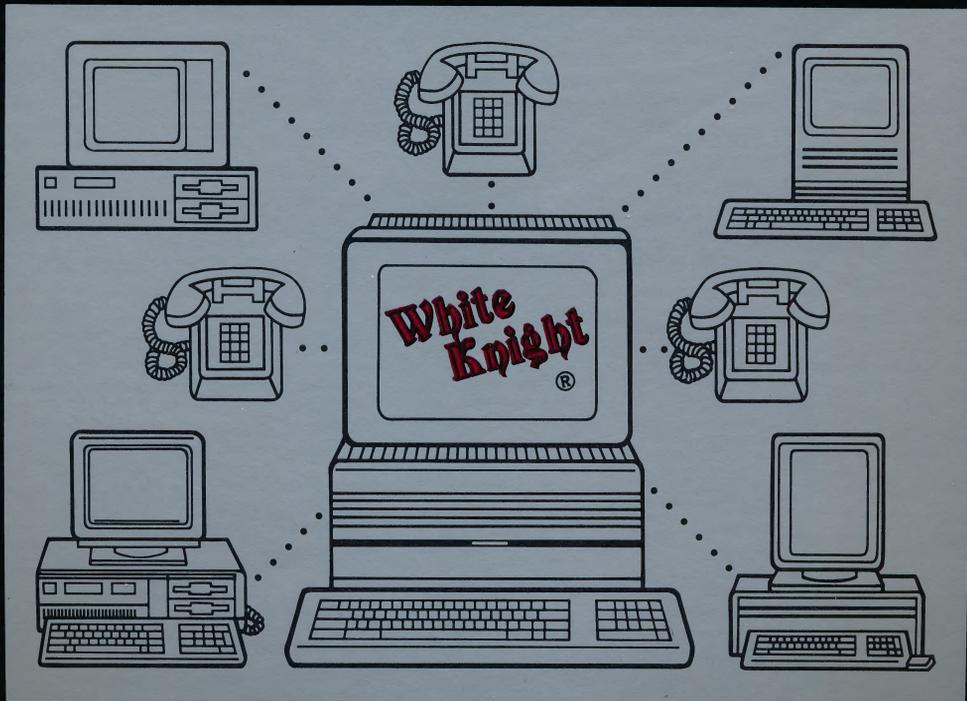


SCOTT WATSON'S

White Knight®

AWARD WINNING COMMUNICATIONS SOFTWARE FOR THE MACINTOSH



FREE INSIDE!

OKYTO
Revolutionary Communication
A \$39.95 Value!
Ask your dealer for a brochure

**New!
Version 11
Enclosed**

The FreeSoft Company

105 McKinley Rd.
Beaver Falls, PA 15004
Phone: (412) 846-1111

Here's What White Knight Will Do For You:

- Supports Macintosh Plus, SE, II and all later models. Not for use on 128K or 512K Macintosh's.
- 270+ page manual has plenty of step-by-step's for beginners. Simple enough for a beginner, yet enough meat to keep experienced users happy for a long, long time.
- Capture all incoming text to a file that can be imported to your favorite word processor.
- Send or receive files using the richest set of error correcting protocols of any Mac communications program, including XMODEM/checksum, XMODEM/CRC, XMODEM/1K blocks, Supercharged XMODEM, YMODEM/CRC, YMODEM/1K blocks, YMODEM-G, ZMODEM, Classic Kermit, Long Packet Kermit, Sliding Windows Kermit, Flash, and CompuServe QuickB protocols.
- Will work completely and transparently in the background under MultiFinder.
- True and complete terminal emulation of TTY, VT52, VT100, and VT102 terminals with exceptionally fast throughput speed makes it easy to connect with other microcomputers, minicomputers, mainframes, time-sharing networks, laboratory equipment, bulletin board systems, etc.
- 30 user-defined Macro Keys reduce multiple actions to a single mouseclick or keypress. Macro Keys can be assigned to any keyboard combination or can be shown in a custom window as icons, double-size icons, color icons, double-size color icons or pictures. You can truly create a custom Macintosh-like interface to any host system.
- 200+ command Procedure Language with a built in two-pass compiler (not to be confused with the simple interpreted "script languages" found in most telecommunications software). Capable of fully automating your communication sessions. Using this language you can create automatic log on and system navigation sequences, customized dialog boxes, menus, and windows - even create your own data files and do input/output. Nothing in the world comes close to White Knight's Procedure Language. We even included an easy to use point-and-click Procedure Editor for beginners.
- "Write A Procedure For Me" features watches what you do and automatically writes a Procedure to perform a mirror image of your actions. You don't even need to know the Procedure language!
- Host Mode with optional password protection allows unattended file transfers from a remote machine.
- Multiple Phonebooks to keep track of your most frequently used services.
- Full text selection and Clipboard support including Copy, Copy Table, Paste To Modem, Copy Then Paste, Append Selected Text To New Or Existing File, Print Selected Text. Import data to or export data from your favorite Macintosh applications with the greatest of ease.
- Full support of all printers (including all Laserwriter models and third party printers).
- Buffered keyboard option lets you compose text (up to 32,000 characters with multiple embedded lines) before sending it. Great for real-time conferences and packet radio applications!
- Fully moveable/sizeable/zoomable windows takes advantage "big screen" monitor sizes. We even let you "colorize" the windows to your own preference (if your machine supports Color QuickDraw).
- Use any font in any point size for text display with no throughput degradation.
- "Special GENie money back guarantee for White Knight users when joining GENie Basic Services. Includes access to the restricted FreeSoft RoundTable. There are literally thousands of Macintosh oriented files on this network for you to download with White Knight!"
- Works with all modems (Hayes compatible or otherwise) and direct connections. Complete modem control via simple pulldown menu choices.
- On-screen meters tell you the elapsed time and money spent in the current session.
- Screen buffer lets you review text that's scrolled off the top of the screen. Find Text feature allows fast searching through the received text.
- Automatic redial busy numbers until connected and dial multiple services in rotation features.
- Exclusive Filters give you full control over each received character.
- Supports 300 to 57,600 baud. High speed modems and hardware handshaking are supported.
- Full MacBinary II standard support for file transfers lets you send any Macintosh application or document and have it arrive in precisely the same condition as it left.

...AND THAT'S JUST THE BEGINNING!

Over 100 new features added to this version! Don't pay more for less!



The FreeSoft Company

105 McKinley Road
Beaver Falls, PA 15001
Phone: (412) 846-7000

White Knight

Version 11



The FreeSoft Company
105 McKinley Road
Beaver Falls, PA 15010
(412) 846-2700

MFD-2DD

Double Sided

MICRO FLOPPYDISK

SONY

28

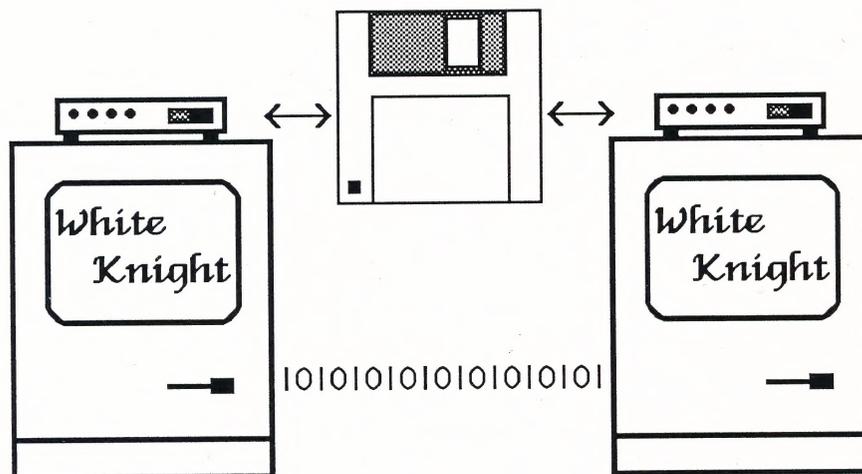
OR6193M

DOUBLE SIDED



Scott Watson's White Knight[®]

An Asynchronous Telecommunications Program For The Apple
Macintosh Using A Modem Or Direct Connection.



Copyright, 1989, The FreeSoft Company, 105 McKinley Road, Beaver Falls, PA 15010. All Rights Reserved. (412)846-2700.



Table Of Contents

General Electric has graciously extended a free subscription to their GENie network for all of our customers. This means they waive their normal \$29.95 sign-up fee. This is available to all customers in the U.S., Canada and Japan.

1) Using 300 or 1200 baud, NO parity, 8 databits, 1 stopbit, and HALF duplex (not FULL!), call the following toll free number: 1 (800) 638-8369. In Toronto, Canada call 1 (416) 858-1230; Calgary, Alberta, Canada call 1 (403) 232-6121; Montreal, Canada call 1 (514) 333-1117; Vancouver, B.C. Canada call 1 (604) 437-7313. In Japan, call 03-454-6909.

2) When connected, immediately type 3 'H' characters and press the Return key.

3) After a couple of seconds you'll get a prompt of "U#=". You should type: XJM11979, FREESOFT and press the Return key.

4) You'll now be taken on a tour of GENie and asked to fill in your billing information. The "800" number does not give you access to the actual GENie network, but is for subscription purposes only. You can and should check out what your local GENie access number (s) are while doing this procedure. Those numbers will be what you'll call after receiving your account to log into GENie.

5) After supplying your billing information, you'll be contacted voice by GENie within the next week with your new account number and password. After you get this, you can log onto GENie through your local access number. Once on, you should type the word "FREESOFT" at any GENie menu prompt.

6) You'll be teleported to the FreeSoft RoundTable, but you won't be allowed to get in. Don't worry, you have to be turned away at least once before I can let you in. Type "MAIL" at the next menu to get the GENie mail section.

7) Send a short note to me (address SWATSON) saying simply, "Let me in". To send the letter you must type an asterisk character ('*') and a 's' character at the beginning of a new line like this: *s.

8) We usually do the unlocking each business day, but don't get upset if it takes us up to three business days. You won't receive any notice about being unlocked, but can check quickly by typing "FREESOFT" at any GENie menu to attempt to enter the FreeSoft RoundTable.

White Knight: Extending White Knight	249
Troubleshooting And Technical Support	256
Even More White Knight Features	263
The ASCII Character Set	263
Hardware Serial Port Circuits	267
Keyword Index	268
Procedure Command Index	274

General Electric has graciously extended a free subscription to their GENIE network for all of our customers. This means they waive their normal \$29.95 signing fee. This is available to all customers in the U.S., Canada and Japan.

1) Using 300 or 1200 baud, NO parity, 8 data bits, 1 stop bit, and HALF duplex (not FULL!), call the following toll free number: 1 (800) 838-8369. In Toronto, Canada call 1 (416) 828-1230; Calgary, Alberta, Canada call 1 (403) 232-6121; Montreal, Canada call 1 (514) 333-1117; Vancouver, B.C. Canada call 1 (604) 437-7313. In Japan, call 03-454-6909.

2) When connected, immediately type 3 'H' characters and press the Return key.

3) After a couple of seconds you'll get a prompt of "H#:". You should type: XMM1179, FREEZ0FT and press the Return key.

4) You'll now be taken on a tour of GENIE and asked to fill in your billing information. The "800" number does not give you access to the actual GENIE network, but is for subscription purposes only. You can and should check out what your local GENIE access number (s) are while doing this procedure. Those numbers will be what you'll call after receiving your account to log into GENIE.

5) After supplying your billing information, you'll be contacted voice by GENIE within the next week with your new account number and password. After you get this, you can log onto GENIE through your local access number. Once on, you should type the word "FREEZ0FT" at any GENIE menu prompt.

6) You'll be returned to the Presoft Roundtable, but you won't be allowed to get in. Don't worry, you have to be turned away at least once before I can let you in. Type "MAIL" at the next menu to get the GENIE mail section.

7) Send a short note to me (address SWATSON) saying simply, "let me in". To send the letter you must type an asterisk character (**) and a 's' character at the beginning of a new line like this: *s.

8) We usually do the unlocking each business day, but don't get upset if it takes us up to three business days. You won't receive any notice about being unlocked, but can check quickly by typing "FREEZ0FT" at any GENIE menu to attempt to enter the Presoft Roundtable.

Table Of Contents

End User License Agreement For White Knight.....	i
Foreword.....	iii
Nomenclature.....	1
Initial Setup.....	4
The Terminal Window.....	9
The Basics Of Computer Communications.....	21
Serial Port Settings.....	23
Terminal Emulation.....	31
Keyboard Support.....	41
Modem Support.....	46
Phonebooks.....	51
The Buffered Keyboard.....	56
Filters.....	58
Macro Keys.....	63
Using A Printer With White Knight.....	74
File Transfers.....	79
Sending Text Files.....	80
Receiving Text Files (File Captures).....	85
Sending And Receiving Files Using Error Correcting Protocols..	89
Batch File Transfers.....	101
XMODEM, YMODEM, And ZMODEM Options.....	102
Kermit Options.....	105
Step By Step File Transfers.....	107
Miscellaneous Features.....	115
Host Mode.....	118
Introduction To Procedures.....	125
Letting White Knight Write A Procedure For You.....	127
The Procedure Editor.....	131
Advanced Procedure Files.....	146
Procedure Commands.....	160
Procedure Examples.....	223
Memory Considerations.....	232
About Filenames.....	233
The "Install Special Keys" Utility.....	235
The GETPARAM And PUTPARAM Procedure Commands.....	238
Using International Or Special Fonts With VT Emulation.....	249
RCMD's: Extending White Knight.....	251
Troubleshooting And Technical Support.....	256
Even More White Knight Features.....	263
The ASCII Character Set.....	265
Macintosh Serial Port Pinouts.....	267
Keyword Index.....	268
Procedure Command Index.....	274

Table Of Contents

1	End User License Agreement For White Knight
111	Foreword
1	Nomenclature
4	Initial Setup
9	The Terminal Window
21	The Basics Of Computer Communications
23	Serial Port Settings
31	Terminal Emulation
41	Keyboard Support
46	Modem Support
51	Phonebooks
56	The Buffered Keyboard
58	Filters
63	Macro Keys
74	Using A Printer With White Knight
79	File Transfers
80	Sending Text Files
85	Receiving Text Files (FileCaptures)
89	Sending And Receiving Files Using Correcting Protocols
101	Batch File Transfers
102	XMODEM, YMODEM, And ZMODEM Options
105	Kernel Options
107	Step By Step File Transfers
115	Miscellaneous Features
118	Host Mode
125	Introduction To Procedures
127	Letting White Knight Write A Procedure For You
131	The Procedure Editor
146	Advanced Procedure Files
160	Procedure Commands
223	Procedure Examples
232	Memory Considerations
233	About Filenames
235	The "Install Special Keys" Utility
238	The GETPARM And PUTPARM Procedure Commands
249	Using International Or Special Fonts With VT Emulation
251	ROM's: Extending White Knight
256	Troubleshooting And Technical Support
263	Even More White Knight Features
265	The ASCII Character Set
267	Mechanosh Serial Port Pinouts
268	Keyword Index
274	Procedure Command Index

End User License Agreement For White Knight

Don't do anything with the White Knight master disks until you've read and agree with the following!

Execution of the White Knight software constitutes your acceptance of the terms and conditions of the End User License Agreement For White Knight. In return for acquiring a license to use the White Knight software contained in this package, you agree to the following terms and conditions. If you do not agree, return the entire contents of this package within 7 days of the date of purchase to the place of purchase for a full refund.

In this license, the words "White Knight" refers to all of the programs and data files recorded on the disks enclosed with this package.

1. The White Knight software and documentation are copyrighted, and you agree to respect all rights and privileges of The FreeSoft Company as the holder of these copyrights. You, as licensee, own the magnetic media on which White Knight is recorded. Title and ownership of White Knight is retained by The FreeSoft Company, including the copy of White Knight recorded on the disks enclosed with this package. The nonexclusive license set forth in this agreement is granted to you by The FreeSoft Company. The license is not a sale of White Knight or any copy. You will take all steps necessary to insure that each licensed copy of White Knight does not reside or execute on more than one Macintosh computer at any given time. You may make and execute as many copies of White Knight as you like on a single Macintosh computer. You may make as many copies of White Knight as you like for archival and backup purposes.

2. You will not use White Knight to support any illegal acts as defined by the laws of your country, state or province, and local governments, including international copyright laws. You will take all steps necessary to insure that White Knight is used only for legal purposes. You will not knowingly allow any copyrighted software to be transmitted by White Knight to another computer without permission from the owner of the copyright.

3. Limited Warranty: The FreeSoft Company warrants to you, the original

licensee of this package, that the disks on which White Knight is recorded and the user manual are free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase. If the disk or manual is defective, The FreeSoft Company will replace the defective item(s) at no charge to you, if you return the defective item(s) to The FreeSoft Company within ninety (90) days from the date of purchase. This paragraph sets forth the only warranty or representation of any kind with respect to White Knight made by The FreeSoft Company or anyone involved in the creation, production, delivery, or licensing of White Knight. Furthermore, any implied warranties are limited in duration to ninety (90) days from the original date of purchase.

4. The FreeSoft Company makes no warranty or representation of any kind, express or implied, with respect to White Knight, or its quality, reliability, performance, or its merchantability or fitness for any particular purpose. White Knight is licensed "as is". You assume the entire risk as to the quality, reliability, and performance of the software and documentation.

5. In no event shall The FreeSoft Company or anyone else involved in the creation, production, delivery, or licensing of White Knight be liable to you for any consequential, special, incidental, direct, or indirect damages of any kind arising out of the use of White Knight or any defect in White Knight, even if The FreeSoft Company or anyone else involved in the creation, production, delivery, or licensing of White Knight has been advised of the possibility of such damages.

6. If you violate any of the terms of this license, you agree to forfeit to The FreeSoft Company, at the request of The FreeSoft Company, all copies of the White Knight software and documentation in your possession.

7. The terms and conditions of this license supersede the terms and conditions of all previous licenses. Use of this software constitutes your agreement to abide by the terms and conditions of this license for all versions in your possession.

So there.

Foreword

It seems like a million years ago, but in fact it was only a few. It was after one in the morning and I had a big problem. But let me back up a bit.

I had just finished my first Macintosh product, a whiz-bang telecommunications program, and all that was missing was a good name for it. Definitely not MacThis or MacThat, certainly not a combination of the usual buzzwords. What I was looking for was something unique and catchy, perhaps even exotic. In fact, the less it had to do with computers or telecommunications in general, the more it would suit the bill.

I was in one of the sleazier bars in St. Louis, but it was close to home and having been without sleep for a few days putting the polish on the user manual, I didn't trust myself to drive. A short walk would lead me to the company of several dozen \$1.25 friends with Budweiser labels on their bellies.

The usual gang of wretches were gathered near the waitress station (the best seats to watch what had to be the largest woman ever to assault a bikini do the hootchy kooch), and when I enlisted their help in suggesting a name for my baby in waiting, they quickly began a championship level game of one-upmanship, each offering something more scatological than the last. By midnight, I came to the conclusion that it could only be the sheer weight of the booze they had inhaled that gave them the center of gravity necessary to stay in their stools.

But good help is hard to find. I wasn't the sort to seek out professional marketing advice, you see. No, for me it had to be a group of miscreants whose only achievement in life was in losing their teeth and most of their body hair from a steady diet of nicotine, discount whiskey, and profanity.

The Hootchy Koochette finally gave up around one A.M., once the bar rats had run out of dollars and were beginning to throw small change, with no lack of force and an uncanny precision that can only come from practice. Deprived of their sport, they teetered out the front door to offer their talents to the nearest Interstate highway.

But I still had a problem. The bartender was giving me that "go home" look, and I still didn't have a name. Finally, I gave in to my exasperation and decided with all conviction that I would name the new program after the next song that came on the jukebox.

When the song came on, I was pleasantly surprised at how much I liked it. The ethereal guitar licks reminded me of Pink Floyd, and the vocals had a hard driven Joe Walsh edge to them. My pleasure was short lived however, when I went to the jukebox and found the listing for the song.

The handwritten listing told me that the song was called "Lunatic Fringe".

Had a nice ring to it, but let's face it folks, it didn't have...shall we say..."Star Potential". I was about to walk away in disgust when I noticed the name of the band who had played the song.

"Red Ryder" - and so I had a name.

An interesting thing that I later found out was that the band's name was actually "Red Rider", but I guess the jukebox owner was an old timer who remembered the cowboy character of the same name and mistakenly changed the "i" to a "y".

The new product wasn't to be released to distributors or dealers, but to local user groups, bulletin board systems, and the large commercial networks CompuServe and The Source. It was released with the instructions that the recipients were to give away as many copies as they could to anyone they wished. If the recipient liked the product and wanted to keep it after an honesty-system 45 day evaluation period, they would send me a check for \$40. This method of distribution was called shareware, and I was only the third company in existence at the time to try it (the other two being in the IBM-PC market).

I didn't have much faith in shareware, but then again, I didn't have much choice either. I didn't have the capital to set up a software publishing house that could print manuals and packaging, set up a customer service and technical support staff, and pay for advertising. The only other option was to contract with an existing software firm to publish the program, with me receiving a royalty for each copy sold.

A year or two earlier, I had contracted with a high profile firm to do just that, with an assembly language coded arcade game that had taken me eight months to write. The entire royalties received was a check for \$31 and some change.

The check bounced.

The shareware program was released at the heyday of piracy paranoia. Very few Macintosh programs were being released without some form of copy protection. My strategy was to encourage exactly what the others were trying to prevent, and to rely on the integrity of the Macintosh market to pay for what they considered valuable. I wouldn't even think about the pirates, who were going to steal from me no matter what I did.

To be honest, I hoped to make just enough to pay off my Macintosh, which I had purchased with an Apple credit card, and which I considered to be the riskiest investment I had ever made.

It's now six months later. During this time I had been working 8-10 hours a day as an editor for a newsletter publisher. At night, the necktie came off and I'd

work 8-10 hours on version 2.0. I'd also spend some time doing technical support and promotion over CompuServe, and would try to fill the week's orders on the weekend. In the original manual, I had strongly encouraged people to send me ideas for improvements, and they came in by the hundreds.

Improving the product became an obsession, and I fell further and further behind on getting the orders out (and the checks to the bank). The 20 hour days had become a way of life, but I always kept frontmost in my mind that my employer came first. During this time, in fact, my work as an editor won the first Newsletter Association award that the company had ever received. It was 100% effort 20 hours a day, and I was starting to get a bit ragged around the edges.

Version 2.0 was released, then version 3.0, then version 4.0.

Not long after, I was in a local pub called The Hidden Cove (not at all sleazy, by the way), explaining to the owner and my good friend Don Killian, how far behind I had got on getting out the orders. Don had helped me out on several occasions in the past, but I must have looked close to a nervous breakdown, because Don offered to close down the bar and come over that night. I can't describe the look on his face when I opened the front door of my house.

Every square inch of the living room floor was covered to a depth of two inches with envelopes containing \$40 checks and money orders. The fireplace hearth and mantle was covered with these envelopes as well as the couch and one chair. Everything, in fact, except for one chair that I sat in to eat dinner and watch television, and the dinette table the Macintosh sat on.

He stayed there for the rest of the night, and since the next day was Saturday, we decided to go ahead and keep working until we got them all covered. We finished late Saturday night, and Don went into a well deserved coma on the now available living room couch.

On Sunday morning, Don showed me the end of the twenty yard long adding machine tape tally of the checks. On Monday morning, I stopped by the bank, went to work, gave my thirty days notice, and hired Don to take phone calls and fill the orders on a percentage basis. I, by the way, became a partner in the Hidden Cove. Everyone needs a hobby.

Versions 5.0, 6.0, 7.0 and 8.0 were released, each to better reviews than the last. The awards and accolades were literally pouring in, and Red Ryder became the benchmark program that all others were compared against. We disregarded every established rule of software businesses that we could think of. We gave money back guarantees, we kept our price low, and we gave very modestly priced updates. Just a couple of guys having fun.

I released version 9.0 at a Macintosh convention in Chicago. Someone suggested a new feature, and since it seemed like a snap and I had brought my

Macintosh with me, I went up to my hotel room. A few hours later, version 9.1 was released. By the end of the show, we were up to version 9.3. It's a good thing the show only lasted two days.

The funny thing about all of this is that I never gave the money any thought. I was having so much fun writing software that people liked that I just didn't care how well or how poorly the business was doing. Actually, it was doing quite well. My accountant advised me that I was getting "raped" taxwise by renting a house, and that I should directly go purchase one.

I didn't find anything I liked in St. Louis, but when visiting my parents in Pennsylvania, fell in love with a log home. I decided that all I really needed was a mailbox and a telephone to be in business, so I purchased a wooded lot that adjoined my dad's apple orchard and hired a company to build the home there.

The user manual had grown too large to put on a diskette, so with great regret, I decided that it was time to leave the shareware market and sell version 10 using the traditional dealer channels. The greatest regret came from the realization that with discounts of from 40-50% given to dealers, I would have to double the price of the product just to keep making what I had in the past.

I arrived in Beaver Falls, Pennsylvania, hired my sister to take Don's place, and released version 10 the same day. Our sales doubled. More people were hired. Three months later, I was in Intensive Care, having popped an ulcer that gave away four pints of my best blood. I decided that I wouldn't work on weekends or holidays anymore. Nevertheless, while in the hospital, I coded in a notebook about 90% of what has now become our Okyto product. I kept the notebook well hidden when my relatives came to visit.

Two days before we were ready to go to the printer with version 11, we received word that the royalty for our trademark license for the name Red Ryder was going to be significantly increased on Jan 1, 1990, when the contract was up for renewal. The best calculations I could do on such short order told me that if we made a bad decision at this critical time, we could lose the farm, so to speak.

In little league football, I learned that when worse came to worst, you put your head down, gritted your teeth, and ran like hell. I reacted in much the same way. I held off going to the printer long enough to hire a trademark attorney and come up with the new names: White Knight for the old Red Ryder, and Second Sight for the old Red Ryder Host. My gut feeling was that even though we had developed a great deal of goodwill with the Red Ryder name, the price increase or elimination of jobs necessary to cover the new license would not have been in the best interest of my company or my customers. I felt that the name had insignificant value compared to the software itself. We'll find out if this is true. All things considered, it is a hell of a risk and certainly not what a marketing specialist would recommend. But then again, look at our history.

The best testament I can give to this strategy is that this product has been the top selling and best known communications program for the Macintosh, and we have never spent a single dime on advertising. Every copy we have ever sold has resulted from word of mouth or published reviews - and that just doesn't happen in this industry. Of course, I do feel that advertising will have to be done to get the word out about the new name, but I'm hoping you'll also help me spread that word. Paradigms have to watch their back.

Tomorrow, I'm going to send this off to the printer. It's the end of two years of very hard work. In more ways than one, I think this is the closest I'll ever come to giving birth. Two years in labor (ouch). But I'm so proud of this product, I have trouble putting it into words without becoming immodest. So I leave that up to you folks, who have always done better than I in that regard.

It's ironic that the first pages of a book are always the last to be written, but at least it allows me to give thanks where due:

To Norm Goodger, who covers for me incessantly on The FreeSoft RoundTable on GENie and who never settles for "good enough".

To my core group here at FreeSoft responsible for White Knight (and my health): Lisa Senkevich (my sister) and Pat Watson (my mom) and Diane Dixon and Dan Romich (who aren't related to me but probably wish they were - grin).

To my "Search And Destroy" team, who stuck their fingers into places they didn't belong in White Knight, just to see what would happen: Doug Acker, Dave Alpert, Bart Barton, Mike Chamberlain, Bob Daniels, Dennis Klatzkin, Eric Larson, Tom Mackie, Ray Terry, Dick Veldhuis, and Steve Winslow.

To Kate Taylor of Finnegan, Henderson, Farabow, Garrett & Dunner (who probably wishes she was related to one of them - big grin), for doing a month's worth of work in a couple of days, and for not laughing at my proposed new names. She also gets the distinction of being the only lawyer I've ever thanked, and I thought I better do so before her bill arrives.

Remember, if you don't tell me what's wrong with version 11, there won't be a version 12. Send in those suggestions - I'm ready to go to work!

Wishing you the best,

Scott Watson, 9/27/89

Nomenclature

Before we do anything (put that master disk down!), I need to explain a bit about how this manual is laid out - the Rosetta Stone, so to speak. Take a few minutes to browse this chapter and it will save you hours of head scratching later on.

Icons

I use several standard Macintosh icons in this manual to point out certain kinds of notes.



points out some sideline information, a/k/a "interesting poop".



points out something important. Take a moment to read this.



STOP! Make sure you read this. VERY IMPORTANT STUFF!

Menu Choices

White Knight makes use of hierarchical menus. If you pull down the **File** menu to **Receive Files Using**, you'll see that a second menu (a hierarchical menu) will drop down from that choice. Hierarchical menus are used when practical to keep like functions grouped for easy location. Note that I say "when practical." As you get into White Knight, you'll see that I kept a couple of very often used functions out of hierarchical menus, even though they *could* have been grouped under a hierarchical menu. This is to reduce the amount of effort necessary to use these functions. You'll know that a menu choice contains a hierarchical menu if it has a right pointing arrow symbol on the right hand side of the menu choice text. In this manual, I will designate menu choices in the following fashion.

A) If the choice is not in a hierarchical menu, I will designate it by the name of the menu, followed by an arrow, followed by the text of the choice. For instance, if you pull down the **Edit** menu, you'll see a choice labeled **Copy**. In this manual, I would refer to this choice as **Edit->Copy**.

B) If the choice is in a hierarchical menu, I will designate it by the name of the menu, followed by an arrow, followed by the choice that contains the hierarchical menu, followed by a second arrow, followed by the text of the choice itself. If you pull down the **File** menu to the choice **Receive Files Using**, a hierarchical menu will drop down with several choices, one of which is **XMODEM Protocol**. I would designate that choice as **File->Receive Files Using->XMODEM Protocol**.

C) Menu choices are always printed in boldface to set them apart from other text in this manual.

If a menu choice ends in ellipsis (three periods, as in **File Transfer Options...**), it indicates that this menu choice will first bring up a dialog box before performing any function. It might be a file selection dialog box, or perhaps a dialog box that allows you to change a particular setting.

Keyboard Menu Command Equivalents

Nearly all of White Knight's menu choices can be selected from the keyboard. The key (or keys) that you must press to execute a menu choice is called a "keyboard menu command equivalent." There are two kinds of keyboard menu command equivalents: single key and double key.

Single key equivalents are shown in menu choices on the far right side of the menu choice text. They are displayed as the cloverleaf symbol (⌘) followed by a single letter. For instance, the **Edit->Paste** menu choice has the symbols "⌘V" to the far right of the text "Paste". To select this choice from your keyboard, you would hold down the ⌘ key and type a "V".

Double key equivalents are something I created for a couple of reasons:

- 1) Because there are more menu choices in White Knight than there are letters in the alphabet.
- 2) Because things were starting to get really awkward with single key equivalents. I was assigning random letters, or worse, giving menu choices bizarre text in order to conform to the menu equivalent letters I had available.

I decided that all hierarchical menus would have double key equivalents. In other words, instead of holding down the ⌘ key and typing a single letter, you would hold down the ⌘ key and type two characters.

This caused a few raised eyebrows at the start of the testing period of White Knight 11, but the testers soon found this to be the most natural and easy to work with arrangement. This is because the double key equivalents are set up to use a mnemonic (memory aiding) system. The first character of a double key equivalent is almost always the first character of the choice that contains the hierarchical menu. The second key is always the first character of the menu choice itself. For example, the keyboard menu command equivalent for the menu choice **Local->Modem->Hang Up** is "%MH". In other words, hold down the cloverleaf key, type a "M", then continue holding down the cloverleaf key and type a "H". Double key equivalents are shown to the left of the menu choice text.

If you get in the middle of a double key sequence and want to bail out, just hold down the % key and type the spacebar.

Quotes

You'll see quotes used often in this manual. I use them to offset words from the rest of the text. If I tell you to type the word "QUIT", for instance, I really want you to type in the letters QUIT, not the quotes. Never type in quotes unless I specifically tell you to include them.

Initial Setup



I have had many technical support calls from people who aren't sure what an icon is, how to create and name a folder, and how to copy files from one disk to another. When a program prompts them to select a file, they don't understand how to navigate through folders to find the file. These are all very basic Macintosh functions, and I can't help but feel sorry for those people who haven't taken the time to learn them. I think the one outstanding thing that sets a Mac apart from the other computer brands is that it is exceptionally fun to use. However, if you haven't learned the basics, I can guarantee that you aren't going to have any fun. You are either going to wonder why you are so stupid, or you'll start throwing insults at software authors for writing a program that "is so difficult that no one can use it." It's really very easy to avoid this whole mess.

If you haven't read completely through your Macintosh Owner's Manual and run through the "Guided Tour" disk that came with your Macintosh, you aren't going to get very far with White Knight (or most other Macintosh applications, for that matter). Some folks are under the impression that User Manuals are optional reading with a Macintosh. This is completely untrue - you have to take the time to learn the fundamentals - not only of the Macintosh, but for each application you'll be using. Plan on taking at least an evening or two just getting used to the Macintosh before you go jumping into any serious work.

If you want to get along easily in White Knight, you must read this manual in the order it is presented. Don't skip around. If you've never used a communications program before and your boss comes in at 3 o'clock and throws a copy of White Knight to you and says "Have this proposal transferred to the VAX in Minneapolis by 5 o'clock.", heed my advice and go directly to the Western Union office, because you're just going to need more time to get your chops together on computer communications.

I've tried my best to lay out this manual in a step-by-step manner. It builds from a very basic level up to very heady stuff at the end. If you need help, don't be afraid to ask for it. There's an Appendix titled "Troubleshooting And Technical Support" in the back of this manual with the most common problems and our Technical Support phone number. Please realize that our Technical Support resources are limited, and we're not going to read the manual to you because

you left it at home, or don't feel like reading it. Give it your best shot and we'll be glad to help you over the brick walls.

Above all, have fun! I promise you that the words "that wasn't so hard" will cross your lips more than once as you learn about computer communications.

Setting It Up

The very first step is to write-protect your White Knight master disk. To do this, hold the disk so that the metal sliding flap is pointing down, and the label is facing away from you. On the top right hand corner you'll notice a rectangular cutout with a small plastic sliding box. Slide the box so that you can see through the rectangular cutout (so that the hole is not covered). Never use the Master Disk to do any work - we'll create a working setup in the paragraphs to follow. Your initial setup will depend on what type of storage equipment your Macintosh contains. There is no System Folder on the White Knight master disk, so you'll need to start up your Macintosh with a disk that contains one.

Floppy Disk Setup

Step 1: If your Macintosh only contains floppy disk drives (and not a hard disk), you should initialize two fresh blank diskettes. Title the first one "WK Working" (without the quotes). Title the second one "WK Extra". You will also need a disk with a System Folder on it to start up your Macintosh with.

Step 2: Insert your White Knight master disk and copy the following files onto the disk titled "WK Working":

- White Knight 11.x (the 'x' will be a digit from 0 to 9, depending on the revision you have received).
- ProcEdit

Step 3: Copy all of the rest of the files on the master disk, except the two mentioned in Step 1, onto the disk titled "WK Extra". As the purpose of these files are discussed later in this manual, you might choose to copy them onto the "WK Working" disk. You might also choose to take ProcEdit off of "WK Working". Don't worry, the only thing that has to be on "WK Working" is White Knight itself.

Hard Disk Setup

Step 1: Start up your Macintosh. Once the startup is complete and your Macintosh is ready to do work, you'll notice an icon underneath the menu bar on the far upper right corner of your screen. Beneath this icon is the name of your hard disk drive. Click on this icon twice (with no pause between clicks) and a window will appear showing the contents of the hard disk. Go to the **File** menu and choose **New Folder**. A folder icon will appear somewhere in the

hard disk's window named "Empty Folder". Rename this folder to "Comm" by dragging your mouse across the words "Empty Folder" and then typing "Comm".



There's nothing magic about the name "Comm" for the folder White Knight is in - I just like to keep my disk and folder names as short as possible (there's a good reason for this that will become clear as you get farther into White Knight). If you leave the folder named "Empty Folder", everything will still work fine.

