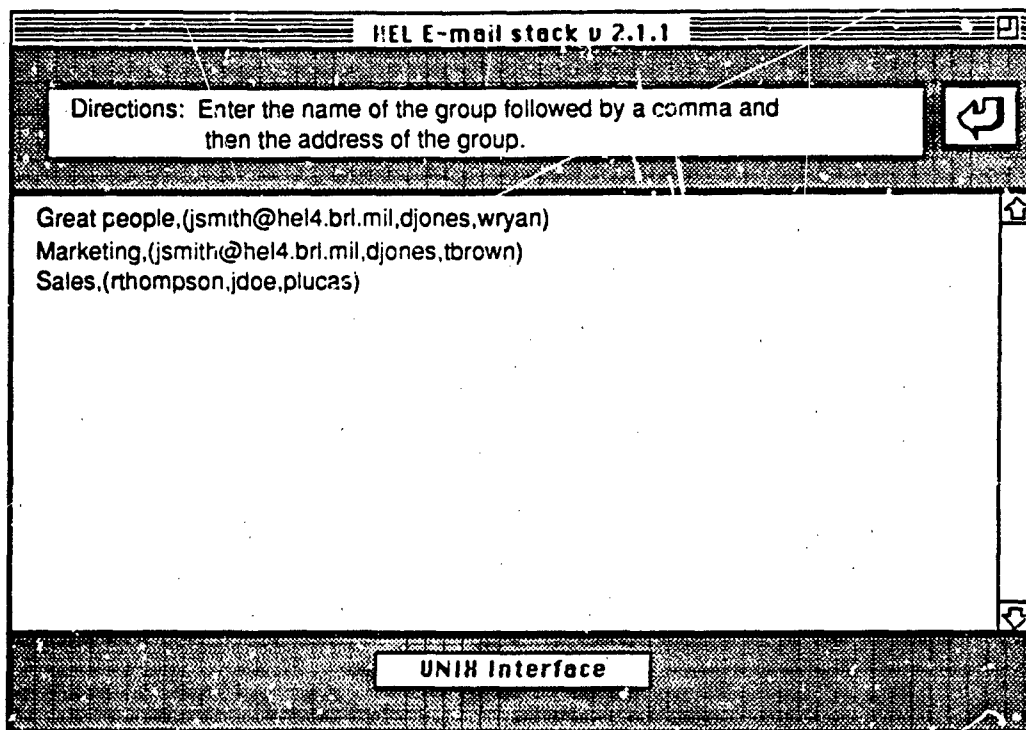
Card name: The user preferences card--MacTCP version.



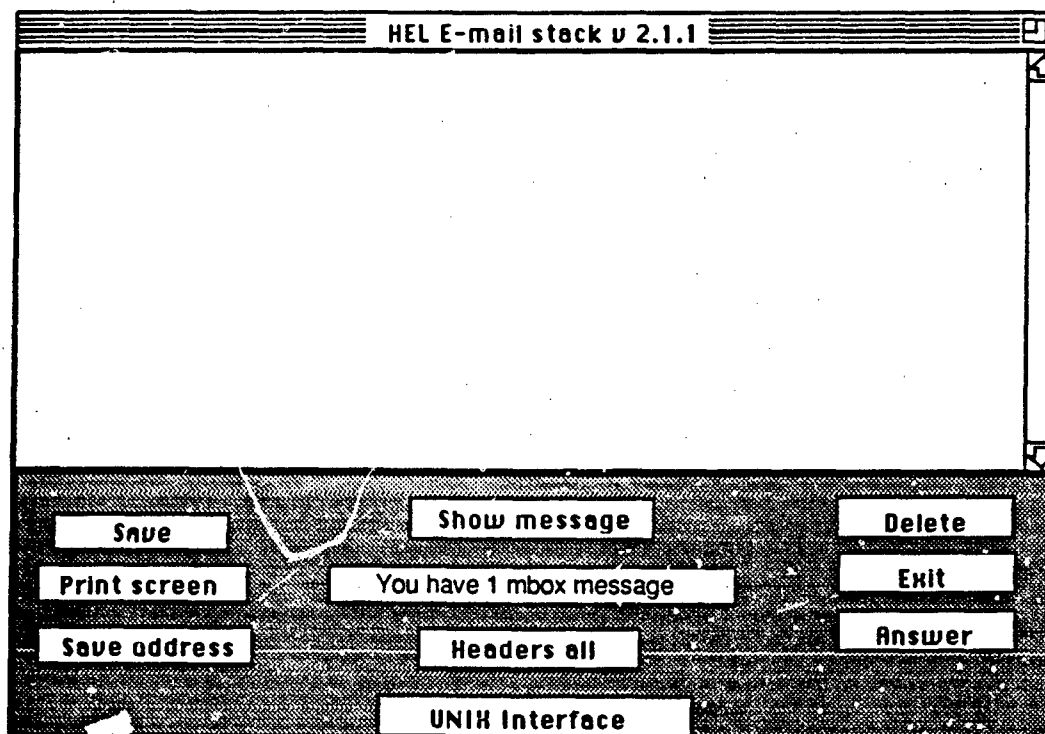
Card name: Interface.



Card name: File view.



Card name: Groups.



Card name: Themessage.

HEL E-mail stack v 2.1.1

To: jsmith@hel4.brl.mil

DD:

Subject:

Options

SEND

UNIX Interface

Saved Addresses

Card name: Mail card.

APPENDIX B

HEL'S HYPERCARD-UNIX E-MAIL SYSTEM (HUES)
SOURCE CODE--MACTCP VERSION

HEL'S HYPERCARD-UNIX E-MAIL SYSTEM (HUES)
SOURCE CODE--MACTCP VERSION

----- STACK: HEL E-mail stack v 2.1.1 -- ----

----- STACK SCRIPT: -----

-- HEL E-mail Stack ©Troy Kelley 1991

-- Many of the login routines and character-handling functions were taken from HyperUNIX

-- v 1.7.3 by Gregg Anderson. My thanks to Gregg for his support. Many of the comments for

-- the MacTCP version are his. His code is set apart by borders.

on openStack

put the seconds into fld theTime of card interface

hide menuBar

set userlevel to 0

go card "User Preferences"

hide msg

-- This is the automatic login routine

answer "login now" with "Cancel" or "OK"

if it is "Cancel" then exit openStack

send mouseUp to card button login

send mouseUp to card button "mail"

--doMenu "Chooser"

end openStack

on closeStack

global TCPconnection

if TCPState(TCPconnection) is "established" then

put "Logging out...." into msg

sendResponse "logout" & return

wait 60

sendResponse "logout" & return

wait 20

TCPRelease(TCPconnection)

wait 60

TCPClose(TCPconnection)

put TCPState(TCPconnection) into fld troystatus of card interface

end if

if the freeSize of this stack ≥ 10960 then -- 40K free space?

put "Compacting the stack" into msg

set userlevel to 0

--send "doMenu Compact Stack" to hypercard

end if

end closeStack

on idle

global TCPconnection,FTPconnection

-- This is an important routine. It checks to see how much time has passed
-- and subtracts it from the time in fld theTime. If a certain amount has
-- passed it simply sends a carriage return to let the SUN know the user
-- is still logged in but simply idle.

put the seconds into nowTime

put fld theTime of card interface into fun

subtract fun from nowTime

if nowTime > 240 then

sendResponse " " & return

end if

```

if FTPconnection is empty then

    put TCPRecvUpTo(TCPconnection,empty,0,empty) into newInput
    if newInput is not "$$$ invalid connection ID $$$" then

        if newInput is not empty then

            set cursor to watch
            put xStrip(newInput,linefeed) after last character in fld screen
            end if
        end if
    else
        put TCPRecvUpTo(FTPconnection,empty,0,empty) into newInput

        if newInput is not "$$$ invalid connection ID $$$" then

            if newInput is not empty then

                set cursor to watch
                put xStrip(newInput,linefeed) after last character in fld screen

                end if
            end if

        end if
    end if
end idle
function UNIXWait
    put 1 into holder
    put WaitForOutPutSilence(60) into fun
    if fun is not empty then
        set cursor to watch
        repeat until holder = empty
            put WaitForOutPutSilence(60) into holder
            put holder after last character in fun
        end repeat
    end if
    --put xStrip(fun,linefeed) into fun
    return fun
end UNIXWait
function WaitForSomething
    put 1 into holder
    put WaitForOutPutSilence(60) into fun
    if fun is empty then
        repeat until fun is not empty,
            put WaitForOutPutSilence(60) into fun
        end repeat
    end if
    return fun
end WaitForSomething

```

```

function flushOutput
    global XMethod,TCPconnection

    get OutputAv()
    if it is 0 then return empty
    if XMethod is "Serial" then
        return recvChars(it)
    else if XMethod is "MacTCP" then
        return TCPRecvChars(TCPconnection,it)
    end if
end flushOutput

```

```

end if
return empty
end flushOutput

```

```

function OutputAv
global XMethod, TCPconnection

if XMethod is "Serial" then
return CharsAvailable()
else if XMethod is "MacTCP" then
return TCPCharsAvailable(TCPconnection)
end if
return 0
end OutputAv

```

```

on telnetCommands s
global TCPconnection

```

```

-- Go through all the new characters, handling Telenet options.
-- Note. This code assumes that both the DO/DON'T/WILL/WON'T
-- and the accompanying option specification are in s. If they
-- get split across a receive, this will break. However, we
-- wait long enough, and they should arrive in the same packet;
-- this should not be a problem.

```

```

repeat while s is not empty
-- Get the character and remove it from s.
get charToNum(char 1 of s)
delete char 1 of s
-- Check for command lead-in character.
if it is 255 then
-- Get the command itself.
get charToNum(char 1 of s)
delete char 1 of s
-- Check for WILL/WON'T/DO/DON'T.
if (it > 250) and (it < 256) then
-- If it's DO/DON'T, answer WON'T.
if it > 252 then
get 252
-- if it's WILL/WON'T, answer DON'T.
else get 254

```

```

-- Send the negotiation response (getting the option char).

```

```

TCPsend TCPconnection, numToChar(255) & numToChar(it) & ~
(char 1 of s)

```

```

-- delete char 1 of s
end if
end if
end repeat
end telnetCommands

```

```

on killConnection
global TCPconnection
if TCPconnection is not empty then

```

```

TCPRelease TCPconnection
if the result is not empty then put "The Result:" && the result
put empty into TCPconnection
end if
end killConnection

```

```

on logout
global XState
-- logout from the host computer

```

```

if XState is "open" then sendResponse "logout" & return
put "Closed" into XState
killConnection
end logout

```

```

function recvUntil terminator
global errorState,XMethod,TCPconnection

put empty into errorState
put empty into serOut
put 1 into lc
put the ticks + 3600 into timeout -- time out after 1 minute
repeat until the ticks > timeout
    put getOutput(serOut) into new
    if new is not serOut then
        put new into serOut
        put the ticks + 600 into timeout
    end if
    if serOut contains terminator then return serOut
end repeat
put "Timed out waiting for " & terminator into errorState
return serOut
end recvUntil

```

```

function WaitForOutputSilence howlong
put empty into output
put the ticks + howlong into timeout
repeat while the ticks < timeout
    get OutputAv()
    if it is not 0 then
        put flushOutput() after output
        put the ticks + howlong into timeout
    end if
end repeat
return output
end WaitForOutputSilence

```

```

on setStatus newStat
    if newStat is not empty then
        put newStat into bg fld troystatus
        hide msg
    end if
end setStatus
on UNIXHost
    global UNIXName,TCPconnection,XState,errorState

```

```

--logout
put "Closed" into XState
put empty into errorState
--
--    CONNECTED TO A UNIX HOST
--
get doLogin()

checkError
--
--    Give the UNIX host time to finish logging us in by waiting
--    for 2 seconds of serial silence.
--
--    Then execute the Bourne shell ('sh') and change the prompt
--    to ' ##done'.
--
setStatus "Starting Bourne Shell"
go to card interface

```

```

end UNIXHost

function doLogin
  global errorState,XMethod,TCPconnection

  put empty into errorState
  put empty into serOut
  put false into gotIn
  put 1 into lc
  put the ticks + 3600 into timeout -- time out after 1 minute
  get the long date && the long time && return
  put "-----" & return after it
  put it into card field "Last Login" of card "User Preferences"

  repeat until ( (the ticks > timeout) or ( serOut contains "% ") or ( serOut contains "=>
  ") )

    put getOutput(serOut) into new

    get "[" & numtochar(1) & "-" & numtochar(12) & numtochar(14) & "-"
    & numtochar(31) & numtochar(128) & "-" & numtochar(255) & "]"
    put rep(it,empty,new) into new

    if new is not serOut then
      put new into serOut
      setStatus(last line of serOut)
      add 600 to timeout
    end if
    put empty into hold
    --
    -- .....
    -- Main comparison loop of login function
    -- .....
    --
    -- This is where you will find the list of
    -- recognized prompts and responses.
    --
    -- .....
    --
    if grep("^login:",serOut) is not empty then

      doName

      put true into gotIn
      put the ticks + 3600 into timeout
      else if serOut contains "WAIT(Y/N)?" then
        sendResponse "n" & return
        setStatus(last line of serOut && "n -- Try later")
        exit to Hypercard
      else if serOut contains "Incorrect" then
        put false into gotIn
        put empty into card field "password" of card "User Preferences"
      else if serOut contains "[ynq]" then sendResponse "q" & return
      else if serOut contains "TERM =" then sendResponse "dumb" & return
      else if serOut contains "WHICH COMPUTER?" or serOut contains "
      WHICH COMUTER?" then doPortSelector
      else if serOut contains "comm-tsl>" then doTerminalServer
      else if serOut contains "BUSY" then exit to Hypercard
      else if serOut contains "NO CARRIER" then exit to Hypercard
      else if serOut contains "CONNECTED TO " then sendReturns
      else
        put serOut into hold
        put empty into serOut
      end if
    end if
  end repeat
end function