Step 2: Double click on the "Comm" folder you just created and an empty window will appear. Copy all of the files and folders on the White Knight master disk into this window.

To Start Up White Knight

When you've completed the Initial Setup procedure, you're ready to execute White Knight for the first time. I suggest that you start up White Knight and refer to it as we go through the next few chapters. To start up White Knight, simply click twice (with no pauses between the clicks) on the "White Knight 11.x" program icon that looks like this:



White Knight 11.0

The White Knight program will then begin execution.

When you quit White Knight, you'll notice that the following file appears in the same folder in which White Knight resides:



WK's 11.0 Stuff

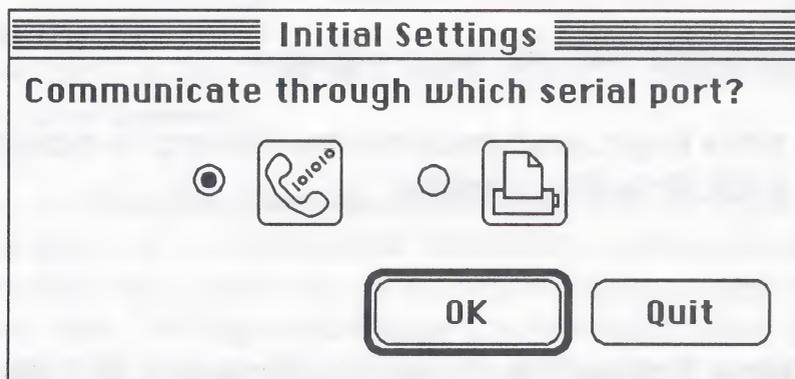
This is the icon for a "Settings File". A Settings File contains all of the settings in White Knight. A Settings File with the name "WK's 11.0 Stuff" in the same folder as White Knight is known as the "Default Settings File". Whichever settings you use in a session with White Knight are remembered by putting them in the "WK's 11.0 Stuff" file. White Knight looks for this file as soon as it is executed, and if it is present, uses the settings in that file to begin with. Don't throw away this file or move it outside of the folder in which White Knight resides or White Knight will create a new "WK's 11.0 Stuff" file.

You might wish to create Settings Files of your own, each containing the custom settings for a specific host system. You can do this by choosing **Customize->Settings->Save**. A standard file definition dialog box will come up asking you to give the Settings File a name. After doing this, click on the "Save" button and a Settings File with that name will be created. You can load a Settings File that was previously saved by choosing **Customize->Settings->Load** and then selecting the desired Settings File. Another way to load a Settings File is to double click on its icon, instead of White Knight's, to start up White Knight. If you do this, White Knight will use the settings in the double-clicked Settings File instead of those in the "WK's 11.0 Stuff" file.



Don't start up White Knight by double-clicking on the "WK's 11.0 Stuff" file. If you wish to use the settings from your last session, just double-click on the White Knight icon.

If White Knight doesn't find a "WK's 11.0 Stuff" file, it creates a new one. This is what will happen the first time you run White Knight. You'll know White Knight is doing this, because as soon as it starts up it will present you with the following dialog box:



Here, you should tell White Knight which serial port to communicate through. The two icons are the same as the icons above each serial port on the back of your Macintosh. Unless the manufacturer of your modem instructs you specifically otherwise, always connect the modem (or direct connection cable) to the port that has the icon of the telephone above it (we call this the "Modem Port").

Once you have selected the proper port, click on the "OK" button and the dialog box will disappear.

The next thing that will appear is White Knight's title page. Read the information on this page and then position your mouse anywhere inside the window and click. The title page will disappear and the Terminal Window will be drawn. Later on, I'll show you how to bypass this title page altogether when starting up White Knight.

The Terminal Window is where 90% of the action in White Knight takes place. The next chapter will discuss the various parts of the Terminal Window. After a bit of basics, we'll have you up and running.



Settings Files created with Red Ryder version 10.3 cannot be used directly by White Knight 11. They must first be converted using the "Convert Settings" application on your White Knight 11 master disk. After starting up the "Convert Settings" application, simply choose the Settings File you wish to convert. The Settings File will be converted to version 11 style and the "Convert Settings" utility will quit back to the Finder. Note that converted Settings Files will no longer work properly for Red Ryder 10.3, so if you need to keep them on hand for any reason, be sure to do the conversion on a backup copy.

Welcome To White Knight!

The Terminal Window

There's really a lot of stuff to digest in this chapter, but it is some of the most important information in this book. Don't be intimidated if some (or a lot) of the information in this chapter sounds like muck. If you're a first-timer, I'm telling you up front that there will be unfamiliar terms used in this chapter without explanation.

Here's how to proceed. As you read through this chapter, when you come to something that isn't clear, make some sort of identifying mark in the outside margin of the page. Don't get stuck here, just move on because all of these unfamiliar terms will be explained in the next few chapters. Once you've read up to the chapter "Phonebooks", come back to this chapter and reread your marked sections so that the blanks are filled in.

After you've had a few online sessions under your belt and are feeling a bit more comfortable about this whole process, I strongly advise you to come back and read this entire chapter again. There will probably be several features you've forgotten about or never took advantage of that will suddenly appear very useful.

Part 1: The Terminal Window Components

The Terminal Window is White Knight's "main" window. It's where all received text (except during file transfers) is displayed. It's made up of several components:

- The Title Bar
- The Status Bar Area
- The Data Area
- The Size Box
- The Horizontal Scroll Bar
- The Vertical Scroll Bar
- The Keyboard Command Display

The Title Bar

The Title Bar is at the very top of the Terminal Window. In the center, the name of the application, "White Knight 11.x", is displayed. Clicking your mouse in this area and dragging allows you to move the window to any position on your screen.

On the left edge of the Title Bar is the Close Box. Clicking in this box quits White Knight and returns to the Finder.

On the right edge of the Title bar is the Zoom Box. Clicking in this box will cause the Terminal Window to expand to fill the entire display screen (except for the menu bar). Clicking in it again returns the Terminal Window to its previous size.



White Knight remembers the size and position of all of its windows between sessions. To override this, just hold down the Option key on your keyboard before the window is drawn, and continue to hold it down until the window is fully drawn. If you do this, the window will be opened in its default size and position.

The Status Bar Area

The Status Bar area is just below the Terminal Window's Title Bar. We'll be discussing the Status Bar in detail in Part 3 of this chapter.

The Data Area

The Data Area is just below the Status Bar Area, and it continues down to the Horizontal Scroll Bar. This is where all received text is displayed. Since the Data Area is where all the action is, and deserves a detailed discussion, we'll go over this in Part 2 of this chapter.

The Size Box

The Size Box is located in the bottom right corner of the Terminal Window. It is used in a standard manner to enlarge or shrink the size of the Terminal Window. In case you're wondering (or have an awfully large display screen), the largest Terminal Window you may have is about 32,000 pixels wide by 32,000 pixels high.

The Horizontal Scroll Bar

The Horizontal Scroll Bar is located at the bottom of the Terminal Window. It is active when the window is not wide enough to display a whole line of text. When this occurs, the Horizontal Scroll Bar is used to scroll left and right as needed to see all of a line.

Here's a trick. Rarely if ever will you receive a full line of the widest character in a font, so there's quite a bit of unused "white" space at the rightmost part of the

scrolling region when you use a proportionally sized font (a font where some characters are wider than others, like Chicago or Geneva). If you hold down the Option key and click your mouse in the "Page Right" area of the Horizontal Scroll Bar (the grey area between the thumb and right arrow box), White Knight will scroll right to the point where the rightmost character in the displayed lines will be flush with the right edge of the Terminal Window. Holding down the Option key and clicking in the "Page Left" part of the Horizontal Scroll Bar is equivalent to dragging the thumb all the way to the left.

The Vertical Scroll Bar

What happens to text that scrolls off the top of the Data Area? It's put in a thing called the **Display Buffer**, where you can scroll back and review it.

Please don't get the Data Area and the Data Buffer confused. The Data Buffer is a block of memory that is used to hold received text. The Data Area is the part of the Terminal Window where a portion of this block is displayed.

The Vertical Scroll Bar is located on the right edge of the Terminal Window. It is used to scroll back through the lines of text in the Data Buffer. White Knight defaults to remembering the last 48 lines of text received, but you can extend this number to cause White Knight to remember as many screens of received text as you have free memory to hold. To change the number of screens White Knight will remember in the Data Buffer, select **Local->Buffer->Size** and then enter the number of screens to allocate in the Data Buffer. A "screen" is defined as 24 lines of text, no matter how big or small you have the Terminal Window sized. Most users find 10 to 20 screens sufficient, but you might want to keep more. Be sure and refer to the "Memory Considerations" chapter before specifying a huge number of screens.

If you want White Knight to remember and restore the contents of the Data Buffer between sessions, select **Local->Buffer->Remember** and the contents of the Data Buffer will be stored in a file in the same folder as White Knight titled **"Saved White Knight Buffer"**. White Knight will look for this file when starting up, and if it exists, will restore the contents of the Data Buffer to what it was at the end of your last White Knight session. Choose **Local->Buffer->Remember** (so that it is no longer checkmarked) again to disable this feature.



The "**Saved White Knight Buffer**" file can be a lot larger than you might guess.

The formula for calculating its size is:

Size (in bytes) = 15+(NUMBER OF SAVED SCREENS * 6480)

Make sure you have enough free disk space to hold it!

Here's another trick. Let's say you save 20 screens of text in the Data Buffer. Until you've received 20 screens of text, the top of the Data Buffer is going to contain a lot of blank lines. If you hold down the Option key and click in the "Page Up" portion of the Vertical Scroll Bar (the grey area between the thumb and the up arrow box), White Knight will position the thumb to the first non-empty line in the Data Buffer (the first line of text received in that session). Holding down the Option key and clicking in the "Page Down" area of the Vertical Scroll Bar is equivalent to dragging the thumb all of the way to the bottom. Typing any character on the keyboard when the Vertical Scroll Bar is anywhere but at its bottom position will automatically move the Vertical Scroll Bar to its bottom position.

Menu Commands Affecting The Terminal Window

If you choose **Local->Window->Bring Others To Top**, all windows belonging to desk accessories will be brought to the top. This is useful if you have the Terminal Window sized to cover the entire screen. This only works under regular Finder, not MultiFinder, which treats desk accessories as if they were applications, in their own separate "layer". To move between applications under MultiFinder, use the commands at the bottom of the  menu, click in the small icon in the top right corner of the menu bar, or click in the desired application's window to bring it frontmost.

Choosing **Local->Window->Hide** will cause the Terminal Window to disappear. Choosing it again (so that it is no longer checkmarked) will cause the Terminal Window to reappear.

Choosing **Local->Window->Colors** brings up a dialog box allowing you to select the foreground, background, and in some cases the highlighting color of a number of White Knight Windows. This choice is only offered on machines that support Color Quickdraw. When in this dialog, checkmark the window types that you wish to add color to, and then click on the button beside each color if you don't like White Knight's default color suggestions. Color choosing

is done through the standard Macintosh Color Picker dialog box. The highlighting color is what White Knight uses for a background color when text is to be shown as selected, so choose something different than the background color for this.

Part 2: Idiosyncrasies Of The Data Area

The Character Cursor

You'll notice that there is a small underline drawn somewhere in the Data Area. I call this the Character Cursor (to differentiate it from the Mouse Cursor). This tells you where the next character that White Knight receives will be displayed.

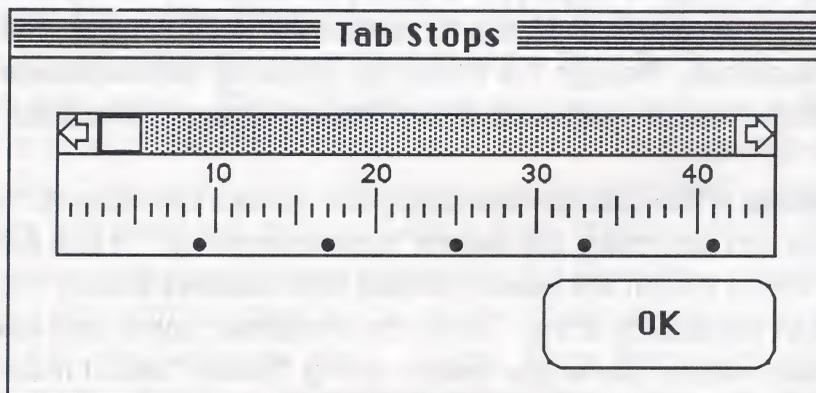
Display Font And Character Size

When using TTY (dumb terminal) emulation, White Knight can display received text using any font installed in your Macintosh, and at any point size. To set the font, select **Local->Window->Font**. First, a **Font** menu will appear for you to select a font. Once you've done this, a **Size** menu will appear for you to select a point size in which to display the font. The installed point sizes are shown in outline style in the **Size** menu, but you can choose any size by selecting **Size->Other Point Size** and then specifying a custom point size in the dialog box that follows (this may yield less than pleasing results). When you are emulating a VT52, VT100, or VT102 terminal, you may use only the TTY-VT52-VT100 font.

Tab Stops

When White Knight receives a tab character, it will interpret this to mean "print spaces up to the next tab stop or the end of the current line." By choosing **Local->Window->Tab Stops**, you can specify at which columns to place tab stops.

When this command is chosen, a dialog box as shown below is displayed on your screen:



The dialog box contains a column ruler, with small black circles showing where

tab stops are located. To set a tab stop, just point your mouse at the desired column in the ruler and click - a small black circle should appear. To remove a tab stop, just point at the column in the ruler containing a small black circle and click. The horizontal scroll bar above the ruler can be used to show columns not currently displayed in the ruler.

There are two important things to remember about tab stops. First, just because you place your tab stops in certain positions does not mean that the remote machine has agreed (or even knows about) these positions. It's similar to taking a typed report from one typewriter in your office and putting it in a second typewriter. If you are capturing the text on the screen to a file, you will probably have to set the tab stops again once the file is loaded into a text editor or word processor.

Secondly, White Knight's tab stop ruler is based on columns, whereas many word processors base their tab stops on inches (or centimeters). Therefore, if you are using a proportionally spaced font (where some characters are wider than others), don't be alarmed if columnar data doesn't line up on the White Knight screen. If you will be receiving columnar data, I suggest you use a monospaced font (where all characters are of equal width), like the Monaco or TTY-VT52-VT100 fonts.

Locating Text In The Data Buffer

You can quickly locate a desired string of characters by choosing **Edit->Locate Text**. White Knight will search from the top of the Data Buffer to the bottom and point out the first occurrence (if there is one) of the string of characters you specify. Whenever the screen is scrolled to display the found text, a zooming rectangle is displayed to help you locate the found text. Whether or not the window scrolls, the found text will flash several times. You can continue searching through the buffer by choosing **Edit->Locate Next** repeatedly until the search reaches the bottommost screen in the Data Buffer.

One of the features in the Locate Text dialog box (where the string to find is specified) is that you can make the search "case insensitive." If this item is not checkmarked, White Knight will search for text that matches exactly the upper and lowercase of the search string. If it is checkmarked, upper and lower case letters are considered the same (the search string "Bobby" would match with the words "bobby", "BOBBY" and "BoBBy" found in the Data Buffer or Data Area).

There are a number of menu choices that work with the Terminal Window and

Data Buffer.

Erasing Text

The command **Local->Window->Erase** erases just that part of the Data Buffer that is displayed in the Data Area. To erase all of the text in the Data Buffer, select **Local->Buffer->Erase**.

Selecting Text

The text in the Data Area is not editable as in a word processor, but you can copy text from the Data Buffer to the Clipboard, so that it can be pasted into some other Macintosh program. You can also paste text from the Clipboard to the modem (which simulates typing the text). Text is selected by dragging the mouse across it with the button held down as you would with a word processor. As text is selected, it is shown highlighted. You can scroll horizontally or vertically if the appropriate scroll bar is active by dragging your mouse outside of the Text Display Area in the direction you wish to scroll.

A quick way to select a block of text is to click your mouse once at the beginning of the block, then hold down the Shift key and click once at the end of the block (scrolling to where the end of the block is first, if necessary). The entire block will then be shown as selected.

If you select text and then change your mind, you can click your mouse anywhere in the Terminal Window and the selection will be undone. Once text is selected, you may choose from a number of choices under the **Edit** menu.

The Copy, Copy Table, and Copy, Then Paste Commands

Under the **Edit** menu, the **Copy** command will copy all of the selected text into the Clipboard. The **Copy Table** command works in much the same way, except that it helps you later import data from the Data Buffer into a spreadsheet or database application. **Copy Table** looks for more than one space character in a row, and converts all of the consecutive spaces to a single tab character. The data in the Clipboard is then referred to as "tab delimited", which many spreadsheets and database programs allow to be imported (consult the user manuals or manufacturers of these programs - not FreeSoft - for more details on how to import tab delimited data to the desired Macintosh application). **Copy Table** leaves all single spaces alone. **Copy, Then Paste** first copies the selected text to the Clipboard, then sends (pastes) the contents of the Clipboard to the modem as if you had typed the text.

The Print Selected Text Command

The **Edit->Print Selected Text** command will send the selected data to your printer. See the "Using A Printer With White Knight" chapter in this manual for more details about printing.

The Append To Command

The **Edit->Append To** command has three variants. Selected data can be saved in a new file, appended to the end of an existing file or appended to the end of the Archive File.

The Archive File is actually titled "Archived Text". It is basically a text file that White Knight will dump blocks of text into with various commands. You can select where the Archive File is to be placed by choosing **File->Archive File Destination**. In the dialog box that follows, simply navigate so that the desired disk volume or folder name is shown in the top center of the dialog box. Otherwise, the Archive File will be placed in the same folder as White Knight.

The "Archived Text" file can be imported into any Macintosh text editor or word processor. For information on how to work with this file, see the section titled "Working With A Received Text File" in the chapter "Receiving Text Files (File Captures)". The information in that section is also valid for the "Archived Text" file.

There are two menu commands for appending text to the "Archived Text" file. Choosing **Local->Window Archive** will save only those lines of text displayed in the Display Area. If you wish to save all of the text in the Data Buffer, choose **Local->Buffer->Archive**. This command will not save any blank lines in the Data Buffer until it comes to the first non-empty line in the buffer.

The Paste Command

Under the **Edit** menu, the **Paste** command will send the contents of the Clipboard to the serial port. What is actually happening here is that White Knight creates a temporary text file and copies the text in the Clipboard into this file. It then uses the **File->Send Text File...** menu choice to send the data to the serial port. When finished, it deletes the temporary file. Therefore, all of the options set under **Customize->Options->Text Transfer** (discussed in a later chapter) will also be used for Pastes.

The Undo, Cut, And Clear Commands

The **Edit->Undo**, **Edit->Cut** and **Edit->Clear** commands are not used by

White Knight. They are there to support those desk accessories which make use of these choices.

Part 3: The Status Bar Area

The Status Bar Area is in the Terminal Window above the Data Area and below the Title Bar. White Knight uses this area for a lot of different things, but we'll start with the General Status Bar (which is what is displayed the first time you run White Knight).



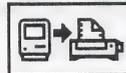
The Status Bar Area takes up some room in the Terminal Window that could be used for displaying extra lines of received text. If you don't need the Status Bar, select **Local->Status Bar->Hide Status Bar** and the Status Bar will disappear so that the Data Area can be extended to the top of the Terminal Window. Turn bring the Status Bar Area back, select this choice again so that it is uncheckmarked.

To select the General Status Bar display, choose **Local->Status Bar->General**. The General Status Bar is composed of six different control areas. From left to right:

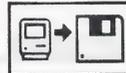
- The Elapsed Time Clock And Billing Timer:

00:47:26
\$3.91

- The Display Screen To Printer button:



- The Display Screen To Archive File button:



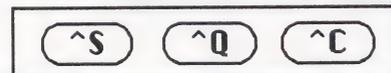
- The Serial Port Settings display:

Serial Port Settings
1200-N-8-1-FULL

- The Pause/Resume Remote Button:

Pause
Remote

- The Control Character Buttons:



The Elapsed Time Clock And Billing Timer

The Elapsed Time Clock and Billing Timer are used to keep tabs on how long you've been connected, and how much money you've spent so far. To reset the

clocks back to 00:00:00 and \$0.00, click in the box in the General Status Bar containing the clocks or select **Local->Timer->Reset To Zero**.

You can calibrate the Billing Clock to the desired charge by selecting **Local->Timer->Billing Cost**. This will display a dialog box prompting you to enter the cost per minutes in 100ths of a cent. In other words, if your service charges you \$5.00 per hour, you would divide 500 cents by 60 minutes to get 8.33 cents per minute. Multiply 8.33 cents by 100 to get 833, which is your cost per minute in 100ths of a cent. This may seem a little convoluted at first, but by specifying the cost per minute in 100ths of a cent, the Billing Clock offers a reasonable degree of precision. The Elapsed Time Clock and Billing Timer both run continuously from the time White Knight is first executed, regardless of whether or not the General Status Bar is selected.



If you leave the billing cost to zero (which is how White Knight starts out), only the Elapsed Time Clock will be displayed, not the Billing Timer.

The Display Screen To Printer Button

Clicking in the "Display Screen To Printer" button will send just the lines of text displayed in the Data Area to the printer. For details in setting up and using a printer with White Knight, see the chapter "Using A Printer With White Knight".

The Display Screen To Archive File Button

Clicking in the "Display Screen To Archive File" button is equivalent to choosing **Local->Window->Archive**, discussed in Part 2 of this chapter.

The Serial Port Settings Display

The Serial Port Settings Display shows you at a glance what baud rate, parity, databits, stopbits, and duplex are being used by the serial port White Knight communicates through. Clicking in this box is equivalent to choosing **Local->Serial Port**. Serial port settings are discussed in detail in the chapter "Serial Port Settings".

The Pause/Resume Remote Button

The Pause/Resume Remote Button is actually two buttons, "Pause Remote" and "Resume Remote" which toggle back and forth each time you click on them. Clicking the "Pause Remote" button will send a CTRL-S (also known as XOFF) control character, which many host machines understand to mean "stop

transmitting". Clicking on the "Resume Remote" button will send a CTRL-Q (also known as XON) control character, which many host machines understand to mean "resume transmitting".

The use of CTRL-S and CTRL-Q to start and stop transmission is known as **XOFF/XON handshaking**. It's not guaranteed that every host machine you'll connect with will support XON/XOFF handshaking, but the vast majority should. Let's say you are connected to a host at 2400 baud and it is displaying some text you are interested in. Unless you're a much faster reader than the average bear, text will probably start scrolling off the top of your screen before you can finish reading it (you'll know this is happening because your head will start to bob up and down uncontrollably). When this happens, click on the "Pause Remote" button and the remote machine should stop transmitting. When you're all caught up, click on the "Resume Remote" button and the remote machine should resume transmitting. This button makes it easy to start and stop the host by positioning the mouse over the Pause/Resume Remote Button and clicking your mouse to throttle the transmission to a more comfortable speed.

The Control Character Buttons

The Control Character Buttons can be modified to send any control character you wish by selecting **Customize->Control Buttons**. These buttons simply send the displayed control character when you click your mouse in them. They are useful for when your hand is on the mouse rather than the keyboard, or when certain control characters are needed but not readily available on the keyboard.

Now that you've been exposed to the basics of White Knight, it's time to start talking about doing some communicating!

The Basics Of Computer Communications

There are four distinct tasks involved in a computer communications session:

- Selecting the proper settings
- Making the connection
- Transferring data
- Disconnecting

Selecting The Proper Settings

Although White Knight contains "billions and billions" of settings, this generally boils down to selecting the Serial Port Settings and Terminal Emulation Settings. The Serial Port Settings control how your computer hardware sends and receives information electronically. The Terminal Emulation Settings control how what is received is displayed on your screen. This step will be discussed in Chapters "Serial Port Settings" and "Terminal Emulation".

Making The Connection

If you're using White Knight with a direct connection cable to a mini-computer or mainframe, you're already connected to the host machine, but you'll probably have to do something to get its attention (consult the System Operator of the host machine for instructions).

If you're using a modem, this step would involve instructing your modem to dial another computer and attempt to connect, or instructing your modem to wait for and accept calls from a remote machine that will be calling yours. Controlling the modem is discussed in the chapter "Modem Support".

Transferring Data

This is what computer communications is all about. By transferring data, I not only mean file transfers, but also sent and received text and typing. This step is covered in the chapters from "File Transfers" through "Step By Step File Transfers"

Disconnecting

When you're finished transferring data, you'll need to disconnect the two machines. If you are directly connected via cable, you'll need to issue some

sort of command to the host machine to disconnect you (consult the host machine's System Operator for instructions).

For a connection via modems, you'll need to instruct the modem to disconnect the call by hanging up the phone. This is covered in chapter "Modem Support".

Now, look over this page again. That's all there is to it, no hidden clauses, no trick questions, no salespeople will call. We're just about to get into the "good stuff", so take a break, water the dog, and fix up a cup of your favorite brew.

Serial Port Settings

If you look at the back of your Macintosh, you'll see that the serial port connectors are rather bland looking. I want you to think of these sockets as your Macintosh's mouth and ears.

When I speak, I use language to transfer selected thoughts to others. My mouth disturbs the air in front of it (no snickers here, please) which sends a waveform representation of the language to the listener's ears. The listener's ears, in turn, pass along these sound waves in a form that the brain reconverts back to language. The listener's brain has been known on occasion to instruct the listener's body to slap me a good one.

A serial port has send and receive channels also. Machine A's send channel is connected via some medium to Machine B's receive channel (and vice versa). Instead of sound waves, the medium reduces the computer's information to the ephemeral bleeps and burps of digital encoding you'll come to know and love. With White Knight, the medium will typically be either a direct cable between machines, or modems will be installed to transfer this information over ordinary telephone lines.

A modem is a device that works with a computer in the same manner that a telephone handset works with your head. Both of these translate the original information into the variety of bleeps and burps that the telephone lines are capable of carrying. The receiving handset or modem then retranslates the bleeps and burps back into voice or electronic signals.

Consider a telephone conversation between people. Both persons must (either implicitly or literally) agree to speak in a language both participants understand, and a certain degree of etiquette (or protocol) must be followed throughout the conversation (such as not hollering wildly at the same time). Using computers and modems is very much the same.

As I have become painfully aware, humans use a variety of languages and dialects to speak to each other. My backwoods noggin has a great deal of trouble understanding English spoken by many French people over the telephone, but not so much in person. I can remember one early morning (which is generally my bedtime) phone call from a certain Apple executive who has a decidedly French accent, and after a dozen or so "Pardon me's?" and

"Could you repeat that's?", I figured I'd best just start agreeing with the man or he'd likely come to the conclusion that I was a first class boob. The next 15 minutes were spent saying "Yes" and "Uh huh" at timed intervals, and I hoped like hell he wasn't asking me if I thought he was an idiot. At any rate, a new Mac II showed up at my doorstep a couple of weeks later, so I guess I lucked out. No telling what I agreed to, though.

The first thing you must know before calling another computer is what language it expects you to use. For computers, we refer to this language as the communications protocol or serial port settings. If you set up your computer to use a different protocol than the computer you call uses, you'll either get "garbage" characters on your screen or nothing at all. Sooner or later, one of you will get frustrated and hang up.

Communications protocol can be divided into several components. Each component must be set up correctly to communicate properly.

Baud Rate - this is how quickly the computers are capable of "speaking" to each other. The choices White Knight offers are 300, 450, 1200, 2400, 4800, 9600, 19200, 38400 and 57600 baud.

Parity - This is a simple form of error checking. White Knight supports NO Parity, ODD Parity, EVEN Parity, MARK Parity, SPACE Parity, and IGNORE Parity.

Data Bits - This size of each character. White Knight supports 5, 6, 7, and 8 data bits.

Stop Bits - Additional signals sent at the end of each character transmitted. White Knight supports 1, 1.5, and 2 stop bits.

Duplex - White Knight supports FULL Duplex, HALF Duplex, ECHO Duplex, and NULL Duplex.



HALF Duplex is what users of some other communications programs will refer to as "local echo", and should not be confused with ECHO Duplex. Just keep in mind that when someone tells you to "turn your echo on", they almost certainly will mean "use HALF duplex".

How To Select The Serial Port Settings

Before I talk about which settings to select, let me first explain how to set them. If you select **Local->Serial Port** (a shortcut is to click in the Serial Port Settings box in the General Status Bar), the following dialog box will appear:

Serial Port Settings

Serial port: Local Serial Port

Modem command inter-character delay: 60th's.

Modem init command:

Baud rate:

Parity:

Databits:

Stopbits:

Duplex:

Don't drop DTR when quitting

Hold DTR low

Use hardware handshaking

Serial port buffer size (100-32767 bytes):

As you can see, the current baud rate, parity, databits, stopbits and duplex are shown in the lower left hand corner of the dialog box. The rectangle that the values for each of these are displayed in is actually a pop-up menu. If you click in the rectangle surrounding the baud rate value, a menu will appear showing all the possible baud rate values for you to select from.

What Settings To Use?

In many situations, you'll be calling a host system and the serial port values it requires will be provided to you by those who administrate that system. If they give you a value for something called "startbits", just ignore that - it's already built into White Knight.

If you are calling an individual, I can give you some guidelines on what serial port settings to use on both sides. You should always use NO parity, 8 databits, 1 stopbit, and HALF duplex. The baud rate will be determined by who has the

"slowest" modem. If one person has a 2400 baud modem and the other person has a 1200 baud modem, both people will have to use 1200 baud because this is the lowest common denominator.



If you are both using 9600 baud modems and they are made by different manufacturers, there's no guarantee that they will be able to connect. The 300, 1200, and 2400 baud standards are pretty much locked in internationally, but for 9600 baud, quite a few manufacturers have gone their own direction, with the result that their modems will connect only with their own brand. If in doubt, call the manufacturers of the modems and discuss what you need to connect with. If worse comes to worse, you can always back down to 2400 baud. While I'm sure this Tower Of Babel situation is likely to improve, there's still a lot of confusion in this area. You should also know that the faster the baud rate you use, the more critical a "clean" phone line connection becomes. There have been many times I was unable to connect internationally at 2400 baud (and at least several domestic calls), and had to back down to 1200 or even 300 baud before I could get a connection.

If you are going to connect with a BBS (bulletin board service), you'll want to use NO parity, 8 databits, 1 stopbit and FULL duplex. Almost all BBS's support 1200 baud, so that's a safe place to start. Once online, you can find out if they support higher baud rates. Approximately 75% of the BBS's I've connected with support 2400 baud, so I usually throw safety to the wind and attempt to connect with unknown systems at 2400, and back down to 1200 if there's a failure to connect. Then again, my phone bill is tax deductible.

Often times you'll hear people talking about XON/XOFF handshaking. There is no special setting you must turn on to make White Knight support this - it is automatically supported.

If you're a beginner, that's about all the information you should need to set up your serial port settings and advance to the next chapter. One more piece of advice before we get into the mid-level difficulty stuff:



If you can't see what you type, turn on HALF duplex. If you see double characters when you type ("HELLO" is displayed as "HHEELLLLLOO") turn on

FULL duplex.

What Serial Port Settings Mean

This section is not for beginners, unless they're desperate for some interesting bathroom reading.

Baud rate is a measure of transmission speed, and it is relative - 1200 baud is four times faster than 300 baud. If you divide the baud rate by 10, you'll know the approximate number of characters per second that can be transmitted.

Now let's talk about databits and parity. Each byte of memory in your machine can hold a single number, ranging from 0 to 255 (a total of 256 values). If you have a good scanning electron microscope, crack open a memory chip and you'll see that each byte is made up of 8 "on or off" electronic switches. A bit of calculation shows that 2 (which is the number of "states" - on or off) raised to the 8th power (the number of "switches" in a byte) is 256. Voila! Each one of these switches is called a "bit".

In the English alphabet, there are only 26 letters. So why do we need 256 values? Remember that there are both upper and lower case letters, punctuation and symbol characters, and the numbers 0 through 9. We also need some special characters (called "control characters") to designate things like "move to the beginning of the next line", or "backspace one character". It turns out that we need about 128 codes to represent all of these characters. Now, 2 raised to the 7th power happens to be 128, so if our data consists only of text, we need only 7 bits to transmit all of our characters. The eighth bit is still transmitted, but it's used for something else.

It's a shame to waste a bit, thought the early computer communicators, so they came up with a very simple form of error checking called parity. Let's start with EVEN parity. The way EVEN parity works is that I count the number of data bits that are turned on. If the sum is even, I turn on the parity bit (the eighth bit in each byte). ODD parity works pretty much the same in reverse. If the sum of the turned on bits is odd, the parity bit is turned on. What's the point? The receiver can also sum up the bits and compare the result to the parity bit. If the parity bit is turned on when it shouldn't be, the receiver knows there's been a transmission error.

MARK and SPACE parity aren't quite as "smart". MARK parity always turns on the parity bit, and SPACE parity always turns it off. Therefore, if MARK parity is

being used and the receiver gets a character with the parity bit turned off, they know a transmission error occurred.

Now we take a left turn. If every byte of memory in your computer held nothing but text, the computer wouldn't be able to do any work. The same bytes of memory must also be capable of holding instructions to tell the computer's "brain" to do something. In fact, the brain in your Macintosh understands many more than 256 instructions, and it can take several bytes (using all eight bits) to represent a single computer instruction.

What to do? Well, parity checking turned out to be a pretty stupid form of error checking. For EVEN and ODD parity, if two data bits were changed by a "glitch", the parity would look correct but the character would be wrong. MARK and SPACE parity wouldn't even catch a single bit "glitch" as long as the parity bit wasn't hit. The answer seemed to be to get rid of parity checking and use all eight bits as data bits.

You've either had a moving revelation, or you're more confused now than ever. If I blew your mind a bit there, don't despair. The bottom line is that if you are using 7 data bits, you'll always specify some form of parity. If you're using 8 data bits, you'll always use NO parity or IGNORE parity.

NO parity and IGNORE parity have a subtle difference which is important. When NO parity is being used, White Knight always strips (turns off) the eighth bit before a character is displayed in the Data Area of the Terminal Window. This is because I've found that some host systems (like CompuServe) will accept your keystrokes using either EVEN parity or NO parity, but will always send text using EVEN parity. Therefore, if I strip off the parity bit, the text will still appear fine on your screen. IGNORE parity will not strip off the parity bit. This is used only when you are using an extended character set that can use all values from 0 to 255.

The Bottom Line



IGNORE parity is used only in special circumstances. Unless you are transmitting textual information that contains international characters like ñ or ô, you should always use NO parity, otherwise, if you are transmitting international characters, use IGNORE parity.

Unless you are specifically told otherwise, you should always set stopbits to 1.

Duplex controls what White Knight does with the characters you send. When you are using FULL Duplex, the characters are sent directly through the serial port without going to the display screen. FULL Duplex is used when the remote machine will echo back the characters you type. When you are using HALF Duplex, White Knight not only sends the characters you type through the serial port, but also to your display screen. This is used when the remote machine does not echo back the characters you type.

ECHO and NULL Duplex are provided for special cases. ECHO Duplex functions like HALF Duplex, except that it also echoes back any characters received. NULL Duplex also acts like HALF Duplex by sending the characters you type to your display screen, but it does not send those characters over the serial port. NULL Duplex is used mainly for local testing purposes.

One More Wild Thing

Just to frustrate me, it seems that a few rare systems actually need 7 data bits and NO parity. Even though I couldn't believe this, I called them up and verified that they would only work with that setting. Apparently, most modem makers were equally aghast at such a suggestion, because at least a dozen different makes of modems I've tested would not respond to 7 data bits and NO parity. The solution was to use 8 data bits and NO parity until the modems connected, and then switch to 7 data bits and NO parity. In order to tell the modem to hang up, I had to switch back to 8 data bits and NO parity. Bizarre? Yup.

Advanced Serial Port Settings

As the serial port receives information, it is placed in a special block of memory called the "serial port buffer". White Knight grabs information from this buffer as it is able to process it. The "Serial port buffer size" option is where you tell White Knight how large a buffer (up to 30,000 bytes) you would like to allocate. I recommend you keep this at the default size of 10,000 bytes unless directed to do otherwise by The FreeSoft Company.

Handshaking

One of the maddening things about Macintosh communications is that there does not seem to be a standard for modem cables. The Mac serial port contains two handshaking lines, which have been given the generic names "Input Handshake" and "Output Handshake". What these lines are connected to, if they are connected at all, seems to vary from modem cable manufacturer to

manufacturer. Before you attempt to use any of the features in this section, I suggest that you contact the manufacturer and find out just what you have.

Most of the time, I've seen Input Handshake left disconnected and Output Handshake connected to the DTR (Data Terminal Ready) line. If this is the case, your modem will automatically hang up when you quit White Knight. You can prevent this from happening by checkmarking the "Don't drop DTR when quitting" option.

If you're using a digital phone system, it might want to use the DTR from some controlling functions. You can instruct White Knight to hold the DTR in a suppressed state by checkmarking the "Hold DTR low" option. You can invert the state of DTR for one second by clicking in the "Invert DTR for one second" button (this might be used for what's called a "Flash" signal for gaining the attention of a local area network or modem equipped PBX system).

Of course, none of these settings will have any effect unless the RS-232 DTR line is connected to the output handshake pin of the serial port White Knight is communicating through, or if the device connected to the serial port is instructed to ignore the state of the DTR line.

With the advent of high speed (9600 baud+) modems, some manufacturers are putting together cables that use the Input and Output Handshake lines connected to the CTS and RTS (Clear To Send and Request To Send) lines. White Knight can be told to use such a cable for high speed connections by checkmarking the "Use hardware handshaking" option.

Terminal Emulation

Before personal computers were available, people communicated with large host systems through a device called a "terminal". Terminals had little if any processing power, they were used mainly as input and output devices. The least intelligent of these terminals were called "TTY" devices, short for teletypewriters. I suppose the reason you hear these clanking monstrosities at the beginning of local news broadcasts is mainly for nostalgia - I'd question any news bureau that is still using one, unless they simply couldn't find the three men and a boy necessary to move the behemoth out.

However, you'll find that a large number of commercial and private host systems support TTY devices (also known as "dumb terminals") for the simple reason that it does function well as a lowest common denominator.

Terminal emulation defines how White Knight reacts to certain characters sent by the host system. TTY emulation, for instance, has a few very simple commands for clearing the screen, moving the cursor to the next line, backspacing, tabbing, and sounding a bell. VT52 emulation adds several additional screen formatting controls, such as those to allow the quick placement of the cursor anywhere on the screen. VT100 emulation is the next most advanced, allowing such things as selective scrolling, character attributes such as boldface and underlining, and tab stop placement under host system control. VT102 emulation adds additional features such as automatic printer control.

Because TTY's offer only the basics of screen display control, manufacturers were quick to provide "next year's model" with more and more advanced features. Modern terminals can cost as much or more than your Macintosh system.

The Macintosh computer is a computer, not a computer terminal. Therefore, if we wish to connect our Macintosh as a terminal, we need to act like some kind of terminal the host system would expect to have connected to it. This is referred to as "terminal emulation".

White Knight offers four different kinds of terminal emulation, meaning it can fool the host system into thinking it's talking to something that it recognizes. These are the terminal types that are most often supported by host systems. Along

with the TTY terminal type, White Knight also supports the emulation of three terminals manufactured by Digital Equipment Corp., known as the VT52, VT100, and VT102.

Which Emulation To Use?

If you are talking to an individual, you will always use TTY emulation on both sides. If you are talking to a host system, you will need to find out what kinds of terminals the host system supports. If you're not sure, start out with TTY emulation. The worst that will happen is that you'll see some "garbage" characters intermixed with the text, in which case you'll know that you should try one of the other terminal types. If you are asked to choose a terminal and don't see "TTY" listed, look for some other generic terminal name like "CRT", "Dumb Terminal", or "Other". If given a choice, choose from the following order of preference: TTY, VT52, VT100, and most desirable, VT102. You should also know that VT100 and VT102 are also known as "ANSI compatible terminals", so if you are given that option, choose one of those terminal types.

How To Select Terminal Emulation

By selecting **Customize->Options->Emulation**, you'll see the following dialog box:

Emulation Options

Display columns (20-132):

Cursor style: Flashing Block

Terminal emulation: TTY VT100 VT52
 VT102

Enable CompuServe 'RLE' graphics

Use filter: Pre-emulation Post-emulation

The "Display columns" options is where you tell White Knight where to place the right margin. Although you can enter any number from 20 to 132 columns, you'll find that most terminals expect either 80 columns (most usually) or 132

columns.

Under the "Cursor style" option, you can specify that the character cursor will either flash or stay solid, or if it should be displayed as a solid block or an underline. Experiment with these and use whatever is most comfortable for you.

Next, under "Terminal Emulation", you tell White Knight which sort of terminal to emulate.

Finally, you can tell White Knight to support the reception of CompuServe's "RLE graphics", and whether or not to filter the incoming data. Both of these will be discussed later - for now, leave them unchecked.

If you're just starting out, you're ready to advance to the next chapter. However, keep in mind that there are several advanced features discussed next in this chapter that you will want to come back to and study later.

Part 2: Advanced Emulation Features

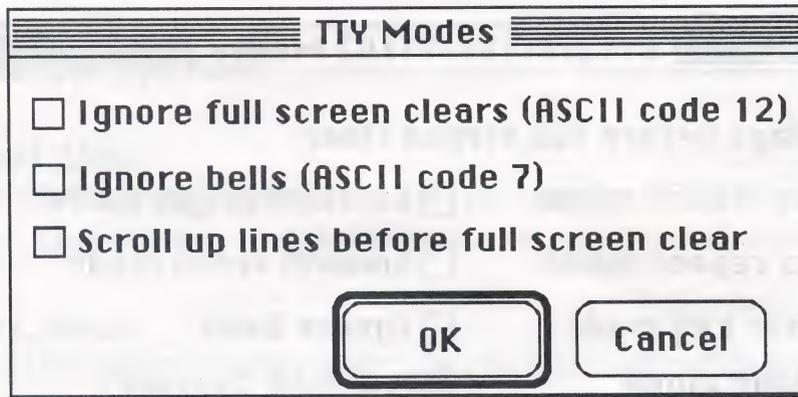
In this section, we're going to look at a few of the advanced features White Knight provides for terminal emulation.

RLE Graphics

When you select **Customize->Options->Emulation**, the dialog box displayed earlier in this chapter allows you to tell White Knight whether or not to support CompuServe's RLE graphics. Because of a conflict, you must be using an emulation other than VT52 to receive these graphics. RLE graphics will be received into a small window that will appear when the transmission begins. A bell will sound when the graphics have been entirely received. It is possible that CompuServe will tell you something like "Your terminal doesn't support RLE graphics, do you wish to go ahead anyway?" If it does, just reply "Y" for "Yes" and the graphics will come over properly. One final note, RLE graphics is not the same as CompuServe's GIF graphics. GIF graphics will have to be downloaded with a file transfer and displayed using a utility program available in the Macintosh forum on CompuServe.

TTY Modes

When you have selected TTY emulation, you will see a **Customize->TTY Modes** choice available. Selecting this brings up the following dialog box:



When White Knight's TTY emulation receives a form feed character, it will erase the bottom 24 lines of the Data Buffer. Once this information is erased, it's gone forever. This circumvents the usefulness of being able to scroll back to look through received information, so White Knight provides a couple of alternate ways to handle received formfeed characters. If you checkmark the "Ignore full screen clears", White Knight will simply ignore them and not erase any lines. However, if you like a nice clean display, but still want the information that would be erased, you should checkmark the "Scroll up lines before full screen clear" option. When this option is checkmarked, White Knight will "push" the bottom 24 lines of the Data Buffer up 24 lines, which has the effect of erasing the bottom 24 lines while still keeping the most recently received 24 lines in the Data Buffer.

If you're working in an environment where you don't want to disturb others, you might want to checkmark the "Ignore bells" option. When this is checkmarked, White Knight will flash the menu bar instead of ringing a bell.

VT Modes

When you have selected VT52, VT100, or VT102 emulation, you'll see that a **Customize->VT Modes** choice is available. Selecting this choice brings up the following dialog box:

UT52/UT100/UT102 Modes	
<input type="checkbox"/> Scroll up lines before full screen clear	
<input type="checkbox"/> UT100 wraparound mode	<input type="checkbox"/> Relative origin mode
<input type="checkbox"/> UT100 auto repeat mode	<input type="checkbox"/> Smooth scroll mode
<input type="checkbox"/> UT100 cursor key mode	<input type="checkbox"/> Ignore Bells
<input type="checkbox"/> UT100 keypad mode	Answerback Message:
<input type="checkbox"/> UT100 newline mode	<input type="text"/>
<input type="checkbox"/> Insert mode (UT102 only)	
UT-Mouse delay speed (0-60):	<input type="text" value="8"/> <input type="checkbox"/> UT-Mouse waits for host
<input type="button" value="OK"/> <input type="button" value="Reset Terminal"/> <input type="button" value="Cancel"/>	

The "Scroll up lines before full screen clear" and "Ignore bells" options work just like their counterparts in the "TTY Modes" dialog discussed previously. However, you should be aware that VT terminals offer a lot of different ways to do a full screen clear. White Knight will catch most of them, but it's impossible to catch all of them.

Several of the modes in this dialog box are native to the VT terminals themselves and have nothing to do with White Knight's emulation of them. In fact, the host system can turn these modes on or off without you even knowing about it. Therefore, it's best to leave these alone and let the host system software set them up the way it wants them. However, I'll give you a brief description of what each of these modes do for your own edification.



Clicking the "Reset Terminal" button will bring back the VT Modes settings that were in place when White Knight was first executed.

Wraparound Mode

When this mode is set (checkmarked), typing a character in the rightmost column (margin) will cause the character cursor to automatically jump down to the beginning of the next line. When this mode is reset (not checkmarked) the

character cursor will stay in the right margin column and overwrite characters in that column as you type them.

Auto Repeat Mode

When this mode is set, most characters on your keyboard will repeat when held down. When it's reset, only one character will be sent for each press of a key.

Cursor Key Mode

The setting of this mode controls what characters are sent when a cursor movement key is typed. You should normally leave this mode reset unless your cursor keys suddenly stop working.

Keypad Mode

The setting of this mode controls what characters are sent when a key on the keypad is typed. Some VT programs use the keypad as a sort of command pad. If this mode is set, typing a key on the keypad will send what's known as an "escape sequence", which might be meaningful to the host software. When it's reset, the numbers printed on the keypad keys are sent instead. You probably shouldn't play with this mode unless the keypad keys don't seem to function as you would expect.

Newline Mode

When set, a received carriage return character tells White Knight to jump to the beginning of the line and then move down a line. When it's reset, a carriage return character tells White Knight just to move to the beginning of the line, and a different character, called a linefeed, tells White Knight to move down a line. Normally, you want to leave this mode reset unless your cursor doesn't go to place you expect when you type the Return key.

Insert Mode

This mode, when set while emulating a VT102, allows you to insert characters (pushes existing characters to the right) in a line. When reset, typing characters in a line overtypes the characters there before. You'll normally want to leave this mode alone.

Relative Origin Mode

This mode is generally set when VT software wishes to use a "split-screen interface" (only part of the screen can be typed in - another part might be used for command or status information). You should normally leave this mode alone unless you find that you can't move your cursor into an area of the screen you

think you should.

Smooth Scroll Mode

When this mode is set, White Knight will smoothly scroll each line up the screen. Although it's aesthetically more pleasing, it also cuts down the throughput dramatically, so play with it if you like, but don't expect lightening fast displays.

Answerback Message

The Answerback Message is a string of characters that White Knight will send out in response to a certain inquiry from the host system. This is typically a password or account identifier. If the administrators of the host system you wish to connect to tell you to put something in your Answerback, this is where you type it in. A couple of pointers - these strings often contain escape and/or carriage return characters. Don't actually type these characters into the answerback string, instead, they are both represented as two-character sequences. Enter an escape by typing "^[" (a caret character - Shift-6 on your keyboard, followed by a left square bracket character). Enter a carriage return by typing a "^M" (a caret character followed by an uppercase M character). For example, if my answerback message was to be escape, followed by my first name, followed by a carriage return, I would enter the following string:

```
^[SCOTT^M
```

The VT-Mouse

You'll find that jack-hammering the cursor movement keys can be quite tiring and hard to follow. Since we've got a mouse on our Macintosh, let's use it! If you hold down your Option key when VT52, VT100, or VT102 emulation is active, you'll see that the mouse cursor turns into a small rectangle whenever it is moved into the Data Area of the Terminal Window as shown in the following picture:

```
people to come to the a  
to come to the [id of tl  
ome to the aid of their
```

You'll find that this rectangle fits snugly around a character of text as shown above. Clicking your mouse will cause White Knight to send the appropriate series of cursor movement keys to move the cursor to that row and column. A great deal of time has been spent to make sure that this routine works as documented, but I've found that some VT100 editors use some very bizarre input throttling and cursor movement optimization methods. The heuristics incorporated into the VT-Mouse feature are as follows:

If the "VT-Mouse waits for host" option is checkmarked:

- 1) When you execute the VT-Mouse, it looks first at where the cursor currently is and then where it needs to go. It sends the proper cursor movement key codes to move in the right direction towards the destination.
- 2) It waits until the host sends the codes back to White Knight which mean "I got your request to move. OK, move it!". If this isn't received within 5 seconds, the VT-Mouse routine gives up.
- 3) It looks at where the cursor is now and repeats this cycle until it gets to where you originally specified you wanted it to go.

If the "VT-Mouse waits for host" option is not checkmarked:

- 1) It looks first at where the cursor currently is and then where it needs to go.
- 2) It sends all of the cursor movement codes needed to move directly to the desired ending location without waiting for the remote machine to respond.

I suggest you checkmark the "VT-Mouse waits for host" option to start with. If it appears that the VT-Mouse is getting caught up in an endless loop, then and only then turn off this option.

Even with this intelligence, it is still possible to overrun the host's ability to process the incoming cursor movement keys. The symptom is a group of seemingly random "garbage" characters dumped in the middle of your text. The solution is to increase the "VT-Mouse delay" value. This delay value ranges from 0 to 60. It specifies in 1/60th of a second increments how long to wait before sending the next cursor movement key.

The VT-Mouse feature is not guaranteed to work inside of all VT host applications. Think of it this way: if it works, hubba! If it doesn't, you're no worse off than any other VT user in the world.

The VT100 Status Bar

The VT100 Status Bar is used for two purposes:

- To display the condition of the four LED's (small lights) found on an actual VT100 terminal, but not on a Macintosh.
- To allow you to simulate typing PF keys, cursor movement keys, and numeric keypad keys found on actual VT52, VT100, and VT102 terminals with your mouse.

The VT100 Status Bar is only used when you are emulating a VT terminal. It has no function when you are emulating a TTY terminal. To display the VT100

Status Bar, choose **Local->Status Bar->VT100 Keys**. The VT100 Status Bar is divided into three control sections:

- The PF keys and LED display:

PF1	PF2	PF3	PF4
L1	L2	L3	L4

- The cursor movement keys:

	↑	
←	↓	→

- The numeric keypad keys:

0	1	2	3	4	-	,
5	6	7	8	9	.	ENTER

With the exception of the four LED's and the two blank boxes surrounding the up cursor movement key, clicking your mouse inside any of the boxes is equivalent to pressing the same key on an actual VT52 or VT100 terminal. If the VT100 Auto-repeat mode is set, the keys in the VT100 Status bar will repeat as long as the mouse button is held down inside them.

The two blank boxes surrounding the up cursor movement key are not used and will just beep at you if you try to select them. The four LED's L1, L2, L3, and L4 will be shown as white characters on a black background when they are "lit". In the above illustration, L2 and L4 are "lit", while L1 and L3 are not "lit".

Filtering

Filters are discussed in detail in the chapter "Filters" (surprise!). What we'll mention here is that if you choose **Customize->Options->Emulation**, you'll notice that you can select filtering to be done either pre-emulation or post-emulation. Typically, you'll always want to do filtering post-emulation, which is to say, after White Knight has already extracted any control characters and emulator commands and is ready to display the text in the Data Area of the Terminal Window. However, if you select pre-emulation filtering, the filtering will be done on the data directly as it comes from the serial port, and before the terminal emulator has a chance to see the data. Pre-emulation filtering is provided mainly for debugging purposes by very experienced communicators.



A Special Note About Fonts For International Users

When using TTY emulation, you can choose any font, and any point size for displaying text in the Data Area of the Terminal Window. When using VT emulation, White Knight will allow you to only use the TTY-VT52-VT100 font. This is for two reasons. First, White Knight uses the highest 128 characters of

this font to hold the boldfaced versions of the normal character set as well as the VT special graphics character set. Secondly, the VT emulations require a non-proportionally spaced font (meaning it needs a font where all characters are the same width) so that the cursor can be positioned anywhere in the Data Area correctly.

This means that if you receive, for instance, a ô or ä character, it would be displayed as some garbage character in the high end of the TTY-VT52-VT100 font. This is definitely not what we're after.

It is possible to use a different font than TTY-VT52-VT100 in conjunction with VT52 or VT100 emulation so that international, diacritical, and other symbolic characters can be used. The method for doing this is explained in the chapter "Using International Or Special Fonts With VT Emulation".

Keyboard Support

One more chapter, and we're all set to make our first connection. Hang in there! In this chapter, we're going to discuss how White Knight supports your keyboard for telecommunications applications. If you're a beginner, there's only a couple of things that are really important.

The first is how to send control characters. In addition to the normal A-Z alphabet and punctuation characters we use in normal written communications, there are many additional characters significant in communicating by computer. What they are and how they are used is really up to each individual service you'll connect with. Many are standard, but don't bet your last dollar that any or all will be recognized until you've read the service's documentation or have had a chance to experiment. Often, the correct characters to press will be displayed on your screen by the remote service at various times, like:

Press CTRL-S to pause, CTRL-Q to resume, or CTRL-C to quit.

The "CTRL", by the way, is a common abbreviation for "control". In this case, the service is telling you that they support CTRL-S for halting transmission, CTRL-Q to resume transmission, and CTRL-C to break out of transmission and return to a previous menu of choices. Control characters are often used in this manner for issuing various commands to the remote service. A 'C' and a CTRL-C are two different characters and should not be confused or used interchangeably.

The newer Apple Desktop Bus keyboards shipped with the Mac SE and II series computers have a "Control" key on the keyboard. The "Control" key is used just like a Shift key. To type a CTRL-C, you would hold down the "Control" key and type a "C". The Macintosh Plus keyboard does not have a Control key, so we have to be a bit sneakier to handle this.

If you have a Mac Plus, select **Customize->Options->Key Mapping** and the following dialog box will appear:

Key Mapping Options	
<input type="checkbox"/> ` key sends ASCII code:	<input type="text" value="27"/>
Backspace key sends:	<input checked="" type="radio"/> Backspace <input type="radio"/> DEL
Shift-Backspace sends:	<input type="radio"/> Short BREAK signal <input checked="" type="radio"/> DEL <input type="radio"/> Backspace
<input type="checkbox"/> Option key sends control characters.	
Shortcuts:	<input checked="" type="checkbox"/> Serial port settings <input checked="" type="checkbox"/> Macro key set
Return key sends:	<input checked="" type="radio"/> carriage return only <input type="radio"/> carriage return and linefeed
Keyboard:	<input type="radio"/> Mac Plus <input type="radio"/> Standard ADB <input checked="" type="radio"/> Extended ADB <input type="radio"/> Other
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

Midway down in this dialog box, you'll see a choice labeled "Option key sends control characters". If you checkmark this box, the "Option" key on your keyboard will send control characters. In other words, to type a CTRL-C, you would hold down the "Option" key and type a "C".



The NULL control character (ASCII code 0) can be sent by holding down your designated Control key (Control or Option) and pressing the Spacebar key.

Modem Break Signals

You may find yourself being instructed to type your "Break" key. On older terminals, the "Break" key was common, but not so much anymore. It did not send an actual character, but a kind of signal that the other machine could detect. There are two kinds of break signals, long and short. The short version is what's most commonly called for, and is what you should start with. To send a break, go to the option in the above dialog box that reads "Shift-Backspace sends:" and select the "Short BREAK signal" choice. You can now send a break

signal by typing a Shift-Backspace on your keyboard.



On some keyboards, the "Backspace" key is actually labeled "Delete". Don't be confused - I use the two words interchangeably in this manual.

If the remote system doesn't respond to a short break signal, try typing ⌘-Shift-Backspace. This will send a long break signal of 4-5 seconds duration.

VT Emulation Keyboard Support

If you are emulating a VT52, VT100, or VT102 terminal, there are some additional keys found on those terminals that are not found on Macintosh keyboards. We'll need to "map" those keys onto keys we do have.

White Knight automatically will recognize the Mac Plus, Standard ADB, and Extended ADB keyboards the first time you execute it on your Macintosh. You'll see at the bottom of the above dialog box next to the "Keyboard:" option the kind of keyboard White Knight believes you have connected. If you move White Knight onto a Macintosh with a different kind of keyboard, you'll need to change the setting here.



Apple will probably release machines with different keyboard types in the future, and there are several third-party keyboards that White Knight doesn't know about. Additionally, you just might not like the way White Knight maps the keyboard for VT emulation. You can map these keys yourself by following the instructions in the chapter "The 'Install Special Keys' Utility".

The four cursor movement keys are mapped to the four cursor movement keys on your keyboard, so nothing is special there. The numeric keypad on your keyboard is mapped in the following physical arrangement:

PF1	PF2	PF3	PF4
7	8	9	-
4	5	6	,
1	2	3	ENTER
0	.		

In other words, to type a "PF1" key, you would type the key on the top left corner of your numeric keypad.

The Standard ADB and Extended ADB keyboards have "Escape" keys, which function just like those on a VT terminal. However, the Macintosh Plus does not have an "Escape" key, so once again, we have to be a bit tricky to accomplish this. In the above dialog box, the topmost option is "` key sends ASCII code:". The ` key is on the top left corner of the keyboard, next to the "1" key. It just so happens that the ASCII code for an Escape key is 27, which you'll see has been entered as a default. If you checkmark this option, the ` key will work just like an Escape key. Another way to type an Escape is to hold down your designated Control key (Control or Option) and type a left square bracket character ([).

Part 2: Advanced Keyboard Options

Most computers will use an ASCII code 8 (CTRL-H) to mean Backspace, but you might run into an oddball system that requires an ASCII code 127 (DELETE) to do this. The "Backspace key sends:" and "Shift-Backspace sends:" options can be configured to send whatever combination of ASCII code 8 or ASCII code 127 you find necessary. You shouldn't have to fool with this unless the remote system doesn't respond to your backspacing as you would expect.

Return Key

The "Return key sends:" option lets you choose whether the Return key sends just a carriage return, or a carriage return followed with a linefeed character. If the characters you type appear perfectly on your screen but the host system does not respond when you press the Return key, or lines seem to overprint each other when you press Return, chances are the remote system probably

requires an additional character after the carriage return called a linefeed character. Don't select this option indiscriminantly, some host systems will not function correctly if this option is turned on.

Keyboard Shortcuts

There are two sets of shortcuts that can save you time and energy when using White Knight, serial port shortcuts and Macro Key set shortcuts.



Both of these shortcuts will interfere with being able to type some international characters - this is why I made these shortcuts an option.

If the "Serial port settings" shortcuts option is checkmarked, you can easily change the serial port settings with the following keyboard strokes:

Shift-Option-1	Change to 1200 baud
Shift-Option-3	Change to 300 baud
Shift-Option-2	Change to 2400 baud
Shift-Option-9	Change to 9600 baud
Shift-Option-7	Change to 7 Databits
Shift-Option-8	Change to 8 Databits
Shift-Option-N	Change to No Parity
Shift-Option-I	Change to Ignore Parity
Shift-Option-O	Change to Odd Parity
Shift-Option-E	Change to Even Parity
Shift-Option-M	Change to Mark Parity
Shift-Option-S	Change to Space Parity
Shift-Option-F	Change to Full Duplex
Shift-Option-H	Change to Half Duplex
Shift-Option-X	Change to Null Duplex
Shift-Option-D	Inverts the DTR modem line for one second
Shift-Option-R	Sends a Delete (ASCII code 127) character

If you have the "Macro key set" shortcuts option checkmarked, you can change the active Macro Key set in the Macros Status Bar by typing Option-1, Option-2, or Option-3, corresponding to the set number you want to have shown in the status bar area.

Modem Support

As soon as White Knight draws the Terminal Window, it is communicating through the serial port. From that moment until you quit White Knight, it will display in the Data Area anything that comes through the serial port, and will send anything you type out the serial port.

If you look through the user manual that came with your modem, you'll probably find that it has dozens of commands of its own. After configuring the serial port settings to something the modem will like, you can just begin typing modem commands into the Data Area, and if all is well, the modem will respond as expected by performing the command.

If you are using a modem that supports the "AT" command set (commonly, yet most often erroneously also known as "Hayes compatible"), you may never have to read your modem manual, because White Knight supplies built in support for the four most often performed tasks:

- Initializing the modem (make sure it's up and talking to us).
- Dialing or automatic redialing of a number and attempt to connect.
- Turn on (or off) automatic answer of incoming calls so that we can connect to someone trying to call us.
- Disconnect modems and hang up a call.

Initializing The Modem

This menu choice **Service->Modem->Initialize** assures us that the modem is connected properly and will respond to further commands. To be honest, this command is provided for the paranoid, because it will be pretty obvious if the other modem support routines fail. However, if this is the first time you've worked with White Knight, it is a real good idea to issue this command after connecting and turning on your modem, just to make sure that we're starting from even ground.

Auto-Answer

If a remote party will be calling your machine, you'll need to instruct your modem first (before the call comes) to automatically answer the incoming call when it comes and attempt to connect with the other modem. To do this, select **Service->Modem->Auto-Answer**. White Knight will send a command to the modem that tells it to pick up the phone on the first ring and attempt to connect. The modem will stay in auto-answer mode until you turn it off or tell it not to

auto-answer. To instruct the modem not to auto-answer, select **Service->Modem->Turn Off Auto Answer**.

Hanging Up

To disconnect your modem from a remote one and hang up the phone, select **Service->Modem->Hang Up**. Here's a hint, though. Only one side of a connection needs to disconnect. Whenever a modem loses contact with the other modem, it will also hang up the phone.

Dialing Or Redialing

To dial or automatically redial a number attempt to connect, select **Service->Dial Or Redial**. The following dialog box will appear:

Dial Or Redial A Number

Number to dial: 555 1212

Telephone line is: tone. pulse.

Wait for answer up to 30 seconds.

Dial only once. Redial until connected or 15 times.

Pause Two Seconds

Redial Options... OK Cancel

You type in the number White Knight is to dial at the top of this dialog box. If you have used previous versions of White Knight, let me point out that you will not precede the number to dial with a modem dialing command. Just type in the number itself. Spaces are ignored, so don't worry about them. If you're mouse oriented, you can click in the number buttons in the graphic telephone touchtone pad. The "Pause Two Seconds" button will be shown in the dialing string as a comma. An example of when this might be useful is when you're going through a switchboard, or using a long distance carrier that requires an identification number. Let's say that in order to get through my switchboard, I

need to dial a 9 and then wait for the second dial tone. From experience, I know that the second dial tone will certainly appear within four seconds, so two of those 2 second pauses should be enough. The number to dial would be:

9,,5551212

which means dial 9, wait four seconds, then dial 555 1212.

Telephone Line Type

You should tell White Knight what sort of telephone line you have (touchtone or pulse dialing) in the "Telephone line is:" item in the above dialog box.

How Long To Wait

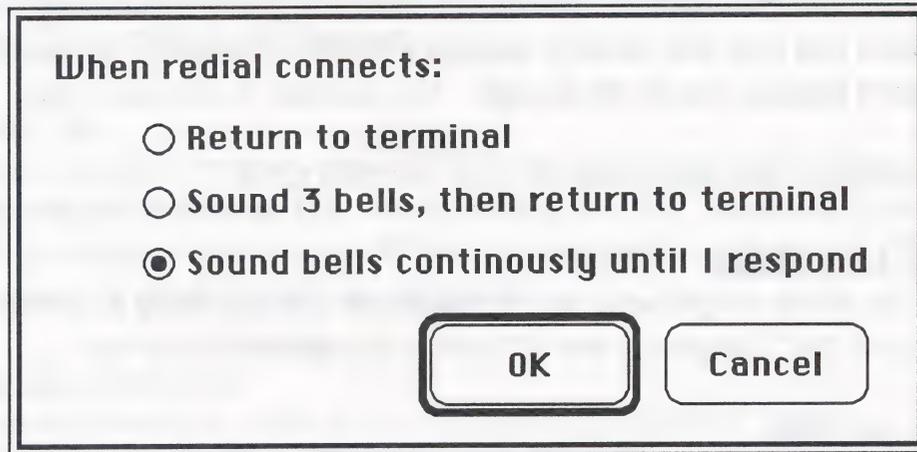
The "Wait for answer up to X seconds" item is where you tell White Knight how long to wait for a connection before giving up. This is 30 seconds by default, but you may need to extend this if you'll be dialing long distance (especially internationally), when a connection can take much longer.

Dial Or Redial

You can tell White Knight to dial the entered phone number only once, or continuously until a connection is established (called redialing). Redialing is extremely useful when you are calling a number that is prone to busy signals. Some countries have federal laws concerning how many times a number can be repeatedly dialed mechanically. In the United States, it's 15 times. You'll notice that the redialing choice reads "Redial until connected or X times". This "X" is called the "redial limit", and it can be a value from 0 to 255. If you are dialing from a country that does not regulate mechanical redialing, or you just have a criminal disposition, you can specify 0 (zero) for the redialing limit and White Knight will redial forever (if you let it).

Redial Options

If you click in the "Redial Options" button in the above dialog box, the following dialog box will appear:



This dialog box affects how White Knight is to act when a redialing results in a connection. You can instruct White Knight to just return to the Terminal Window, sound three bells first, or display a dialog box telling you that a connection has been made and sound bells continuously until you respond.



During a dial or redial, the dialing status box can be dragged to any position on the screen, so you could, for instance, keep an eye on it while working in another application under MultiFinder.

Part 2: Advanced Modem Support Options

If you select **Local->Serial Port**, the dialog box shown in the chapter "Serial Port Settings" is displayed. This dialog box has two items which affect White Knight's built-in modem support routines.

The "Modem init command" is the string White Knight sends to initialize the modem. The default string is "ATE1Q0V1" (that's a zero). If you consult the command list in your modem manual, you'll find that this command tells the modem to echo the characters White Knight sends as part of a modem command, and to respond in a verbal (not numeric) fashion. It shouldn't be changed except under the direction of The FreeSoft Company. If you can't resist playing with it, make sure and restore the default string above if things go "boom".

One item you can and should play with is the "Modem command intercharacter

delay" value. This is a value from 0 to 60 in 60th's of a second. Some modems that I've worked with have the peculiar trait that they won't accept a command at the full baud speed. The command must be slowed down to typing speed. This is what the intercharacter delay is for. What I would suggest is that you try a value of zero (no delay) and see if everything still works well. If it does, then the modem support routines will work much faster.

Modem Drivers

White Knight has the ability to support non-Hayes compatible modems (or other devices) through what is known as a Modem Driver. Modem Drivers can also be used to supplement the built in routines with up to 10 brand specific features, each of which will appear as a standard choice in the **Modem** menu. Modem Drivers are installed by selecting **Service->Modem->Use Driver** and selecting the Modem Driver file. Modem Drivers are a pretty tricky subject, and are not covered in this manual. If you are interested in how they work and how they are written, please read the chapter "Even More White Knight Features".

Phonebooks

Once you start taking advantage of the thousands of commercial and "free" services available by modem, you'll no doubt end up losing or misplacing phone numbers, account numbers, and passwords or notes. White Knight contains a phonebook feature that allows you to safely keep all of these entities collected in neat order much as you do in a mechanical desktop phone directory.

If you desire, you can create as many phonebooks as you wish, sorted by whatever criteria you wish. To create a phonebook, select the menu choice **Service->Phonebook->New**. You will be presented with a standard file definition dialog box asking you to name the phonebook.

A phonebook has an icon that looks like this: 

Phonebook

After you give it a name and click on the **Save** button, you will be presented with a dialog box asking you if you wish to password protect the phonebook. If you do not wish to, simply click on the **OK** button. Otherwise, type in the password you wish to use and then click on the **OK** button.

The password can include up to 20 letters, characters, or symbols. It is **very** important that you type the password slowly and deliberately. If you make a mistake or are unsure, type a backspace and everything you've typed will be deleted, allowing you to start over from scratch.

WARNING! WARNING! WARNING!

Choose a password that you will remember, because password protected phonebooks are encrypted in a very devious manner. This makes them safe from even low-level hacking by unscrupulous sorts and completely useless to those who do not possess the password. If you forget your password, your phonebook and everything in it is lost forever - that's an up front guarantee. **The FreeSoft Company will not under any circumstance remove password protection or unencrypt a password protected phonebook for any price or ANY reason, even if you can "prove" that it is yours.**

For security reasons, a phonebook that is not password protected when it is created can not later have password protection added. In addition, a phonebook that is password protected can not later have that protection

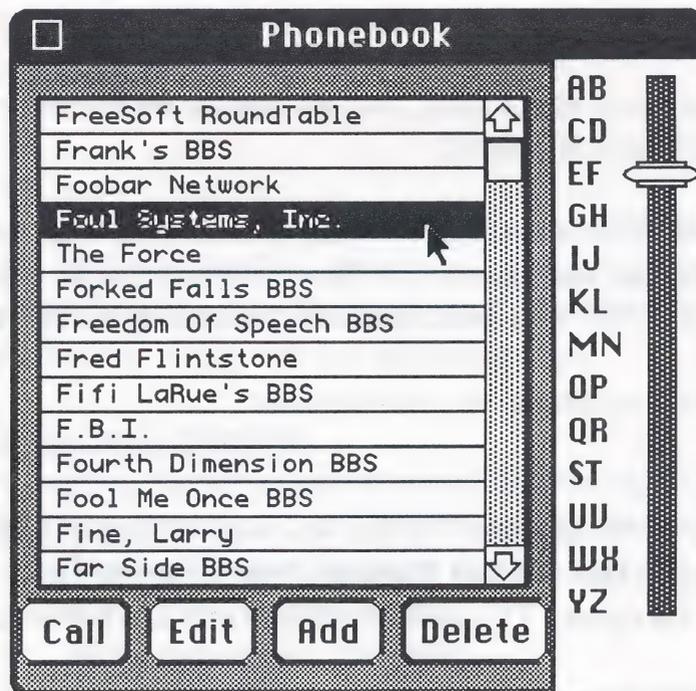
removed. Once you make the decision whether or not to password protect a phonebook, you are stuck with it until you or the phonebook dies.

After you create a phonebook with the **New Phonebook...** menu choice, you can open it up by selecting **Service->Phonebook->Open**. You'll then be presented with a file selection dialog box which prompts you to select the phonebook file to open.



Many users will not need more than one phonebook, so the multiple steps needed to open a phonebook can be antagonizing. When White Knight first executes, it will look in the same folder as itself for a Phonebook file titled "Phonebook". If a Phonebook exists with that exact name, a new choice, **Open Default Phonebook** will be added to the end of the **Phonebook** menu. This menu choice will immediately open up the "Phonebook" file.

A phonebook looks like this when opened:



You can click inside the Close Box to get rid of a phonebook at any time, or select **Service->Phonebook->Close**. The phonebook is modeled after its mechanical desktop equivalent. The name of the phonebook (in this case, I

created it with the title "Phonebook") is centered in the Title Bar. The right edge contains a sliding control knob that lets you open the phonebook to one of thirteen alphabetically arranged "pages" of entries. The last page the phonebook was opened to is remembered, and the phonebook will open to that page the next time it is opened. To move the phonebook to a specific page, simply drag the sliding knob with your mouse until it is next to the letters you desire. To display a different page, drag the control to a different pair of letters.

The **Call**, **Edit**, and **Delete** buttons are inactive until you select an entry by clicking your mouse in it. When selected, an entry is displayed highlighted, as in the entry "Foul Systems, Inc." in the above illustration.