```

```

    put serOut after card field "Last Login" of card "User Preferences"
    put hold into serOut
end repeat
put serOut after card field "Last Login" of card "User Preferences"
if gotIn then return serOut
put "Timed out in login" into errorState
return serOut
end doLogin

on doName
    global passforftp
    get card field "UNIX login" of card "User Preferences"
    if it is empty then get ask("Login:")
    set cursor to watch
    sendResponse it & return
    get WaitForOutputSilence(60)

    setStatus last line of it
    get card field "password" of card "User Preferences"
    if it is empty then get ask("secret", "Enter the user password:")
    if it is empty then
        put empty into card fld "UNIX login" of card "User Preferences"
    end if
    put it into passforftp
    sendResponse passforftp & return
    get "*****" -- overwrite password immediately
    set cursor to watch
end doName

on checkError
    global errorState

    if errorState is not empty then
        setStatus errorState
        exit to Hypercard
    end if
end checkError

function getOutput old
    global XMethod, TCPconnection

    if XMethod is "Serial" then
        put recvUpTo(return, 60, old) into new
    else if XMethod is "MacTCP" then
        put TCPrecvUpTo(TCPconnection, return, 10, old) into new

        if numToChar(255) is in new then telnetCommands new
    end if
    put replace(linefeed, empty, new) into new

    return new
end getOutput

on sendResponse what
    global TCPconnection, XMethod, XState, sendCtrlU, FTPconnection

    if XMethod is "Serial" then
        if sendCtrlU is "true" and XState is "Open" then
            SendSPort numToChar(21)
        end if
        sendSPort what
        repeat until sendSPortDone()
        end repeat
    else if XMethod is "MacTCP" then

```

```

put the seconds into fld theTime of card interface

if FTPconnection is empty then

    TCPSend TCPconnection,what
else
    TCPSend FTPconnection,what & return & linefeed
end if
end if

end sendResponse

```

```

----- BACKGROUND: bkgnd id 2753 -----

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

on mouseUp
-- This code allows the user to click in the field to select a file name
put the short name of this card into cardname
if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
    - top of me)/textheight of me))+1 into theLine
    -- get the line number
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun
    if last character of fun = "/" then
        -- see if it is a directory or not
        sendResponse "cd " & fun & return
        put empty into fld screen

        set cursor to watch
        sendResponse "ls -l" & return
        UNIXWait

        set the scroll of me to 1
    else

        sendResponse "more -f -l" & fun & return
        --sendResponse "more" & fun & return
        put fun into card fld filename of card "file view"
        set lockscreen to true
        set the rect of me to 0,57,512,297

        go to card "file view"

        get UNIXWait()
        put it into fld screen of card "file view"
    end if

end if

hide msg
end mouseUp

```

----- BACKGROUND SCRIPT: bkgnd id 4774 -----

```
on FTPprogress bytes
  set cursor to busy
  set the loc of the msg to 10,300
  put "Bytes transferred:" && bytes into msg
end FTPprogress bytes
on CloseConn
  global dataID,FTPconnection

  TCPRelease dataID
  wait 50
  TCPClose dataID
  wait 50
  TCPRelease FTPconnection
  wait 50
  TCPClose FTPconnection
  wait 50
  put empty into ftpconnection
  put empty into dataID
  put empty into fld screen
  get loc of card button "list"
  click at it
end CloseConn
```

----- BACKGROUND BUTTON SCRIPTS -----

----- BUTTON: bkgnd button "Home"

```
on mouseUp
  global logoutme
  -- don't let them go anywhere without logging out
  if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
  end if
```

```
  visual effect iris open
  go Home
end mouseUp
```

----- BUTTON: bkgnd button "Strip control characters"

```
on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp
```

----- BUTTON: bkgnd button "Strip parity bit"

```
on mouseUp
  if the hilite of me then configureSPort stripOn
```

```

else configureSPort stripOff
end mouseUp

```

```

----- BACKGROUND FIELD SCRIPTS -----

```

```

----- FIELD: bkgnd field "screen"

```

```

on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseloc -
    - top of me)/textheight of me))+1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun
    if last character of fun = "/" then

      sendResponse "cd "&fun & return
      put empty into fld screen

      set cursor to watch
      sendResponse "ls -lf" & return
      --UNIXWait

      set the scroll of me to 1
      exit mouseUp

    end if
    sendResponse "cd "&fun & return
    put empty into fld screen
    set cursor to watch
    get UNIXWait()
    put it into card fld remove

    get FindInfield(card fld remove,"directory",false)
    if it is "0,0,0,0" then
      sendResponse "ls -lf" & return
      exit mouseUp
    end if

    sendResponse "more -f -l"&fun & return
    --sendResponse "more"&fun & return
    CgRet fun,60
    put fun into card fld filename of card "file view"
    set lockscreen to true
    set the rect of me to 0,57,512,297
    go to card "file view"
    --get UNIXWait()
    --put it into fld screen of card "file view"

  end if
end if
hide msg
end mouseUp

```

```

----- BACKGROUND: bkgnd id 6418 -----

```

```

----- BACKGROUND BUTTON SCRIPTS -----

```

----- BUTTON: bkgnd button "Home"

```
on mouseUp
  global logoutme

  if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
  end if
```

```
  visual effect iris open
  go Home
end mouseUp
```

----- BUTTON: bkgnd button "Strip control characters"

```
on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp
```

----- BUTTON: bkgnd button "Strip parity bit"

```
on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp
```

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

```
on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me / textheight of me) * (item 2 of the mouseloc -
    - top of me) / textheight of me)) * 1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun

    sendSPort "cd " & fun & return
    put empty into fld screen
    set cursor to watch
    set the loc of msg to 10,300
    put "Checking for file or directory type..." into msg
    wait 300

    put charsAvailable() into z
    put recvChars(z) into gotit

    if last word of line 2 of gotit is "directory" then

      sendSPort "more -f -1" & fun & return
      put fun into card fld filename of card "file view"
      set the rect of me to 0,57,512,297
      go to card "file view"
      put empty into fld screen
```

```

else

    sendSPort "ls -l" & return
    set the scroll of me to 1
end if
end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 3806 -----
----- BACKGROUND BUTTON SCRIPTS -----
----- BUTTON: bkgnd button "Home"

on mouseUp
    global logoutme

    if logoutme is not empty then
        answer "You must logout to go home" with "OK"
        exit mouseUp
    end if

    visual effect iris open
    go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

on mouseUp
    if the hilite of me then configureSPort stripControlsOn
    else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
    if the hilite of me then configureSPort stripOn
    else configureSPort stripOff
end mouseUp

----- BACKGROUND FIELD SCRIPTS -----
----- FIELD: bkgnd field "screen"

on mouseUp
    put the short name of this card into cardname
    if cardname = "interface" then

        put (trunc((scroll of me / textheight of me) * (item 2 of the mouseloc -
        - top of me) / textheight of me)) + 1 into theLine
        select line theLine of fld screen

        put last word of line theLine of fld screen into fun

        sendSPort "cd " & fun & return
    end if
end mouseUp

```

```

put empty into fld screen
set cursor to watch
set the loc of msg to 10,300
put "Checking for file or directory type..." into msg
wait 300

put charsAvailable() into z
put recvChars(z) into gotit

if last word of line 2 of gotit is "directory" then

    sendSPort "more -f -l" & fun & return
    put fun into card fld filename of card "file view"
    set the rect of me to 0,57,512,297
    go to card "file view"
    put empty into fld screen
else

    sendSPort "ls -l" & return
    set the scroll of me to 1
end if
end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 7265 -----
----- BACKGROUND BUTTON SCRIPTS -----
----- BUTTON: bkgnd button "Home"

on mouseUp
global logoutme

if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
end if

visual effect iris open
go Home
end mouseUp

----- • BUTTON: bkgnd button "Strip control characters"

on mouseUp
if the hilite of me then configureSPort stripControlsOn
else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
if the hilite of me then configureSPort stripOn
else configureSPort stripOff
end mouseUp

```

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkqnd field "screen"

```
on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
    - top of me)/textheight of me))+1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun

    sendSPort "cd "&fun & return
    put empty into fld screen
    set cursor to watch
    set the loc of msg to 10,300
    put "Checking for file or directory type..." into msg
    wait 300

    put charsAvailable() into z
    put recvChars(z) into gotit

    if last word of line 2 of gotit is "directory" then

      sendSPort "more -f -l"&fun & return
      put fun into card fld filename of card "file view"
      set the rect of me to 0,57,512,297
      go to card "file view"
      put empty into fld screen
    else

      sendSPort "ls -l" & return
      set the scroll of me to 1
    end if
  end if

  hide msg
end mouseUp
```

----- BACKGROUND: bkqnd id 8414 -----

----- BACKGROUND BUTTON SCRIPTS -----

----- BUTTON: okqnd button "Home"

```
on mouseUp
  global logoutme

  if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
  end if

  visual effect iris open
  go Home
end mouseUp
```

----- BUTTON: bkgnd button "Strip control characters"

```
on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp
```

----- BUTTON: bkgnd button "Strip parity bit"

```
on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp
```

----- BUTTON: bkgnd button "Prev"

```
on mouseUp
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"
```

```
    exit mouseUp
  end if
```

```
  put the short name of this card into cardname
  if cardname = "file view" then
    if fld screen of this card = empty then
      put empty into fld screen of this card
      CntrZ
      put the result into fun
      sendSPort fun
```

```
    wait 100
    sendSPort "ls -l" & return
  end if
```

```
end if
```

```
set lockscreen to true
go to prev card
```

```
hide msg
end mouseUp
```

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

```
on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseloc -
    - top of me)/textheight of me))+1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun

    sendSPort "cd " & fun & return
```

```

put empty into fld screen
set cursor to watch
set the loc of msg to 10,300
put "Checking for file or directory type..." into msg
wait 300

put charsAvailable() into z
put recvChars(z) into gotit

if last word of line 2 of gotit is "directory" then

    sendSPort "more -f -l" &&fun & return
    put fun into card fld filename of card "file view"
    set the rect of me to 0,57,512,297
    go to card "file view"
    put empty into fld screen
else

    sendS: "ls -l" & return
    set the scroll of me to 1
end if

hide msg
end mouseUp

----- CARD SCRIPT: User Preferences -----

on openCard
    put empty into fld troystatus
    hide fld screen
end openCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button    qin"

on mouseUp

    global TCPconnection,XState,XMethod
    put "MacTCP" into XMethod

    set cursor to watch

    if TCPconnection is not empty then
        logout
    end if

    setStatus "Opening a MacTCP connection to UNIX host"

    -- Connect to the 'telnet' port of the UNIX host.
    -- Changing the "23" here can change the type of connection
    put TCPActiveOpen(item 1 of card field "IP address", 23, 0)~
    into TCPconnection

    if TCPconnection contains "fail" then
        put "The Result:" && TCPconnection
        put empty into TCPconnection
        put "Closed" into XState
        exit mouseUp
    end if

```

```

end if

--
-- Connect to the UNIX host & we're done
--
UNIXHost
setStatus "MacTCP connection to UNIX host established"
put "Open" into XState

end mouseUp

----- BUTTON: card button "New Button"

on mouseUp
  show card fld "last login"
end mouseUp

----- BUTTON: card button "New Button"

on mouseUp
  show card fld help
end mouseUp

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "Last Login"

on mouseUp
  hide me
end mouseUp

----- FIELD: card field "help"

on mouseUp
  hide me
end mouseUp

----- CARD SCRIPT: interface -----

on openCard
  global TCPconnection
  put empty into fld screen
  if TCPState(TCPconnection) is "established" then

    sendResponse "ls -lf" & return
    -- this type of listing is used to list the directories in single
    -- columns with no file sizes
  end if

  set lockscreen to true
  set the rect of fld screen to 0,34,260,342
end openCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Send UNIX Command"

```

```

on mouseUp

    ask "Send what UNIX command"
    put empty into fld screen
    sendResponse it & return
    get UNIXWait()
    put it into fld screen of this card

end mouseUp

----- BUTTON: card button "List"

on mouseUp
    global ftp,thereis
    if ftp = 1 then

        --put thereis into fld title
        --put empty into card fld FTPsite
        put empty into fld screen

        sendResponse "ls -lf" & return
        --get UNIXWait()
        --put it into fld screen

        set the scroll of fld screen to 1

    end if
    if ftp = 1 then
        set cursor to watch
        put empty into fld screen
        --put "Sending command" into card fld mystatus
        sendResponse "ls -lf" & return
        put empty into fld screen
        get WaitForOutputSilence(250)
        if it is empty then
            repeat until it ≠ empty
                get WaitForOutputSilence(60)
                --put "Waiting for response" into card fld Diag
            end repeat
        end if

        put xStrip(it,linefeed) into fun
        delete last line of fun
        put fun into fld screen

    end if

    --put "Ready" into card fld mystatus

end mouseUp

----- BUTTON: card button "Logout"

on mouseUp