Unlike the mechanical variety, the pages in White Knight's phonebooks are endless. When a page contains more than 14 entries, the vertical scroll bar on the right edge of the page becomes active so you can scroll through the various entries to the desired one.

To get rid of an entry, select it and then click on the **Delete** button.

To add a new entry, click on the **Add** button. A dialog box will appear asking you to supply the following information about the entry:

- **Name:** This is the name of the entry, that will appear on the selected page when the phonebook door is opened.
- **Dialing command:** This is the modem dialing command followed by the number to dial. Let's say you have a touchtone phone and wish this entry to dial the number 555 1212. You would enter as the dialing command:
ATDT 555 1212
if you had a pulse dialing phone, you would enter:
ATDP 555 1212
- **Settings file:** This is the Settings File that you wish to load in before the redialing commences. Use the **Get Settings File** button to easily fill in this entry with a proper file name. This item is optional and can be omitted.
- **Procedure file:** This is the compiled Procedure that you wish to execute after the redialing is terminated. Use the **Get Procedure File** button to easily fill in this entry with a proper filename. This item is optional and can be omitted.
- **Notes:** This can be anything you want, up to 80 characters of personal notes

about the entry, account numbers, passwords, etc. If you are going to put sensitive information in this section, I strongly advise that you password protect the phonebook so that it can not be compromised if stolen or copied. This item is optional and can be omitted.

Once an entry has been added, you can modify the information in it (or view the "Notes" data) by selecting the item and then clicking in the **Edit** button.

After selecting an entry and clicking in the **Call** button, the following steps take place in chronological order:

- 1) If the "Settings file" portion of the entry is not empty, the appropriate Settings File is loaded in.
- 2) The dialing command is passed to the **Dial Or Redial** function, and a redial until connection begins as explained in the chapter "Modem Support".
- 3) When the redial terminates, if the "Procedure file" portion of the entry is not empty, the appropriate Procedure is immediately executed. If it is empty, White Knight will respond in the same manner you have "Redial Options" set as explained in the chapter "Modem Support".



If you do have a Procedure selected for that entry, the YES/NO flag will be set in the same manner as for the REDIAL Procedure command, explained in the chapter "Procedure Commands". This is so the Procedure will know if the redialing was successful or not and can act accordingly.

Phonebook Keyboard Support

Many of the phonebook functions can be done from the keyboard, in addition to the mouse. The Phonebook supports the following keystrokes:

- | | |
|------------------|---|
| 'A' through 'Z': | Move to Phonebook page corresponding to the letter. |
| Up Arrow: | Move page up one notch. |
| Down Arrow: | Move page down one notch. |
| Left Arrow: | Move selected item up one entry. |
| Right Arrow: | Move selected item down one entry. |
| Return: | Same as clicking on Phonebook "Call" button. |

Gang Dialing

Because Bulletin Board Systems are frequently busy, I've added an interesting feature to the Phonebook called "Gang Dialing". This feature allows you to continuously redial a group of remote systems in round robin fashion until a connection is made.

To specify that an entry is to be made a member of the "gang", simply hold down the Option key on your keyboard and click on the entry. A small black dot will appear to the left of the entry's name, which tells you that it is a gang member. Option-click on it again to remove it from the gang. You can have as many gang members as you want on any number of different pages of the Phonebook, but you can have only one gang per Phonebook.

To execute the Gang Dialing routine, select **Service->Phonebook->Gang Dial**. You'll notice that in the redialing status dialog box, the name of the Phonebook entry will be shown rather than the number being dialed. This is a really useful feature, so if you frequent bulletin board systems, I strongly suggest you check it out.

When Gang Dialing, White Knight will first load in any Settings File associated with the phonebook entry, and then place the call, if a connection is established, it will then execute the Procedure associated with that entry, or if none was entered, will alert you to the connection.

The Buffered Keyboard

The Buffered Keyboard was added mainly for use with the real time conferencing features of several commercial timesharing services like CompuServe, GENie, and Delphi. A real time conference is where two or more (sometimes several dozen or even hundred) people get together and type to each other in a CB-radio fashion, exchanging opinions, tips, bad jokes, etc. The value of the Buffered Keyboard is that it gives you the opportunity to correct mistakes (or obscene punch lines) before sending them, and it sends the characters typed all at once, so that even the slowest typist doesn't hamper the faster typists in the crowd. If you take advantage of real time conferences (I suggest you do - they can be very educational and a lot of fun), you'll find the Buffered Keyboard indispensable.

To display the Buffered Keyboard, select **Local->Status Bar->Buffered Keyboard**. Whenever the Buffered Keyboard is displayed, the characters that you type are not sent to the serial port immediately, but are instead held in an editable buffer until the Return key is pressed. When that happens, the characters are "flushed" out of the buffer and sent through the serial port all at once.

The exception to this is that all control characters, except tabs and carriage returns, and characters from sources other than the keyboard (such as Macro Key Strings, VT100 Status Bar keys, and Procedure TYPE commands) are sent immediately through the serial port without effect to the buffer. It is not possible to put control characters in the buffer. Tabs are expanded as spaces (according to where you have tab stops set) just as inside the Display Area.

The text inside the buffer is editable, which means it can be selected with the mouse, and then cleared, copied, or cut to the Clipboard. To use the **Cut**, **Copy**, **Paste**, and **Clear** commands under the **Edit** menu with the buffer, hold down the Option key when selecting the desired command from under the **Edit** menu. The command must be selected from the **Edit** menu with the mouse - keyboard menu command equivalents won't work for this.

For example, holding down the Option key and selecting **Paste** from under the **Edit** menu tells White Knight to paste the contents of the Clipboard to the Buffered Keyboard, rather than directly to the serial port.

Some services have a limit of how many characters you can type before sending a carriage return. Exceed that line length limit, and the service will probably choke, send you an obscure error message, and pitch what you typed into oblivion. For this reason, White Knight supports a modifiable right margin for the Buffered Keyboard that it won't let you type past. To change the position of this margin, the dialog box brought up by **Customize->Options->White Knight** contains the choice **Buffered keyboard right margin column**. The horizontal scroll bar above the buffer is activated (if necessary) to allow you to scroll left and right for viewing all of the characters in the buffer.

You'll also notice that to the left of the Buffered Keyboard's horizontal scroll bar, your current right margin column and the number of characters in the buffer is displayed. As you type past the right edge of the Terminal Window, the buffer is automatically scrolled left for you so you can keep your eyes on what you're typing. Once you hit the right margin, a bell is sounded and no more characters will be accepted until you either delete one or more characters, or send the contents of the buffer by pressing the Return key.

The characters in the buffer are always displayed in White Knight's TTY-VT52-VT100 font at 9 point size.

You can also embed what I call "soft carriage returns" in the Buffered Keyboard to send multiple lines of text. To do this, type a Shift-Return, and instead of sending the contents of the buffer, a downward pointing arrow character will be inserted in the buffer at that point. When you type the Return key, these "soft carriage returns" will be converted to true carriage returns as the text is sent.



Note for advanced users. It is possible to tell White Knight to delay a certain amount of time after sending each soft carriage return in the buffer. To set this up, you'll need to be able to write a short Procedure. Look at the description for parameter #500 in the chapter "The GETPARAM And PUTPARAM Procedure Commands".

Filters

Every character of information that White Knight sends and receives has a numeric value from 0 to 255. When I type an "A" character, for instance, White Knight sends the value 65. This value is known as the "ASCII code" for that value. ASCII (pronounced "ass' key") stands for American Standard Code for Information Interchange. It's simply a standard that all microcomputer (and most mini and mainframe) manufacturers have agreed to use so that text typed on one computer can be displayed on another. Therefore, when White Knight receives a value of 65, it displays an "A". To see which values yield which characters, refer to the chapter "The ASCII Character Set".

Some ASCII values don't display actual characters, they're used for other reasons like ringing a bell or causing a tab, backspace, screen clear, or carriage return. These are known as "control characters" and have ASCII values between 0 and 31 (inclusive) and 127. The alphanumeric characters range from 32 through 126 (inclusive). Above code 127, the ASCII standard comes to a halt. It's really up to each manufacturer as to what character these values represents. On the Macintosh, these can be different for each font. To get an idea, choose the KeyCaps desk accessory and hold down the Option or Shift-Option keys to see what Apple did with these "undefined" values.

You might find that on machine X, an ASCII value of 150 yields the "±" character, but in the font you're using in the Data Area of the Terminal Window of White Knight, this character has an ASCII value of 177. Somehow, you'll need to be able to translate any received code 150 into a code 177.

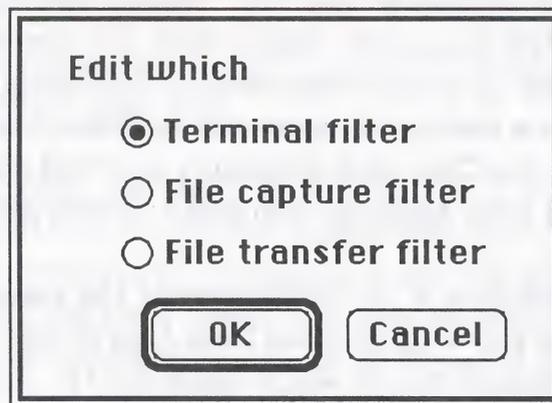
Another problem comes with control characters. Although there are 32 control characters, very few of them are actually used nowadays. There are few standards even among the few commonly used. ASCII code 10, called a "linefeed", has no meaning on a Macintosh and is displayed as a rectangular "garbage" character. However, this character is very important on an IBM-PC (and many other computer brands), so you're likely to get a lot of them if you are communicating with such a machine. Somehow, we'll need to get rid of any code 10's that we receive.

But when we see these garbage characters in our received documents, how do we know which characters to tell White Knight to strip out? We'll need some way to examine the ASCII codes of suspect characters in order to find out which

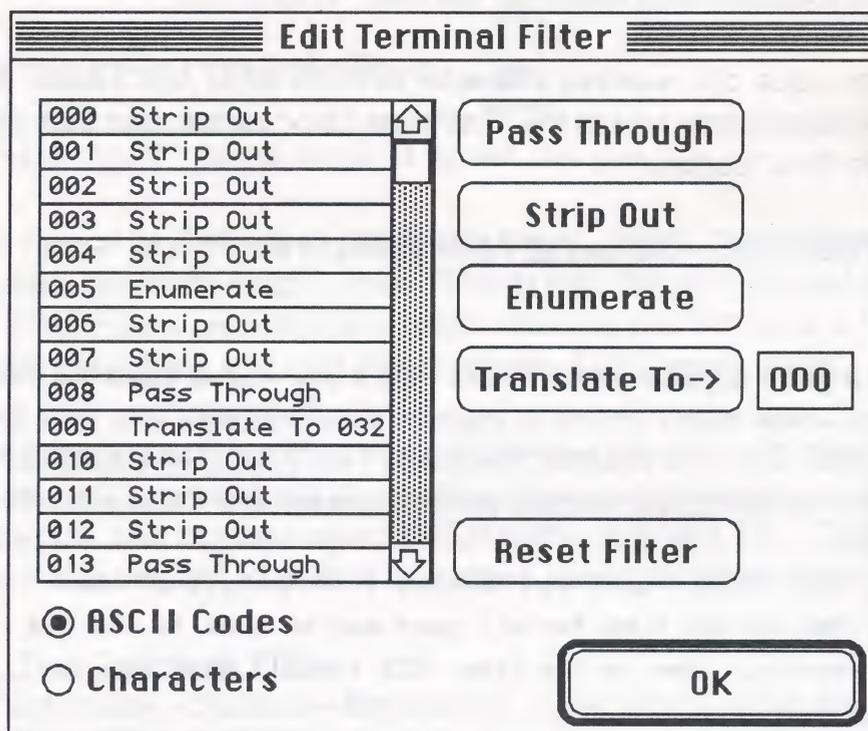
are the offenders.

White Knight has the ability to look at incoming data, character by character, and at your command strip out the character, change it to some other character, show the ASCII code of the character rather than the character itself, or just pass the character through unchanged. To do this, we use a thing called a "Filter". White Knight has three different filters. The "Terminal Filter" is used to filter information that will be displayed in the Data Area of the Terminal Window. The "File Capture" filter is used to filter characters that will be captured to a disk file. The "File Transfer" filter is used to filter characters that have been received via a file transfer protocol. Each of these filters can be turned on or off independently of each other. You can choose to use one, two, three, or none of the filters at any given time.

To look at or edit a filter, select **Customize->Filter->Edit** you will be presented with the following dialog box:



Select the filter you wish to edit and click on the "OK" button. The contents of the filter will be shown in a dialog box like this:



The left hand side of the dialog box is where each ASCII code (0 through 255) is shown. Use the vertical scroll bar to move to codes not displayed. Next to each code is displayed what White Knight will do to each character. To change this, click on the character line you wish to modify so that it is displayed highlighted (white text on a black background). Then click on one of the buttons on the right hand side of this dialog box (we'll explain what these buttons mean in a moment).

If you want to switch between displaying the ASCII codes to the actual character that each code represents, click on the "Characters" button at the bottom of this dialog box. To display the code values, click on the "ASCII Codes" button.

To let a particular code go through the filter unchanged, highlight the code and then click on the "Pass Through" button. "Pass Through" will then be shown for the highlighted code value.

To tell White Knight to strip that character out so that it is not saved or displayed (depending on the filter type), click on the "Strip Out" button.

To have the ASCII code be displayed rather than the character itself, click on the "Enumerate" button. The code will then be shown between a less than

character and a greater than character like this: "<127>".

To change the value of a received character to some other ASCII code, first type in the desired target code next to the "Translate To->" button, and then click on the "Translate To->" button.

The "Reset Filter" button changes all ASCII codes in the Filter to "Pass Through".

Let's look at a good application for Filters. Let's say you've used the **File->File Capture->New** menu choice to capture incoming data to a disk file (this process is explained in the chapter "Receiving Text Files (File Captures)". You then open that file using your favorite word processor and it looks like this:

```
Now is the time for all good men to come □ to the aid of their
□ country. Now is the time for all good men to come to the aid
□ of their country. Now is the time □□□ for all good men to
□come to the aid...
```

Obviously, those rectangular characters are garbage. To solve this. Go into White Knight, edit the File Capture Filter, and change all ASCII values from 0 to 31 (except for 9 and 13 - tab and carriage return) to "Enumerate". Then, choose **Customize->Options->TEXT Transfer** and checkmark the "Use Filter" option. Capture the file again and then open it with the word processor. It will look something like this:

```
Now is the time for all good men to come <7> to the aid of their
<10> country. Now is the time for all good men to come to the aid
<10> of their country. Now is the time <4><4><4> for all good men to
<10>come to the aid...
```

We can see by the values enclosed in the brackets that the control characters with ASCII codes of 4, 7, and 10 are the offenders. All that's left to do is to edit the File Capture Filter, click on the "Reset Filter" button to set everything back to "Pass Through", and then click on "Strip Out" for codes 4, 7, and 10 so that these are no longer saved.

Filters can be saved to disk by selecting **Customize->Filter->Save**, selecting which Filter to save, and then giving the file a name.

A filter file has an icon that looks like this:



It's a blender. Get it? Some sense of humor, this guy.

To load a Filter File that's been previously saved, select **Customize->Filter->Load**, tell White Knight which Filter to load the file into, and then select the Filter File. Filters are interchangeable, meaning you can save a Filter File from the Terminal Filter and load it back into the File Capture Filter.

There are two sample Filter Files supplied on your White Knight master disk. "Show Control Characters" will enumerate all control characters except backspace, tab, and carriage return (ASCII codes 8, 9, and 13). "Zap Control Characters" will strip out all control characters except for those three.

To Use The Terminal Filter

Select **Customize->Options->Emulation**. Checkmark either the "Pre-emulation" to filter all characters, or "Post-emulation" to filter only those characters not used by the emulation. The Terminal Filter is also discussed in the chapter "Terminal Emulation".

To Use The File Capture Filter

Select **Customize->Options->TEXT Transfer**. Checkmark the "Use filter" item. The File Capture Filter is also discussed in the chapter "Receiving Text Files (File Captures)".

To Use The File Transfer Filter

Select **Customize->Options->File Transfer**. Checkmark the "Use filter" item. The File Transfer Filter is also discussed in the chapter "Sending And Receiving Files Using Error Correcting Protocols".

Macro Keys

Don't attempt to utilize the information in this chapter until you've had several online sessions under your belt, and are feeling pretty comfortable with the basics of White Knight. If you are a beginner, feel free to skip ahead at this point to the next chapter, but don't forget to come back later and dig into this. This chapter illustrates one of the most powerful and useful features of White Knight.

What Is A Macro Key?

A Macro Key is a user defined gizmo that can perform one of three different functions:

- 1) Send a string of up to 240 characters as if you were typing them.
- 2) Load a different Macro Keys file.
- 3) Execute a compiled Procedure file.

For instance, after using a particular service for awhile, you might find that a lot of what you type in each session is redundant. You might set up some Macro Keys to automatically type in your account number, or to send commands to navigate to various places in the host system. By using the capabilities of Macro Keys, White Knight allows you to build a custom interface to a host system.

By using Macro Keys to execute compiled Procedure files, you can set up a "point and click" interface that allows a very complicated set of actions to be undertaken by even a rank beginner.

Macro Keys are saved in two different kinds of files, Settings Files and Macro Key Files. The advantage of using Macro Key Files is that only the Macro Keys, and no other settings, are changed. To save your Macro Keys, select **Customize->Macro Keys->Save**. You'll be prompted to supply a name for the file and to define where it is to be saved. To select a Macro Key File to be loaded, choose **Customize->Macro Keys->Load**.

A Macro Key File has an icon that looks like this:



Macro Keys

There are 30 different Macro Keys available at any given time. But since a given Macro Key can load in a whole new set of Macro Keys from a disk file, there is really no limit to what you can do with them.

OK, let's start talking about the specifics of Macro Keys. You can execute a Macro Key in one of three ways:

- 1) By mouse clicking on the desired Macro Key button in the Macros Status Bar.
- 2) With a keystroke.
- 3) By mouse clicking on the desired Macro Key entry in the Macros Window.

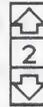
Let's look at all three of these one at a time.

The Macros Status Bar is displayed by selecting **Local->Status Bar->Macro Keys**. It is composed of three control areas, from left to right:

- The Macro Key Buttons:



- The Set Selection And Display Control:



- The Elapsed Time And Billing Clocks:

01:42:49
\$8.49

The Elapsed Time And Billing Clocks operate exactly like their counterparts in the General Status Bar, and are described in Part 3 of the chapter "The Terminal Window".

Because of space constraints, only ten Macro Keys can be displayed in the Macros Status Bar at any given time. Therefore, the 30 Macro Keys have been divided into 3 Sets of ten Macro Keys each, which we refer to as Set #1, Set #2, and Set #3. Each of the three Sets contain Macro Keys numbered 0 to 9.

A Set containing a Macro Key must be **active** before that Macro Key can be executed from the Macros Status Bar. You know that a Macro Key Set is active when its Set number is shown in the middle of the Set Selection And Display Control - in the above illustration, Set #2 is active. To make a different set active, click in the arrows in the Set Selection And Display Control. The new active set will pop into view and its Set number will be displayed in the middle of the Set Selection And Display Control.

In the Macros Status Bar, the top row of Macro Key Buttons contains buttons #0, #1, #2, #3, and #4 of the active Set. The bottom row displays Macro Key Buttons #5, #6, #7, #8, and #9 of the active Set.

To execute a Macro Key, you hold down the ⌘ key and type a numeral from 0 to 9 corresponding to the Macro Key you wish to send (i.e. type ⌘ -1 to execute Macro Key #1 in the currently active Set), or click in the corresponding Macro Key Button in the Macros Status Bar.



You can activate a Macro Key with the keyboard when any Status Bar, not just the Macros Status Bar, is selected.

Each Macro Key consists of two different strings of characters, the **macro key label** and the **macro key string**. The label is used to remind you what the Macro Key does. Macro Keys become visible in the Macros Status Bar when they have a label. If the label is empty, the Macro Key Button is not displayed in the Macros Status Bar and Macros Window and is not executable.

Before we go any farther, let's talk about how to set up a Macro Key. If you select **Customize->Macro Keys->Edit**, the following dialog box is displayed:

Select Macro Key To Edit

Set Number: **Macro Key Number:**

1 0 1 2 3 4

2 5 6 7 8 9

3

Window title:

Macro Keys

Remember window size and position

We'll refer to this dialog box as the "Macro Key Selection Dialog". At the top of this dialog box, there are two groups of radio buttons labeled "Set Number" and "Macro Key Number". Click your mouse in the desired set and number that you wish to edit, and then click on the "Edit" button. Now, a second dialog box like the one below will be displayed:

Edit Macro Key

Set #1 Macro Key #0

String:

Label: Invisible in Status Bar

Display as: Button Invisible in Macro Key Window

Icon Don't respond to mouse clicks



Double-size icon

PICT

We'll refer to this dialog box as the "Macro Key Editing Dialog". You'll notice that the set and macro key number that is being edited is displayed at the top. Just below the two buttons are two editable text items. A Macro Key is made up of two parts, the "label" and the "string". The label portion is what is displayed inside the Macro Key button in the Macros Status Bar. The string portion is what tells White Knight what to do when that Macro Key is activated.

The label can be up to 13 characters long. You can type anything you want in this box to remind you what that key does. However, if you don't specify a label, the Macro Key button will not be displayed in the Macros Status Bar or Macros Window and will not be executable.

Let's say you want the Macro Key to type a string of characters. Simply type the

characters into the "String:" box.

A carriage return character is not automatically sent at the end of a Macro Key string. However, it just so happens that there is a way to embed control characters in a Macro Key string, and there is a certain control character that is the same as what your RETURN key sends. This control character is CTRL-M. You specify control characters by preceding it with a caret character, like ^M for CTRL-M. To send a caret character, type in two carets side by side (^ ^).

Here's an unlikely (but illustrative) example:

```
HELLO^ETHERE^BOB^M
```

This example would send the characters "HELLO", followed by a CTRL-E, followed by the characters "THERE" followed by a single caret character, followed by the characters "BOB", followed by a carriage return (which is the same a CTRL-M).

If the above makes perfect sense, you're in good shape. Otherwise, you need to go back and review the last few paragraphs until it is clearer.

For the most part, the only control character you'll probably need to ever use in a Macro Key is CTRL-M for carriage returns. Just remember that:

```
HELLO THERE^M
```

would send a carriage return at the end of "HELLO THERE", and:

```
HELLO THERE
```

would not. For those of you who have found the need to send linefeeds after carriage returns, your "magic" sequence becomes:

```
HELLO THERE^M^J
```

Macro Keys are useful for holding passwords, commonly used commands, and even quick and dirty modem dialing commands. If you have a Hayes compatible modem you could set up a Macro Key that contains:

```
ATDT 555-1212^M
```

When you sent that Macro Key, it would issue the appropriate dialing command to dial touch-tone the number 555-1212. Please change that number before experimenting or you'll have a Directory Assistance operator who thinks they have a "breather" on the line while you're busy celebrating success.

Here are some other goodies I've thrown in:

<u>Sequence</u>	<u>Function</u>
^\$	Sends a DEL character (ASCII code 127)
^!	Sends a short modem break signal
^@	Sends a long modem break signal
^#	Inverts the DTR serial port line for 1 second
^[Sends an ESCAPE character (ASCII code 27)

Remember that each Macro Key String can only hold 240 characters, including any caret characters. If you try to enter more than 240, the surplus will be "lopped off" into oblivion. So there.

Later on you'll learn about a programming language contained in White Knight which can be used to write something called a Procedure. Procedure files are used to completely automate a portion of, or even an entire online session. If the first character of a Macro Key String is a backslash ("\ - which is not to be confused with the slash ("/) character) White Knight will use the rest of the characters in the Macro Key String to be the file name of a Procedure file you wish to execute immediately. No characters are sent through the serial port by the Macro Key itself when used in this manner. This is a much faster way of executing frequently used Procedures than going through the **Initiate Procedure...** menu command and then navigating through disks and folders to find the Procedure File you wish to run.

In other words, if the Procedure File named CALL COMPUSERVE is located on the disk volume named PROCEDURES, you would need to set up the Macro Key String to read:

```
\PROCEDURES:CALL COMPUSERVE
```

The colon character is what separates the volume and file names.

One of the most common problems in using Macro Keys to execute Procedure Files is getting the file name of the Procedure File correct. Bad spelling aside, the Macintosh allows you to have leading and trailing spaces in front of disk, folder, and file names, and these spaces must be present in the Macro Key String or else you'll get a "Procedure File Not Found" error. In other words, unless you have X-ray vision, it can be real difficult to get such file names correct. For this reason, instead of typing in a file name, use the "Get Procedure File" button provided in the dialog box. When clicked, you will be presented

with a standard file selection dialog box and prompted to select a Procedure File. Once a Procedure File is selected, White Knight will construct a valid path name to the file and insert it in the "String" box (complete with the leading backslash character).

If a Macro Key begins with two slash characters ("\\"), White Knight will take the filename that follows to be the name of a Macro Keys file that you wish to load. Again, a better way than typing in the filename is to click on the "Get Macro Keys File" button and select the file from the dialog that's presented.

When you're finished editing a Macro Key, click on the "OK" button and you'll be returned back to the Macro Key Selection Dialog. You'll see that the selection buttons will automatically advance forward to the next Macro Key. If you don't want to do any further editing, click on the "Done" button.

If you want to erase the contents of all of the Macro Keys, click on the "Erase All" button.

Setting The Keyboard Equivalent

You'll see in the Macro Key Selection Dialog that just under the Macro Key number selection buttons is a button labeled "Set Keyboard Equivalent". Once you have the Set and Macro Key number buttons set up properly, you can click on this button to give that Macro Key a keyboard identity. After clicking on this button, a dialog box will appear telling you to type the keyboard sequence that you wish to activate that Macro Key. You can use any key on your keyboard, and you can also use your Shift, Control, Option, and ⌘ keys in any combination you wish. Those of you with keyboards that have function keys at the top will appreciate being able to use them for a change. The Keyboard Equivalent will activate any Macro Key at any time, not just when the Set containing that key is active.

The Macros Window

When I said earlier that White Knight can be set up to have a "point and click" interface to a host system, I really meant it. There is one more way that Macro Keys can be displayed and selected that's a lot of fun to work with. It's called the Macros Window. To show the Macros Window, select **Customize->Macro Keys->Window**. Unlike the Macros Status Bar, the Macros Window allows you to display all 30 Macro Keys simultaneously, if you so desire. But the really neat part is the number of ways you can display Macro Keys in the Macros Window. You can display any given Macro Key as a pushbutton, an icon, a

double-size icon, a color icon, a double-size color icon, or a picture (whew!).

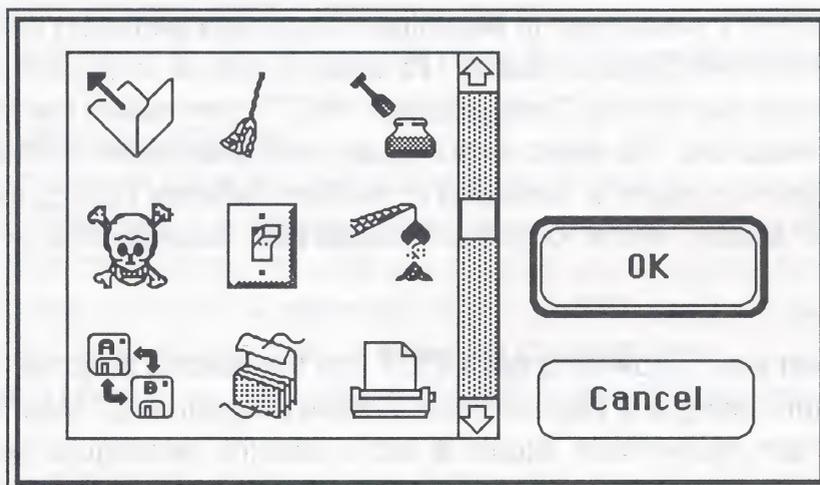
Slip your hand back to the page showing the Macro Key Editing Dialog. Next to the "Display As:" item, you'll see four radio buttons, labeled "Pushbutton", "Icon", "Double-size icon" and "PICT". To the left of these is an example of how the item will appear in the Macros Window. If it's a picture, it will be scaled down to fit in this area (with admittedly limited degrees of legibility).

Let's say we want to show a particular Macro Key as an icon. Click your mouse in the radio button marked "Icon". A file selection dialog box will appear asking you which file to look inside of for icons. Almost all Macintosh applications contain some icons, so they're a good place to start hunting for one that suits your purposes. Alternatively, there are several public domain and shareware icon editors that you can use to create your own icons, and many bulletin board services offer files full of nothing but icons. You can even "steal" icons from HyperCard or one of its stacks.



Never, ever, select the "WK's 11.0 Stuff" file to choose icons or pictures from. However, feel free to open any other Settings or Macro Keys File.

Once you've chosen a file, if the file has any icons in it, they will be displayed in what I call an "Item Selector" dialog that looks like this:



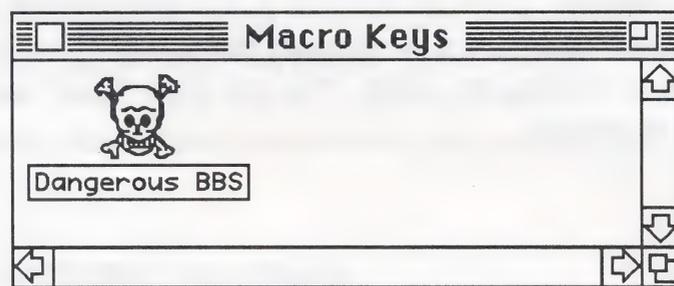
Icons are shown 9 at a time (4 at a time for double-size icons), and you can use the vertical scroll bar to scroll through those not currently displayed. When you

find an icon you like, click on it so that it is shown highlighted, and then click on the "OK" button.



A Macro Key which does not have a label will not be displayed in the Macros Window, even if you have selected an icon or picture for it. Therefore, fill in the "Label:" box in the Macro Key Editing Dialog before choosing an icon or picture for that Macro Key.

The item would be displayed as in the window illustration below:



The Macro Key's label is shown centered below the icon. To activate this Macro Key, you would just click on the icon. White Knight allows you to arrange the items in the Macros Window in any manner you wish. To move an item, just hold down the Option key, click on the item, and drag it to the desired location.

Pictures (called "PICT resources" in Macintosh Codehead parlance) are shown one at a time in the Item Selector dialog. To select a picture, click on the radio button in the Macro Key Editing Dialog labeled "PICT", then select the file which contains PICT resources. To select a PICT, just use the scroll bar so that a portion of the desired picture is displayed in the Item Selector Dialog, and then click on the "OK" button. White Knight can handle both black & white as well as color pictures.

Here's the easiest way I know of to get a PICT into the Macros Window. First, create your picture using any popular paint or draw program (e.g. MacPaint, MacDraw, FullPaint, SuperPaint, Studio 8, etc.). Use the rectangular selection tool to select the desired part of the picture, and then copy the selection to the Clipboard. Next, open the Scrapbook desk accessory and paste the contents of the Clipboard to it. Now, the Scrapbook file can be opened directly by the Item Selector dialog and the picture can be selected. Imagine placing a call to your

boss just be clicking on a digitized picture of his "mug"!

You may wish to keep certain Macro Keys hidden in the Macros Status Bar and/or Macros Window. The two checkbox items in the Macro Keys Editing Dialog labeled "Invisible In Status Bar" and "Invisible In Macro Key Window" will allow you to do this. There are also Procedure commands that will hide and show individual items that will be covered in a later chapter.

The item "Don't respond to mouse clicks" is used to set up a "background picture" in the Macros Window. In other words, you might wish to place a logo, or some other information in the Macros Window that you don't want to be a selectable Macro Key. By checkmarking this choice, you can accomplish that. The item can still be positioned (by holding down the Option key and dragging the item), but not executed. You will still need to fill in the "Label:" portion of the Macro Key so that it will be visible, but you can leave the "String:" portion blank since this Macro Key doesn't actually do anything. A couple of hints here. First, this feature only works in the Macros Window, so you'll probably want to use it in conjunction with the "Invisible in Status Bar" choice. Secondly, White Knight draws items in the Macros Window in the order of lowest Macro Key to highest, so if you choose to use a "background picture", install it into Set #1, Key #0 so that all other Macro Key items will be drawn on top of it.

As I mentioned earlier, you can hold down the Option key and click on an item to drag it to a new location in the Macros Window. White Knight also has two built-in automatic arrangement routines that can be activated by clicking in the buttons labeled "Arrange Items Horizontally" and "Arrange Items Vertically". Experiment with these to see if they give you a more pleasing layout.

About The Macros Window

There are a couple of more features that have to do with the Macros Window itself. If you checkmark the "Remember window size and position" item in the Macro Key Selection Dialog, the size and position of the Macros Window will be recorded in a Macro Keys File when you save it to disk, and it will be restored when that file is loaded from disk by selecting **Customize->**

Macro Keys->Load or by executing a Procedure MACRO command.

You can type in what you'd like to have displayed in the Title Bar of the Macros Window under the "Window title:" item in the Macro Keys Selection Dialog. Here's a trick. If you leave the "Window title:" item blank, the Macros Window will be drawn without a Title Bar. You can still drag the window to a new

location by holding down the Shift key, clicking in the window, and dragging. Since there's no Close Box, the window will have to be closed either with a Procedure WINDOW OFF command or by choosing **Customize->Macro Keys->Window**, but it might be useful if screen space is at a premium, or if the drawbacks of having a Title Bar outweigh the benefits.

Using A Printer With White Knight

White Knight supports a printer connected to your Macintosh (either directly or over a network) in a myriad of ways. The ways that we've already discussed in previous chapters are:

- Selecting text in the Display Area with your mouse and then choosing **Edit->Print Selected Text**.
- Clicking on the "Send Display Area To Printer" button in the General Status Bar.

In this chapter, I'll first show you two new ways to get information to the printer, and then I'll go into a discussion of how you should set up White Knight to work with your printer. You shouldn't try to do any actual printing until you've read that second section!

Printing TEXT Files

To print out a TEXT file stored on your disk, select **Local->Print TEXT File**, and then select the file to be printed. Because the printer will be slower than White Knight, you might want to save information to a text file (using the Archive functions or **File->File Capture** as discussed in the chapter "Receiving Text Files (File Captures)") and then print it out once you're offline.

Printing Data As It Arrives

White Knight can simultaneously send to the printer everything that it sends to the Data Area of the Terminal Window. By doing this, you can get a hardcopy record of an entire session. To enable this, select **Local->Printer Echo**. Everything that White Knight receives will also be sent to the printer. Select this choice again to uncheckmark it and toggle it off. You might wish to toggle this choice on and off multiple times during a session to print out just what you want to keep instead of everything that is received.

Part 2: Setting Up White Knight To Work With Your Printer

Because there are different kinds of printers that you can hook up to your Macintosh, and certainly many more to come in the future, White Knight needs to decide the best way to send your data to the printer. We refer to the two printing methods that White Knight supports as "low level" and "high level" printing. Each has its own advantages and disadvantages, so let's look at which you should use for your particular printer.

By selecting **Customize->Options->Printer**, the following dialog box will appear:

The dialog box contains the following elements:

- A checkbox labeled "Print time/date heading on:" followed by two radio button options: "first page only." and "each page." (the latter is selected).
- A section labeled "Printing style:" with two radio button options: "Low level (direct connect ImageWriter or equivalent)" and "High level (AppleTalk printers)" (the latter is selected).
- Two buttons: "Low level options..." and "High level options..."
- An "OK" button.

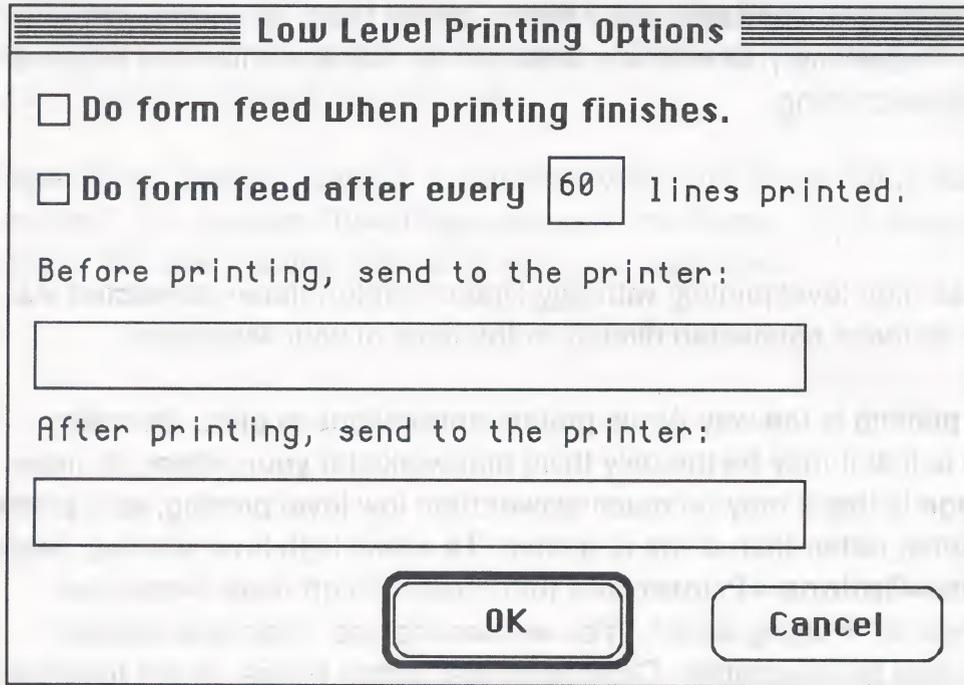
If you checkmark the "Print time/date heading on:" choice, White Knight will start each page with a line stating the current time, date, and page number. You can also specify that this heading is to be printed either on the "first page only" or on "each page". This feature works the same in both low level and high level printing.

Low Level Printing

Low level printing is never used with printers connected to your Macintosh via a network (such as AppleTalk). This includes LaserWriters and AppleTalk ImageWriters. However, if you have a printer connected directly to the back of your Macintosh, you might want to use low level printing.

The main advantage of low level printing is that it prints each line as it is received, rather than page by page. This means that the printer will do a much better job of keeping up with received data during a Printer Echo. The main disadvantage is that it will not work with many printers (even some non-Apple directly connected printers and possibly future versions of ImageWriter system software). Don't worry, it will be immediately obvious if low level won't work. You'll probably get a system crash. Therefore, don't experiment with low level printing when you have anything at stake should the system choke and puke.

To select low level printing, click in the item next to "Printing style" labeled "Low level (direct connect ImageWriter or equivalent)". You'll see that the "Low Level Options" button at the bottom of this dialog box will then become selectable. Clicking on that button brings up the following dialog box:



The dialog box is titled "Low Level Printing Options". It contains two checkboxes: "Do form feed when printing finishes." and "Do form feed after every 60 lines printed." Below these are two text input fields labeled "Before printing, send to the printer:" and "After printing, send to the printer:". At the bottom are "OK" and "Cancel" buttons.

Low Level Printing Options

Do form feed when printing finishes.

Do form feed after every lines printed.

Before printing, send to the printer:

After printing, send to the printer:

OK **Cancel**

If you checkmark the "Do form feed when printing finishes", White Knight will instruct your printer to move to the top of the next page when it is finished printing. Since this can waste paper, you may or may not want this.

If you are using cut sheet paper rather than fanfold, or if you want to prevent printing on the perforation of fanfold paper, you'll want to checkmark the "Do form feed after every X lines printed" option. Although 60 lines is usually right for this choice, you might want to try other values to best suit the kind of paper you are using. Also, if for some reason White Knight will only print out the first page and not anything further, try checkmarking this option.

The "Before printing, send to the printer:" and "After printing, send to the printer:" items are provided to suit those with special needs. Because White Knight always prints in draft mode using the printer's draft mode font, you might use these options to tell the printer to use condensed print (if you are receiving 132 characters per line, for instance), or to print everything in boldface (if your ribbon is getting weak). The special commands to do these kinds of things are listed in your printer's User Manual. You can specify control characters in the strings

sent by these options just as you do in a Macro Key string, by preceding the character with a caret character. To send an ESC (also known as Escape) character, for example, you would type `^[` (caret-left square bracket).

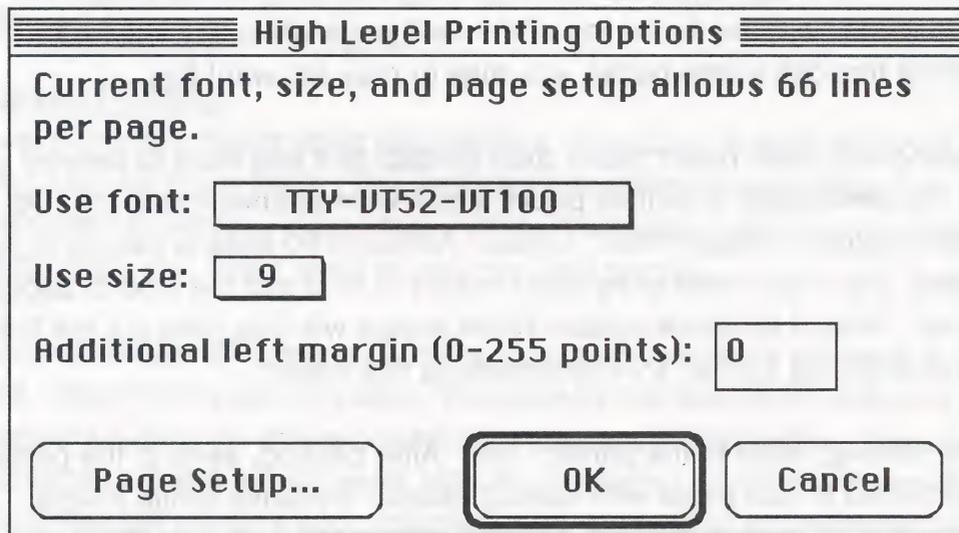
High Level Printing

High level printing is used with any network printer (such as a LaserWriter or AppleTalk ImageWriter), or with any other printer that won't function properly using low level printing.



You can use high level printing with any kind of printer, those connected via AppleTalk, or those connected directly to the back of your Macintosh.

High level printing is the way Apple prefers applications to print. Its main advantage is that it may be the only thing that works for your printer, its main disadvantage is that it may be much slower than low level printing, as it prints a page at a time, rather than a line at a time. To select high level printing, select **Customize->Options->Printer** and then select "High level (AppleTalk printers)" next to "Printing style:". You will see that the "High level options" button will then be selectable. Clicking on that button brings up the following dialog box:



High Level Printing Options

Current font, size, and page setup allows 66 lines per page.

Use font:

Use size:

Additional left margin (0-255 points):

Low level printing uses the font and point size "built into" the printer. High level, on the other hand, allows you to use any font and any point size for printing, even a different font and size than you are using in the Data Area of the

Terminal Window. The top line in this dialog box tells you how many lines can be printed on a page with the currently selected font and point size. The font and size are selected from the pop-up menus that appear when you click inside the boxes next to the items "Use font:" and "Use size:"

The "Additional left margin" item is used to tell White Knight to start printing somewhere right of the left margin of the paper. This might be useful for leaving some extra room for binding the printouts.

The "Page Setup" button brings up a standard Macintosh Page Setup dialog box, just like if you choose **File->Page Setup** in the Finder. This dialog box is discussed in the user manual that came with your Macintosh.

File Transfers

File transfers are used to accomplish several goals:

- Dumping a text file from your disk over the modem
- Capturing what comes over the modem to a text file on your disk.
- Transferring a file (text or otherwise) that resides on a remote machine to your machine with error checking and correction to insure that it arrives safely.
- Transferring a file (text or otherwise) that resides on your disk to a remote machine with error checking and correction to insure that it arrives safely.

File transfers are usually where beginners get pretty confused and frightened, because they are supported with what seems like a bewildering array of options. Actually, transferring files is not difficult at all in the vast majority of cases, and after a few test runs, you'll be zipping things back and forth like a pro. The options are there only for special circumstances, such as with dealing with a rather "stupid" communications program on the other end, or for transferring files between unlike machines.

In communications jargon, "uploading" a file means sending it from your machine to a remote. "Downloading" means receiving a file on your machine from a remote. Therefore, if I send a file to you, I am uploading and you are downloading.

We'll begin by looking at the first two objectives listed above, which we refer to as "text file transfers". A text file is defined as being composed only of alphanumeric characters. A text file cannot contain non-alphanumeric characters except for certain control characters, like tabs and carriage returns (we'll discuss this in more detail later). What it boils down to is that if you choose **Send Text File...** and the file you want to send is not displayed in the file selection dialog box, it isn't a text file and you'll have to use one of the file transfer protocols to send it (and hope the other side can do something with the file once they've received it).

Sending Text Files

The menu choice **File->Send Text File** is used to send a text file from your disk over the modem. This is a "raw" dump of the file to the modem which means that the file is sent through the serial port character by character without any error checking or correction. Therefore, telephone line interference can actually transform characters into garbage, delete characters, or add garbage characters. There's no way of knowing whether or not the file arrived in the same condition it was sent (other than by the observation of the receiver), so critical data should not be sent by this method (use a file transfer protocol as discussed in the chapter "Sending And Receiving Files Using Error Correcting Protocols") when there is a chance of such interference.

You should also know that this type of transfer does not insure that the receiver actually gets the file on their disk. The receiver must instruct their communications software to capture the incoming data to a text file before you send it, and must close the file immediately after the last characters are sent to avoid capturing any additional (and possibly undesirable) characters in the received file.

When **File->Send Text File** is chosen, a standard file selection dialog box will appear on your screen. Only files of type TEXT will be shown, to prevent you from sending anything other than a text file. If the file you desire to send does not show up in this dialog box, then you haven't saved the file as "text only" from your word processor.



You can't just "Save" a file in your word processor and use "Send Text File" in White Knight to send it. Most word processors save files in a non-textual format, containing such information as font, point size, style, justification, rulers, tab stops, and pictures. In order to save a file as "text only", a special saving sequence must be followed in your word processor. This is different for all word processors, and as a rule, I don't keep this kind of information on hand. If it isn't spelled out in the word processor's manual, contact the manufacturer directly - not FreeSoft Technical Support. Refer to the section "Helpful Hints For Sending Text Files" later in this chapter.

Once you've chosen a file, it will be opened and immediately sent, character

by character. During the send, the **Send Text File...** choice becomes **Cancel File Send**, which allows you to cancel a send in progress. Be aware that your keyboard is not locked out during a file send. Any characters you type when the Terminal Window is frontmost will also be sent.

During a text file send, you may or may not get any feedback as to the progress of the transfer, depending on the Duplex serial port setting. If you are using HALF duplex, the characters will be shown in the Data Area of the Terminal Window as they are sent. If you are using FULL duplex, the characters will not be displayed unless the remote machine echoes the received characters back.

Text File Send Options

A single dialog box contains all of the options for both sending and receiving text files. Choosing **Customize->Options->Text Transfer** brings up the following dialog box:

TEXT File Transfer Options

Wait after each line sent for : character

line sent for 1 (1-9) seconds

Delay after each: character sent for 10 (1-60) 60ths

Received TEXT file creator: MACA

Wordwrap line ends in sent files at 79 columns

Use filter

Paragraph format

OK Cancel

There are four options in this dialog box that control how a text file is sent:

- **Wait after each line sent for "X" character.** If this option is checkmarked, after each line is sent, White Knight will patiently wait for the remote machine to send the character "X" (which you fill in the dialog box with your choice) before sending the next line of text from the file. The prompting character must be alphanumeric - no control characters are allowed for this.

Here's an example of when you might use this option. Let's say that you've typed an electronic mail message into your word processor and saved it as a text file. Now you use White Knight to connect and log into your favorite electronic mail service, and wish to upload the message. This saves connect time (and perhaps long distance charges) because White Knight can send the file much faster than you can type it while online. For the sake of our example, let's say that the electronic mail service's editor works like this (it happens to be CompuServe's format, by the way):

```
1> Dear Bob:
2> This is a short test message.
3> _
```

In this example, the editor prompts me for each line by typing the current line number followed by a greater-than symbol (>). Knowing this, I could get into the service's editor, wait for it to prompt me for the first line of entry (it would send a "1>") and then start sending my file. Of course, I would be clever enough to first tell White Knight to wait after each line sent for a > character.

• **Line-by-line or character-by-character pacing.** Sometimes when we are forced to connect to archaic remote machines, we find that they are unable to keep up with the speed which we send them information. The typical result is lost characters. White Knight is equipped to automatically use a handshaking method called **XON/XOFF** which you'll find most up-to-date equipment will deal with in a nice way with no loss of information. If you do discover lost characters, however, don't despair. White Knight has two other methods for slowing things down for the remote, with the "Delay after each" option in the above dialog box. When this option is checkmarked, White Knight will introduce delays at certain points during the transfer, depending on how the rest of this option is filled out.

You can select "Delay after each line sent for X seconds", where X is a value from 1 to 9 seconds, and White Knight will wait that number of seconds after each line is sent before sending the next line of text from the file. Or, you can select "Delay after each character sent for X 60ths", where X is a value from 1 to 60, and White Knight will wait that number of 60ths of a second after each character is sent before sending the next character in the file.

How do you decide which method (line-by-line or character-by-character) of pacing to use? If in the received file (on the other machine) you discover that characters are lost only at the beginning of lines, you should pace on a line-by-

line basis. If characters are dropping out at various places in the lines, you should pace character-by-character. To anticipate your next question, I really can't be much help on what values to use for the delays. If a system is quirky enough to require using this option, only experimentation (and perhaps even a reduction in baud rate if possible) will yield the proper values to use.

• **Word-wrapping.** Up until now, we really haven't discussed too much what a text file is, other than that it's just a file containing text. The way that each line is terminated, however, is very important because some systems must have their cake and eat it too, or they'll just refuse to work with you. Most systems you will encounter will not allow an unlimited line length. Send them a line of text longer than they can handle (usually 80 or 132 characters, but we've seen all sorts of bizarre limits), and they'll choke and puke and make life miserable for you. Therefore, we sometimes will need to make sure that our lines of text do not exceed that limit. In the above dialog box, you'll see the option "Wordwrap line ends in sent files at X columns", where X is a value up to 132 (White Knight's maximum line length). If this choice is checkmarked, White Knight will load each line of text from the file into memory before sending it. If the line is longer than the limit, White Knight will send that line up to the end of the word before that limit (in order not to chop any words in half), a carriage return, and then the rest of the line. The result is a nice paragraph reformatted to the desired column number without any words broken in the middle.

Helpful Hints For Sending Text Files

The difference between a text editor and a word processor is that a text editor works with nothing but characters, but a word processor is capable of including graphics, superscripts, subscripts, and character attributes such as fonts, character point sizes, styles, etc. As you might guess, this type of information is not what we would call "plain text". In order to use the **Send Text File** choice, we must have a file on our disk in plain text format, but there are some things you need to be aware of to accomplish this.

1) Choosing **Save...** from your word processor's **File** menu will probably save the file in a binary (non-text) format which only that word processor can recognize. If you will be transferring a file to a Macintosh equipped with the same word processor (or one that is able to read your word processor's documents), you should use a file transfer protocol (discussed in the chapter "Sending And Receiving Files Using Error Correcting Protocols" rather than **Send Text File** to transfer the file. In order to get a document down to a "plain text" level that can be sent with the **Send Text File** command, most word

processors give you this option when you choose **Save As...** from their **File** menus.

2) The end-of-line character(s) can be very important for a successful transfer. When the Macintosh was created, a design decision was made that makes Macintosh text files not very compatible with non-Macintosh machines. Most non-Macintosh machines (including IBM-PC's, minicomputers and mainframes) we've worked with need two special characters to designate that end of a line, a carriage return character and a linefeed character. The Macintosh only needs a carriage return character, and in fact will deal rather poorly with linefeed characters, displaying them as rectangles at the beginning of each line of text. If the target machine seems to choke on a file you send it, or the receiver tells you that the lines "print over the top of each other", you'll know that the receiver needs carriage return/linefeed pairs at the end of each line. You can instruct White Knight to add linefeeds after carriage returns in sent files by selecting **Customize->Options->Key Mapping**, and in that dialog box, select the option "Return key sends carriage return and linefeed".

Receiving Text Files (File Captures)

There are two ways to capture incoming data to a text file. This is useful for when the remote machine is doing the equivalent of White Knight's **File->Send Text File** command discussed in the last chapter, or when you want to capture a part or all of the text the remote machine sends in a session.

To Capture Incoming Text To A New File

Select **File->File Capture->New**. You will be asked to name the file and tell White Knight where to save it. Once you've done this, everything White Knight receives will be saved in that file until you choose **File->File Capture->Close**, which closes the file and tells White Knight not to save anything further.

To Append Incoming Data To The End Of An Existing File

Select **File->File Capture->Append To**. You will be asked to select an existing file. Once you've done this, everything White Knight receives will be appended to the end of that file until you choose **File->File Capture->Close**, which closes the file and tells White Knight not to save anything further.

Pausing The File Capture

Sometimes during a file capture, you'll want to stop and resume capturing, without having to close and reopen the file. Typically this is because the remote system is sending information that is redundant or unneeded. To pause the file capture, select **File->File Capture->Pause** and White Knight will no longer capture incoming data until you choose this choice again (to remove the checkmark in the menu choice).

Since anything and everything received is captured to the disk file, you could be recording an entire session (or parts of a session) for later perusal and editing. Just like with sending text files, no error checking or correction is done. "Garbage" characters caused by phone line interference will be captured along with everything else, so this is not a good way to capture information of a critical nature if you need any sort of reliability.



Working With A Received Text File

How you get a text file into your word processor is different for each program. However, I have found that the following process works well for most of the word processors I have used:

- 1) Double-click on the word processor to start it up.
- 2) Once the word processor is executing, if it puts up a blank "Untitled" window, close that window.
- 3) Go to the word processor's **File** menu and choose **Open**. If the file selection dialog box has choices for different kinds of files, choose "Text" or "Plain Text".
- 4) Select the text file and the word processor should open it.

These are just guidelines, you may need to contact the manufacturer of your word processor to find out how to load a "plain text" file. Very often, just double-clicking on the text file will produce the error message "Application missing or busy", which tells you that the Finder doesn't know which word processor created the file and therefore doesn't know which to execute. I'll tell you how to solve this problem a bit later so that you can just double-click on the text file and the Finder will load up the word processor of your choice and open the file.

By choosing **Customize->Options->Text Transfer**, the following dialog box is displayed:

TEXT File Transfer Options

Wait after each line sent for : character

line sent for (1-9) seconds

Delay after each: character sent for (1-60) 60ths

Received TEXT file creator:

Wordwrap line ends in sent files at columns

Use filter

Paragraph format

There are three options in this box that concern file captures:

- **Received Text File Creator.** Every Macintosh file is given two signatures when it is created, the **file type**, and the **file creator**, both of which are four alphanumeric characters. For text files, the **file type** will always be **TEXT**. The **file creator** tells the Finder what application to start up when you open a document on the Desktop. In other words, if you double-click on a MacWrite document, the Finder looks at the document, sees that it has the same **file creator** as MacWrite, and therefore knows that it needs to start up MacWrite and pass along the name of that document to open. Since there's no way for White Knight to know what you use as your favorite word processor, it gives you the ability to define what **file creator** to assign when it creates text files.

You can type in the four characters you wish to assign any text file created by White Knight's **Capture Incoming Data To Text File...** command. Although the following list is not by far complete, it lists a few of the most commonly used text editor/word processor applications' **file creators**.

<u>Application</u>	<u>File Creator</u>
MacWrite	MACA
Microsoft Word 3.0 (and up)	MSWD
Apple MDS Edit	EDIT
WriteNow 2.0	nX^n

If you use an application other than the ones listed here, a quick call to the manufacturer's technical support people should yield the proper **file creator** to use. Upper and lowercase letters are different in file creators, so they must be typed in exactly.

- **Use Filter:** if checkmarked, this tells White Knight to use the "File Capture Filter" on received data before the data is saved. Filters are discussed in the chapter "Filters". This feature can be extremely helpful. Often, you'll find that when you open the capture file there will be "garbage" characters mixed in with your desirable data. The most common garbage character is a linefeed character (ASCII code 10), which many computer brands need and use, but Macintosh word processors abhor. You'll know immediately if linefeeds are your problem because when you load the file into your word processor, there will be rectangle character at the beginning of each line. Here's how to fix that.
 - 1) Select **Customize->Options->Text Transfer**. Checkmark the "Use filter" option and click on the "OK" button.
 - 2) Select **Customize->Filter->Edit**. In the dialog box that appears, click on

the "File capture filter" option and then click on the "OK" button.

3) The File Capture Filter will now appear. Since a linefeed is ASCII code number 10, click on the box in the scrolling list that has "010" on the leftmost part of the line, it will then be shown inverted (white text on black background).

4) Now click on the "Strip Out" button, and the click on the "OK" button.

By doing this, you are telling White Knight to not save any linefeeds in the captured files.

Paragraph Format

Most non-Macintosh computers put a carriage return (or carriage return and linefeed) at the end of each line. Most Macintosh word processors, on the other hand, use something called "paragraph format", which means that they put a carriage return only at the end of each paragraph. If the file has a carriage return at the end of each line, it can be a real pain to try to get the word processor to format and wordwrap the file to something that looks nice. Before White Knight 11, users had to go through and manually backspace out the carriage returns at the end of every line. However, by checkmarking the "Paragraph format" option, White Knight will try to intelligently strip out carriage returns in the text as it is received on the following basis:

- If two carriage returns in a row are received, only one carriage return will be saved. This will delineate paragraphs.
- If a carriage return is not preceded by a tab or at least three spaces (common paragraph beginners), it is stripped out. This is to keep the lines inside of a paragraph together so they'll wordwrap properly when loaded into your word processor.

This algorithm may not be perfect or work in all cases, but in many cases it will save you a lot of work. You'll appreciate the "Paragraph format" option the first time you try to do some editing on a file that came from a non-Macintosh system.

Sending And Receiving Files Using Error Correcting Protocols

Performing file transfers in the "raw" sense of just dumping characters to the serial port or capturing whatever comes over the serial port is at best risky. Because of telephone line interference, characters can be changed or deleted, and "garbage" characters can be introduced. For this reason, error correcting file transfer protocols were created to insure that data arrives in the same condition it was sent. A protocol is simply a collection of rules for breaking a file up into blocks of data, sending each block, and doing comparisons on the block at both ends. The sending portion of the protocol will continue to retransmit a block until both sides agree that it arrived safely. Error correcting protocols offer a high degree of reliability (over 99%) and should be used whenever the data is important, and always when the file is non-textual in content.

White Knight offers a wide choice of the most commonly used error correcting protocols: Kermit, XMODEM, YMODEM, ZMODEM, and one special purpose protocol called Flash. These protocols are not interchangeable, meaning that if one machine is sending a file using XMODEM protocol, the other side must receive the file using XMODEM, and not one of the other protocols. From your point of view, however, they will operate in a very similar fashion to each other.

Now I'm going to give you a warning up front. A lot of people get really hung up on transferring files, and I hate to see that happen. I like to compare using file transfer protocols to learning how to drive an automobile. The first time, you're going to feel apprehensive and awkward. After your first success, you're going to start thinking "Gee, that wasn't so hard." Then you'll actually start to enjoy file transfers (believe it or not), and finally, they become "old hat" and you'll hardly give it a second thought. Everybody feels like a dummy the first time they go through this, so just stick to it and you will get over the hump.

As we go through the list of file transfer protocol options that White Knight offers, it will be easy for you to become intimidated. What I want you to remember is that the default settings are done to provide the widest possible flexibility. There is a strong likelihood that you will not have to change anything in order to transfer files. The myriad of options are there for when you've become confident with transferring files, and are ready to take advantage of them. In fact, I'm willing to bet that you could skip forward right now to the section at the

end of this chapter titled "Step By Step File Transfers" and with no further instructions, start transferring files like a champ. Feeling brave?



Do not "play around" with the various file transfer protocol settings until you have read about them in this chapter. If you checkmark the "Supercharged XMODEM receive" option indiscriminately, for instance, you will likely find that all XMODEM transfers will fail.

Let's look at the various file transfer protocols that White Knight offers.

XMODEM Protocol

XMODEM and Kermit are the grandfathers of file transfer protocols, and for this reason are the lowest common denominators on virtually all systems that are able to transfer files. XMODEM is used mainly for microcomputer to microcomputer file transfers, and for this reason you'll find that nearly all microcomputer communications programs support it. You'll also find it on all Bulletin Board Systems (BBS's) and nearly all of the commercial host services that cater to microcomputer users like CompuServe, GENie, and Delphi.

White Knight offers four "flavors" of XMODEM. **Classic XMODEM** is the original version of the protocol, and it should work with any system that supports XMODEM. **XMODEM - CRC** was an improvement made to classic XMODEM's error correction algorithm to offer even higher reliability, and you'll also find it widely supported. **XMODEM - 1K Blocks** was the last major improvement made to classic XMODEM. It extended the size of the "blocks" of data sent from 128 bytes to 1024 bytes. A good number of systems support this variant, and it will be more efficient than classic XMODEM unless you have a particularly "dirty" connection, in which case more data will have to be retransmitted for each error encountered.

Kermit Protocol

Like XMODEM, Kermit is also widely used. Although there are communications programs that support Kermit for nearly all major microcomputer brands, Kermit is rarely used for microcomputer to microcomputer file transfers, because it is less efficient than XMODEM. Kermit is mostly used for mainframe to microcomputer file transfers, because it is especially good at dealing with the idiosyncrasies of this task. Therefore, if you'll be calling a mainframe as a terminal, you'll almost certainly find that it supports Kermit, which White Knight

refers to as **classic Kermit**. Classic Kermit had a rather small block size (blocks are called "packets" in Kermit parlance), so one of the first enhancements made to this protocols was **Long Packet Kermit**, which increased the size of the data packet from a maximum of 94 bytes to a maximum of 9024 bytes. An additional enhancement, called **Sliding Windows Kermit** (also known as Super Kermit), has been steadily gaining popularity among Kermit users because of its increased throughput efficiency on networks that suffer from periodic response delays. All of the Kermit variants can send batches of files, rather than XMODEM's one file at a time limit.

Another nice thing about Kermit is that you can get a Kermit package for almost all major micro, mini, and mainframe brands at bargain basement prices by contacting the Kermit Distribution Center at Columbia University in New York City, NY. Highly recommended if your host system wants to talk to small computer models.

YMODEM Protocol

The **classic YMODEM** protocol was the first major overhaul of the XMODEM protocol, and it has been widely accepted. Classic YMODEM's major contribution is the ability to send a batch of files, rather than XMODEM's one file at a time limit. As you might guess, the next enhancement to classic YMODEM was **YMODEM - 1K Blocks**, which, like XMODEM - 1K Blocks, increased the size of the data block from 128 bytes to 1024 bytes. **YMODEM - G** (Lord only knows what the "G" stands for) was the final enhancement to classic YMODEM. YMODEM - G does not do error correction. For this reason, it is extremely fast (up to 98% efficiency), but also extremely risky. If a single error occurs, the file transfer will fail, and this generally happens after you've received about 300,000 bytes of a 300,500 byte file. Therefore, it should only be used with error-correcting (MNP is an example) modems or with extremely clean connections.

ZMODEM Protocol

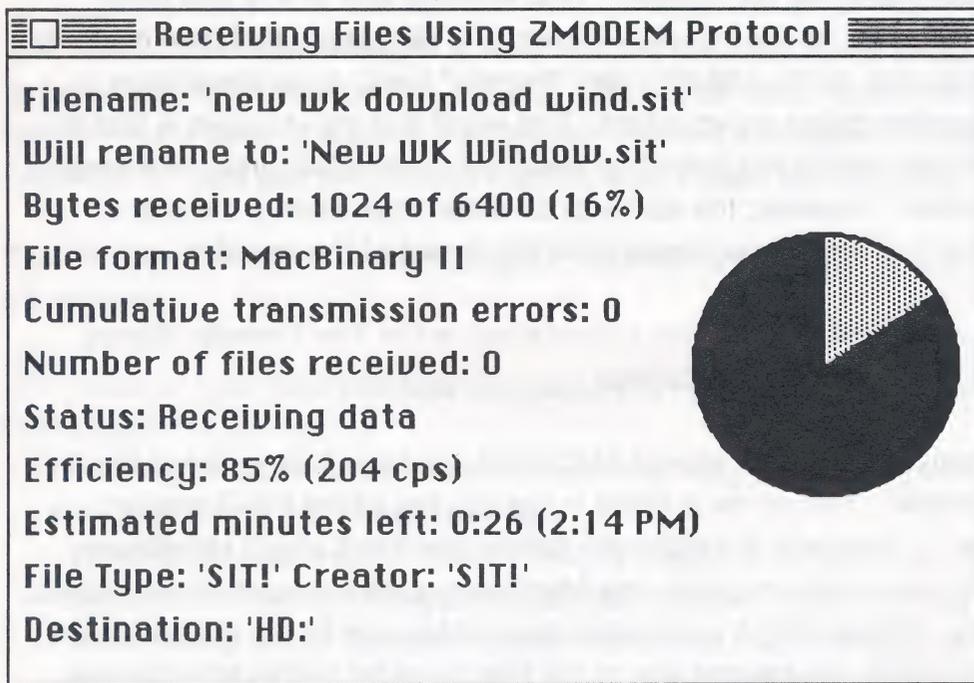
ZMODEM Protocol is the most recent major overhaul to classic XMODEM, to the degree that I can say, having seen it from the inside, there are few similarities to it and XMODEM remaining. ZMODEM supports batch file transmissions, it is nearly as fast as YMODEM-G but still offers error correction, and it can even resume an interrupted file transfer at a later date from the point of the interruption. All in all, ZMODEM is a wonderful protocol and definitely the one to choose if it is available. Many BBS's are supporting ZMODEM as well as at least one major commercial Host system (GENie) at the time of this writing.

Flash Protocol

I wrote this special purpose protocol to take advantage of the new error correcting modems that have come onto the market recently. Since the modems are doing error correcting, it would be superfluous for the file transfer protocol to also do this. Flash protocol is a "least overhead possible" type protocol that should be used only when you have two error correcting modems connected. When this situation is available, and the other side is also using White Knight 11, you'll find that Flash protocol offers the highest transmission speed possible of all of the file transfer protocols. It also supports sending batches of files.

The File Transfer Status Window

During a file transfer, The Terminal Window will become inactive and a new window, called the File Transfer Status Window, will be displayed. Here's what the this window looked like when I was receiving a file using ZMODEM protocol:



As you can see, the type of file transfer (sending or receiving) and the protocol being used is displayed in the Title Bar of this window. When you position your mouse inside of this window (except in the Title Bar) the mouse cursor will change to a "spinning beach ball". This (with tongue in cheek) is provided for the paranoid, to assure you that the protocol is still working, and your system hasn't froze up during a lengthy block transfer or error recovery. You can drag the File Transfer Status Window to any place on your screen, it will always

appear in the same position that you left it in during the last file transfer.

To cancel a file transfer in progress, click in the Close Box in the File Transfer Status Window's Title Bar. Although this usually works fine, sometimes it can be an "iffy" sort of thing. You might have to click several times in this box, and you might have to wait several seconds for the cancel to take place (it's a lot harder than you might think to cancel a transfer "cleanly"). When the transfer has been cancelled, the File Transfer Status Window will disappear.

Another thing I want to point out is that your Macintosh is not "locked up" during a file transfer. You'll notice that the  menu is available for you to work with desk accessories (the **Edit** menu is also active during file transfers to support desk accessories) during a file transfer. If you are running under MultiFinder, you can also put White Knight in the "background" (by bringing some other application's window to the top) and you can work in other applications while White Knight continues the file transfer. Yes, working with D.A.'s and other applications will cause the file transfer efficiency to degrade. Just how much it will degrade depends on how MultiFinder "friendly" the D.A. or application is. This bears experimentation on your part. The worst that can happen is that the file transfer will degrade to the point that it will completely fail, and you'll have to restart the transfer. However, the ability to do other work while a transfer is taking place might of more importance than the speed of the transfer.

Now let's look at the information that is contained in the File Transfer Status Window (refer to the illustration above).

All of the file transfer protocols except XMODEM send the file's name in the first block of the transfer. This name is listed in the top line of the File Transfer Status Window. If the file is in MacBinary format (we'll talk about MacBinary format in detail later in this chapter), the MacBinary data will contain the original name of the file, White Knight will always give preference to the original name, and if this is the case, the second line of the File Transfer Status Window will show that White Knight will rename the file to this original name at the end of the transfer.

The next line shows how many bytes have been transferred. If you are sending the file, or if the file is being received and is in MacBinary format, White Knight will know the total size of the file and can tell you a number of additional things about the transfer:

- The total number of bytes to be transferred, what percentage has been

transferred, and a graphic pie chart representation of the percentage transferred.

- The file transmission efficiency, both in percentage (100% being absolutely perfect efficiency) and in characters per second.
- The estimated time left for the transmission and the estimated time of day when the transfer will be finished.
- The file's type and creator signatures.



When a file being received is not in MacBinary format, the above items will be displayed as unknown.

The "File format:" item will tell you the format of the file, MacBinary, MacBinary II, or Not MacBinary.

The "Cumulative Transmission Errors" will tell you how many errors have been detected during the current file transfer (this will be reset to zero between each transfer in the case of a batch file transfer). If the file transfer completes successfully, you can be assured that the errors were all corrected.

The "Number of files (received or sent)" item will tell you the progress of a batch file transfer.

The "Status:" item tells you how the transfer is running. This item will show one of the following strings:

- Holding for start - which means that White Knight is waiting for the other side to give the "go ahead" to begin.
- Waiting for data - which usually means that an error occurred and White Knight is attempting to resynchronize the transfer. It can also be shown at the beginning of the transfer, which means that the other side is taking an unusual amount of time to give us the "go ahead" to begin.
- "Sending Data or Receiving Data - which means that the transferring is progressing as expected without problems.

If you are receiving a file, the bottom line of the File Transfer Status Window will show you the "Destination" of the file. White Knight, as a default, stores all received files in the root level of the disk volume that White Knight resides on. We'll show you how to change this destination a bit later in this chapter.



The "Estimated Time Left" is just that, White Knight's best estimate. This time will change rapidly at the beginning of a file transfer, and then settle down and become more accurate as the transfer progresses. The "Efficiency" is also an estimate, and should not be used as a "benchmark". In other words, treat it as an indication of relative performance, rather than gospel. Other programs may calculate efficiency in a much different manner that is not meaningful when compared to what White Knight reports. As you gain experience with the various file transfer protocols, you'll find that the efficiency depends on a number of things:

- The protocol being used - XMODEM, for instance, degrades rather extremely as the baud rate is increased. Additionally, some protocols are just plain more efficient than others. Kermit and XMODEM are at the low end of the efficiency spectrum, ZMODEM, YMODEM-G, and Flash protocol are at the high end.
- Network delays - you'll almost always find that BBS's run faster than commercial services.
- User intervention - the more you do with your Macintosh during a file transfer, the less time White Knight has to process each block of data. If you work with a desk accessory or a different application under MultiFinder, the performance will degrade. Even dragging the File Transfer Status Window can affect the efficiency!
- Disk drive performance - hard disks allow higher efficiency than floppies.
- Telephone line quality - transmission errors will adversely affect efficiency.

The MacBinary Format

We've mentioned the "MacBinary Format" a few times thus far. Now let's talk about it in detail. Macintosh disk files are very different from files on any other current microcomputer in at least one regard. Most non-Macintosh computers store all file data in one big clump. The Macintosh, however, actually divides each disk file up into two subfiles, which are called **forks**. The **data fork** is equivalent to what non-Macintoshes use to hold the contents of a file. But a Macintosh file can contain such things as icons, pictures, dialog boxes, and a whole mess of other things that don't have an equivalent on non-Macintosh machines. For this reason (and others we don't need to get into), the designers of the Macintosh decided to give each disk file a separate storage area, called the **resource fork** to hold such things.