```

```
global logoutme
answer "Logout now?" with "NO" or "Yes"
if it is "Yes" then
```

```
    put empty into logoutme
    set cursor to watch
    put empty into fld screen
    sendResponse "logout" & return
    wait 200
    sendResponse "logout" & return
    wait 200
    logout
    answer "You have been logged out"
end if
```

```
end mouseUp
```

```
----- BUTTON: card button "Move Up"
```

```
on mouseUp
```

```
    sendResponse "cd .." & return
    wait 100
```

```
    put empty into fld screen
```

```
    sendResponse "ls -lf" & return
    --UNIXWait
```

```
    set the scroll of fld screen to 1
```

```
end mouseUp
```

```
----- BUTTON: card button "Rename file"
```

```
on mouseUp
```

```
    global TCPconnection
```

```
    put empty into card fld remove
    put "Please wait... gathering data" into msg
```

```
    set cursor to busy
    put xgetScreen(fld screen,return) into card fld remove
```

```
--delete item 1 of card fld remove
put "Choose file...." into msg
put card field "remove" into theList
DoList 999,theList,one
```

```
    -- get the file list from the field
    put the result into thelist
    if thelist is not empty then
```

```
        put item 2 of thelist into thelist
        answer "Rename file:&&thelist&&?" with "OK" or "Cancel"
        if it is "Cancel" then
            hide msg
```

```

        exit mouseUp
    end if

    ask "Enter new name (one word please)" with thelist
    put it into new
    if new = thelist then
        answer "Not one word"
        exitmouseUp
    end if

    put the number of words in new into numword
    if numword = 1 then
        SendResponse "mv" & thelist & new & return
        answer "File has been renamed"
        put empty into fld screen
        wait 50
        send mouseUp to card button list
    else
        answer "Not one word"
    end if

end if
hide msg
end mouseUp

----- BUTTON: card button "Mail"

on mouseUp
    global messnum, TCPconnection
    put empty into messnum

    answer "Check mail or send it?" with "Cancel" or "Send" or "Check"
    if it is "Cancel" then
        exit mouseUP
    end if

    if it is "Send" then
        go to card "mail card"
        exit mouseUp
    end if
    set cursor to watch
    --put empty into fld screen
    put empty into card fld remove
    SendResponse "msg" & return
    -- interface to the mail system on the SUN (msg)

    wait 100

    put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput
    repeat until newInput is empty

        put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into nowInput

        put newInput after last character of card field remove

    end repeat

```

```

set the scroll of fld screen to 1
-- try to get the number of messages with this routine, or if there are any
-- messages to read
get FindInField(card fld remove,"total","false",0)
if it = "0,0,0,0" then
    get FindInField(card fld remove,"empty","false",0)
    if it = "0,0,0,0" then
        answer "Error in checking mail, try again"

        exit mouseUp

    end if

    answer "No new messages"
    --put empty into fld screen
    --sendSPort "ls -l" & return

set the scroll of fld screen to 1
exit mouseUp

else
    get FindInField(card fld remove,"binary","false",0)

    if it = "0,0,0,0" then
        get FindInField(card fld remove,"message","false",0)
        put item 2 of it into linenum
        put the number of words in line linenum of cd fld remove into temp
        subtract 2 from temp
        put word temp of line linenum of cd fld remove into theCount

        set the loc of the msg to 10,300
        if theCount = "1" then

            put "You have "&theCount&" message" into cd fld msgname of card themessage
        else
            put "You have "&theCount&" messages" into cd fld msgname of card themessage
        end if

        go to card themessage
        put empty into card fld save
        put empty into card fld theList
        put theCount into messnum
        repeat with x = 1 to theCount

            put x&";" after last character in card fld theList

        end repeat

    else
        get FindInField(card fld remove,"message","false",0)
        put item 2 of it into linenum

        put word 2 of line linenum of cd fld remove into theCount
    end if
end if

```

```

set the loc of the msg to 10,300
if theCount = "1" then
    put "You have "&theCount&" message" into cd fld msgname of card themessage
else
    put "You have "&theCount&" messages" into cd fld msgname of card themessage
end if

go to card themessage
put empty into card fld save
put empty into card fld theList
put theCount into messnum
repeat with x = 1 to theCount

    put x&" " after last character in card fld theList

end repeat

end if
end if

hide msg

end mouseUp

```

----- BUTTON: card button "Delete File"

```

on mouseUp
    global TCPconnection
    put empty into card fld remove

    put empty into card fld remove
    put "Please wait... gathering data" into msg

    set cursor to busy
    put fld screen into dataget

    put xgetScreen(fld screen,return) into card fld remove

    put "Choose file...." into msg
    put card field "remove" into theList
    DoList 999,theList,one
    put the result into theList
    if theList is not empty then

        put item 2 of theList into theList
        answer "Remove file"&&theList&&"?" with "OK" or "Cancel"
        if it is "Cancel" then
            hide msg
            exit mouseUp
        end if

        sendResponse "rm "&quote&theList&quote & return

    wait 100

    sendResponse "y" & return
    put empty into trash
    put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput

```

```

repeat until newInput is empty

    put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput

    put newInput after last character of trash

end repeat

put empty into fld screen
answer "File has been removed" with "OK"

put empty into fld screen

sendResponse "ls -lf" & return

set the scroll of fld screen to 1

end if

hide msg box
end mouseUp
--if the optionkey is down then pass mouseup
--put card field list into thelist
--DoList 999,thelist,one
--put the result
--end mouseUp

----- BUTTON: card button "New Button"

on mouseUp
    get the loc of card button mail
    click at it
end mouseUp

----- BUTTON: card button "Transfer files"

on mouseUp
-- This routine transfers a file. There is a binary to hex external command
-- that is used to transfer the file to the SUN. The code opens up a seperate
-- FTP type connection in addition to the Telnet type connection which the
-- user has already established.
global FTPconnection,dataID,passforftp
set lockscreen to true
answer "Do you want to send or receive a file" with "Send" or "Receive" or "Cancel"
put it into ans
if ans is "Cancel" then exit mouseUp
if ans is "Receive" then
    put "Please wait... gathering data" into msg
    put xgetScreen(fld screen,return) into card fld remove
    put "Choose file...." into msg
    put card field "remove" into thelist
    DoList 999,thelist,one
    put the result into thelist
    if thelist is not empty then

        put item 2 of thelist into theFile
        get length(theFile)
        get char it-2 to it of theFile
        if it is "qx" then
            put "H" into xType
        else

```

```

        put "A" into xType
        put NewFileName("Save as:", theFile) into theFile
    end if

    else
        hide msg
        exit mouseUp
    end if

end if

if ans is "Send" then
    answer "Is the file Text or an Application" with "Text" or "Appl"
    put it into xType
    if xType is "Appl" then

        put fileName() into theFile
    else
        put fileName("TEXT") into theFile
    end if

    if theFile is empty then exit mouseUp
    put "stack" & quote & theFile & quote into remFile

    gimelast remFile
    put the result into remFile
    if xType is "Appl" then

        put ".hqx" after remFile
    else
        put ".txt" after remFile
    end if
    put Translate( " ;/\\,/()[]<>!@#$%^&*()-~'+' " & quote, "_", ~
LeafName( remFile ) ) into remFile
    ask "Name to save as on the host:" with remFile
end if

put TCPActiveOpen(item 1 of card field "IP address" of card "User Preferences", 21, 0) ~
into FTPconnection
put empty into fld screen
wait 100
put "Opening transfer connection..." into msg
put TCPCharsAvailable(FTPconnection) into x
put TCPRecvChars(FTPconnection,x) after last character in fld screen

get FindInField(fld screen,"220","false",0)
if it = "0,0,0,0" then
    put card fld "UNIX login" of card "User Preferences" into username
    sendResponse "USER" & username
    wait 100
    put empty into fld screen
    put TCPCharsAvailable(FTPconnection) into x
    put TCPRecvChars(FTPconnection,x) after last character in fld screen
    get FindInField(fld screen,"331","false",0)
    if it = "0,0,0,0" then

        sendResponse "PASS"&passforftp
        wait 100
        put TCPCharsAvailable(FTPconnection) into x
        put x into msg
        put TCPRecvChars(FTPconnection,x) after last character in fld screen
        put TCPPassiveOpen( 0, 0, 0 ) into dataID -- create a new data conn.
    end if
end if

```

```

    if dataID contains "fail" then
        put empty into dataID
        exit to Hypercard -- don't continue with file transfer
    end if

    wait until TCPState( dataID ) contains "waiting" -- happens asynch.

    put dataID into msg
    sendResponse "PORT" && TCPGetAddr( dataID, "local" ) -- get local address

    put char 1 of getLine() into resp

    put "A" into typeCode
    if resp is "2" then
        sendResponse "TYPE" && typeCode -- specify the transfer mode
        put char 1 of getLine() into resp

        if resp is "2" and typeCode is "I" then
            sendResponse "TYPE L 8" -- set byte length to 8 for Binary
            get getLine() -- ignore the response
        end if
    end if

    if resp is not "2" then
        TCPClose dataID
        exit to Hypercard -- don't continue with file transfer
    end if

end if
end if
wait 200
if ans is "Send" then
    SendResponse "STOR" && remFile
    if char 1 of getLine() is not "1" then
        TCPClose dataID
        TCPRelease dataID
    end if
    if WaitForDataCon() then
        if xType is "Appl" then
            FTPHqXfer dataID, "send", theFile
            if char 1 of the result is "$" then
                answer the result
                closeConn
            end if
        else
            FTPxfer dataID, "send", "ascii", theFile
            if char 1 of the result is "$" then
                answer the result
                closeConn
            end if
        end if
    end if
end if
end if

if ans is "Receive" then
    SendResponse "RETR" && theFile
    if char 1 of getLine() is not "1" then

```

```

        closeConn

    end if
    if WaitForDataCon() then
        if xType is "H" then
            FTPHqXfer dataID, "receive"
            if char 1 of the result is "$" then
                answer the result
                closeConn
            end if
        else
            FTPxfer dataID, "receive", "ascii", theFile
            if char 1 of the result is "$" then
                answer the result
                closeConn
            end if
        end if
    end if
    CloseConn
    answer "Transfer complete"
    hide msg
end mouseUp

function getLine
    global FTPconnection

    repeat 15 times
        set the cursor to busy
        get TCPRecvUpTo( FTPconnection, linefeed, 60, empty )
        if it is not empty then exit repeat
    end repeat

    return it
end getLine
function WaitForDataCon
    global dataID

    repeat until TCPState( dataID ) is "established"
        if TCPState( dataID ) contains "close" then return false
        idleCursor
    end repeat

    return true -- signal connection is established
end WaitForDataCon

*----- BUTTON: card button "mbox"

on mouseUp
    global TCPconnection, mbox
    put empty into card fld remove
    put 1 into mbox
    SendResponse "msg mbox" & return
    wait 100

    put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput
    repeat until newInput is empty

        put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput

    put newInput after last character of card field remove

```

```

end repeat

--put xStrip(card fld remove,return) into card fld remove

set the scroll of fld screen to 1

get FindInField(card fld remove,"total","false",0)

if it = "0,0,0,0" then
    get FindInField(card fld remove,"empty","false",0)
    if it = "0,0,0,0" then
        answer "Error in checking mail, try again"

        exit mouseUp
    end if

    answer "No new messages"
    --put empty into fld screen
    --sendSPort "ls -l" & return

    set the scroll of fld screen to 1
    exit mouseUp
else
    get FindInField(card fld remove,"binary","false",0)

    if it = "0,0,0,0" then
        get FindInField(card fld remove,"message","false",0)
        put item 2 of it into linenum
        put the number of words in line linenum of cd fld remove into temp
        subtract 2 from temp
        put word temp of line linenum of cd fld remove into theCount

        set the loc of the msg to 10,300
        if theCount = "1" then

            put "You have " & theCount & " mbox message" into cd fld msgname of card themessage
        else
            put "You have " & theCount & " mbox messages" into cd fld msgname of card themessage
        end if

        go to card themessage
        put empty into card fld save
        put empty into card fld theList
        put theCount into messnum
        repeat with x = 1 to theCount

            put x & ";" after last character in card fld theList

        end repeat
    else

```

```

get FindInField(card fld remove,"message","false",0)
put item 2 of it into linenum

put word 2 of line linenum of cd fld remove into theCount

```

```

set the loc of the msg to 10,300
if theCount = "1" then

```

```

    put "You have "&theCount&" mbox message" into cd fld msgname of card themessage
else
    put "You have "&theCount&" mbox messages" into cd fld msgname of card themessage
end if

```

```

go to card themessage
put empty into card fld save
put empty into card fld theList
put theCount into messnum
repeat with x = 1 to theCount

```

```

    put x&" " after last character in card fld theList

```

```

end repeat

```

```

end if
end if

```

```

hide msg

```

```

end mouseUp

```

```

----- CARD FIELD SCRIPTS -----

```

```

----- FIELD: card field "HEX"

```

```

on mouseUp
    hide me
end mouseUp

```

```

----- CARD SCRIPT: Mail Card -----

```

```

on opencard
    hide bg btn 1

```

```

    hide fld "receiving"
    hide fld "screen"
end openCard

```

```

on closeCard
    show bg btn 1
    show fld screen
    show fld "receiving"

```

end closeCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Send"

on mouseUp

-- This sends a message in a HyperCard field to the SUN and is send as a
-- message. The important thing here is to strip out any carriage returns that
-- might be in the fields and add them in the correct places for the SUN. The
-- fields in hypercard have a word wrap function so a message can be typed
-- with no carriage returns. The CgRet external command adds carriage returns
-- to the end of a variable of so many characters, in this case 60. It also
-- checks to make sure that it does not break up any words, or ruin any
-- existing carriage returns.