Finally, it is desirable to know other things about a file (called **Finder information**) that should be included in a file transfer. Some of these things are pretty esoteric as far as you're concerned, but they do include some things like the original file name, the file's creation and last modification dates, which icon should be displayed by the Finder for that document, and the file type and file creator (these two things were discussed previously in this chapter).

What this boils down to is that when we send a Macintosh file, we need to include three separate entities: the data fork, the resource fork, and the Finder information. The authors of all major Macintosh communications software met and agreed up a standard for doing this. This standard is called **MacBinary format**, and guarantees that a file sent by one Macintosh will arrive in the same condition when received by any Macintosh communications program that supports this standard.

The beautiful thing about MacBinary is that it is completely invisible to a non-Macintosh, meaning that you could send a Macintosh application or document to a non-Macintosh, and then that non-Macintosh could then send the file to a second Macintosh. Since the non-Macintosh does not recognize MacBinary, it simply stores the file chronologically on its disk as it is received. However, the second Macintosh recognizes that the file is in the MacBinary format as it receives it, and does the conversions necessary to restore the file to its original condition.

Whenever White Knight receives a file that is not in MacBinary format, it gives it a file type signature of TEXT as a default.

File Transfer Options

There are two kinds of file transfer options in White Knight, those that are supported for all of the protocols, and those that are specific to a certain protocol. Let's look first at those that are supported for all protocols. By selecting **Customize->Options->File Transfer**, the following dialog box is displayed:

File Transfer Options	
Use MacBinary format for:	
<input checked="" type="radio"/> All files except type 'TEXT'	<input type="radio"/> All files
<input type="radio"/> No files	
If received MacBinary format file exists:	
<input checked="" type="radio"/> Destroy old file	<input type="radio"/> Create unique filename
At end of MacBinary format receive:	
<input checked="" type="radio"/> Rename file to original filename	<input type="radio"/> Don't rename
Received non-MacBinary format file conversion:	
<input checked="" type="radio"/> No conversion	<input type="radio"/> Paragraph format
<input type="checkbox"/> Use filter	
At completion of transfer, ring bell <input type="text" value="3"/> times.	
Received file destination:	
<input type="text" value="HD"/>	
Default creator:	<input type="text" value="MACA"/>
<input type="checkbox"/> Auto-receive	<input type="checkbox"/> Final report
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

As you can see, there are quite a number of options, but don't be put off by this quantity, this is exactly what gives White Knight its power! We'll take them one at a time so you can see how simple they really are.

Use MacBinary Format For

This option is used to tell White Knight what kinds of files to use the MacBinary format for when sending (White Knight automatically recognizes files in MacBinary format when receiving). You can specify that White Knight is to use MacBinary format for "All files except type 'TEXT'" (the usual setting), "All files" (which includes those of type 'TEXT'), or "No files". You will select "All files" when you will be sending a TEXT file to another Macintosh, and want things like the original filename, creation date, or precise file size to be transmitted. However, don't send MacBinary format files to a non-Macintosh and expect that machine to be able to do anything with the file.

The "No files" option is used when you just want to send just the data fork of the file without the MacBinary information. I know that probably sounds like goop, but let me explain when you'll want to use it. A number of Macintosh

applications, like PageMaker, Microsoft Word, and Word Perfect, have equivalent versions running on other microcomputer brands, like the IBM-PC. These programs offer the ability to save a document in the format that the other version can load in and work with. When you are transferring such a file, you will want to select the "No files" option to suppress the MacBinary format.

If Received MacBinary Format File Exists

This option tells White Knight what to do if a MacBinary format file (which you'll remember will contain the original filename of the file) is being received, and a file by that same name already exists in the received file destination folder. You can tell White Knight to either destroy the old file in favor of the the new (the default setting), or you can tell White Knight to leave the old file alone and create a unique filename for the new file (this is done by adding "#1", "#2", etc. to the end of the new file's name).



In order for ZMODEM's "resume interrupted file transfer" function to work, this option must be set to "Destroy old file".

At End Of MacBinary Format File Receive

This option tells White Knight whether or not to rename a received MacBinary format file to the name provided by the MacBinary format information, or to use the name provided by the protocol.



All of the protocols supply a filename for the file being transferred except for XMODEM. XMODEM file receives begin by creating a file called "Untitled". If the file is not in MacBinary format, or if you have this option set to "Don't rename", a dialog box will appear at the end of the transfer allowing you to optionally give the file some other name than "Untitled".

Received Non-MacBinary Format File Conversion

This option allows you to specify that a non-MacBinary file is either to be saved exactly as it is received, or that it should be converted to Paragraph Format. Paragraph Format can be very useful for getting text files into your favorite word processor with least amount of trouble, and it is described in detail at the end of the chapter "Receiving Text Files (File Captures)".

You can also tell White Knight to use the File Transfer Filter to filter the non-MacBinary file before it is saved to disk. Filters are discussed in detail in the chapter "Filters". Here's an example of when you might want to use this feature. If you are receiving text files from a non-Macintosh computer, you might find that when the file is loaded into your word processor, it has a rectangle "garbage" character at the beginning of each line. This is linefeed character (ASCII code 10). Here's how to get rid of them:

- 1) Select **Customize->Options->File Transfer**. Checkmark the "Use filter" option and click on the "OK" button.
- 2) Select **Customize->Filter->Edit**. In the dialog box that appears, click on the "File transfer filter" option and then click on the "OK" button.
- 3) The File Transfer Filter will now appear. Since a linefeed is ASCII code number 10, click on the box in the scrolling list that has "010" on the leftmost part of the line, it will then be shown inverted (white text on black background).
- 4) Click on the "Strip Out" button, and then click on the "OK" button.



Don't use the File Transfer Filter when receiving non-textual files that are not in the MacBinary format or the file will be damaged!

At Completion Of Transfer, Ring Bell X Times

This tells White Knight how many times to ring the bell to alert you that a file transfer has finished successfully. If a file transfer fails for some reason (except you cancelling it), a dialog box will appear telling you (if White Knight can figure it out) why the file transfer failed.

Received File Destination

This tells you where White Knight will save received files. The default destination is the root level of the disk volume White Knight resides on (not inside any folders). You can change this default location by selecting **File->Received File Destination**. A folder selection dialog box will appear. Use the scrolling box to navigate to the desired destination folder. Open this folder and the folder name will appear at the very top of this dialog box. Then, click on the "Select" button and White Knight will save all future received files in this folder.

Default Creator

This option tells White Knight what creator signature to give a non-MacBinary

format file (the file type signature for these files is always TEXT). This creator signature tells the Finder which word processor to start up when you double-click on the file's icon. Although the following list is not by far complete, it lists a few of the most commonly used text editor/word processor applications' file creators.

<u>Application</u>	<u>File Creator Signature</u>
MacWrite	MACA
Microsoft Word 3.0 (and later)	MSWD
Apple MDS Edit	EDIT
WriteNow 2.0	nX^n

Auto Receive

This feature is used as a time-saving device when you are connected to another Macintosh running White Knight. If both users have Auto Receive checkmarked, only the sender has to do anything to execute a file transfer. When the sender selects the menu choice to send a file, White Knight will automatically instruct the remote White Knight program to automatically start up the file transfer receive for the selected protocol. This choice should normally not be checkmarked when you are not connected to another Macintosh running White Knight. It should also not be checkmarked on a White Knight that is in Host Mode (discussed later), or when communicating with a White Knight that is in Host Mode.

Final Report

If this option is checkmarked, White Knight will print a line of information about a file transfer in the Data Area of the Terminal Window when the file transfer ends successfully. The line will look like this:

```
File:'New WK Window.sit', Time: 0:50, Size: 6400, Efficiency: 54%
```

It lists the name of the file transferred, the time it took to transfer the file, the size of the file in bytes, and the final efficiency of the file transfer. In the case of a batch file transfer, a final report line will be printed for each file transferred.

Batch File Transfers

Sometimes you will find the need to transfer a number of files, rather than just one. All of White Knight's file transfer protocols, with the exception of XMODEM, allow you to send or receive numerous files without having to reinvoke the protocol for each file. This is called a "batch file transfer".

To receive a batch of files, you don't need to do anything special. White Knight will automatically recognize when the sender is sending a batch of files, and will react appropriately.

To send a batch of files, you must first identify the files to send. This is done through what we call a "batch file". A batch file is simply a file that contains the filenames of other files. When you choose **File->Send File** and select a file, White Knight checks to see if the selected file is a batch file. If it is not, White Knight sends that file. However, if the selected file is a batch file, White Knight does not send that file, but instead opens the file, extracts the filenames contained in it one by one, and sends those files.

To create a batch file, select **File->Create Batch File**. You will first be shown a file creation dialog. Type the name of the batch file you wish to create. It is this filename that you'll select later via **File->Send File** to send the batch.

Next, White Knight will cycle continuously through a file selection dialog. You simply choose each file to send as part of the batch and then click on the "OK" button. After you've selected the last file to be sent, click on the "Cancel" button.

A batch file has an icon that looks like this:



Batch File



A special RCMD module called "FolderBatch" is included to quickly create a batch file containing all of the files in a particular folder. Instructions on how to use this feature are in the chapter "RCMD's: Extending White Knight".

XMODEM, YMODEM, And ZMODEM Options

By selecting **Customize->Options->X-Y-ZMODEM**, the following dialog box is displayed:

XMODEM, YMODEM, ZMODEM Options

XMODEM Options:

XMODEM - Classic XMODEM - CRC XMODEM - 1K Blocks

Supercharged XMODEM Receive

YMODEM Options:

YMODEM - Classic YMODEM - 1K Blocks YMODEM - G

1K Block Options:

Send only if requested Send automatically

XMODEM, YMODEM, ZMODEM block timeout: seconds.

Keep partial ZMODEM receives for later completion

All of the features in the remainder of this chapter will refer to this dialog box. As I mentioned previously, the default settings in this dialog box should work in the majority of cases, and you should make changes to them only after you have read this chapter thoroughly and are comfortable with the basics of XMODEM, YMODEM, and ZMODEM file transfers.

XMODEM Options

In this option, you may tell White Knight to use classic XMODEM (this is generally listed on host systems as "XMODEM" or "XMODEM-checksum"), XMODEM with CRC error checking, or XMODEM with 1K blocks.

YMODEM Options

In this option you may tell White Knight to use classic YMODEM (generally listed on host systems as just "YMODEM"), YMODEM with 1K blocks, or the special

YMODEM-G non-error correcting variant.

1K Block Options

This option was added because somewhere along the road, the authors of various microcomputer communications programs took either side of a fork. This should be left on "Send only if requested" unless you know that the remote system supports XMODEM - 1K blocks or YMODEM - 1K blocks, yet White Knight seems to only want to send 128 byte blocks. In that case, experiment by setting this option to "Send automatically". The worst that can happen is that the file transfer will simply fail, which would indicate that the other side does not, in fact, support the 1K block option.

XMODEM, YMODEM, ZMODEM Block Timeout

This is the amount of time (in seconds) that these protocols will wait in the case of inactivity before "timing out" and attempting to resynchronize a transfer. This should always be left at 5 seconds except in a special case. Some commercial host systems, especially CompuServe, will experience occasional network delays that are well above this 5 second limit. If this is the case, and only if you are getting all sorts of file transfer errors on what appeared to be a clean connection, try upping this value to 30 seconds.



Always set this value to 5 seconds for doing microcomputer to microcomputer file transfers.

Keep Partial ZMODEM Receives For Later Completion

The ZMODEM protocol has the ability to resume an aborted file transfer at a later date. If you want White Knight to support this feature, checkmark this option. If this option is not checkmarked, White Knight will delete a partially received file upon premature cancellation of a transfer.



In order for this feature to work, you must have in the dialog box brought up by **Customize->Options->File Transfer** the "If received MacBinary format file exists" option set to "destroy old file".

When this feature is active, when a ZMODEM file transfer is cancelled, White

Knight will create an additional file containing the information necessary for later resumption of the transfer. This file will have the characters ".ZRES" appended to the name of the file that was being received (a file by that name with a generic icon will also be present in the same folder). Don't rename or move either of these files! When you do a ZMODEM receive, White Knight will check for the existence of a ".ZRES" file for the file being received, and if it does exist, will attempt to resume the transfer from the point where it had left off. At the end of a resumed file receive, White Knight will delete the ".ZRES" file

A ".ZRES" file has an icon that looks like this:



Cancelled File.ZRES

Kermit Options

By choosing **Customize->Options->Kermit**, the following dialog box is displayed:

Kermit Options

Classic Kermit - packet size (10-94): 94

Long Packet Kermit - packet size (10-9024): 1024

Sliding Windows Kermit - window count (1-31): 31

Treat outgoing files as: Text Binary data

Packet timeout: 3 seconds.

Use line turnaround handshake character: ^Q

Start of packet character: ^A

End of packet character: ^M

OK Cancel

Like the X-Y-ZMODEM options, the default settings should work without problems in the majority of cases and should not have to be changed unless you are experiencing difficulty, or want to take advantage of one of the more advanced Kermit protocols. The only exception to this rule is the "Treat outgoing files as" option, which you'll want to read over before using Kermit the first time.

Kermit Style

The first three choices are where you tell White Knight what "flavor" of Kermit to use: Classic Kermit, Long Packet Kermit, or Sliding Windows Kermit. Any system that supports Kermit will support Classic Kermit, but you should use Long Packet Kermit or Sliding Windows Kermit only if the host system specifically mentions support of those.

Treat Outgoing Files As

Kermit has the built-in feature that if a file being sent is plain text, it will add linefeed characters after carriage return characters for those systems that need them. You should have this option set to "Text" only if you are sending or

receiving plain text files. In the case of receiving files, you might want to strip out linefeed characters via the "File Transfer Filter" described in the chapter "Filters". The filter will only be used in Kermit transfers when this option is set to "Text". When this option is set to "Binary", no linefeeds are added to outgoing files, and no filtering is done to incoming files (if filtering is selected, that is).

Packet Timeout

Since Kermit "negotiates" the abilities of the remote side in the first packets, you shouldn't have to change this value except under specific directions of your host system's administrator.

Use Line Turnaround Handshake Character

This option is used only in special cases. The only time I have ever seen it used is with certain IBM mainframe implementations of Kermit.



A special tip for those who will be using Kermit to communicate with an IBM mainframe. If you can't get Kermit to start a transfer, first try switching the serial port settings to 7 data bits and MARK parity (you were probably using 8-N or 7-E - just trust me on this). If it still won't go, try checkmarking the "Use Line Turnaround Handshake Character" option. If it still fails, leave this option checkmarked and switch back to your original serial port settings. If it still ignores you, consider taking up heavy drinking.

Start Of Packet And End Of Packet Characters

These should never be changed except under the direct instructions of the host system's administrator. Indiscriminate meddling will almost certainly cause the two Kermits to ignore each other. So there.

Step By Step File Transfers

Experience is the best teacher, so in this chapter, I'm going to take you step by step with some actual file transfer sessions I've done.

Sending Files To An Individual Microcomputer User

Person A wants to send a file to Person B using XMODEM protocol. Person B will be calling Person A's computer (it doesn't make any difference who calls who, by the way).

Step 1: Both persons will set their serial port settings (White Knight users will choose **Local->Serial Port**) using the highest baud rate that both persons' modem will support, and NO parity, 8 data bits, 1 stop bit, and HALF duplex.

Step 2: Person A puts their machine in auto-answer mode (White Knight users would choose **Service->Modem->Auto-Answer**). Person B places the call to Person A's machine (White Knight users would choose **Service->Dial Or Redial**).

Step 3: Once the connection is made, both users type a simple "Howdy, Bub" type message to make sure that both can receive what the other is sending. Once they're satisfied, Person A will type "I'm ready to send the file, are you ready?". Person B types "Ready, GO!".

Step 4: Person A tells their software to send the file using XMODEM (White Knight users select **File->Send File Using->XMODEM Protocol**). Person B tells their software to receive a file using XMODEM protocol (White Knight users select **File->Receive File Using->XMODEM Protocol**).

Step 5: Sit back and watch. Don't get disturbed if the file transfer takes up to a minute to start (the other person may be slower than you). When the file transfer ends, and if there are no further files to send using the same process in Steps 3 and 4, disconnect the modems (White Knight users select **Service->Modem->Hang Up**).



The above example would be precisely the same for any of the other protocols, there's nothing special about XMODEM except that virtually all microcomputer

communications packages support it. If the other person's software supports a faster protocol, by all means use it! Just remember that both users have to select the same file transfer protocol.

Downloading From A Host System Using XMODEM, YMODEM, or ZMODEM

Step 1: Get connected and logged into your Host system. In this example, I'll use CompuServe to download a file using XMODEM. Navigate to the area where the files available for downloading are available. In this example, I'll go into the MACPRO forum. What I type will be in boldface.

```
CONNECT 2400
^C (that's a CTRL-C, folks)
```

```
User ID: 73176,61
Password: (none o' yo beezwax!)
CompuServe Information Service
14:05 EDT Tuesday 22-Aug-89 P
```

```
Last access: 15:17 15-Aug-89
```

```
Copyright (c) 1989
CompuServe Incorporated
All Rights Reserved
```

```
You have Electronic Mail waiting.
GO RATES for information on new
connect time FREE services and
the Membership Support Fee
```

What's New This Week

- 1 Please Participate in Free Online Survey
- 2 MacWarehouse Has Grand Opening in The Mall
- 3 Game Challenge Forum Supports Modem Players
- 4 OAG Electronic Edition Adds World Travel Guide
- 5 Center Weather Advisories in EMI Aerobriefs
- 6 IQuest Management and Manufacturing SmartSCANS Free
- 7 Consumer Reports Adds Articles
- 8 Neptune Encounter Explored in Two Forums
(Above Articles are Free)
- 9 Online Today Daily Edition
- 10 Uploads: New Forum Files
- 11 Forum Conference Schedules

```
Enter choice !go macpro
CompuServe MACPRO
```

```
One moment please...
Welcome to MAUG (tm) MAC PRO Forum, V. 4D(101)
```

```
Hello, Scott Watson
Last visit: 15-Aug-89 15:23:17
```

Forum messages: 157052 to 200085
Last message you've read: 197291

Press <CR> ! (I type Return key)

MAUG (tm) MAC PRO Forum Menu

- 1 INSTRUCTIONS
- 2 MESSAGES
- 3 LIBRARIES (Files)
- 4 CONFERENCING (0 participating)

- 5 ANNOUNCEMENTS from sysop
- 6 MEMBER directory
- 7 OPTIONS for this forum

Enter choice !3

MAUG (tm) MAC PRO Forum Libraries Menu

Libraries Available:

- 1 Forum Business
- 2 Community Square
- 3 Programming
- 4 Utilities
- 5 DAs/FKEYs/INITs
- 6 S'ware Aids/Addons
- 7 Hardware
- 8 Telecommunications
- 9 CIS Navigator/VMCO
- 10 Mac II Software
- 11 Mac II Hardware
- 12 Desktop Publishing
- 13 'Early' Macs
- 14 Fonts

Enter choice !4

Step 2: Now look through the files until you find the one you want, and then tell the host system to send that file.

MAUG (tm) MAC PRO Forum Library 4

Utilities

- 1 BROWSE thru files
- 2 DIRECTORY of files

- 3 UPLOAD a new file
- 4 DOWNLOAD a file

- 5 LIBRARIES

Enter choice !1

Enter keywords (e.g. modem)
or <CR> for all: (I type Return key)

Oldest files in days
or <CR> for all: (I type Return key)

[73377,2457]

DENCAN.SIT/binary

21-Aug-89 2432

Accesses: 18

Keywords: TRASH CAN RESEDIT FINDER

This is a new Full Trash Can. I decide to make my Trash Can Icon when it was Full, a beat up old dented can with trash falling out of it. It reminds me of my wife all way telling me to take our trash out. Hope you like it. It requires ResEdit.

Press <CR> for next or type CHOICES !CHOICES
MAUG (tm) MAC PRO Forum Library Disposition

- 1 READ this file
- 2 DOWNLOAD this file
- 3 DESCRIPTION
- 4 RETURN to library menu

Enter choice or <CR> for next !2

Step 3: Now that we've told them to send it to us, we'll need to tell them which protocol to use. After that, we need to wait for them to tell us to begin our download. Don't tell White Knight to receive the file until they've told us to begin!

Library Protocol Menu

Transfer protocols available -

- 1 XMODEM
- 2 CompuServe B+ and original B
- 3 CompuServe A
- 4 DC2/DC4 (Capture)
- 5 YMODEM
- 6 CompuServe QB (B w/send ahead)
- 7 Kermit

0 Abort transfer request

Enter choice !1
Starting XMODEM transfer

Please initiate XMODEM transfer
and press <CR> when the transfer
is complete.

(This is where I choose File->Receive File Using->XMODEM
Protocol!)

The file transfer should now begin normally. Don't fret if it takes up to a minute

to get going. When it's finished, if the remote system doesn't send out any text, type a carriage return or two. I'm finished now, so I'll just log off.

[73377,2457]

DENCAN.SIT/binary

21-Aug-89 2432

Accesses: 18

Press <CR> for next or type CHOICES !bye

Here's a session with GENie, downloading a file from The FreeSoft RoundTable using ZMODEM. I'll forego the step-by-step comments in this example.

CONNECT 2400

HHH

U#=**xxx99999,password** (it's bogus, folks)

** Thank you for choosing GENie **

The Consumer Information Service
from General Electric
Copyright (C), 1989

Genie Logon at: 14:47 EDT on: 890822

Last Access at: 13:13 EDT on: 890822

\$ PRIME TIME Rate in Effect (\$18/hr) \$

* What's new at the Family History *
Library in SLC....Type "GENEALOGY"

* Visit "MICROSOFT" KnowledgeBase. *
-> No Surcharge During August!! <-

* "RENT" Mother Nature in GENie Mall *
Lease a Maple Tree: \$39.00!

* Meet A-Maze-ing Designer GCORSON *
Type "SCORPIA" for Details

* "SPORTS" RTC: General Sports talk. *
Give your two cents! 9/23 10PM EDT

You have 13 LETTERS WAITING.

Genie TOP Page 1
GE Information Services

- | | |
|--------------------|-------------------|
| 1. GENie Users' RT | 2. Index - Info |
| 3. Billing/Setup | 4. GE Mail & Chat |
| 5. Computing | 6. Travel |
| 7. Finance | 8. Shopping |
| 9. News | 10. Games |
| 11. Professional | 12. Leisure |
| 13. Reference | 14. Logoff |

Enter #, or <H>elp? **freesoft**

Number: 1347 Name: NEW WK DOWNLOAD WIND.SIT

Address: DICKVELDHUIS Date: 890815

Approximate # of Bytes: 7560

Number of Accesses: 2 Library: 7

Description:

Couple suggestions from Dick Veldhuis for decreasing the size of the file transfer windows for those of us with small screen Macs that like to do file transfers in the background. Document is in MacPaint format...

Keywords: Window, File Transfer

File: NEWWK DOWNLOADWIND.SIT
is a BINARY File.

Press <RETURN> to skip, <D>ownload, <L>ist, or <Q>uit.
?d

Select Download Protocol

1. XMODEM
2. XMODEM (w/1K blocks)
3. YMODEM
4. ZMODEM

Which item, or <RETURN> to skip? 4

File is ready. Start your ZMODEM receive file.
(this is where I choose File->Receive File Using->ZMODEM Protocol. When it finished, I typed a carriage return.)

rowse, <R>epeat, or <Q>uit?Q

Quitting ...

Genie

Page 586

Freesoft Software Library
Library: ALL Libraries

1. Description of this Library
2. Directory of files
3. Search File Directory
4. Browse through files
5. Upload a new file
6. Download a file
7. Delete a file you own
10. Set Software Library
11. Save Current Software Library
12. Instructions for Software Exchange
14. Directory of New Files
15. Join/Ignore Library Category

Item #, or <RETURN> for more? bye

Downloading From A Host System Using Kermit

Step 1: Since your logon sequence will depend on the mainframe, I'll forego that part of this sample session. However, the main thing is that you'll want to start up the Kermit program in its non-server mode (do not use the Kermit server

mode with White Knight). When you enter Kermit, you should get a prompt like the one below (depending on the Kermit version you are using):

```
VMS Kermit-32 version 3.3.117
Default terminal for transfers is: _LTA5704:
Kermit-32>
```

Step 2: To download (receive) a file from the host system, you must first tell them to send the file. The Kermit command for doing this is "send", followed by a space, followed by the filename you want to receive, followed by a carriage return:

```
Kermit-32> send sample.txt
```

Step 3: After you've typed this command, most Kermit programs I've worked with won't explicitly tell you to start. Instead, after a few seconds, you'll see a line of "garbage" characters that look like this:

```
, Sp/ @-#Y1~Y
```

When you see this line, immediately select **File->Receive File Using->Kermit Protocol** and the transfer should begin within a few seconds. When the transfer completes, you'll be back at the Kermit prompt. To exit the Kermit program, just type the command "quit" at the next Kermit prompt.

To upload (send) a file to the host system, you follow the above process except that you first type the Kermit command "receive" followed by a space, followed by the filename you want the host system to save the file with. When you see the line of garbage, choose **File->Send File Using->Kermit Protocol**, and then White Knight will prompt you for the file to send.



Before uploading any files using Kermit, review the section "Treat outgoing files as" in the chapter "Kermit Options". If you have this option set as "Text", the first command you should type to the Kermit host system (before the "receive" command) should be "set filetype text". If you have this option set as "Binary", the first command you should type to the Kermit host should be "set filetype binary". Also, make sure that the file has been given a name that is compatible with the host system. If you will be uploading to a Unix or VAX machine, a file named "This Is A Mac File" will cause them to choke and puke. Rename the file to something compatible with the target operating system, like "MACFILE.TXT" before uploading it.

Miscellaneous Features

This chapter will discuss a few handy features that don't fit well in any other chapter.

Selecting **File->Delete A File** allows you to remove files from any mounted disk volume. This is a handy alternative to having to quit to the Finder, put files in the Trash, empty the Trash, and then return to White Knight.

Selecting **Customize->Options->White Knight** presents you with a plethora of options in the following dialog box:

White Knight Options	
<input type="checkbox"/>	Screen saving and printing functions ignore blank lines
<input type="checkbox"/>	"Delete A File" continuously cycles until "Cancel" is pressed
<input checked="" type="checkbox"/>	"Locate Text" uses zooming rectangle when window shifts
<input type="checkbox"/>	Using MultiFinder
<input type="checkbox"/>	Display title screen at program startup
<input type="checkbox"/>	Cancel Procedure in progress when new Procedure is executed
<input type="checkbox"/>	Locate Buffered Keyboard at bottom of window
Confirm choice before:	<input checked="" type="checkbox"/> Deleting a file
<input type="button" value="OK"/>	<input checked="" type="checkbox"/> Sending screen to printer button
<input type="button" value="Cancel"/>	<input checked="" type="checkbox"/> Sending screen to disk button
	<input type="checkbox"/> Close box in main window
	<input type="checkbox"/> Reset time and billing clocks buttons
Buffered keyboard right margin column:	<input type="text" value="80"/>

Screen saving and printing functions ignore blank lines: if this option is checkmarked, the "Display Screen To Printer" and "Display Screen To Archive File" buttons, the **Edit->Append To**, **Local->Window->Archive**, and **Local->Buffer->Archive** menu selections, and the SCREENPRINT and SCREENDISK Procedure commands. will all ignore blank lines of text.

"Delete A File" continuously cycles until "Cancel" is pressed: If this

option is checkmarked, you can delete multiple files without having to continue to select **File->Delete A File**. To stop deleting files, click on the "Cancel" button. If it is not checkmarked, you will have to select **File->Delete A File** for each file you wish to delete.

"Locate Text" uses zooming rectangle when window shifts: If this option is checkmarked, a zooming rectangle will be drawn whenever the vertical or horizontal scroll bars must be repositioned to display the located text. If it is not checkmarked, the located text will just flash several times.

Using MultiFinder: If you are running White Knight under MultiFinder, you must have this option checked in order for the Procedure "RUN" command to work properly. This setting must be selected properly for the kind of Finder you are working with.

Display title screen at program startup: Federal copyright laws advise that a legal copyright message must be displayed clearly. This is the primary purpose of White Knight's title screen. Once you've read and understand this message, you may tell White Knight to bypass this screen by checkmarking this option.

Cancel Procedure in progress when new procedure is executed: If this option is checkmarked, White Knight will allow you to interrupt an executing Procedure file when execute a second Procedure using a Macro Key. If it is not checkmarked, White Knight will display a dialog box telling you that a Procedure is already executing, and that you should cancel it first before executing a new Procedure.

Locate Buffered Keyboard at bottom of window: If this item is checkmarked, White Knight will position the Buffered Keyboard at the bottom of the Terminal Window instead of in the Status Bar area. If you have a large display screen, this will allow you to use the Buffered Keyboard, display one of the other Status Bars, and still have plenty of space left for incoming text display. If this option is checkmarked, the Buffered Keyboard will not be hidden when "Hide Status Bar" is checked (but the Status Bar displayed in the Status Bar area will be).

Confirm choice before: If you are a beginner, you will come to appreciate that White Knight uses confirming dialogs ("Do you really, really want to...?") for many functions. Once you've become confident with White Knight's operation,

however, you'll probably come to find that at least some of these dialogs are a pain in the rear. This option allows you to selectively tell White Knight to use confirming dialogs for deleting a file, the "Display Screen To Printer" button, the "Display Screen To Archive File" button, the Close Box in the Terminal Window (which quits White Knight), and resetting the elapsed time and billing clocks.

Buffered keyboard right margin column: This option allows you to tell White Knight the number of characters to allow you to type into the Buffered Keyboard. Once you've reached this "right margin" limit, White Knight will beep at you and allow no further characters to be entered until you backspace, delete characters, or press the Return key to send the contents of the keyboard buffer.

Host Mode

White Knight has a feature called "Host Mode" that allows it to run in an unattended state to send and receive files. You could, for instance, leave your Macintosh running when you leave your office for the evening, call it later from your IBM-PC at home, and transfer files to and from White Knight using any standard communications program on the PC side. When White Knight is in Host Mode, it is watching for certain commands to be typed. When it receives a command that it recognizes, it responds in an appropriate manner.

White Knight's Host Mode is what I call "down and dirty". It does not hold hands with a beginner, check for passwords, identify the caller, switch baud rates to that of the incoming call, provide menus, or any of the other niceties inherent to a full blown host system. My recommendation is that the Host Mode is used for personal use only, or at least a very small core group of experienced communicators. If you need a full blown host system or bulletin board system, The FreeSoft Company sells one. It is called "Second Sight" and it is a highly customizable standalone host system program that can do all of the things mentioned above, plus manage electronic and private mail, provide an audit trail for each session, and has a custom scripting language just for question and answer type procedures. If the remote machines will all be Macintosh computers, you don't need all of the features of a full blown host program, but you do like the friendly Macintosh interface, you should examine the capabilities of FreeSoft's "Okyto" program. Please contact us for details if you require such a program. This isn't so much a plug as a warning not to use White Knight's Host Mode for something it wasn't intended.

To put White Knight into Host Mode, follow these steps:

- 1) Set up your serial port settings to be NO parity, 8 data bits, 1 stop bit, and HALF duplex. The baud rate must be set to a baud rate that all incoming callers can support - White Knight will not switch baud rates. The people that call your White Knight should have their serial port settings set up precisely the same. They should also be emulating a dumb terminal (TTY) device.



If they complain that they can't see what they're typing, then they are not using HALF duplex, also known as "Local Echo" in some communications software.

2) Put your modem in auto-answer mode by selecting **Service->Modem->Auto Answer**.

3) Put White Knight into Host Mode by selecting **Local->Host Mode->Become Host**. When White Knight is in Host Mode, you'll see a message saying so in the Status Bar area of the Terminal Window. To exit the Host Mode, just click your mouse in this Status Bar area.



All Host Mode functions will work fine in the background under MultiFinder.

Using Host Mode

Host Mode does not "know" when someone is connected. It is simply waiting patiently for a string of characters to come over the serial port that it understands. It will not react to any other non-command strings.

When a user connects, they should type a couple of carriage returns to clear out any connection garbage. They they should begin typing the necessary commands to accomplish what they're after.



Many of the Host Mode commands require what is called a "full pathname" to designate a file. If you are not familiar with this term, you must read about this in the chapter "About Filenames" before attempting to use Host Mode.

All Host Mode commands are given in uppercase letters and must begin in the first column of a new line. Host Mode will ignore lowercase or mixed case commands like "Sendx" or "sendx", but will respond to "SENDX". When a pathname is used, there is no difference between upper and lower case letters in the pathname itself.

Part 2: Host Mode Commands

To Send A File To The White Knight Running Host Mode

To send a file from the calling machine to the White Knight in Host Mode, the user will first type one of the following commands:

RECX filename to use XMODEM protocol
RECY filename to use YMODEM protocol
RE CZ filename to use ZMODEM protocol
RECK filename to use Kermit protocol
RECF filename to use Flash protocol

If the "filename" is valid, Host Mode will respond to the caller with the string "Ready to receive". At that point, the sender will instruct their communications program to send the file using the selected protocol.

You can also instruct Host Mode to capture incoming text to a file. The command:

RECN filename

creates a new file (destroying an existing file by that name if one already exists) and captures all text it receives to that file. The command:

RECA filename

will append all text it receives to the end of the file if it exists, or will create a new file by that name if one doesn't yet exist. To close the file and tell Host Mode to stop capturing, the following command is used:

CLOSE

Both asterisks are important, and there are no spaces in this command. When Host Mode closes the file, it will send back the response "Closed...". Remember that this command must be typed at the beginning of a new line.



White Knight will "rewind" the captured file so that the file will end with the data just before the *CLOSE* command.

To Receive A File From The White Knight Running Host Mode

To have Host Mode send a file to the calling machine, one of the following commands are used:

SENDX filename to use XMODEM protocol
SENDY filename to use YMODEM protocol
SENDZ filename to use ZMODEM protocol
SENDK filename to use Kermit protocol
SENDF filename to use Flash protocol

If the filename is valid, Host Mode will not send a response, but will immediately begin sending the file. At this point, the caller should instruct their communications software to receive the file using the selected protocol. To

have White Knight send a text file without using an error checking protocol, use the command:

SENDA filename

To Tell Host Mode To Disconnect The Call

To instruct Host Mode to disconnect the current call, the user should type the following command:

HANG UP

To Tell White Knight To Exit Host Mode and Execute A Procedure File

The command:

DO filename

will exit Host Mode and execute the compiled Procedure file in "filename". It will not automatically return to Host Mode when the Procedure file finishes, so you might want to use the Procedure command "HOST" to accomplish this.

To Examine The Contents Of The Disk Drives

There are two commands used to look at the connected disk drives, and the files and folders that are contained on them. The command

DRIVES

tells White Knight to print out the volume names of all mounted disk volumes. The volume names that Host Mode responds with are enclosed in brackets so that any leading or trailing spaces are obvious (as you'll learn, leading and trailing spaces are significant and must be supplied when giving a full pathname).

Once you know the names of the mounted disk volumes, you can look at the contents of a particular volume, or a folder contained on that volume with the command:

DIR pathname

This command tells White Knight to print out the names of all folders and files contained in the volume or folder "pathname". When Host Mode lists them, file and folder names are enclosed in brackets so that leading and trailing spaces are obvious. Folders are designated with a colon as the first character to set them apart from files. Some examples of this command would be:

DIR HD

would show all files and folders on the disk volume titled "HD".

DIR HD:Comm

would show all files and folders in the folder "Comm" which

resides on the disk volume "HD".

DIR HD:Comm:BBS

would show all files and folders in the folder "BBS" which resides in the folder "Comm" which resides on the disk volume "HD".

Special Host Mode Commands

There are two special commands for implementing optional features (that is, you decide whether or not to support them). The command:

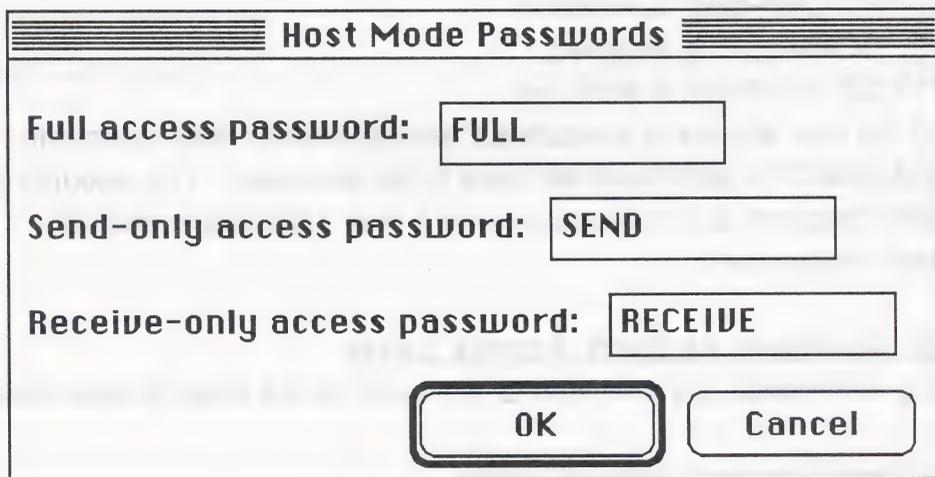
HELP

will cause Host Mode to look for a file titled "Host Mode Help" that resides in the same folder as White Knight. If one by that name exists, Host Mode will send that file in the same manner as it would respond to a SENDA command. This file should contain only plain text.

The command:

PASSWORD string

is used to take advantage of Host Mode's password capability. The "string" contains the user's password. Host Mode will grant certain allowances to that user based on the password. Here's how it works. By selecting **Local->Host Mode->Passwords** the following dialog box is displayed:



The dialog box is titled "Host Mode Passwords" and has a double-line border. It contains three rows of labels and text input fields. The first row is "Full access password:" followed by a text box containing "FULL". The second row is "Send-only access password:" followed by a text box containing "SEND". The third row is "Receive-only access password:" followed by a text box containing "RECEIVE". At the bottom of the dialog are two buttons: "OK" and "Cancel".

As you can see, Host Mode offers three levels of password protection, full, send-only, and receive-only. In the above dialog box, you'll see that I've entered the passwords "FULL", "SEND", and "RECEIVE", which would be the strings the caller would have to enter with the PASSWORD command to achieve the level associated with that string (hopefully, you'll come up with less easily guessed passwords!). As a default, these three passwords are blank,

which tells White Knight that any caller is to be given FULL access.

When the user enters the PASSWORD command, Host Mode will respond with one of the following strings: "Full access granted", "Send-only access granted", "Receive-only access granted", or "Bad password".



If you are going to specify a password for one of the levels, you should specify one for all three. Don't leave any of the password levels blank unless all three are left blank, because if any are left blank, full access is automatically given to the caller.

The password may contain up to 8 characters. Upper and lowercase letters are considered the same.

How Passwords Affect Access

If passwords have been declared, Host Mode will lower the access level to no-access on the following conditions:

- 1) The DO command is executed.
- 2) Host Mode is entered (even with a Procedure HOST command).
- 3) The string "NO CARRIER" is received.
- 4) The string "CONNECT" is received.
- 5) The HANG UP command is executed.

Therefore, if the user wishes to execute a Procedure which later returns to Host Mode, the PASSWORD command will have to be reentered. This security is not by any means foolproof, but it should provide a very satisfactory level of insulation from malcontents.

Commands Available At Each Access Level

The following commands are available to the caller at the listed access level:

No Access Level (or any Access Level)

HANG UP
PASSWORD

Any Access Level (except No Access)

HELP

Full Access Only

DRIVES

DIR

DO

Receive-Only Or Full Access

RECX

RECA

RECN

RECK

RECZ

RECF

RECY

CLOSE

Send-Only Or Full Access

SENDA

SENDX

SENDY

SENDZ

SENDK

SENDF

Host Mode will ignore any commands not available at the user's given access level.

Introduction To Procedures

At this point, you should be getting pretty comfortable with White Knight, and are ready to start automating some of the tasks you perform repetitiously. A great deal of White Knight's power comes from its very complete Procedure language, so please don't shy away from this feature just because it looks complex at first glance. We're going to introduce you to Procedures in the most painless way possible in the next couple of chapters, progressing finally until you are a full fledged telecommunications hacker (that's a compliment).

What Is A Procedure?

A Procedure is a set of instructions that guide White Knight to perform a series of actions. There is virtually nothing that you can do with White Knight that the Procedure language cannot also perform unattended. Let's take the following bogus session as an example. What I typed will be in boldface.

(**First, I dialed the number, which is 555-9999**)

CONNECT 2400

(**Now, I type a carriage return**)

Welcome To The Computer Connection BBS!

Please enter your first name? **SCOTT**

Now, enter your last name? **WATSON**

Now, enter your password? **DOGLIPS**

You're in! Please remember to check your mail and delete any private messages after you've read them.

Computer Connection BBS Main Menu

I can break down the above actions into the following instructions:

- 1) Dial the number.
- 2) Wait for the message "CONNECT 2400".
- 3) Type a carriage return.
- 4) Wait for the string "first name?".
- 5) Type the string "SCOTT".
- 6) Wait for the string "last name?".
- 7) Type the string "WATSON".
- 8) Wait for the string "password?".
- 9) Type the string "DOGLIPS".

By using White Knight's Procedure language, I could encode these instructions

and have White Knight perform this logon sequence automatically! Additionally, I could create other Procedures to automatically navigate to various areas of the host system, and perform necessary actions like capturing messages to a disk file (for later perusal), downloading files, just about anything you can think of.

Remember that one of the things a Macro Key can be programmed to do is execute a Procedure. By taking advantage of this, I can exploit the full power of a host system just by clicking on Macro Key buttons (or typing a keystroke I've assigned to a Macro Key).

Imagine having a digitized picture of my friend's face in the Macros Window. Just by clicking on his head (!), I could have White Knight dial up his computer and exchange files.

The possibilities of Procedures make my head spin, and believe me, I've seen some doozies. As you brave forth into the following chapters, I promise that you're going to really have fun with telecommunications.

There are three ways that White Knight Procedures are created:

- 1) By telling White Knight to watch what you do and then create a Procedure to mirror image your actions.
- 2) By using the Procedure Editor, which implements a point and click environment with a subset of the most often used Procedure commands.
- 3) By using the Procedure Editor, or a text editor/word processor of your choice, and taking advantage of the full 200+ Procedure command language.

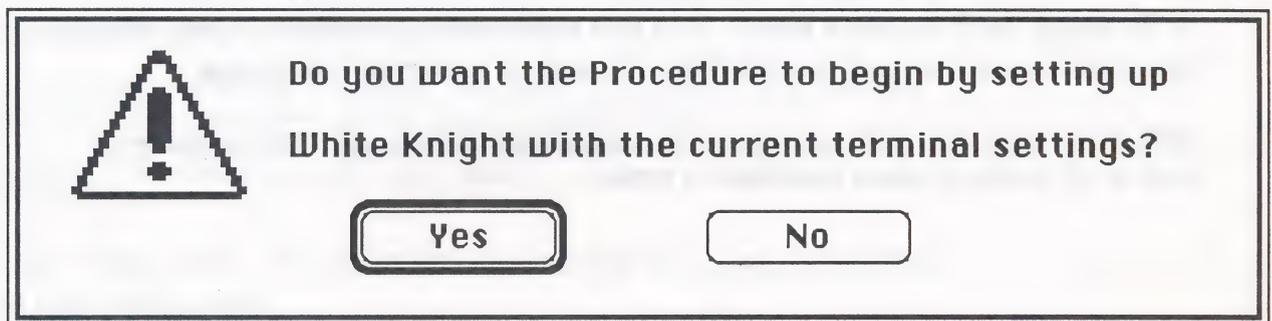
With each step, the difficulty (and the capabilities) increases. We're going to look at all three of these methods in order.

Letting White Knight Write A Procedure For You

Imagine saying to your machine, "Yo, White Knight! Watch what I do and learn it by heart, because I'm going to ask you to do this later." If that sounds like something out of Star Wars, you might be surprised to find out that White Knight actually has the capability of watching what you do, and then generating the Procedure language instructions to perform those actions.

This is the easiest way to get involved with the Procedure language, because you actually don't even need to know the Procedure language! However, as you'll see, there are some limitations to this. So, I think you'll find this to be a useful feature (I use it myself frequently, and I wrote the Procedure language!), but it should not be a stopping point.

To instruct White Knight to write a Procedure for you, select **Service->Procedure->Write For Me**. White Knight will then ask you to give this Procedure a name. This is the file where White Knight will store the instructions it generates. After you've named the file, the following dialog box appears:



If you click on the "Yes" button, White Knight will generate all of the Procedure instructions to set up the current serial port settings, terminal emulation style, and line length. This is typically used only in a Procedure that will connect to and log you into a service (rather than in a Procedure that is used once you are already connected).

From this point on, White Knight will watch the data that comes over the screen, and watch how you respond with your typing. It will generate the proper "wait for this string" and "type this response" instructions to mirror image your actions.

In addition, it will also generate commands to duplicate any of the following actions you perform:

- Dialing or redialing a number
- Loading a Settings File
- Loading a Macro Key File
- Loading a Filter File
- Changing the serial port settings
- Resetting the Elapsed Time and Billing Clocks.
- Starting and ending file captures
- Sending a TEXT file.
- Turning on or off Printer Echo
- Printing files
- Deleting files
- Sending or receiving files using a file transfer protocol



To dial or redial a number, it is imperative that you use the **Service->Dial Or Redial** menu choice in order for White Knight to generate the proper instructions. Do not use a Phonebook entry or type a dialing command directly to the modem.

When you've arrived at the point you want the Procedure to later take you to, instruct White Knight to stop generating instructions by selecting **Service->Procedure->Stop Writing**.

Now, here comes the important part. The file that you just created contains the plain text version of the Procedure commands. I'll show you how to open up this file, look it over, and perhaps even make changes to it in the next chapter. This is not an executable Procedure File, we refer to it as a "Source Code File".

The Source Code File, as I said, is not executable. To create an executable Procedure File, we run the Source Code File through a gizmo called a Procedure Compiler. The Compiler reads in the Source Code File, and creates a second executable file. It's this second file that we call a Procedure File.

To compile a Source Code File into a Procedure File, select **Service->Procedure->Compile TEXT File**. The compiler will first prompt you to select your Source Code File. Once you've done that, the compiler will ask you to give the resulting Procedure File a name. It will automatically

suggest a name for the Procedure File, which is the name of the Source Code File with the letters ".PROC" appended to the end.

After you've named, or have accepted the suggested name for the Procedure File, the compiler window will appear. You'll see the Source Code File instructions scrolling very quickly (too quickly to read, in fact) up this window, and when the compiler finishes, the compiler window will disappear.

You now have an executable Procedure File. To execute it, select **Service->Initiate Procedure** and select the Procedure File. If all goes well, the Procedure File will imitate your previous actions perfectly.

A Procedure File has an icon that looks like this:



Sample.PROC

Hints And Tips

1) While generating a Procedure file, type slowly and deliberately. Typing errors, backspacing, etc. will be repeated verbatim for the next billion years (or however long you continue to use that Procedure).

2) When a menu command and typing action are to take place after the same prompt from the host (like opening a receive file and then typing the command to the host system to list all new messages), choose the menu action first if possible. A typing sequence must never be interrupted by a menu command action - do one before or after the other. However, if you're going to be sending or receiving a file using a file transfer protocol, you'll find that the keyboard is locked out after the menu command is executed, so in this case the typing sequence would come first, followed by the menu action.

3) Break signals, Macro Keys, VT100 cursor key and numeric keypad sequences, and automatic VT100 cursor key positioning are not supported in Procedure generation. Typing sequences are limited to normal ASCII alphanumeric and control keys.

Limitations Of Letting White Knight Write The Procedure

There are, in fact, certain situations that the Procedure that White Knight generates will fail dismally to perform as expected. Here are three times when you'll see this happen:

- When the host system does not perform consistently. If the host system does not provide the same prompts as when you created the Procedure, White Knight will "get lost" and not know what to do.

- When things are based on time delays, rather than prompt and response type sequences.
- When you need to do something (like sending a modem break signal) that this feature doesn't support.

When this feature works, it works great. If the Procedure does not perform as expected, it's time to get your hands dirty and modify it so that it will perform right. I'm going to tell you up front that doing modifications can get pretty complex, and requires a strong understanding of the Procedure language. If you're still a beginner, I would suggest instead that you write the Procedure from scratch, rather than modifying the one that White Knight wrote. As you'll find out in the next chapter, this isn't nearly as difficult as you might expect.

The Procedure Editor

In this chapter, we're going to introduce you to White Knight's Procedure Editor, and use it to write Procedure Files.

The Procedure Editor is contained in the file titled "ProcEdit" on your White Knight master disk. For it to be available, you must have this file residing in the same folder as White Knight and you must not rename it. It is executed by selecting **Service->Procedure->Edit Procedure** (if the ProcEdit file is not in the same folder as White Knight, this menu choice will not exist).

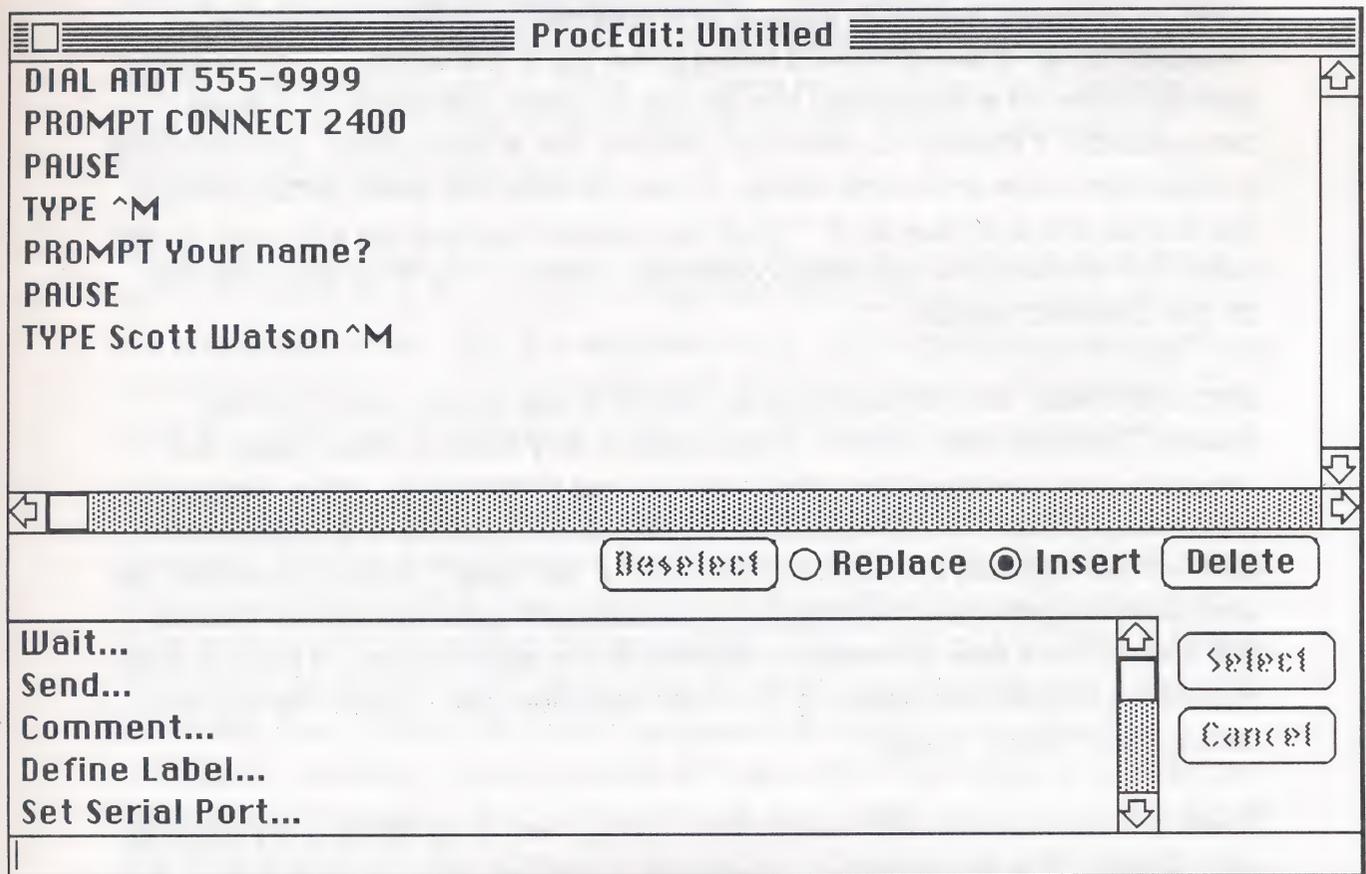
You may notice that the ProcEdit file has an icon that looks just like a Procedure File. This is one of the unique properties of the Procedure Editor and a testament to the power of White Knight's Procedure Language. The Procedure Editor, is in fact a Procedure File! The **Service->Procedure->Edit Procedure** menu command is just a built-in shortcut of choosing **Service->Initiate Procedure** and then selecting ProcEdit. Because it is modular, and not a built-in part of White Knight itself, we could write a Procedure Editor that is icon-based, for instance. In fact, I think you'll be seeing a variety of Procedure Editors in the months to come.

To create a new Source Code File, select **File->New**. An "Untitled" Window will appear. To view or modify an existing Source Code File, select **File->Open**. To close a Source Code File without saving changes, select **File->Close**. The **File->Save** command is used to save the changes or additions you've made to the Source Code File. **File->Save As** is used to save a copy of the Source Code File to a different filename. **File->Revert** is used to lose any changes you've made and reload the Source Code File from disk back into ProcEdit.

Once you've got the Source Code File the way you like it and have saved it, it's time to compile it. Select **File->Compile** to do this. The Procedure File that is generated will have the same name as the Source Code File with the letters ".PROC" appended to the end. If you like, you can choose **File->Compile, Then Initiate** and the Procedure File will be automatically executed when the compiler finishes. If there is an error during compilation, the compiler will relaunch ProcEdit, and ProcEdit will display the Source Code File with the offending line highlighted.

Of course, choosing **File->Quit** exits ProcEdit back to White Knight.

Here is what ProcEdit's window looks like. We'll be referring to this illustration for the rest of the chapter:



The Title Bar shows the name of the Source Code File being edited. Clicking in the Close Box is equivalent to choosing **File->Close**.

The area beneath the Title Bar displays the actual Procedure instructions. I know that those in the illustration look like garbage, but you'll soon begin to understand what they mean. If these instructions run further than the right edge of the window, you can use the horizontal scroll bar to scroll left and right. The vertical scroll bar is used to view lines that are above or below the displayed lines.

The buttons and checkboxes below the instruction display control how instructions are entered (we'll discuss them in detail later). Below this is the point and click command selection box. The vertical scroll bar is used to scroll through this list of items.

Just below the point and click area is what I call the textual entry area. I'll also explain its use a bit later.

Entering And Editing Commands

When a command has been entered and accepted, it is displayed at the "selection point" in the command listing at the top of the window. The selection point is shown as a highlighted line (shown as white characters on a black background). If there is no selection point (no line is highlighted), the command is appended at the end of the listing. To set the selection point, simply scroll to the desired line and click on it. To set the selection point to the end, click on the blank line following the last listed command. Another way to do this is to click on the "Deselect" button.

Just underneath the command listing box you'll see the two radio controls labeled "Replace" and "Insert". These buttons only have an effect when the selection point is somewhere other than the end of the listing. If the "Replace" button is selected, the command highlighted at the insertion and replacement point will be replaced by the new command. If the "Insert" button is selected, all lines from the command highlighted at the selection point are moved forward one line, and the new command is inserted in the resulting gap. If you just want to delete a line without replacing it, set the selection point to that line and then click on the "Delete" button.

There are two ways to enter commands: typing them in by hand, or by pointing and clicking. The fastest way is to type the commands into the textual entry area at the absolute bottom of the window. However, this requires a knowledge of White Knight's native Procedure language, which is not very English-like. The point and click area is used to take you step by step in plain English to select the desired command.

The point and click area has one major benefit and one major drawback. The benefit is that it is much more suitable to guide a beginner through a subset of the most commonly used Procedure commands in plain English than to just dump a stack of 200-plus arcane Procedure commands in your lap and expect you to wheedle your way through them.

The chief drawback is that using the point and click environment is much slower than typing in the commands by hand, especially once you've become familiar with the command language. In other words, the very thing that was constructed to help you will sooner or later become an obstacle. As you work

with ProcEdit, you'll begin to learn which point and click sequences generate which commands. As the commands are generated, it will help you to look at their descriptions in the chapter "Procedure Commands" to find out exactly what they do - these commands are listed in alphabetical order in the back of this manual to help you locate their descriptions. To understand the nomenclature used in the command descriptions, you'll want to first read the information in the chapter "Advanced Procedure Files". At some point in time, you'll be able to type the commands directly in the textual entry area without having to go through the point and click routine.

You'll also notice that when you set the selection point by clicking on a line in the command listing, the contents of that line will automatically be displayed in the textual entry area. This is a very fast way to make small typing changes to a line without having to go through the point and click replacement process. To enter the command in the textual entry area, simply press your Return key.

The point and click area takes you step by step into selecting a desired command in White Knight's native Procedure language. To select a command, look at the list of words in the point and click area and click your mouse on the most appropriate one so that it is highlighted. Then, click on the "Select" button. Depending on the command, there may be further lists of words that subdivide the earlier item, a dialog box might be presented asking you to fill in the information needed by that command, or if no further information is needed, the command will appear in the command list at the selection point (or at the end of the command list if there is no selection point). If you get into a sublevel in the point and click area and wish to return to the top level, just click in the "Cancel" button.



A shortcut for clicking on the desired point and click item and then clicking on the "Select" button is to doubleclick on the desired item.

Point And Click Commands

We'll first go through the point and click sequences for each command, as they appear in order in the point and click area, and describe what each command does. In the section following this, we'll discuss how these commands fit together to form a Procedure File.

Wait

This item has four sublevels: "Wait for text", "Wait for control character", "Wait until time has elapsed", and "Wait until time of day".

Wait for text: This command allows you to enter a string of characters (upper and lowercase are different and must be exact) that the Procedure is to wait to receive before executing any further commands. This sequence generates a "PROMPT" command.

Wait for control character: This command allows you to enter in a control character that the Procedure is to wait to receive before executing any further commands. This sequence generates a "PROMPT ^" command.

Wait until time has elapsed: This command prompts you to enter in the number of 60th's of a second the Procedure is to delay before executing any further commands. This sequence generates a "PAUSE" command.

Wait until time of day: This command prompts you to enter in a time of day in 24 hour HH:MM:SS format. HH is the hours, from 00 to 23, and (like MM and SS) must have a leading zero if it is less than ten. MM is the minutes, from 00 to 59. SS is the seconds, from 00 to 59. The Procedure will wait until that time of day before executing any further commands. This sequence generates a "WAIT" command.

Send

This item has three sublevels: "Send text", "Send long break signal", and "Send short break signal".

Send text: This command prompts you to enter a string of characters that the Procedure is to send. Control characters can be embedded in the string in the same manner as they would be in a Macro Key. A carriage return is not automatically supplied, so if you want the line to be terminated with a carriage return, you would end the line with a "^M" sequence (a caret character followed by an upper case M, which stands for CTRL-M). This sequence generates a "TYPE" command.

Send long break signal: This command causes White Knight to send a long (3 to 4 second) modem break signal. This command generates a "LONG BREAK" command.

Send short break signal: This command causes White Knight to send a

short (233 millisecond) modem break signal. This command generates a "SHORT BREAK" command.

Comment

There are no sublevels to this item. This item prompts you to enter a string of characters in the form of a comment. Comments are useful for documenting what each section of your Procedure is used for. The comment text will be preceded by a "(" character. It's not necessary (but looks better) to end the comment with a ")" character.

Define label

There are no sublevels to this item. This item prompts you to enter a string of characters to define a label. The label text will be preceded by a ":" character. Labels are used to designate specific places in the Procedure code. They're used by other Procedure commands to specify where execution is to branch to under certain circumstances.

Set serial port

There are no sublevels to this item. This item prompts you to specify the baud rate, parity, databits, stopbits, and duplex that White Knight is to switch the serial port to use. This sequence generates a "COMM" command. As an interesting tidbit, you only have to specify the setting that you wish to change. In other words, if you only want to change the duplex, but not the baud rate, parity, databits, or stopbits, you can select just a duplex setting in the dialog box and ignore the others.

Branch

This item has seven sublevels: "Branch to label", "Execute Procedure", "Nest Procedure", "Return from nested Procedure", "Whenever text is received", "Branch to subroutine", and "Return from subroutine".

Branch to label: This item prompts you to enter a label (which was or will be supplied using the "Define label" sequence described earlier) that the Procedure is to immediately branch to and continue execution at that point. This sequence generates a "JUMPTO" command.

Execute Procedure: This item prompts you to enter the filename of a Procedure file to immediately execute. This sequence generates a "DO" command.



All of the sequences that prompt you to enter a filename will have a "Select From Dialog" button that will construct and insert the correct filename. It is strongly suggested that you use this to avoid entering a faulty pathname. If the file does not yet exist, or for some other reason you need to type in the pathname, be sure to read the chapter "About Filenames" before attempting to do this.

Nest Procedure: This item prompts you to enter a filename of a Procedure File and generates a "NEST" command. A NEST command allows your Procedure to run a second Procedure as a subroutine, returning to the instruction following the NEST command. Procedures should be nested no more than five deep, although six is the actual limit (nesting six deep can cause problems if you are also using a Modem Driver).

Return from nested Procedure: This item generates a "NESTEND" command. If the Procedure was executed by a "NEST" command, the "NESTEND" returns to the calling Procedure. If it was not executed by a "NEST" command, the Procedure file simply stops execution.

Whenever text is received: This item prompts you to enter a string of characters, and a label that the Procedure is to branch to whenever that text is received. There are three priority levels, and each priority level must be used only once. This sequence generates an "ALERT1", "ALERT2", or "ALERT3" command, depending on the priority level. Whenever a "PROMPT" or "ALERT" command is satisfied (the text has been received), all other "PROMPT" and "ALERT" commands are turned off (and the priority levels may then be reused). Unlike the "PROMPT" command, the Procedure continues to execute instructions after this command, so this command is generally used before a "PROMPT" command in order to look for more than one string of characters.

Branch to subroutine: This item prompts you to enter a label that the Procedure should branch to immediately and continue execution from. This sequence generates a "GOSUB" command. Unlike the "JUMPTO" command, the "GOSUB" command is used in conjunction with the "RETURN" command (described next) to return execution to the instruction following the "GOSUB" command. Therefore, the same subroutine can be called by different parts of your Procedure File with subsequent return back to the calling part.

Return from subroutine: This item generates a "RETURN" command, which is used to terminate a subroutine. The "RETURN" command branches to the instruction following the "GOSUB" command that called the subroutine.

Modem

This item has four sublevels: "Dial or redial a number", "Turn auto answer on/off", "Hang up modem", and "Initialize modem".

Dial or redial a number: This item brings up a dialog box exactly like the one in White Knight when you choose **Service->Dial Or Redial**. It generates either a "DIAL" or "REDIAL" command. Note: The "REDIAL" command will be preceded by a "REDIAL LIMIT" command.

Turn auto answer on/off: This item causes White Knight to instruct the modem whether or not to auto-answer and attempt to connect with incoming calls. It generates either an "AUTOANSWER ON" or "AUTOANSWER OFF" command.

Hang up modem: This item causes White Knight to instruct the modem to disconnect and hang up the phone line. It generates a "HANGUP" command.

Initialize modem: This item causes White Knight to initialize the modem so that it is prepared to accept further instructions. It generates an "INITMODEM" command.

File

This item has six sublevels: "Send file using protocol", "Receive file using Protocol", "Load File", "Send Text File", "File Capture", and "Delete File".

Send File Using Protocol

This item has five sublevels: "Send file using XMODEM protocol", "Send file using YMODEM protocol", "Send file using ZMODEM protocol", "Send file using Kermit protocol", and "Send file using Flash protocol". Each of these sublevels prompts you to select the file (or batch file) to send. It generates a "SENDX", "SENDY", "SENDZ", "SENDK", or "SENDF" command, depending on the protocol selected.

Receive File Using Protocol

This item has five sublevels: "Receive file using XMODEM protocol", "Receive

file using YMODEM protocol", "Receive file using ZMODEM protocol", "Receive file using Kermit protocol", and "Receive file using Flash protocol". Each of these sublevels prompts you to select the file to receive. It generates a "RECX", "RECY", "REZ", "RECK", or "RECF" command, depending on the protocol selected.

Load File

This item has two sublevels: "Load Settings File" and "Load Macro Keys File".

Load Settings File: This item prompts you to select the Settings File White Knight is to load when this instruction is executed. It generates a "LOADSET" command.

Load Macro Keys File: This item prompts you to select the Macro Keys File White Knight is to load when this instruction is executed. It generates a "MACRO" command.

Send Text File

This item prompts you to select a text file that White Knight is to send when this instruction is executed. It generates a "SENDA" command.

File Capture

This item has four sublevels: "Capture to new file", "Capture to end of existing file", "Close capture file", and "Pause capture on/off".

Capture to new file: This item prompts you to enter the file that White Knight is to create and save all further incoming data to when this instruction is executed. It generates a "RECN" command.

Capture to end of existing file: This item prompts you to select the existing file that White Knight is to open and append all further incoming data to the end of when this instruction is executed. It generates a "RECA" command.

Close capture file: This item closes the capture file opened with a previous "RECN" or "RECA" command. It generates a "CLOSE" command.

Pause capture on/off: This item turns on or off the **File->File Capture->Pause** menu choice, which has the effect of either suspending or resuming capturing of data. It generates either a "SUSPEND ON" or "SUSPEND OFF" command.

Delete File

This item prompts you to enter a filename that White Knight will delete from the disk when this instruction is executed.

End Procedure

This item has no sublevels. It generates an "END" command, which causes White Knight to stop execution of the Procedure File.

Printer

This item has two sublevels: "Turn printer echo on/off", and "Print file".

Turn printer echo on/off: This item causes White Knight to either turn on or off printer echo. It generates either an "ECHO ON" or "ECHO OFF" command.

Print file: This item prompts you to select a text file that White Knight is to print when this instruction is executed. It generates a "WRITE" command.

Timer

This item has two sublevels: "Reset timers to zero" and "Set billing timer cost".

Reset timers to zero: This item generates a "RESET" command, which causes White Knight to reset the elapsed time and billing clocks to zero when it is executed.

Set billing timer cost: This item prompts you to enter the cost per minute in 100th's of a cent with which White Knight is to increment the billing clock. It generates a "SETCOST" command.

Status Bar

This item has nine sublevels, "General Status Bar", "Macros Status Bar (currently active set)", "Macros Status Bar (active set #1)", "Macros Status Bar (active set #2)", "Macros Status Bar (active set #3)", "Buffered Keyboard Status Bar", "VT100 Keys Status Bar", "Hide Status Bar", and "Show Status Bar".

General Status Bar: This command generates a "GBAR" command, which causes White Knight to display the General Status Bar which this command is executed.

Macros Status Bar: The next four items cause White Knight to display the

Macros Status Bar when the instruction is executed. There are four variants, each of which affect which active set is displayed in the Status Bar area, either whatever set is currently active, or specifically set #1, #2, or #3. It generates either an "MBAR", "MBAR 1", "MBAR 2", or "MBAR 3" command.

Buffered Keyboard Status Bar: This item generates a "BBAR" command, which causes White Knight to display the Buffered Keyboard Status Bar whenever this instruction is executed.

VT100 Keys Status Bar: This item generates a "VBAR" command, which causes White Knight to display the VT100 Keys Status Bar whenever this instruction is executed.

Hide Status Bar: This item generates a "NOBAR ON" command, which causes White Knight to hide the Status Bar display when this instruction is executed.

Show Status Bar: This item generates a "NOBAR OFF" command, which causes White Knight to display the Status Bar when this instruction is executed.

Erase Display Screen

This item has no sublevels, it generates a CLEAR command, which erases the lines of text in the Terminal Window.

Set PANIC Time For Next PROMPT

This item has no sublevels. This item prompts you to enter in the number of seconds to wait for the string of characters to appear that are specified in the next "PROMPT" or "ALERT" command to be executed. If those characters are not received within the specified number of seconds, a "PANIC" condition exists. What White Knight will do when the "PANIC" condition exists depends on what you specify in the next command below to be described. This sequence generates a "PANICAFTER" command.

If PANIC Condition Exists

This item has no sublevels. This item will prompt for the label that White Knight is to branch to when the "PANIC" condition exists. It generates an "ONPANIC" command.

Quit White Knight

This item has no sublevels. This item generates the "QUIT" command which will cause White Knight to quit back to the Finder when this instruction is executed.

Sound A Bell

This item has no sublevels. This item generates the "BELL" command, which causes White Knight to sound a bell when this instruction is executed.

Part 2: Creating A Procedure To Automate A Session

Let's look at a sample connection and logging on sequence, and then I'll show you how you would use the Procedure Editor to automate that process. What I typed is in boldface.

```
(First, I set my serial port settings to 2400-N-8-1-FULL)
(Now, I dial the number 555-0000)
CONNECT 2400
^C (I typed a CTRL-C)
```

```
User ID: 73176,61
Password: (I type my password "GADZOOKS" and a carriage return)
CompuServe Information Service
14:05 EDT Tuesday 22-Aug-89 P
```

```
Last access: 15:17 15-Aug-89
```

```
Copyright (c) 1989
CompuServe Incorporated
All Rights Reserved
```

```
You have Electronic Mail waiting.
GO RATES for information on new
connect time FREE services and
the Membership Support Fee
```

```
What's New This Week
```

- 1 Please Participate in Free Online Survey
- 2 MacWarehouse Has Grand Opening in The Mall
- 3 Game Challenge Forum Supports Modem Players
- 4 OAG Electronic Edition Adds World Travel Guide
- 5 Center Weather Advisories in EMI Aerobriefs
- 6 IQuest Management and Manufacturing SmartSCANS Free
- 7 Consumer Reports Adds Articles
- 8 Neptune Encounter Explored in Two Forums
(Above Articles are Free)
- 9 Online Today Daily Edition
- 10 Uploads: New Forum Files
- 11 Forum Conference Schedules

```
Enter choice !go macpro
```

Now let's look at how I would use the Procedure Editor to write a Procedure to

do all of this automatically.

Step 1: Set the serial port settings

Scroll in the point and click area to "Set Serial Port" and select it. In the dialog box that is displayed, select 2400 baud, NO parity, 8 data bits, 1 stop bit and full duplex. Click on the "OK" button and you'll see the command

COMM 2400-N-8-1-FULL

appear in the command listing area.

Step 2: Dial the number

Scroll in the point and click area to "Modem" and select it. Select "Dial or redial a number". Click the item labeled "Dial this number only once". Enter the number "555-0000" in the box and click on the "OK" button. You'll see the command:

DIAL ATS7=30DT 555-0000

appear in the command listing area.

Step 3: Respond to the connection message

When I get the "CONNECT 2400" message, I want to type a control-C character. One important thing I need to teach you is that you always want to pause one second after receiving a prompt before typing anything. It's a good rule of thumb and it has always served me well. At any rate, here we go. First, scroll to "Wait" and select it. Select "Wait for text" and in the dialog box that appears, type in "CONNECT 2400" (without the quotes and in all uppercase - these strings have to be exactly like the text that will appear). You'll see the command:

PROMPT CONNECT 2400

appear in the command listing area. Now, again select "Wait", select "Wait until time has elapsed, and then type in 60 in the dialog box that appears. Since this is in 60th's of a second, the delay time will be one second. You'll see the command:

PAUSE 60

appear in the command listing area. Now, scroll to "Send" and select it. Select "Send Text" and in the dialog box that appears, type a "^C". That's a caret character (shift-6 on your keyboard) followed by an uppercase C. This is how we tell White Knight to send a control character. You'll see the command:

TYPE ^C

appear in the command listing area.