global includedfile, TCPconnection
set the loc of the msg to 10,300

if includedfile is empty then
 answer "Send this message?" with "No" or "Yes"
 if it is "No" then
 exit mouseUp
 end if
end if

set cursor to watch
if card fld "To" is empty then
 answer "There is no name to send to!" with "Sorry"
 exit mouseUp
end if

put card fld "To" into person
put xStrip(person,return) into temp
stripLastReturn temp
put the result into cd fld "To"
put card fld "To" into person

put card fld "cc" into x
put xStrip(x,return) into x
stripLastReturn x
put the result into card fld "cc"
put card fld "cc" into carbon

put "Sending message...." into msg

put card fld "text" into y
if y is empty then
 answer "There is no text to send!"
 exit mouseUp
end if

CgRet y,60 -- put carriage returns at the end of every 60 characters
put the result into fun
put y into card fld "text"

```

sendResponse "send" & return
wait 100
-- this checks to see if they are sending the mail to more than one person
put the number of items in card fld "To" into linenum
if linenum > 1 then
    repeat with x = 1 to linenum

        SendResponse item x of cd fld "To"&"\"& return
        wait 60

        put "Sending to:"&item x of cd fld "To" into msg
        wait 60

    end repeat
    sendResponse "" & return
    get UNIXWait()
    put it into card fld error
else
    SendResponse person & return
end if
wait 100

get UNIXWait()
put it into card fld error

put the number of items in card fld "cc" into linenum
if linenum > 1 then
    repeat with x = 1 to linenum

        SendResponse item x of cd fld "cc"&"\"& return
        wait 60

        put "Sending to:"&item x of cd fld "cc" into msg
        wait 60

    end repeat
    sendResponse "" & return
    get UNIXWait()
    put it into card fld error
else
    SendResponse carbon & return
end if
wait 100

put card fld "Subject" into sub
put xStrip(sub,return) into temp
stripLastReturn temp
put the result into cd fld "Subject"
put card fld "Subject" into sub

SendResponse sub & return

wait 100

SendResponse fun & return
get UNIXWait()
put it into card fld error
wait 100

CntrD -- this sends a control D to the SUN to end the message routine

```

```

put the result into fun
SendResponse fun & return
get UNIXWait()
put it into card fld error
wait 100
put "sending control d..." into msg

if includedfile is not empty then
  answer "include file " & includedfile & "?" with "No" or "Yes"
  if it is "Yes" then
    set cursor to watch
    SendResponse "file include" & return
    wait 100
    SendResponse includedfile & return
    put empty into includedfile
  end if
end if

SendResponse "send" & return
get UNIXWait()
put it into card fld error

wait 100
--put empty into cd fld error
--put TCPCharsAvailable(TCPconnection) into fun
--put TCPRecvChars(TCPconnection, fun) into card fld error
--get UNIXWait()
--put it into card fld error

put TCPRecvUpTo(TCPconnection, return, 0, empty) into newInput
repeat until newInput is empty

  put TCPRecvUpTo(TCPconnection, return, 0, empty) into newInput

  put newInput after last character of card field error
end repeat

put FindInField(card fld error, "Message Posted", false) into errortime
if errortime = "0,0,0,0" then
  beep 1
  answer "Message was sent successfully" with "Great!"
else
  beep 1
  SendResponse "quit" & return
  wait 100
  SendResponse "y" & return
  answer "Sorry an error of some kind, see field below" with "Darn"
  set lockscreen to true
  show card fld error
end if

put "****Click on field to close****" after last line in card fld error
put the number of lines of card fld error into fun
multiply fun by 10
set the scroll of card fld error to fun
hide msg
end mouseUp

---- -- BUTTON: card button "Options"

```

```

on mousedown
  -- the global here is to check and see if they have already included a file
  -- to send
  global includedfile, TCPconnection
  put the mouseLoc into myPlace
  put item 1 of myPlace - 10 into horiz
  put item 2 of myPlace - 20 into vert

  get PopUpMenu("Groups;Edit Groups;Include File;Change Directory;Move Up Directory;Clear
Field",5, vert, horiz)
  if it is 0 then answer "This is a pop up menu"
  if it is 1 then
    set cursor to busy
    set lockscreen to true

    go to card "groups"

    put empty into card fld fun
    repeat with x = 1 to the number of lines in card fld "group name"
      put item 1 of line x of card fld "group name" & "," after last character of card fld fun
    end repeat

    put card fld fun into theList

    DoList 999,card fld fun,one
    put the result into fun
    if fun is empty then
      go card "mail card"

    exit mouseDown

  end if

  put item 2 of fun into gotit

  get FindInField(card field "group name",gotit,"true",0)
  put item 2 of it into linenumber

  if it is 0 then
    answer "error of some kind"
  end if
  put 0 into start
  repeat forever
    get FindInField(card fld "group name", "(","true",start)

    if item 2 of it = linenumber then

      exit repeat
    else
      put item 1 of it into start
    end if
  next repeat
end repeat
put item 1 of it into firstspot
get FindInField(card fld "group name",")","true",firstspot)
put item 1 of it into secondspot

```

```

put character firstspot to secondspot of card fld "group name" into address
delete character 1 of address
delete last character of address
put empty into card fld "To" of card "mail card"
put address into card fld "To" of card "mail card"
go to card "mail card"

```

```

set lockscreen to false
end if

```

```

if it is 2 then

```

```

    go to card "groups"
end if

```

```

if it is 3 then

```

```

    set cursor to busy
    set lockscreen to true
    put fld screen of card interface into dataget
    if dataget is empty then

```

```

        SendResponse "Is -lf" & return
        wait 100

```

```

        put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput
        repeat until newInput is empty

```

```

            put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput

```

```

            put xStrip(newInput,linefeed) after last character of field screen of card interface
            end repeat
        end if

```

```

put empty into card fld remove

```

```

put fld screen of card interface into dataget

```

```

-- xgetScreen is an external command to get the contents of a field and
-- pass it to DoList as quickly as possible. I have done this with
-- regular HyperTalk scripts and it can be very slow.

```

```

put xgetScreen(dataget,return) into card fld remove

```

```

put "Choose file...." into msg

```

```

put card field "remove" into theList

```

```

if theList is empty then

```

```

    hide msg

```

```

    beep 1

```

```

    answer "No files in directory, try changing directory"

```

```

    exit mousedown

```

```

end if

```

```

DoList 999,theList,one

```

```

put the result into gotit

```

```

if gotit is empty then

```

```

    go card "mail card"

```

```

hide msg
exit mousedown
end if

```

```

put item 2 of gotit into includedfile
answer "include" && includedfile with "Cancel" or "Ok"
if it is "Cancel" then

```

```

hide msg
exit mouseDown
end if

```

```

go to card "mail card"

```

```

answer "file" && includedfile && "included, send now?" with "Cancel" or "OK"
if it is "OK" then

```

```

get the loc of card button "send"
click at it
end if

```

```

hide msg
set lockscreen to false
end if

```

```

if it is 4 then
set cursor to busy
set lockscreen to true

```

```

put empty into card fld remove
put fld screen of card interface into dataget
if dataget is empty then

```

```

SendResponse "Is -if" & return
wait 100
put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput
repeat until newInput is empty

```

```

put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput

```

```

put xStrip(newInput, linefeed) after last character of field screen of card interface
end repeat
end if

```

```

put fld screen of card interface into dataget
put xgetScreen(dataget, return) into card fld remove

```

```

if card fld "remove" = empty then
answer "Try moving up a directory" with "OK"
go card "mail card"

```

```

exit mouseDown
end if

```

```

put card field "remove" into theList
put "Choose directory...." into msg
DoList 999, theList, one
put the result into gotit

```

```

if gotit is empty then

```

```

        go card "mail card"

        hide msg
        exit mousedown
    end if

    put item 2 of gotit into gotit

    answer "change directory to"&gotit with "No" or "ok"
    if it is "No" then
        hide msg

        exit mousedown
    end if

    SendResponse "cd "&gotit& return
    wait 50
    put empty into fld screen of card interface
    SendResponse "ls -lf" & return
    wait 200
    go to card "mail card"

    hide msg

    wait 100

    set lockscreen to false
    answer "Directory changed" with "Good"
    end if
    if it is 5 then
        answer "Move up a directory?" with "Yes" or "Cancel"
        if it is "Cancel" then exit mousedown

        SendResponse "cd .."& return
        wait 100
        set lockscreen to true
        put empty into fld screen of card interface
        SendResponse "ls -lf"& return
        wait 200

        answer "Directory changed" with "OK"
    end if

    if it is 6 then
        put empty into card fld text
        put empty into card fld "cc"
        put empty into card fld "subject"
    end if

end mousedown

----- BUTTON: card button "UNIX Interface"

on mouseUp

```

```

global TCPconnection
put the short name of this card into cardname
if cardname = "file view" then
    if fld screen of this card ≠ empty then
        put empty into fld screen of this card
        sendResponse "q"

        wait 100
        --sendSPort "ls -lf" & return
        go card interface
    end if
end if
if cardname = "read mail" then
    if fld screen of this card ≠ empty then
        put empty into fld screen of this card
        sendResponse "q"

        wait 100
        --sendSPort "ls -lf" & return
        if card fld head of this card ≠ empty then

            get the loc of card button "save messages"
            end if

            click at it
        end if
    end if
    set lockscreen to true
    go card interface
    set lockscreen to false
end mouseUp

----- BUTTON: card button "Saved Addresses"

on mousedown

    put the mouseLoc into myPlace
    put item 1 of myPlace - 10 into horiz
    put item 2 of myPlace - -20 into vert

    get PopUpMenu(card fld savedaddress,5, vert, horiz)
    if the optionKey is down then
        put it into fun
        delete item fun of card fld savedaddress
        answer "Address was removed"
        exit mouseDown
    end if

    if it is 0 then
        answer "this is a pop up menu"
        exit mousedown

    end if
    put it into fun
    put item fun of card fld savedaddress into card fld "To"
end mousedown

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "error"

on mouseUp

```

hide me
end mouseUp

----- CARD SCRIPT: file view -----

on opencard
 global morecount
 set lockscreen to true
 set the lockText of fld screen to false
 put empty into morecount
 set the rect of fld screen to 0,57,512,297

 set lockscreen to false
end opencard
on closeCard
 set lockscreen to true
 set the lockText of fld screen to true
 set the rect of fld screen to 0,0,261,342

 set lockscreen to false
end closeCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Show More"

on mouseUp
 global TCPconnection
 if fld screen of this card is not empty then

 put fld screen into dataget
 --put xStrip(dataget,linefeed) into dataget
 --CgRet dataget,60
 put dataget into fld screen
 end if

 put number of characters in fld screen into numofchars
 -- Since HyperCard fields can only hold some many characters you need to
 -- check it every once and awhile and empty the field out.

 if numofchars > 25000 then
 answer "Field will be cleared this time, it's getting full!"
 with "Cancel" or "OK"
 if it = "Cancel" then
 exit mouseUp
 else
 put empty into fld screen
 put empty into morecount
 end if
 end if

 --delete last line of fld screen
 delete last character of fld screen
 --delete last character of fld screen
 sendResponse ""
 set cursor to watch
 wait 100
 --delete last character of fld screen

 sendResponse ""

end mouseUp

----- BUTTON: card button "print"

```
on mouseUp
  put fld screen of this card into x
  printText x
  hide msg
end mouseUp
```

----- BUTTON: card button "UNIX interface"

on mouseUp

```
  put the short name of this card into cardname
  if cardname = "file view" then
    if fld screen of this card = empty then
      put empty into fld screen of this card
      sendResponse "q"

      wait 100

      set lockscreen to true
      go card interface
    end if
  end if

  set lockscreen to true
  go card interface
  set lockscreen to false
end mouseUp
```

----- CARD SCRIPT: themessage -----

```
on openCard
  set lockscreen to true
  put empty into card fld error
  put empty into fld screen of this card
  set the rect of fld screen to 0,0,512,210
end openCard
on CloseCard
  global mbox
  put empty into mbox
end CloseCard
```

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Show message"

on mousedown

```
  global includedfile
  put the mouseLoc into myPlace
  put item 1 of myPlace - 10 into horiz
  put item 2 of myPlace - 20 into vert
  put card fld thelist into list
  get PopUpMenu(list,5, vert, horiz)
```

```

if it is 0 then
    answer "Click and hold button, this is a pop-menu"
    exit mousedown
end if

if it > 0 then
    put empty into fld screen of this card
    SendResponse "t" & it & return
    put "message" & it into card fld msgname
else

    exit mousedown
end if

end mousedown

----- BUTTON: card button "Save"

on mouseUp
    global messnum, TCPconnection

    put empty into lax

    answer "Save messages ?" with "Yes" or "cancel"
    if it is "cancel" then exit mouseUp
    if it is "Yes" then
        put 1 into count
        repeat messnum times
            answer "Do you want to save message" & count & "?" with "No" or "Yes"
            if it is "No" then
                put 1 into lax
                add 1 to count
            next repeat
        end if

        ask "Name message" & count & "(one word)"
        if it is empty then exit mouseUp
        put it into namer
        put the number of words in namer into goof
        if goof > 1 then
            answer "I told you one word"

            put word 1 of namer into namer
            answer "File is named" & namer
        end if

        SendResponse "m" & count & return
        wait 100

        SendResponse namer & return
        wait 100
        put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput
        repeat until newInput is empty

            put TCPRecvUpTo(TCPconnection, linefeed, 0, empty) into newInput

```

```

        put newInput after last character of card field error
    end repeat
    put FindInField(card fld error,"Confirm",false) into fun
    if fun is not "0,0,0,0" then

        SendResponse "y" & return
    end if
    add 1 to count
    put empty into card fld error
end repeat
end if

wait 50
SendResponse "e"& return
answer messnum&"message(s) taken care of"
if lax = 1 then

    answer "Unsaved messages are in the user mbox"
end if

--put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput
--repeat until newInput is empty

--put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput
beep 1
--put newInput into trash
--end repeat
put 1 into card fld save
go to card interface

end mouseUp

----- BUTTON: card button "Delete"

on mouseUp
    global messnum,TCPconnection,mbox
    if mbox = 1 then

        subtract 1 from messnum
        answer "Can't delete mbox, one message will be left" with "Yes"
        put empty into mbox
    else

        answer "Delete all messages?" with "Yes" or "cancel"
    end if

    if it is "cancel" then exit mouseUp
    if it is "Yes" then
        answer "Last chance" with "OK" or "Cancel"
        if it is "Cancel" then exit mouseUp
        repeat with x = 1 to messnum
            wait 50
            SendResponse "d"&x & return
            wait 20
        end repeat
    end repeat
end repeat

```

```

    answer "All messages deleted"
    put empty into messnum
    SendResponse "e"&return
    wait 20
    put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput
    repeat until newInput is empty

        put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput

        put newInput into trash
    end repeat

    put empty into fld screen
    set lockscreen to true
    set cursor to watch
    go to card interface

    put 1 into card fld save of card "themessage"

end if
end mouseUp

----- BUTTON: card button "Headers all"

on mouseUp

    put empty into fld screen of this card
    wait 50
    SendResponse "ha" & return
    set scroll of fld screen to 1
end mouseUp

----- BUTTON: card button "UNIX Interface"

on mouseUp
    if fld receiving > 0 then
        answer "Wait a second, still listing data"
        exit mouseUp
    end if

    put the short name of this card into cardname
    if cardname = "themessage" then
        if card fld save is empty then
            answer "You need to save or delete messages"
            exit mouseUp
        else
            go to card interface
        end if
    end if
end mouseUp

----- BUTTON: card button "Print screen"

on mouseUp
    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"
    end if
end mouseUp

```

```

        exit mouseUp
    end if

    put fld screen into x
    printText x
    hide msg
end mouseUp

----- BUTTON: card button "Exit"

on mouseUp
    global TCPconnection

    SendResponse "e" & return
    wait 50
    put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput
    repeat until newInput is empty

        put TCPRecvUpTo(TCPconnection,linefeed,0,empty) into newInput

    put newInput into trash
    end repeat
    answer "Messages are in the user mbox"
    set lockscreen to true
    go to card interface
    put 1 into card fld save of card themessage
end mouseUp

----- BUTTON: card button "Answer"

on mouseUp

    get FindInField(fld screen,"From","false",0)
    if it is "0,0,0,0" then
        answer "Try showing the message"
        exit mouseUp
    end if

    put item 2 of it into linenum
    put word 2 of line linenum of fld screen into address
    answer "Send to" & address with "OK" or "No"
    if it is "No" then
        exit mouseUp
    end if
    SendResponse "e" & return
    wait 100
    go to card "mail card"
    put address into card fld "To"
    put empty into card fld subject
    put 1 into card fld save of card themessage

end mouseup

----- BUTTON: card button "Save address"

on mouseUp
    get FindInField(fld screen,"From","false",0)

```

```

if it is "0,0,0,0" then
    answer "Try showing the message"
    exit mouseUp
end if

```

```

put item 2 of it into linenum
put word 2 of line linenum of fld screen into address
put address$"," after last character of card fld savedaddress of card "Mail card"
answer "address has been saved"
end mouseUp

```

----- CARD SCRIPT: Groups -----

```

on openCard
    hide fld screen
    hide fld receiving
    hide bg button home
end openCard
on closeCard
    show bg button home
    show fld screen
    show fld receiving
end closeCard

```

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "New Button"

```

on mouseUp
    go card "mail card"
end mouseUp

```

----- BUTTON: card button "UNIX Interface"

```

on mouseUp

    put the short name of this card into cardname
    if cardname = "file view" then
        if fld screen of this card * empty then
            put empty into fld screen of this card
            CntrZ
            put the result into fun
            sendResponse fun

            wait 100
            --sendSPort "ls -l" & return
            go card interface
        end if
    end if
    if cardname = "read mail" then
        if fld screen of this card * empty then
            put empty into fld screen of this card
            CntrZ
            put the result into fun
            sendResponse fun

            wait 100
            --sendSPort "ls -l" & return
            if card fld head of this card * empty then

```

```
        get the loc of card button "save messages"  
    end if  
  
    click at it  
end if  
end if  
set lockscreen to true  
go card interface  
set lockscreen to false  
end mouseUp
```

APPENDIX C

HEL'S HYPERCARD UNIX E-MAIL SYSTEM (HUES)
SOURCE CODE - MODEM VERSION

HEL'S HYPERCARD UNIX E-MAIL SYSTEM (HUES)
SOURCE CODE - MODEM VERSION

```
-- The source code for this version is very similar to the MacTCP version
-- although it is much simpler, especially the login routines, and
-- character handling code.
-- This stack constantly checks to see if any characters are available
-- in it's idle handler. If so, it simply puts them into the screen field.
----- STACK: HUES-ModemVersion -----
```

```
----- STACK SCRIPT: -----
```

```
on openStack
  put the seconds into card fld theTime of card interface

  hide menubar
  global SPortGlobals,logoutme
  put empty into logoutme
  set lockscreen to true
  go to card Preferences
  show message box
  set userModify to true
  configureSPort modemPort,baud1200,data8,stop10,parityOff,stripOn,-
  stripControlsOff
  setSPortBufferSize 10240
  set hilite of card button "300 baud" of card Preferences to false
  set hilite of card button "1200 baud" of card Preferences to true
  set hilite of card button "2400 baud" of card Preferences to false
  set hilite of card button "9600 baud" of card Preferences to false
  set hilite of card button "Strip control characters" of card Preferences to false

  set hilite of card button "Strip parity bit" of card Preferences to true
  set the rect of fld screen to 125,160,414,237
  put empty into fld screen
  hide card button "Login"
  hide fld receiving
  put empty into cd fld msgname of card themessage
end openStack

on closeStack
  global logoutme
  put empty into card fld text of card "Mail Card"
  if logoutme is not empty then
    put "Must log you out to quit...." into msg
    put empty into logoutme
    set cursor to watch
    put empty into fld screen
    sendSPort "logout" & return
    wait 200
    sendSPort "logout" & return
    wait 200
    sendSPort "" -- send the proper modem commands
    wait 500
    closeSPort
    if the result is not empty then answer the result with "OK"
    wait 50
    sendSPort "ath" & return -- send the proper modem commands

    wait 50
    sendSPort "ath" & return
    wait 50
    answer "You have been logged out." with "OK"
```

```

        closeSPort
        if the result is not empty then answer the result with "OK"
        exit closeStack
    end if

end closeStack
on openBackground
    push recent card
end openBackground

on idle

    -- This is the main character handling function of the stack, which is much
    -- more simple than the TCP version.

    put recvUpTo(empty,0,empty) into newInput

    if newInput is not empty then
        put charsAvailable() into fld "receiving"

        put xStrip(newInput,linefeed) after last character in fld screen.

    end if

end idle

on returnKey
    sendSPort message box & return
    put empty into message box
end returnKey

on test
    repeat until the mouse is down
        sendSPort "1234567890." & return & linefeed
    end repeat
end test

on login
    global logoutme

    put card fld "login name" into name
    sendSPort name & return
    wait 200
    put CharsAvailable() into x
    put recvChars(x) into trash

    get ask("secret","Enter the user password:")
    sendSPort it & return

    wait 300
    put empty into card fld look
    put CharsAvailable() into x
    put recvChars(x) into cd fld look

    put FindInField(card fld look,"incorrect",false) into fun

    if fun is "0,0,0,0" then
        put 1 into logoutme
        set cursor to watch

        go to card interface
        put empty into fld screen of this card
    end if
end login

```

```

exit login

else
  put card fld "login name" into name
  sendSPort name & return
  wait 50
  put CharsAvailable() into x
  put recvChars(x) into trash

  get ask("secret","Password Incorrect:")
  sendSPort it & return
  wait 300
  put empty into card fld look
  put CharsAvailable() into x
  put recvChars(x) into cd fld look
  put FindInField(card fld look,"incorrect",false) into fun

  if fun is "0,0,0,0" then
    put 1 into logoutme
    set cursor to watch
    go to card interface
    exit login
  else
    set the loc of msg to 10,300
    put "incorrect passwords given" into msg
    sendSPort "+++"
    wait 500
    put "Closing connection" into msg
    closeSPort
    sendSPort "ATH" & return
    wait 100
    sendSPort "ATH" & return
    hide card button login
    hide msg
    exit login
  end if
end if
end login

```

----- BACKGROUND: bkgnd id 2599 -----

----- BACKGROUND BUTTON SCRIPTS -----

----- BUTTON: bkgnd button "Home"

```

on mouseUp
  global logoutme

  if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
  end if

```

```

visual effect iris open
go Home
end mouseUp

```

```

----- BUTTON: bkqnd button "Strip control characters"

on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp

```

```

----- BUTTON: bkqnd button "Strip parity bit"

on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp

```

```

----- BUTTON: bkqnd button "Prev"

on mouseUp
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"

    exit mouseUp
  end if

  put the short name of this card into cardname
  if cardname = "file view" then
    if fld screen of this card ≠ empty then
      put empty into fld screen of this card
      CntrZ
      put the result into fun
      sendSPort fun

      wait 100
      sendSPort "ls -l" & return
    end if

```

```

  end if

```

```

  set lockscreen to true
  go to prev card

```

```

  hide msg
end mouseUp

```

```

----- BACKGROUND FIELD SCRIPTS -----

```

```

----- FIELD: bkqnd field "screen"

```

```

on mouseUp
  -- This is essentially same as the MacTCP version (See appendix A)
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
    - top of me)/textheight of me))+1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun

    sendSPort "cd " & fun & return

```

```

put empty into fld screen
set cursor to watch
set the loc of msg to 10,300
put "Checking for file or directory type..." into msg
wait 300

put charsAvailable() into z
put recvChars(z) into gotit

if last word of line 2 of gotit is "directory" then

    sendSPort "more -f -l"&fun & return
    put fun into card fld filename of card "file view"
    set the rect of me to 0,57,512,297
    go to card "file view"
    put empty into fld screen
else

    sendSPort "ls -l" & return
    set the scroll of me to 1
end if
end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 8414 -----
----- BACKGROUND BUTTON SCRIPTS -----
----- BUTTON: bkgnd button "Home"

on mouseUp
    global logoutme

    if logoutme is not empty then
        answer "You must logout to go home" with "OK"
        exit mouseUp
    end if

    visual effect iris open
    go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

on mouseUp
    if the hilite of me then configureSPort stripControlsOn
    else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
    if the hilite of me then configureSPort stripOn
    else configureSPort stripOff
end mouseUp

```

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

on mouseUp

put the short name of this card into cardname
if cardname = "interface" then

put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
- top of me)/textheight of me))+1 into theLine
select line theLine of fld screen

put last word of line theLine of fld screen into fun

sendSPort "cd \"%fun & return
put empty into fld screen
set cursor to watch
set the loc of msg to 10,300
put "Checking for file or directory type..." into msg
wait 300

put charsAvailable() into z
put recvChars(z) into gotit

if last word of line 2 of gotit is "directory" then

sendSPort "more -f -1" & return
put fun into card fld filename of card "file view"
set the rect of me to 0,57,512,297
go to card "file view"
put empty into fld screen
else

sendSPort "ls -l" & return
set the scroll of me to 1

end if
end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 7265 -----

----- BACKGROUND BUTTON SCRIPTS -----

----- BUTTON: bkgnd button "Home"

on mouseUp

global logoutme

if logoutme is not empty then
answer "You must logout to go home" with "OK"
exit mouseUp
end if

visual effect iris open
go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

```
on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp
```

----- BUTTON: bkgnd button "Strip parity bit"

```
on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp
```

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

```
on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
    - top of me)/textheight of me))+1 into theline
    select line theline of fld screen

    put last word of line theline of fld screen into fun

    sendSPort "cd " & fun & return
    put empty into fld screen
    set cursor to watch
    set the loc of msg to 10,300
    put "Checking for file or directory type..." into msg
    wait 300

    put charsAvailable() into z
    put recvChars(z) into gotit

    if last word of line 2 of gotit is "directory" then

      sendSPort "more -f -l" & fun & return
      put fun into card fld filename of card "file view"
      set the rect of me to 0,57,512,297
      go to card "file view"
      put empty into fld screen
    else

      sendSPort "ls -l" & return
      set the scroll of me to 1
    end if
  end if

  hide msg
end mouseUp
```

----- BACKGROUND: bkgnd id 5891 -----

----- BACKGROUND BUTTON SCRIPTS -----

```

----- BUTTON: bkgnd button "Home"

on mouseUp
  global logoutme

  if logoutme is not empty then
    answer "You must logout to go home" with "OK"
    exit mouseUp
  end if

  visual effect iris open
  go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp

----- BACKGROUND FIELD SCRIPTS -----

----- FIELD: bkgnd field "screen"

on mouseUp
  put the short name of this card into cardname
  if cardname = "interface" then

    put (trunc((scroll of me /textheight of me) + (item 2 of the mouseloc -
    - top of me)/textheight of me))+1 into theLine
    select line theLine of fld screen

    put last word of line theLine of fld screen into fun

    sendSPort "cd " & fun & return
    put empty into fld screen
    set cursor to watch
    set the loc of msg to 10,300
    put "Checking for file or directory type..." into msg
    wait 300

    put charsAvailable() into z
    put recvChars(z) into gotit

    if last word of line 2 of gotit is "directory" then

      sendSPort "more -f -l" & fun & return
      put fun into card fld filename of card "file view"
      set the rect of me to 0,57,512,297
      go to card "file view"
      put empty into fld screen

```

```

else

    sendSPort "ls -l" & return
    set the scroll of me to 1
    end if
end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 4774 -----
----- BACKGROUND BUTTON SCRIPTS -----
----- BUTTON: bkgnd button "Home"

on mouseUp
    global logoutme

    if logoutme is not empty then
        answer "You must logout to go home" with "OK"
        exit mouseUp
    end if

    visual effect iris open
    go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

on mouseUp
    if the hilite of me then configureSPort stripControlsOn
    else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
    if the hilite of me then configureSPort stripOn
    else configureSPort stripOff
end mouseUp

----- BACKGROUND FIELD SCRIPTS -----
----- FIELD: bkgnd field "screen"

on mouseUp
    put the short name of this card into cardname
    if cardname = "interface" then

        put (trunc((scroll of me /textheight of me) + (item 2 of the mouseloc -
        - top of me)/textheight of me))+1 into theLine
        select line theLine of fld screen

        put last word of line theLine of fld screen into fun
        if last character of fun = "/" then

```

```

sendSPort "cd %fun & return
put empty into fld screen
set cursor to watch
sendSPort "ls -l" & return
set the scroll of me to 1

else

sendSPort "more -f -l"%fun & return
put fun into card fld filename of card "file view"
set lockscreen to true
set the rect of me to 0,57,512,297

go to card "file view"
put empty into fld screen
end if

end if

hide msg
end mouseUp

----- BACKGROUND: bkgnd id 2845 -----
----- BACKGROUND BUTTON SCRIPTS -----
----- BUTTON: bkgnd button "Home"

on mouseUp
global logoutme

if logoutme is not empty then
answer "You must logout to go home" with "OK"
exit mouseUp
end if

visual effect iris open
go Home
end mouseUp

----- BUTTON: bkgnd button "Strip control characters"

on mouseUp
if the hilite of me then configureSPort stripControlsOn
else configureSPort stripControlsOff
end mouseUp

----- BUTTON: bkgnd button "Strip parity bit"

on mouseUp
if the hilite of me then configureSPort stripOn
else configureSPort stripOff
end mouseUp

----- BACKGROUND FIELD SCRIPTS -----

```

----- FIELD: bkgnd field "screen"

on mouseUp

put the short name of this card into cardname
if cardname = "interface" then

put (trunc((scroll of me /textheight of me) + (item 2 of the mouseLoc -
- top of me)/textheight of me))+1 into theLine
select line theLine of fld screen

put last word of line theLine of fld screen into fun

sendSPort "cd " & fun & return
put empty into fld screen
set cursor to watch
set the loc of msg to 10,300
put "Checking for file or directory type..." into msg
wait 300

put charsAvailable() into z
put recvChars(z) into gotIt

if last word of line 2 of gotIt is "directory" then

sendSPort "more -f -l" & fun & return
put fun into card fld filename of card "file view"
set the rect of me to 0,57,512,297
go to card "file view"
put empty into fld screen
else

sendSPort "ls -l" & return
set the scroll of me to 1
end if
end if

hide msg
end mouseUp

----- CARD SCRIPT: Preferences -----

on openCard

hide card button "300 baud"
hide card button "1200 baud"
hide card button "2400 baud"
hide card button "9600 baud"
hide card button "Strip control characters"
hide card button "Strip parity bit"
show card button "Advanced Users Only"
put ath into msg
hide msg
set lockscreen to true
put empty into fld screen
show fld screen
set the rect of fld screen to 126,165,393,208
hide fld receiving
set lockscreen to false

end openCard

on closeCard

show fld receiving

end closeCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Dial"

```
on mouseUp
  put empty into fld screen
  put ath into msg
  hide msg
  answer "Dial phone number?" with "No" or "Yes"
  if it is "Yes" then
    answer "Click on the login button after connected" with "Cancel" or "OK"
    if it is "Cancel" then exit mouseUp
    put empty into fld "screen" of card interface
    put card fld "phone number" into pnumber
    sendSPort "ATDT"&pnumber & return

    show card button "login"
  end if
end mouseUp
```

end mouseUp

----- BUTTON: card button "300 baud"

```
on mouseUp
  configureSPort baud300
  set hilite of cd button "1200 baud" to false
  set hilite of cd button "2400 baud" to false
  set hilite of cd button "9600 baud" to false
  set hilite of cd button "300 baud" to true
end mouseUp
```

----- BUTTON: card button "1200 baud"

```
on mouseUp
  configureSPort baud1200
  set hilite of cd button "300 baud" to false
  set hilite of cd button "2400 baud" to false
  set hilite of cd button "9600 baud" to false
  set hilite of cd button "1200 baud" to true
end mouseUp
```

----- BUTTON: card button "2400 baud"

```
on mouseUp
  configureSPort baud2400
  set hilite of cd button "300 baud" to false
  set hilite of cd button "1200 baud" to false
  set hilite of cd button "9600 baud" to false
  set hilite of cd button "2400 baud" to true
end mouseUp
```

----- BUTTON: card button "9600 baud"

```
on mouseUp
  configureSPort baud9600
  set hilite of cd button "300 baud" to false
  set hilite of cd button "1200 baud" to false
```

```

set hilite of cd button "2400 baud" to false
set hilite of cd button "9600 baud" to true
end mouseUp

```

```

----- BUTTON: card button "Strip control characters"

on mouseUp
  if the hilite of me then configureSPort stripControlsOn
  else configureSPort stripControlsOff
end mouseUp

```

```

----- BUTTON: card button "Strip parity bit"

on mouseUp
  if the hilite of me then configureSPort stripOn
  else configureSPort stripOff
end mouseUp

```

```

----- BUTTON: card button "New Button"

on mouseUp
  show card fld help
end mouseUp

```

```

----- BUTTON: card button "Advanced Users Only"

on mouseUp
  hide me
  show card button "300 baud"
  show card button "1200 baud"
  show card button "2400 baud"
  show card button "9600 baud"
  show card button "Strip control Characters"
  show card button "Strip parity bit"
end mouseUp

```

```

----- BUTTON: card button "Login"

on mouseUp
  global logoutme
  put empty into card fld remove of card interface
  put empty into card fld look
  -- Must wait for a connection before the code can execute any of the
  -- login routines.
  put FindInField(fld screen,"Connect",false) into fun
  if fun is "0,0,0,0" then
    sendSPort " " & return
    wait 100
  put FindInField(fld screen,"Login",false) into fun
  if fun is "0,0,0,0" then
    sendSPort " " & return
    wait 100

  put FindInField(fld screen,"Password",false) into fun
  if fun is "0,0,0,0" then
    put "Could not get login prompt... Proceeding with care" into msg
    wait 50
    login

```



```

hide bg button home
end openCard
on closeCard
  show bg button home
  show fid screen
  show fid receiving
end closeCard

```

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "New Button"

```

on mouseUp
  go card "mail card"
end mouseUp

```

----- BUTTON: card button "UNIX Interface"

```

on mouseUp

  put the short name of this card into cardname
  if cardname = "file view" then
    if fid screen of this card = empty then
      put empty into fid screen of this card
      Cntr2
      put the result into fun
      sendSPort fun

      wait 100
      --sendSPort "is -1" & return
      go card interface
    end if
  end if
  if cardname = "read mail" then
    if fid screen of this card = empty then
      put empty into fid screen of this card
      Cntr2
      put the result into fun
      sendSPort fun

      wait 100
      --sendSPort "is -1" & return
      if card fid head of this card = empty then
        get the loc of card button "save messages"
        end if

        click at it
        end if
      end if
      set lockscreen to true
      go card interface
      set lockscreen to false
    end mouseUp

```

----- BUTTON: card button "New Button"

```

on mouseUp

  show card fid help

```

end mouseUp

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "Help"

on mouseUp
hide me
end mouseUp

----- CARD SCRIPT: the message -----

on openCard
set lockscreen to true
put empty into card fld error
put empty into fld screen of this card
set the rect of fld screen to 0,0,512,210
end openCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Show message"

on mousedown
if fld receiving > 0 then
answer "Wait a second, still gathering data!"

exit mouseUp
end if

global includedFile
put the mouseLoc into myPlace
put item 1 of myPlace - 10 into horiz
put item 2 of myPlace - 20 into vert
put card fld theList into List
get PopupMenuList, 5, vert, horiz

if it is 0 then
answer "Click and hold button, this is a pop-menu"
exit mousedown
end if

if it > 0 then
put empty into fld screen of this card
SendSPort ":" & it & return
put "message" & it into card fld msgname
else

exit mousedown
end if

end mousedown

----- BUTTON: card button "Save"

```

on mouseUp
  global messnum
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"

    exit mouseUp
  end if
  put empty into lax

  answer "Save messages ?" with "Yes" or "cancel"
  if it is "cancel" then exit mouseUp
  if it is "Yes" then
    put 1 into count
    repeat messnum times
      answer "Do you want to save message"&count&"?" with "No" or "Yes"
      if it is "No" then
        put 1 into lax
        add 1 to count
      next repeat
    end if

    ask "Name message"&count&" (one word)"
    if it is empty then exit mouseUp
    put it into namer
    put the number of words in namer into goof
    if goof > 1 then
      answer "I told you one word"

      put word 1 of namer into namer
      answer "File is named"&namer
    end if

    sendSPort "m"&count & return
    wait 100

    sendSPort namer & return
    wait 100
    put rcvUpTo(linefeed,0,empty) into newInput
    repeat until newInput is empty

      put rcvUpTo(linefeed,0,empty) into newInput

    put newInput after last character of card field error
    end repeat
    put FindInField(card fld error,"Confirm",false) into fun
    if fun is not "0,0,0,0" then

      sendSPort "y" & return
    end if
    add 1 to count
    put empty into card fld error
    end repeat
  end if

  wait 50
  sendSPort "e" & return
  answer messnum&"message(s) taken care of"
  if lax = 1 then

    answer "Unsaved messages are in the user mbox"
  end if

```

```

put recvUpTo(linefeed,0,empty) into newInput
repeat until newInput is empty

    put recvUpTo(linefeed,0,empty) into newInput

    put newInput into trash
end repeat
put 1 into card fld save
go to card interface

```

end mouseUp

***** BUTTON: card button "Delete"

```

on mouseUp
    global messnum
    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"

        exit mouseUp
    end if
    get FindInField(card fld msgname,"-box",true,0)
    if item 1 of it > 0 then
        subtract 1 from messnum
        answer "Can't delete mbox, one message will be left"
    end if

```

answer "Delete all messages?" with "Yes" or "cancel"

```

if it is "cancel" then exit mouseUp
if it is "Yes" then
    answer "Last chance" with "OK" or "Cancel"
    if it is "Cancel" then exit mouseUp
    repeat with x = 1 to messnum
        wait 50
        SendSPort "d"&x & return
        wait 20
    end repeat

```

end repeat

```

answer "All messages deleted"
put empty into messnum
SendSPort "e"&return
wait 20
put recvUpTo(linefeed,0,empty) into newInput
repeat until newInput is empty

    put recvUpTo(linefeed,0,empty) into newInput

    put newInput into trash
end repeat

put empty into ld screen
set lockscreen to true
set cursor to watch
go to card interface

```

```

    put 1 into card fld save of card "themessage"

    end if
end mouseUp

----- BUTTON: card button "Headers all"

on mouseUp
    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"

        exit mouseUp
    end if

    put empty into fld screen of this card
    wait 50
    SendSPort "ha" & return
    set scroll of fld screen to 1
end mouseUp

----- BUTTON: card button "UNIX InterFace"

on mouseUp
    if fld receiving > 0 then
        answer "Wait a second, still listing data"
        exit mouseUp
    end if

    put the short name of this card into cardname
    if cardname = "themessage" then
        if card fld save is empty then
            answer "You need to save or delete messages"
            exit mouseUp
        else
            go to card interface
        end if
    end if
end mouseUp

----- BUTTON: card button "Print screen"

on mouseUp
    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"

        exit mouseUp
    end if

    put fld screen into x
    printText x
end mouseUp

----- BUTTON: card button "Exit"

on mouseUp

```

```

if fld receiving > 0 then
  answer "Wait a second, still gathering data!"

  exit mouseUp
end if
sendSPort "e" & return
wait 50
put recvUpTo(linefeed,0,empty) into newInput
repeat until newInput is empty

  put recvUpTo(linefeed,0,empty) into newInput

  put newInput into trash
end repeat
answer "Messages are in the user mbox"
set lockscreen to true
go to card interface
put 1 into card fld save of card themessage
end mouseUp

----- BUTTON: card button "Answer"

on mouseUp

  get FindInField(fld screen,"From","false",0)
  if it is "0,0,0,0" then
    answer "Try showing the message"
    exit mouseUp
  end if

  put item 2 of it into linenum
  put word 2 of line linenum of fld screen into address
  answer "Send to" & address with "OK" or "No"
  if it is "No" then
    exit mouseUp
  end if
  sendSPort "e" & return
  wait 100
  go to card "mail card"
  put address into card fld "To"
  put empty into card fld subject
  put 1 into card fld save of card themessage

end mouseUp

----- BUTTON: card button "New Button"

on mouseUp

  show card fld help
end mouseUp

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "Help"

on mouseUp

```

```

hide me
end mouseUp

```

```

----- CARD SCRIPT: Mail Card -----

```

```

on opencard
    hide bg btn 1
    put empty into fld screen
    put the time into card fld time
    show card fld "time out"
    hide fld "receiving"
    hide fld "screen"
end openCard

```

```

on closeCard
    show bg btn 1
    show fld screen
    show fld "receiving"
end closeCard

```

```

----- CARD BUTTON SCRIPTS -----

```

```

----- BUTTON: card button "Send"

```

```

-- This sends a message in a HyperCard field to the SUN and is send as a
-- message. The important thing here is to strip out any carriage returns that
-- might be in the fields and add them in the correct places for the SUN. The
-- fields in hypercard have a word wrap function so a message can be typed
-- with no carriage returns. The CgRet external command adds carriage returns
-- to the end of a variable of so many characters, in this case 60. It also
-- checks to make sure that it does not break up any words, or ruin any
-- existing carriage returns.

```

```

on mouseUp

```

```

    global includedfile
    set the loc of the msg to 10,300
    put the time into card fld time
    if includedfile is empty then
        answer "Send this message?" with "No" or "Yes"
        if it is "No" then
            exit mouseUp
        end if
    end if

```

```

    set cursor to watch
    if card fld "To" is empty the
        answer "There is no name to send to!" with "Sorry"
        exit mouseUp
    end if
    -- xStrip and stripLastReturn both strip out the carriage returns
    put card fld "To" into person
    put xStrip(person,return) into temp
    stripLastReturn temp
    put the result into card fld "To"
    put card fld "To" into person

    put card fld "cc" into x

```

```

put xStrip(x,return) into x
striplastrReturn x
put the result into cd fld "cc"
put cd fld "cc" into carbon

put "Sending message...." into msg

put card fld "text"&return into y
if y is empty then
    answer "There is no text to send!"
    exit mouseUp
end if

CgRet y,60
put the result into fun
put y into card fld "text"

SendSPort "send" & return
wait 100
-- This routine is used if the person is sending e-mail to more than
-- one person.

put the number of items in card fld "To" into linenum
if linenum > 1 then
    repeat with x = 1 to linenum
        SendSPort item x of cd fld "To"&"\ "& return
        wait 100
        put "Sending to:"&item x of card fld "To" into msg
        wait 45
    end repeat
    SendSPort "" & return
else
    SendSPort person & return
end if
wait 100

SendSPort person & return
wait 100

put the number of items in card fld "cc" into linenum
if linenum > 1 then
    repeat with x = 1 to linenum
        SendSPort item x of cd fld "cc"&"\ "& return
        wait 100
        put "Sending to:"&item x of card fld "cc" into msg
        wait 45
    end repeat
    SendSPort "" & return
else
    SendSPort carbon & return
end if
wait 100

put card fld "Subject" into sub
put xStrip(sub,return) into temp
striplastrReturn temp
put the result into cd fld "Subject"
put card fld "Subject" into sub

SendSPort sub & return

```

```

wait 100

put "Sending message ..." into msg
SendSPort fun & return
wait 100

CntrD
put the result into fun
SendSPort fun & return
wait 100
put "sending control d..." into msg

if includedfile is not empty then
    answer "include file " & includedfile & "?" with "No" or "Yes"
    if it is "Yes" then
        set cursor to watch
        SendSPort "file include" & return
        wait 100
        SendSPort includedfile & return
        put empty into includedfile
    end if
end if

SendSPort "send" & return

wait 500
put empty into cd fld error

put recvUpTo(linefeed,0,empty) into newInput
repeat until newInput is empty

    put recvUpTo(linefeed,0,empty) into newInput

    put newInput after last character of card field error
end repeat

put FindInField(card fld error,"Message Posted",false) into errortime
if errortime = "0,0,0,0" then
    beep 1
    answer "Message was sent successfully" with "Great!"
else
    beep 1
    SendSPort "quit" & return
    wait 100
    SendSPort "y" & return
    answer "Sorry an error of some kind, see field below" with "Darn"
    set lockscreen to true
    show card fld error
end if

put "*****Click on field to close*****" after last line in card fld error
put the number of lines of card fld error into fun
multiply fun by 10
set the scroll of card fld error to fun

hide msg

```

end mouseUp

```
----- BUTTON: card button "Update"
-- must do this so that the SUN knows we are still logged in.
on mouseUp
  put the time into card fld time
  sendSPort " " & return
end mouseUp
```

```
----- BUTTON: card button "Options"
```

```
on mousedown
  if fld receiving > 0 then
    answer "What a second, still gathering data"
    exit mousedown
  end if
```

```
global includedfile
put the mouseLoc into myPlace
put item 1 of myPlace - 10 into horiz
put item 2 of myPlace - -20 into vert
```

```
get PopUpMenu("Address;Edit Addresses;Include File;Change Directory;Move Up Directory;Clear
Field",5, vert, horiz)
if it is 0 then answer "This is a pop up menu"
if it is 1 then
```

```
  set lockscreen to true
  put card fld time into savedtime
  go to card "groups"
```

```
  put empty into card fld fun
  repeat with x = 1 to the number of lines in card fld "group name"
    put item 1 of line x of card fld "group name"&" " after last character of card fld fun
  end repeat
```

```
  put card fld fun into theList
```

```
  DoList 999,card fld fun,one
  put the result into fun
  if fun is empty then
    go card "mail card"
    put savedtime into card fld time
    hide card fld "time out"
    exit mouseDown
```

```
  end if
```

```
  put item 2 of fun into gotit
```

```
  get FindInField(card field "group name",gotit,"true",0)
  put item 2 of it into linenumber
  if it is 0 then
    answer "error of some kind"
  end if
  put 0 into start
  repeat forever
    get FindInField(card fld "group name","(", "true",start)
```

```

    if item 2 of it = linenumber then
        exit repeat
    else
        put item 1 of it into start

        next repeat
    end if
end repeat
put item 1 of it into firstspot
get FindInField(card fld "group name", "1", "true", firstspot)
put item 1 of it into secondspot

put character firstspot to secondspot of card fld "group name" into address
delete character 1 of address
delete last character of address
put empty into card fld "To" of card "mail card"
put address into card fld "To" of card "mail card"
go to card "mail card"

put savedtime into card fld "time out"
hide card fld "time out"
set lockscreen to false
end if

if it is 2 then
    go to card "groups"
end if

if it is 3 then
    set cursor to watch
    put "Please wait gathering file names..."
    set lockscreen to true
    if fld screen of this card is empty then
        sendSPort "ls -lf" & return
        wait 100
        put recvUpTo(linefeed, 0, empty) into newInput
        repeat until newInput is empty

            put recvUpTo(linefeed, 0, empty) into newInput

        put newInput after last character of field screen
    end repeat
end if

put empty into card fld remove
put fld screen of this card into dataget
put xStrip(dataget, linefeed) into fld screen of this card
put xgetScreen(fld screen, return) into dataget
put "Choose file...." into msg

put dataget into theList

```

```

if theList is empty then
    hide msg
    beep 1
    answer "No files in directory, try changing directory"
    exit mousedown
end if

DoList 999,theList,one
put the result into gotit

if gotit is empty then

    go card "mail card"
    hide card fld "time out"
    hide msg
    exit mousedown
end if

put item 2 of gotit into includedfile
answer "include" && includedfile with "Cancel" or "OK"
if it is "Cancel" then
    hide card fld "time out"
    hide msg
    exit mouseDown
end if

go to card "mail card"
hide card fld "time out"
answer "file" && includedfile && "included, send now?" with "Cancel" or "OK"
if it is "OK" then
    get the loc of card button "send"
    click at it
end if
hide msg
set lockscreen to false
end if
if it is 4 then

    set lockscreen to true
    if fld screen of this card is empty then
        go to card interface
        put fld screen into needit
        go to card "mail card"
        put needit into fld screen of this card
    end if

    put empty into card fld remove
    put fld screen of this card into dataget
    put xStrip(dataget,linefeed) into dataget
    put xgetScreen(dataget,return) into dataget

    if dataget = empty then
        answer "Try moving up a directory" with "OK"
        go card "mail card"
        hide card fld "time out"
        exit mouseDown
    end if

```

```

put dataget into theList
put "Choose directory...(words with a / are directories)..." into msg
DoList 999,theList,one
put the result into gotit

```

```

if gotit is empty then

```

```

    go card "mail card"
    hide card fld "time out"
    hide msg
    exit mousedown
end if

```

```

put item 2 of gotit into gotit
delete last character of gotit
answer "change directory to"&gotit with "No" or "ok"
if it is "No" then
    hide msg
    hide card fld "time out"
    exit mousedown
end if

```

```

SendSPort "cd "&gotit& return
wait 50
put empty into fld screen of this card
SendSPort "ls -lf" & return
wait 200
go to card "mail card"
hide card fld "time out"
hide msg

```

```

wait 100

```

```

set lockscreen to false
answer "Directory changed" with "Good"
end if
if it is 5 then
    answer "Move up a directory?" with "Yes" or "Cancel"
    if it is "Cancel" then exit mousedown

```

```

SendSPort "cd .."& return
wait 100
set lockscreen to true
put empty into fld screen
SendSPort "ls -lf"& return
wait 200

```

```

answer "Directory changed" with "OK"
hide card fld "time out"
end if

```

```

if it is 6 then
    put empty into card fld text
end if

```

```

end mousedown

```

----- BUTTON: card button "UNIX InterFace"

on mouseUp

```
    put the short name of this card into cardname
    if cardname = "file view" then
        if fld screen of this card * empty then
            put empty into fld screen of this card
            Cntr2
            put the result into fun
            sendSPort fun

            wait 100
            --sendSPort "ls -l" & return
            go card interface
        end if
    end if
    if cardname = "read mail" then
        if fld screen of this card * empty then
            put empty into fld screen of this card
            Cntr2
            put the result into fun
            sendSPort fun

            wait 100
            --sendSPort "ls -l" & return
            if card fld head of this card * empty then

                get the loc of card button "save messages"
            end if

            click at it
        end if
    end if
    set lockscreen to true
    go card interface
    set lockscreen to false
end mouseUp
```

----- BUTTON: card button "New Button"

on mouseUp

```
    show card fld help
end mouseUp
```

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "error"

```
on mouseUp
    hide me
end mouseUp
```

----- FIELD: card field "time out"

```
on mouseUp
  hide me
end mouseUp
```

----- FIELD: card field "Help"

```
on mouseUp
  hide me
end mouseUp
```

----- CARD SCRIPT: interface -----

```
on openCard
  put empty into fld screen
  sendSPort "ls -lf" & return
  set lockscreen to true
  set the rect of fld screen to 0,34,260,342
  set the loc of fld receiving to 450,27
  show fld receiving
  set lockscreen to false
end openCard

on mouseUp
  if fld receiving of this card > 0 then
    answer "Wait a second, still receiving data"
    exit mouseUp
  end if
end mouseUp
```

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Send UNIX Command"

```
on mouseUp
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"

    exit mouseUp
  end if

  ask "Send what UNIX command"
  put empty into fld screen
  sendSPort it & return
end mouseUp
```

----- BUTTON: card button "List"

```
on mouseUp
  put empty into fld screen
  sendSPort "ls -lf" & return
  wait 50

  set the scroll of fld screen to 1
end mouseUp
```

----- BUTTON: card button "Logout"

```
on mouseUp
  global logoutme
  answer "Logout now?" with "NO" or "Yes"
  if it is "Yes" then

    put empty into logoutme
    set cursor to watch
    put empty into fld screen
    sendSPort "logout" & return
    wait 200
    sendSPort "logout" & return
    wait 200
    sendSPort "+++"
    wait 500
    closeSPort
    if the result is not empty then answer the result with "OK"
    wait 50
    sendSPort "ath" & return
    wait 50
    answer "You have been logged out." with "OK"

    closeSPort
    if the result is not empty then answer the result with "OK"
  end if

end mouseUp
```

----- BUTTON: card button "Move Up"

```
on mouseUp
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"

    exit mouseUp
  end if

  sendSPort "cd .." & return
  wait 100
  put empty into fld screen
  sendSPort "ls -lf" & return
  set the scroll of fld screen to 1

end mouseUp
```

----- BUTTON: card button "Transfer File"

```
-- The transfer file routine is similar to the MacTCP version only it is
-- more simple because a separate FTP connection does not have to be opened
-- first. This uses xmodem protocol. I think other protocols can be used
-- but I have not tried them.

on mouseUp
  global logoutme
  if fld receiving > 0 then
    answer "Wait a second, still gathering data!"

    exit mouseUp
```

```

end if

put empty into secon
put empty into card fld remove
answer "Send to or Receive from UNIX system" with "Send" or "Receive" or "Cancel"
if it is "Cancel" then
    exit mouseUp
end if

if it is "Send" then
    answer "Has the file been converted to HEX format" with "What?" or "No" or "Yes"
    if it is "What?" then
        show card fld "HEX"
        exit mouseUp
    end if
    if it is "No" then
        answer "Open the BinHex conversion application?" with "Cancel" or "No" or "Yes"
        if it is "Cancel" then exit mouseUp
        if it is "Yes" then
            put empty into logoutme
            set cursor to watch
            put empty into fld screen
            sendSPort "logout" & return
            wait 200
            sendSPort "logout" & return
            wait 200
            sendSPort "+++"
            wait 500
            closeSPort
            if the result is not empty then answer the result with "OK"
            wait 50
            sendSPort "ath" & return

            wait 50
            sendSPort "ath" & return
            wait 50
            answer "You have been logged out." with "OK"

            closeSPort
            if the result is not empty then answer the result with "OK"

            open "binHex 4.0"
            exit mouseUp
        end if
    end if

    put filename() into theFile
    put theFile into stripper
    if theFile is empty then
        hide msg
        exit mouseUp
    end if

    put fileLength(theFile) into howlong
    if howlong = "0" then
        beep 1
        answer "Error cannot transfer this file"
        exit mouseUp
    end if

    put "stack"&&quote&theFile&quote into theFile

```

```

put "stack" quote stripper quote into stripper

-- Calculate out how long the transfer will take.
divide howlong by 100

if howlong > 60 then
  divide howlong by 60
  put round(howlong) into howlong
  answer "Transfer will take approx." howlong "minutes" with "Continue" or "Cancel"
  if it is "Cancel" then
    exit mouseUp
  end if
else
  put 1 into secon
  answer "Transfer will take approx." howlong "seconds" with "Continue" or "Cancel"
  if it is "Cancel" then
    exit mouseUp
  end if
end if
put empty into moreoneword
set cursor to watch

striplast stripper
put the result into path

gimelast theFile
put the result into filename
put the number of words in filename into x
-- Make sure the file name is one word so the SUN can handle it.
if x > 1 then
  put 1 into moreoneword
  put 1 into count
  put filename into filenamesaved
  repeat with x = 1 to the number of words in filename
    put word count of filename "_" after last character in fun
    add 1 to count
  end repeat

  delete last character of fun
  put fun into filename
end if
--put quote filenames quote into filename

delete first character of path
put path ":" into path
-- Here is the xmodem protocol.
if moreoneword = 1 then

  sendSPort "xmodem rb" filename &return
  wait 200
  set loc of msg to 10,300
  if secon = 1 then

    put "Transfer begun at " the time "Transfer will take approx." howlong "seconds"
  else
    put "Transfer begun at " the time "Transfer will take approx." howlong "minute(s)"
  end if

  xmodem "send", path, filenamesaved

beep 1

```

```

    answer "Transfer of" & filename & "completed" with "OK"
    hide msg
    put empty into fld screen of this card
    sendSPort "ls -lf" & return

    exit mouseUp

else

    sendSPort "xmodem rb" & filename & return
    wait 200
    set loc of msg to 10,300
    if secon = 1 then

        put "Transfer begun at" & the time & "Transfer will take approx." & howlong & "seconds"
    else
        put "Transfer begun at" & the time & "Transfer will take approx." & howlong & "minute(s)"
    end if

    xmodem "send", path, filename
    put path & filename into msg

    answer "Transfer of" & filename & "completed" with "OK"
    get the loc of card button List
    hide msg
    put empty into fld screen of this card
    sendSPort "ls -lf" & return
    exit mouseUp

end if

end if

if it is "Receive" then
    put "Gathering data.." into msg
    put fld screen into dataget
    put xStrip(dataget, linefeed) into dataget
    put xgetScreen(fld screen, return) into dataget
    put "Choose file...." into msg
    put dataget into theList
    DoList 999, theList, one
    put the result into thelist
    if thelist is empty then
        hide msg
        exit mouseUp
    end if
    put FolderName("Choose place to put file") into path
    if path is empty then
        hide msg
        exit mouseUp
    end if

    put item 2 of thelist into thelist
    put FindInField(fld screen, thelist, false) into where
    put item 2 of where into theline
    put line theline of fld screen into theline
    put xStrip(theline, linefeed) into theline
    put word 1 of theline into howlong
    multiply howlong by 1000
    divide howlong by 100
    put round(howlong) into howlong
    if howlong > 60 then
        divide howlong by 60

```

```

    put round(howlong) into howlong
    put 1 into secon
    answer "Transfer will take approx. " & howlong & "minutes" with "Continue" or "Cancel"
    if it is "Cancel" then
        hide msg
        exit mouseUp
    end if
else
    answer "Transfer will take approx. " & howlong & "seconds" with "Continue" or "Cancel"
    if it is "Cancel" then
        hide msg
        exit mouseUp
    end if
end if

set loc of msg to 10,300
if secon is empty then

    put "Transfer begun at" & the time & "." & "Transfer will take approx." & howlong & "seconds"
else
    put "Transfer begun at" & the time & "." & "Transfer will take
approx." & howlong & "minute(s)"
end if

sendSPort "xmodem sb" & the list & return
wait 300
xmodem "receive", path, the list
beep 1
answer "Transfer complete" with "Great"
end if
hide msg
end mouseUp

----- BUTTON: card button "Mail"

on mouseUp
    global messnum
    put empty into messnum
    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"

        exit mouseUp
    end if

    answer "Check mail or send it?" with "Check" or "Send" or "Cancel"
    if it is "Cancel" then
        exit mouseUp
    end if

    if it is "Send" then
        go to card "mail card"
        exit mouseUp
    end if
    set cursor to watch
    --put empty into fld screen
    put empty into card fld remove
    SendSPort "msg" & return

```

```

wait 100
--put CharsAvailable()&" after last character in msg
--put CharsAvailable() into x
--put recvChars(x) into cd fld remove
--put CharsAvailable()&" after last character in msg

put recvUpTo(linefeed,0,empty) into newInput
repeat until newInput is empty

    put recvUpTo(linefeed,0,empty) into newInput

    put newInput after last character of card field remove

end repeat

set the scroll of fld screen to 1

get FindInField(card fld remove,"total","false",0)
if it = "0,0,0,0" then
    get FindInField(card fld remove,"empty","false",0)
    if it = "0,0,0,0" then
        answer "Error in checking mail, Try a different phone#"

        exit mouseUp

    end if

    answer "No new messages"
    --put empty into fld screen
    --sendSPort "ls -l" & return

set the scroll of fld screen to 1
exit mouseUp

else
    get FindInField(card fld remove,"binary","false",0)

    if it = "0,0,0,0" then
        get FindInField(card fld remove,"message","false",0)
        put item 2 of it into linenum
        put the number of words in line linenum of cd fld remove into temp
        subtract 2 from temp
        put word temp of line linenum of cd fld remove into theCount

set the loc of the msg to 10,300
if theCount = "1" then

    put "You have "&theCount&" message" into cd fld msgname of card themessage

else
    put "You have "&theCount&" messages" into cd fld msgname of card themessage

and if

go to card themessage
put empty into card fld save
put empty into card fld theList
put theCount into messnum
repeat with x = 1 to theCount

```

```

        put x&" after last character in card fld theList

    end repeat

else
    get FindInField(card fld remove,"message","false",0)
    put item 2 of it into linenum

    put word 2 of line linenum of cd fld remove into theCount

    set the loc of the msg to 10,300
    if theCount = "1" then

        put "You have "&theCount&" message" into cd fld msgname of card themessage

    else
        put "You have "&theCount&" messages" into cd fld msgname of card themessage
    end if

    go to card themessage
    put empty into card fld save
    put empty into card fld theList
    put theCount into message
    repeat with x = 1 to theCount

        put x&" after last character in card fld theList

    end repeat

end if

hide msg
end mouseUp

```

```

----- BUTTON: card button "Delete File"

```

```

on mouseUp

    if fld receiving > 0 then
        answer "Wait a second, still gathering data!"

        exit mouseUp
    end if

    put empty into card fld remove
    put "Please wait... gathering data" into msg

    put fld screen into dataget
    put xStrip(dataget,linefeed) into dataget
    put xgetScreen(fld screen,return) into dataget
    delete item 1 to 2 of dataget
    put "Choose file...." into msg

```

```

put dataget into theList
DoList 999,theList,one
put the result into thelist
if thelist is not empty then

    put item 2 of thelist into thelist
    answer "Remove file"&thelist&"?" with "OK" or "Cancel"
    if it is "Cancel" then
        hide msg
        exit mouseUp
    end if

    SendSPort "rm "&quote&thelist&quote & return

    wait 100

    SendSPort "y" & return
    put charsAvailable() into test
    put recvChars(test) into thetest
    put empty into fld screen
    answer "File has been removed" with "OK"

    put empty into fld screen

    SendSPort "ls -l" & return

    set the scroll of fld screen to 1

end if

hide msg box
end mouseUp
--if the optionkey is down then pass mouseup
--put card field list into thelist
--DoList 999,theList,one
--put the result
--end mouseUp

----- BUTTON: card button "New Button"

on mouseUp
    get the loc of card button mail
    click at it
end mouseUp

----- BUTTON: card button "mbox"
-- This code is the same as the MacTCP version. It checks to see how many
-- messages the user has, then goes to the read mail card and waits for
-- the user to pick a message number to view.
on mouseUp
    global messnum
    put empty into card fld remove

    SendSPort "msg mbox" & return
    wait 100

    put recvUpTo(linefeed,0,empty) into newInput
    repeat until newInput is empty

        put recvUpTo(linefeed,0,empty) into newInput

```

```

    put newInput after last character of card fld remove
end repeat

--put xStrip(card fld remove,return) into card fld remove

set the scroll of fld screen to 1
get FindInField(card fld remove,"total","false",0)

if it = "0,0,0,0" then
    get FindInField(card fld remove,"empty","false",0)
    if it = "0,0,0,0" then
        answer "Error in checking mail, try again"

        exit mouseUp
    end if

    answer "No new messages"
    --put empty into fld screen
    --sendSPort "ls -l" & return

    set the scroll of fld screen to 1
    exit mouseUp
else
    g-t FindInField(card fld remove,"binary","false",0)

    if it = "0,0,0,0" then
        get FindInField(card fld remove,"message","false",0)
        put item 2 of it into linenum
        put the number of words in line linenum of cd fld remove into temp
        subtract 2 from temp
        put word temp of line linenum of cd fld remove into theCount

        set the loc of the msg to 10,300
        if theCount = "1" then
            put "You have "&theCount&" mbox message" into cd fld msgname of card themessage
        else
            put "You have "&theCount&" mbox messages" into cd fld msgname of card themessage
        end if

        go to card themessage
        put empty into card fld save
        put empty into card fld theList
        put theCount into messnum
        repeat with x = 1 to theCount

            put x&";" after last character in card fld theList

        end repeat
    end if
end repeat

```

```

else
  go. FindInField(card fld remove,"message","false",0)
  put item 2 of it into linenum

  put word 2 of line linenum of cd fld remove into theCount.

  set the loc of the msg to 10,300
  if theCount = "1" then

    put "You have {theCount} mbox message" into cd fld msgname of card themessage

  else
    put "You have {theCount} mbox messages" into cd fld msgname of card themessage
  end if

  go to card themessage
  put empty into card fld save
  put empty into card fld theList
  put theCount into messnum
  repeat with x = 1 to theCount

    put x ";" after last character in card fld theList

  end repeat

end if
end if

hide msg

```

end mouseUp

----- BUTTON: card button "New Button"

on mouseUp

```

  show card fld help
end mouseUp

```

----- CARD FIELD SCRIPTS -----

----- FIELD: card field "HEX"

on mouseUp

```

  hide me
end mouseUp

```

----- FIELD: card field "Help"

on mouseUp

```

  hide me
end mouseUp

```

```

----- CARD SCRIPT: file view -----
-- This repositions the background field "screen" for each card.
on opencard
  global morecount
  set lockscreen to true
  set the lockText of fld screen to false
  put empty into morecount
  set the rect of fld screen to 0,57,512,297
  set the loc of fld "receiving" to 57,39
  set lockscreen to false
end opencard
on closeCard
  set lockscreen to true
  set the lockText of fld screen to true
  set the rect of fld screen to 0,0,261,342
  set the loc of fld receiving to 319,323
  set lockscreen to false
end closeCard

----- CARD BUTTON SCRIPTS -----

----- BUTTON: card button "Show More"

on mouseUp
  if fld screen of this card is not empty then

    put fld screen into dataget
    put xStrip(dataget,linefeed) into dataget
    put dataget into fld screen
  end if

  put number of characters in fld screen into numofchars
  -- Check to make sure the field doesn't get too full of characters
  if numofchars > 25000 then
    answer "Field will be cleared this time, it's getting full"--
    with "Cancel" or "OK"
    if it = "Cancel" then
      exit mouseUp
    else
      put empty into fld screen
      put empty into morecount
    end if
  end if
  delete last line of fld screen
  delete last character of fld screen
  --delete last character of fld screen
  sendSPort " "
end mouseUp

----- BUTTON: card button "New Button"

on mouseUp
  put fld screen of this card into x
  printText x
end mouseUp

----- BUTTON: card button "UNIX InterFace"

on mouseUp
  if fld receiving > 0 then

```

```
answer "Wait a second, still gathering data!"
```

```
exit mouseUp  
end if
```

```
put the short name of this card into cardname  
if cardname = "file view" then  
  if fld screen of this card = empty then  
    put empty into fld screen of this card  
    Cntr2  
    put the result into fun  
    sendSPort "q"
```

```
    wait 100  
    --sendSPort "ls -lf" & return  
    set lockscreen to true  
    go card interface  
  end if  
end if
```

```
set lockscreen to true  
go card interface  
set lockscreen to false  
end mouseUp
```

```
----- BUTTON: card button "New Button"
```

```
on mouseUp
```

```
  show card fld help  
end mouseUp
```

```
----- CARD FIELD SCRIPTS -----
```

```
----- FIELD: card field "Help"
```

```
on mouseUP  
  hide me  
end mouseUp
```