Step 4: Respond to the "User ID:" prompt

You'll find out that a lot of what you do in a Procedure is simply waiting for a prompt and then responding to it. For this one, select "Wait", select "Wait for text", and type in "User ID:". The command:

PROMPT User ID:

will appear. Now select "Wait", select "Wait until time has elapsed", and type in "60". The command:

PAUSE 60

appears. Finally, select "Send", select "Send text", and type in "73176,61^M". The "^M" at the end of this string is very important. A CTRL-M character happens to be a carriage return (what your "Return" key sends). If we neglected to put that in, CompuServe would just wait patiently for the carriage return, and our Procedure would be stuck in limbo. The command appears as:

TYPE 73176,61^MStep 5: Respond to the "Password:" prompt

Select "Wait", select "Wait for text" and enter "Password:". The command appears as:

PROMPT Password:

Now, select "Wait", select "Wait until time has elapsed" and type in 60 so that the command:

PAUSE 60

appears. Finally, select "Send", select "Send text", and type in "GADZOOKS^M". This appears as:

TYPE GADZOOKS^M

in the command listing area.

Step 6: Respond to the top level menu prompt

When we get to the first menu, we want to issue the command "go macro". Here's how we do it. Select "Wait", select "Wait for text", and type in "Enter choice !". Notice the space between "choice" and "!". When I say these things have to be exact, I really mean it. The command:

PROMPT Enter choice !

appears in the command listing area. Now select "Wait", select "Wait until time has elapsed", and type in "60" so the command:

PAUSE 60

appears. Finally, select "Send", select "Send text", and type in "go macro^M" so the command:

TYPE go macro^M

appears.

If everything went well, the command listing area should contain the following commands from top to bottom:

```
COMM 2400-N-8-1-FULL  
DIAL ATS7=30DT 555-0000  
PROMPT CONNECT 2400  
PAUSE 60  
TYPE ^C  
PROMPT User ID:  
PAUSE 60  
TYPE 73176,61^M  
PROMPT Password:  
PAUSE 60  
TYPE GADZOOKS^M  
PROMPT Enter choice !  
PAUSE 60  
TYPE go macpro^M
```

You would now save the Procedure, compile it, and execute the compiled Procedure to see if it works as you expect. By the way, the above sample happens to be the exact logon script I use for CompuServe (with a bogus phone number and password, of course). By substituting your own phone number, user ID, and password, (and possibly slowing down the baud rate) you'll have a ready to run CompuServe logon script.



It's possible (if you have Electronic Mail waiting for instance) that CompuServe will transfer you to some other menu rather than the top level menu after you log on. Since all CompuServe prompts end with a "!", you could change the above

PROMPT Enter choice !

command to just

PROMPT !

and this Procedure would get you to the "MacPro" forum from any menu you arrived at.

With just a few commands, you can create very complex Procedures. After you get comfortable writing a few "Wait for and respond" type Procedures like the one above, start looking at the descriptions (in the chapter "Procedure Commands") of the other commands available in the point and click area and see if they might be useful.

Advanced Procedure Files

A Procedure is simply a set of instructions you want White Knight to perform. Sound like a computer program? It is exactly that, and White Knight is equipped with a powerful, yet easy to learn and use, computer language of its own with over two hundred different commands. White Knight's Procedure language was originally intended to be used for automatic dialing and log on for remote services. It quickly became evident that the possibilities are literally endless, and the Procedure language has grown in quantum leaps and bounds.

Earlier versions of White Knight's Procedure language, as well as all other current telecommunications programs on the market use what computer hackers refer to as an interpreter to execute the instructions. White Knight now sports the next technical leap up: a true two-pass compiler. By compiling the written commands (called "source code") into a very efficient compiled form (called "object code"), White Knight offers the following advantages over interpreted commands:

- Extremely fast execution speed, as the translation of the commands to something meaningful to White Knight has already been done by the compiler and need not be done redundantly each time the commands are executed.
- Source code privacy. The source code commands are translated to a non-human readable format. In addition, a very devious method of encryption may be performed on the object code at compile time to prevent prying eyes from hacking the object code to find passwords or other critical text strings. With a few lines of code to add password protection (as shown in Example #12 in the chapter "Procedure Examples"), you can insure that a compiled Procedure File will be absolutely useless to unauthorized persons.

White Knight's Procedure Language is something many beginners get intimidated by because of its obvious power and depth, and put off learning to a later date. Please don't confuse this power with complexity. As you learned in previous chapters, a Procedure file can be as simple as a two-liner that sets the communications parameters and dials a phone number, or a wild bugger that automates an entire session while you sleep. It's up to you how complex you want to get. Start simple, experiment, learn, and have fun! With the examples provided, there's no reason you can't be programming like a pro after just a few

hours of leisurely study.

Just like learning any other computer language, the best way to learn Procedure commands is to browse over the following command descriptions (not trying to memorize them), and then look at some examples, referring back to the command descriptions for a more complete understanding. Learn a few commands at a time, biting off more as you feel comfortable. The beauty of it is that you can make your procedures as simple or complex as suits your needs and ingenuity, so don't overdo it by moving too fast.

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

- By choosing **Service->Initiate Procedure** and selecting the compiled Procedure File to execute.
- By double-clicking the mouse button (or single-clicking and choosing **Open** from the Finder's **File** menu) on a compiled Procedure File icon in the Finder. If you start up White Knight this way, the Procedure file will be immediately executed.
- By naming a compiled Procedure File with the special name "WKJ\$" (without the quotes) in the same folder as White Knight - White Knight will execute this file immediately upon starting up if it exists. If you create one of these, don't start up White Knight by double-clicking anything other than White Knight's icon, because what you double-clicked on will be ignored in favor of the WKJ\$ Procedure. The WKJ\$ file can be used to bypass the title screen, or to set up a default configuration you like instead of using one left over from the last session contained in the "WK's 11.0 Stuff" Settings File. Remember, the executable Procedure must be named "WKJ\$" - "WKJ\$.PROC" would not work.
- By using the Host Mode command **DO**, which is explained in detail in the chapter "Host Mode". There is also a Procedure File command **DO** for executing one Procedure File from another.
- Through a macro key. The first character of the macro key should be a backslash ("\") followed by the Procedure filename. Don't get the backslash confused with the slash ("/"). Installing a Procedure File filename into a macro key is explained in the chapter "Macro Keys".
- Through a Phonebook entry, which is explained in the chapter "Phonebooks".

During the execution of a Procedure File, the Title Bar of the Terminal Window will show the filename of the Procedure File executing. If at any time you wish to cancel a Procedure File in progress, you will find that the **Initiate Procedure** choice under the **Service** menu has changed into **Cancel Procedure**. Choose **Cancel Procedure**, and execution will end immediately.

If the **Service->Procedure->Monitor Execution** choice is checkmarked, Procedure commands are shown in the Status Bar area as they are executed to let you see what is happening. Because of the nature of compiled files, some of the commands (like the **JUMPTO** and **GOSUB** commands) that are shown may differ from the way they appear in the source code (labels will be changed to numbers). The **Monitor Execution** command may be overridden with the **QUIET** and **LOUD** Procedure commands during the Procedure's execution, and it will always be automatically uncheckmarked and disabled (for security reasons) whenever an encrypted Procedure File is executed.

Creating A Procedure File

The first step in creating a Procedure File is to type in the list of Procedure commands to execute into a text file. That file is then compiled into object code, and the object code may be executed. A few words about the source code file. A Procedure source code file is created with your favorite editor/word processor.



Each line of a Procedure source code file must be 79 characters long or less, and each line must end with a carriage return.

The gymnastics involved in coercing some Macintosh word processing applications to do this can be frightening. Thankfully, you can avoid this altogether by using the Procedure Editor (described in the section "Working With Received Text Files" in the chapter "Receiving Text Files (File Captures)") to just type in the commands in the textual entry area at the bottom of the editor window.

Once you've created and saved your Procedure source code file, it's time to compile it into executable object code. To do this, select **Service->Procedure->Compile TEXT File** (this is equivalent to choosing **File->Compile** in the Procedure Editor). You will be prompted to select the text file to

compile, and then for the name of the resulting Procedure File. Don't worry about the compiler doing anything to your source code file - it only reads it and does not change it in any way. A special window will appear during compilation, and you can click on the **Cancel** button inside the compiler window to stop compiling the file immediately. If you choose **Service->Procedure->Syntax Check**, the compiler will perform normally except that it will not create a compiled object code file. This is a good way to check for errors in the source code file without creating an executable file.



Don't throw away your Procedure source code file! You can not make any changes to a compiled Procedure File. This only way to make a change is to the source code file, which is then recompiled.

During compilation, the last several lines that have been compiled will be shown in the center of the compiler window. The bottom-most line is the one being currently compiled. **Any error messages you may see during the compile refer to the bottom-most line displayed in the compiler window.**

Because the syntax of Procedure commands is very consistent, there aren't that many different kinds of errors that can happen during compilation (only eight). Most of the time, simple observation of the offending line will disclose a spelling error.

Compiler Errors

Error: Syntax Error (Parameter #XXX)

Probable cause: Missing or unknown parameter, or an illegally terminated parameter in a list. Spelling error.

Error: Parameter #XXX should be a YYY

Probable cause: Wrong parameter type - it should be the type denoted by YYY in the error message.

Error: Out Of Label Reference Space

Probable cause: Too many label references. To increase the number of allowed label references, choose **Customize->Options->Compiler** and put a larger number in the "Max. number of label references" item.

Error: Out Of Label Definition Space

Probable cause: Too many label definitions. To increase the number of allowed label definitions, choose **Customize->Options->Compiler** and put a larger number in the "Max. number of label definitions" item.

Error: No Such Command

Probable cause: Misspelled or illegal Procedure command.

Error: Missing Command

Probable cause: Two part conditional command (such as **IF YES** or **ALERT**) lacks second command to execute when condition is true.

Error: No Such Label <XXX>

Probable cause: The label XXX was referenced but never defined.

Error: Line Is Over 79 Characters Long

Probable cause: Command line is too long or is not terminated with a carriage return.

Procedure Command Syntax

A Procedure command has either one or two parts. There will always be the command name, a label definition, or a comment. Many Procedure commands will also have a parameter list of one or more parameters.

All Procedure commands must begin flush left on the left margin. Do not indent any lines or add any leading or trailing spaces.

Comments

Any line that begins with a left parenthesis is considered to be a comment, not a command. Since comments are not compiled or included in the object code file, they do not affect execution speed of the Procedure. Therefore, it's wise to make liberal use of comments in your source code file. Don't try to put a comment after a command on the same line, as in:

```
MBAR 3      (Show the Macros Status Bar set #3)
```

as this would be construed by the compiler as a syntax error. Instead, put the comment directly before or after the line it describes, whichever suits you best.

```
(Show the Macros Status Bar set #3)
```

```
MBAR 3
```

By the way, the right parenthesis at the end of a comment is not necessary, but

does makes the source code prettier and more readable.

Label Definitions And References

It is often desirable to branch to different locations in the Procedure File depending on what the remote machine does. We use labels to mark the locations that can be branched to in a Procedure File. A label definition (where execution will branch to) begins with a colon, followed by the label name. The label name may be as long as you like (as long as it follows the 79 character Procedure line limit), but only the first 20 characters are significant. The label name can contain any characters, including spaces, but it should not begin with a space. You'll notice that in all example Source Code listings in this manual, I boldface all Procedure Commands to set them apart from the parameter list. This is not how they'll appear in your Source Code file. Here are two examples of label definitions as they might appear in a Procedure source code file:

(The following line is a label definition)

```
:FIRST  
BELL  
JUMPTO SECOND  
END
```

(The following line is a label definition)

```
:SECOND  
GOSUB THIRD  
END
```

In this example, the labels FIRST and SECOND are defined. Two commands, JUMPTO SECOND and GOSUB THIRD, make references to labels. Notice that the colon is not included in a label reference, only in the label definition. If you've got a sharp eye, you probably realize that this Procedure would not compile correctly because the label THIRD was referenced but never defined. Labels can be defined but not referenced, but the reverse is a no-no.

Parameter Types

There are several different parameter types used by Procedure commands:

Label

A label, as described above, is abbreviated in Procedure command listings as LABEL.

Numeric Constant

A numeric constant is abbreviated in Procedure command listings as NUM_CON. It is a number from -2,147,483,648 to 2,147,483,647. Although this is a huge range of numbers, a numeric constant must be a whole number, not a decimal or fraction.

Examples:

123

10

-200

0

44

Character Constant

A Character constant is abbreviated in Procedure command listings as CHAR_CON. A character constant is a single capital letter from A to Z.

Examples:

A

B

C

D

Z

String Constant

A string constant is abbreviated in Procedure command listings as STR_CON.

A string constant is a group of one or more alphanumeric characters. Typed spaces are considered by the compiler to be a valid part of the string constant.

Examples:

H

HE

HELLO THERE

H1234

23ZZH

Numeric Variable

A numeric variable is abbreviated in Procedure command listings as NUM_VAR. A numeric variable is a capital letter from A to Z followed by a number from 1 to 9 (not zero!) followed by a percent sign. There are therefore 234 available numeric variables. Numeric variables are used to hold a numeric constant, and therefore have the same range as numeric constants of the size of numbers they can hold. Numeric variables retain their values even after a

Procedure stops execution so that they can be used to pass values between different Procedure Files.

Examples:

A1%

B9%

C6%

D3%

Z4%



To maintain upwards compatibility with Procedures created in previous versions of White Knight, you may choose to omit the number from the variable name. If the number is omitted, White Knight will treat it as if there were a "1" there. In other words, A% is the same as A1%, B% is the same as B1%, and C% is the same as C1%. You'll see that I omit the numbers from all of my sample Procedure instructions in this manual as a time-saver.

String Variable

A string variable is abbreviated in Procedure command listings as STR_VAR. A string variable is a capital letter from A to Z followed by a number from 1 to 9 (not zero!) followed by a dollar sign. There are therefore 234 available string variables. String variables are used to hold a string constant or a character constant, and can hold up to 132 characters, which is by no coincidence the size of the longest line White Knight can display in the Data Area of the Terminal Window. There are also 3 special string variables referred to as @ variables, because they are designated as the @ symbol followed by a 0, 1, or 2. @ variables are used by certain Procedure commands for displaying information to the user in a special Status Bar. They are just like string variables, but have a limit of 80 characters. String variables retain their values even after a Procedure stops execution so that they can be used to pass values between different Procedure Files.

Examples:

A1\$

C9\$

D4\$

Z6\$

@0

@1

@2



Like numeric variables, you can omit the number from the variable name (except @ variables) and White Knight will insert a "1" for you). In other words, A\$ is the same as A1\$, and Z\$ is the same as Z1\$. I have taken advantage of this and have omitted numbers from all string variable names in the examples in this manual.

Numeric Expression

A numeric expression is abbreviated in Procedure command listings as NUM_EXP. Anytime you see a numeric expression parameter, it means that you can use your choice of either a numeric constant or a numeric variable for that parameter.

String Expression

A string expression is abbreviated in Procedure command listings as STR_EXP. Anytime you see a string expression parameter, it means that you can use your choice of either a string constant, string variable, @ variable, or character constant.

Numeric Test Operator

A numeric test operator is abbreviated in Procedure command listings as NUM_TEST_OP. They are used only with the TEST Procedure command. There are eight different symbols that can be used where a numeric test operator is specified. They are:

- > (Greater than)
- < (Less than)
- = (Equal to)
- <= (Less than or equal to)
- >= (Greater than or equal to)
- <> (Not equal to)
- & (Binary AND)
- | (Binary OR)

ON Or OFF Expression

An on or off expression is abbreviated in Procedure command listings as ON_OR_OFF_EXP. Where an on or off expression is specified, you should supply either the word ON or the word OFF.

Filetype Expression

A filetype expression is abbreviated in Procedure command listings as FILETYPE_EXP. A filetype expression is used to specify a four character file type as used by the Finder. Therefore, it is four alphanumeric characters, and uppercase and lowercase letters are considered different.

Examples:

AAAA

aaaa

TEXT

MACA

WORD

EDIT

Procedure Command

A Procedure command is abbreviated in Procedure command listings as PROC_CMD. Certain Procedure commands are actually made up of two Procedure commands. If a certain condition exists (depending on the command) the second Procedure command is executed, otherwise it is not executed. For example:

```
IF YES JUMPTO PROCESS
```

is actually constructed of two Procedure commands, IF YES and JUMPTO PROCESS. If the "yes" condition exists (which is explained in more detail later), the JUMPTO PROCESS command is immediately executed.

Flags

There are two internal flags used by some Procedure commands. These flags are used to reflect a certain result of an operation, so that the Procedure execution may branch to perform certain instructions based upon what has happened thus far. The first flag is referred to as the **YES/NO** flag. It is either set to a "YES" condition, or a "NO" condition. The second flag is called the **ERROR** flag. It is either set to an "ERROR" condition, or a "NO ERROR" condition. How and where these commands are used are described in the next chapter in the descriptions of Procedure commands that affect or use them.

Whenever a Procedure File is executed, the state of the flags is undefined (unless you know the state of them as passed from the Phonebook or a previous Procedure). Please refer to Example #9 in the chapter "Procedure Examples" for instructions on how to set the flags to a known state.

How Procedure Commands Are Documented

In the next chapter, each Procedure Command is documented as in the following fictitious command example:

MELTDOWN NUM_VAR,NUM_EXP

Description: This command blows up the Macintosh attached at AppleTalk node number NUM_VAR only if NUM_EXP is a negative number. Otherwise, it merely laughs at you.

Example:

TYPE Thar she blows!

MELTDOWN A%,-23

The first line gives the syntactic description of the command with the command itself in boldface and the parameter list (if any) in plain face type. The parameters are listed in order, with any delimiting characters shown - in the above example, we know that NUM_VAR and NUM_EXP must be separated with a comma. If two or more parameters are of the same type, the parameters are numbered to differentiate them, as in NUM_VAR1 and NUM_VAR2. If a parameter is optional (meaning it can be left out in the source code and White Knight will use a described default value), it is enclosed with square brackets. Next comes a description of what the command does and what each parameter designates. Finally, a short usage example of the command is given.

The Procedure commands are separated into 9 categories grouped according to primary function. The categories are:

- Screen And Keyboard Input And Output
- Variable Manipulation
- File Transfers
- Disk File Manipulation
- Disk File Input And Output
- Printer Manipulation
- Settings Manipulation
- Procedure Execution Modifiers
- User Defined Menus

- User Defined Dialog Box Commands
- User Defined Windows

Final Hints And Tips

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

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

- 1) Not prompting for the correct string ("YOU HAVE MAIL", "You Have Mail" and "you have mail" are three different strings because of the upper and lowercase letters).
- 2) Inserted a leading, trailing, or unneeded space in the string.
- 3) Never receive the string you are prompting for.

In 99% of the cases where somebody says "my Procedure won't work", the problem has been traced to a faulty PROMPT command. Please, please check these very carefully before calling for help. Turn on **Service->Procedure->Monitor Execution** and see that the string you are looking for is in fact being received after the PROMPT command is executed that looks for that string.

Unless a Procedure is ended by a command (such as QUIT, RUN, or DO), it will stop normally after executing the physically last Procedure command line or when an error occurs. Although not all of the Procedure commands are used in the "Procedure File Examples" chapter, each command is given a short usage example in its listing in the "Procedure Commands".

Remember that a string or numeric expression can be either a constant or a variable, but not a combination of both. If you wanted to type a constant, a variable, and finally a carriage return, the command:

```
TYPE This is my name: A$^M
```

would not work. It would need to be broken up into the commands:

```
TYPE This is my name:
```

```
TYPE A$
```

```
TYPE ^M
```

Procedure Compiler Options

A number of things about how White Knight compiles your source code into Procedure Files can be modified. By choosing **Customize->Options->Compiler**, the following dialog box is displayed:

Compiler Options

Compiler encrypts procedure files

Max. number of label definitions:

Max. number of label references:

Show compiler window during compilation

- **Compiler encrypts procedure files:** If this option is checkmarked, a very devious scheme of encryption is used on the Procedure File, which renders the file impervious to the likes of file editing programs and pseudo-hackers searching for password strings in the file. If you choose encryption, the **Service->Procedure->Monitor Execution** choice is uncheckmarked and disabled when the Procedure File is executed. Only explicit **LOUD** Procedure commands will allow tracing of an encrypted Procedure File.
- **Max. number of label definitions:** This value tells White Knight how much memory to set aside to keep track of label definitions while compiling. If you get "ERROR: Out of label definition space" errors from the Procedure compiler, you should increase the number in this value.
- **Max. number of label references:** This value tells White Knight how much memory to set aside to keep track of label references while compiling. If you get "ERROR: Out of label reference space" errors from the Procedure compiler, you should increase the number in this value.
- **Show compiler window during compilation:** Because of the overhead involved in printing each line as it is compiled (as well as its corresponding line number), you can improve the speed of compilation by uncheckmarking this option. When not checkmarked, the compiler window is not shown during compilation, and errors are signalled only by a beep sound. In that event, you could then turn this option back on and then recompile to find out where and what the error was that was signalled.

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



The following sections describe the individual Procedure commands. Each section includes a description of the command, its syntax, and an example of its use. The commands are listed in alphabetical order. The first section is for the 'Add' command, which is used to add a new procedure to a project. The second section is for the 'Delete' command, which is used to delete a procedure from a project. The third section is for the 'Edit' command, which is used to modify an existing procedure. The fourth section is for the 'Run' command, which is used to execute a procedure. The fifth section is for the 'Save' command, which is used to save a procedure. The sixth section is for the 'Undo' command, which is used to revert a procedure to its previous state. The seventh section is for the 'Redo' command, which is used to restore a procedure to its state after an undo. The eighth section is for the 'Copy' command, which is used to copy a procedure to the clipboard. The ninth section is for the 'Paste' command, which is used to paste a procedure from the clipboard. The tenth section is for the 'Print' command, which is used to print a procedure. The eleventh section is for the 'Close' command, which is used to close a procedure. The twelfth section is for the 'Quit' command, which is used to exit the application. The thirteenth section is for the 'Help' command, which is used to access the user manual. The fourteenth section is for the 'About' command, which is used to view information about the application. The fifteenth section is for the 'Preferences' command, which is used to configure the application. The sixteenth section is for the 'Settings' command, which is used to configure the application. The seventeenth section is for the 'Options' command, which is used to configure the application. The eighteenth section is for the 'Tools' command, which is used to access the application's tools. The nineteenth section is for the 'Windows' command, which is used to manage the application's windows. The twentieth section is for the 'View' command, which is used to manage the application's view. The twenty-first section is for the 'Edit' command, which is used to manage the application's editing. The twenty-second section is for the 'Format' command, which is used to manage the application's formatting. The twenty-third section is for the 'Style' command, which is used to manage the application's style. The twenty-fourth section is for the 'Theme' command, which is used to manage the application's theme. The twenty-fifth section is for the 'Language' command, which is used to manage the application's language. The twenty-sixth section is for the 'Locale' command, which is used to manage the application's locale. The twenty-seventh section is for the 'Region' command, which is used to manage the application's region. The twenty-eighth section is for the 'Currency' command, which is used to manage the application's currency. The twenty-ninth section is for the 'Date' command, which is used to manage the application's date. The thirtieth section is for the 'Time' command, which is used to manage the application's time. The thirty-first section is for the 'Number' command, which is used to manage the application's number. The thirty-second section is for the 'Text' command, which is used to manage the application's text. The thirty-third section is for the 'Image' command, which is used to manage the application's image. The thirty-fourth section is for the 'Audio' command, which is used to manage the application's audio. The thirty-fifth section is for the 'Video' command, which is used to manage the application's video. The thirty-sixth section is for the 'Network' command, which is used to manage the application's network. The thirty-seventh section is for the 'Security' command, which is used to manage the application's security. The thirty-eighth section is for the 'Performance' command, which is used to manage the application's performance. The thirty-ninth section is for the 'Stability' command, which is used to manage the application's stability. The fortieth section is for the 'Reliability' command, which is used to manage the application's reliability. The forty-first section is for the 'Scalability' command, which is used to manage the application's scalability. The forty-second section is for the 'Flexibility' command, which is used to manage the application's flexibility. The forty-third section is for the 'Interoperability' command, which is used to manage the application's interoperability. The forty-fourth section is for the 'Compatibility' command, which is used to manage the application's compatibility. The forty-fifth section is for the 'Portability' command, which is used to manage the application's portability. The forty-sixth section is for the 'Accessibility' command, which is used to manage the application's accessibility. The forty-seventh section is for the 'Usability' command, which is used to manage the application's usability. The forty-eighth section is for the 'Learnability' command, which is used to manage the application's learnability. The forty-ninth section is for the 'Efficiency' command, which is used to manage the application's efficiency. The fiftieth section is for the 'Effectiveness' command, which is used to manage the application's effectiveness. The fifty-first section is for the 'Productivity' command, which is used to manage the application's productivity. The fifty-second section is for the 'Quality' command, which is used to manage the application's quality. The fifty-third section is for the 'Value' command, which is used to manage the application's value. The fifty-fourth section is for the 'Cost' command, which is used to manage the application's cost. The fifty-fifth section is for the 'Risk' command, which is used to manage the application's risk. The fifty-sixth section is for the 'Return' command, which is used to manage the application's return. The fifty-seventh section is for the 'Investment' command, which is used to manage the application's investment. The fifty-eighth section is for the 'Profit' command, which is used to manage the application's profit. The fifty-ninth section is for the 'Loss' command, which is used to manage the application's loss. The sixtieth section is for the 'Gain' command, which is used to manage the application's gain. The sixty-first section is for the 'Revenue' command, which is used to manage the application's revenue. The sixty-second section is for the 'Expense' command, which is used to manage the application's expense. The sixty-third section is for the 'Income' command, which is used to manage the application's income. The sixty-fourth section is for the 'Outflow' command, which is used to manage the application's outflow. The sixty-fifth section is for the 'Inflow' command, which is used to manage the application's inflow. The sixty-sixth section is for the 'Asset' command, which is used to manage the application's asset. The sixty-seventh section is for the 'Liability' command, which is used to manage the application's liability. The sixty-eighth section is for the 'Equity' command, which is used to manage the application's equity. The sixty-ninth section is for the 'Debt' command, which is used to manage the application's debt. The seventieth section is for the 'Capital' command, which is used to manage the application's capital. The seventy-first section is for the 'Equity' command, which is used to manage the application's equity. The seventy-second section is for the 'Debt' command, which is used to manage the application's debt. The seventy-third section is for the 'Capital' command, which is used to manage the application's capital. The seventy-fourth section is for the 'Equity' command, which is used to manage the application's equity. The seventy-fifth section is for the 'Debt' command, which is used to manage the application's debt. The seventy-sixth section is for the 'Capital' command, which is used to manage the application's capital. The seventy-seventh section is for the 'Equity' command, which is used to manage the application's equity. The seventy-eighth section is for the 'Debt' command, which is used to manage the application's debt. The seventy-ninth section is for the 'Capital' command, which is used to manage the application's capital. The eightieth section is for the 'Equity' command, which is used to manage the application's equity. The eighty-first section is for the 'Debt' command, which is used to manage the application's debt. The eighty-second section is for the 'Capital' command, which is used to manage the application's capital. The eighty-third section is for the 'Equity' command, which is used to manage the application's equity. The eighty-fourth section is for the 'Debt' command, which is used to manage the application's debt. The eighty-fifth section is for the 'Capital' command, which is used to manage the application's capital. The eighty-sixth section is for the 'Equity' command, which is used to manage the application's equity. The eighty-seventh section is for the 'Debt' command, which is used to manage the application's debt. The eighty-eighth section is for the 'Capital' command, which is used to manage the application's capital. The eighty-ninth section is for the 'Equity' command, which is used to manage the application's equity. The ninetieth section is for the 'Debt' command, which is used to manage the application's debt. The ninety-first section is for the 'Capital' command, which is used to manage the application's capital. The ninety-second section is for the 'Equity' command, which is used to manage the application's equity. The ninety-third section is for the 'Debt' command, which is used to manage the application's debt. The ninety-fourth section is for the 'Capital' command, which is used to manage the application's capital. The ninety-fifth section is for the 'Equity' command, which is used to manage the application's equity. The ninety-sixth section is for the 'Debt' command, which is used to manage the application's debt. The ninety-seventh section is for the 'Capital' command, which is used to manage the application's capital. The ninety-eighth section is for the 'Equity' command, which is used to manage the application's equity. The ninety-ninth section is for the 'Debt' command, which is used to manage the application's debt. The one hundredth section is for the 'Capital' command, which is used to manage the application's capital.

Procedure Commands

Screen And Keyboard Input And Output

@ ON_OR_OFF_EXP

Description: The "@" command turns on or off, according to ON_OR_OFF_EXP the display of the Special Status Bar. The "@ ON" command does not show the contents of the three @ variables, it simply prepares for the "SHOW@" command by saving internally the contents of the Status Bar before the "@ ON" command was executed. After the "SHOW@" command is executed, the Special Status Bar displays in white text on a black background from top to bottom the three @ variables: @0, @1, and @2. It's useful for displaying pertinent information to the user during the progress of a Procedure, as a dialog box would stop execution and wait for a user response. After the "@ ON" command is executed, the previous Status Bar contents can be restored by ending or cancelling the Procedure, or by executing an "@ OFF" command.

Example: Refer to Example #1 in the chapter "Procedure Examples" for sample usage of "@ ON", "@ OFF", and "SHOW@".

SHOW@

Description: The "SHOW@" command is also used whenever you change the contents of one or more of the three @ variables, and wish to have the Special Status Bar updated with the new messages. In other words, just changing the contents of an @ variable does not automatically change what is shown in the Special Status Bar, an explicit "SHOW@" command must be executed for the new contents to be displayed.

Example: Refer to Example #1 in the chapter "Procedure Examples" for sample usage of "@ ON", "@ OFF", and "SHOW@".

ALERT1 STR_EXP/PROC_CMD

ALERT2 STR_EXP/PROC_CMD

ALERT3 STR_EXP/PROC_CMD

Description: These three commands let you look for up to three different strings of characters (specified by the STR_EXP parameter) to be received, and if and when they are, the Procedure command PROC_CMD is

immediately executed. STR_EXP must be less than 20 characters and upper and lowercase letters are considered different and must match between STR_EXP and what is received. These commands do not hold up execution of the Procedure while waiting for the string to come over the modem like the "PROMPT" and "PROMPT ^" commands. Control characters cannot be embedded in STR_EXP. PROC_CMD is a full and valid Procedure command to be executed immediately when STR_EXP is received. Anytime an "ALERT" or "PROMPT" STR_EXP is received or a "PROMPT ^" CHAR_CON is received, all active "ALERT", "PROMPT", and "PROMPT ^" commands are disabled. There is a single space between the "ALERT" number (1, 2, or 3) and STR_EXP, but no spaces between STR_EXP, the slash character, and PROC_CMD. Don't confuse the slash character "/", with the backslash character "\".

Example: Refer to Example #2 in the chapter "Procedure File Examples" for sample usage of "ALERT1".

BELL

Description: Causes the Macintosh to emit a short beep sound. Useful to alert you audibly during various stages of a Procedure execution.

Example:

```
TYPE One ringy-dingy!^M
BELL
```

CLEAR

Description: Clears all lines of text in the Data Area of the Terminal Window.

Example:

```
COMM 300-N-8-1-FULL
DIAL ATDT 555-1212
PROMPT CONNECT
CLEAR
```

DIAL STR_EXP

REDIAL STR_EXP

Description: These commands work just like the **Service->Dial Or**

Redial menu choice. **DIAL** dials the number only once, and **REDIAL** dials the number continuously until the redial limit is reached or a connection is made. You can embed control characters in **STR_EXP** just as you would in a Macro Key, but you don't need to put a ^M at the end (the carriage return is supplied automatically). **STR_EXP** must be 70 characters long or less. The number to be dialed must be preceded by a modem dialing command (i.e. "ATDT" if you are doing touch-tone dialing, or "ATDP" if you are doing pulse dialing). You can tell the result of a **DIAL** or **REDIAL** command by examining the state of the YES/NO and ERROR flags immediately after the **DIAL** or **REDIAL** instruction:

YES/NO flag = YES and ERROR flag = NO ERROR: Connection established.

YES/NO flag = NO and ERROR flag = NO ERROR: Cancelled by user.

YES/NO flag = NO and ERROR flag = ERROR: Didn't connect (redial limit exceeded or dialed once and didn't connect).

YES/NO flag = YES and ERROR flag = ERROR: Modem not responding.

Example:

(Set the serial port)

COMM 300-N-8-1-FULL

(Dial the number)

DIAL ATDT 555-1212

INITMODEM

Description: This command is equivalent to choosing **Service->Modem->Initialize**. The ERROR flag is set to "NO ERROR" if the modem responded properly, or is set to "ERROR" if the modem didn't respond.

Example:

(Initialize the modem)

INITMODEM

(If there was an error, get out of here)

IF ERROR JUMPTO CANTDIAL

HANGUP

Description: This command is equivalent to choosing **Service->Modem->Hang Up**.

Example:

(Hang up the phone)

HANGUP

AUTOANSWER ON_OR_OFF_EXP

Description: "AUTOANSWER ON" is equivalent to choosing Service->Modem->Auto-Answer. "AUTOANSWER OFF" is equivalent to choosing Service->Modem->Turn Off Auto-Answer. If the modem responded properly, the ERROR flag is set to "NO ERROR", otherwise it's set to "ERROR".

Example:

(Turn on auto-answer to accept future incoming calls)

AUTOANSWER ON

USEDRIVER STR_EXP

Description: This command tells White Knight to load and use the Modem Driver File specified in "STR_EXP".

Example:

(Load and use the Modem Driver File named "HAYES.DVR")

USEDRIVER HAYES.DVR

LONG BREAK

SHORT BREAK

Description: The "LONG BREAK" command is used to send a long (3 1/2 to 4 second) modem break signal, and is functionally equivalent to a "TYPE ^@" command. The "SHORT BREAK" command is used to send a short (233 millisecond) modem break signal, and is functionally equivalent to a "TYPE ^!" command.

Example:

(Send a short break signal)

SHORT BREAK

GETLINE NUM_EXP,STR_VAR

Description: The "GETLINE" command is used to read a line of text directly from the Data Area of the Terminal Window into a string variable. NUM_EXP must be a positive number from 1 to 24, and is the line number of the last 24 lines received (the most recently received line being line number 24) you wish to read into the string variable STR_VAR.

Example:

(Read in line #1 to variable A\$)

```
GETLINE 1,A$
```

PROMPT ^CHAR_CON

Description: The "PROMPT ^" command works just like the regular "PROMPT" command, except that it is used to look for a single control character to come over the serial port, rather than a string of characters. CHAR_CON is an uppercase letter from A to Z, representing the control character you wish to look for. Execution of the Procedure stops until the control character is received. Once the control character is received, execution of the Procedure continues normally.

Example:

(Wait for a carriage return (CTRL-M) before continuing)

```
PROMPT ^M
```

(Some systems choke if we type something right after they send data,)

(so we'll put a short PAUSE here to let 'em settle down)

```
PAUSE
```

```
TYPE And away we go...
```

PROMPT STR_EXP

Description: The "PROMPT" command is used when you want a Procedure to hold up until a certain string of characters (designated in STR_EXP) to be received before the next command is executed. STR_EXP can be up to 20 characters long. Upper and lower case are considered significant, and must match between STR_EXP and the string of characters received in order to be recognized as a match. Control characters may not be

embedded in STR_EXP (use the **PROMPT** ^).

Example:

(Dial a phone number, wait for the CONNECT message from my modem, and)
(then type my password and a carriage return)

DIAL ATDT 555-1212

PROMPT CONNECT

(It's always a good idea to put a short delay in before typing)

PAUSE

TYPE My-Password^M

TYPE STR_EXP

Description: The "**TYPE**" command is used to simulate typing on your keyboard. In other words, if you want the Procedure to send a certain string of characters, you put that string of characters in STR_EXP. STR_EXP can be a mixture of regular and control characters, just as is allowed in a Macro Key. Four special character strings can be placed in STR_EXP to yield the following functions:

<u>Sequence</u>	<u>Function</u>
^\$	Sends a DEL character (ASCII code 127)
^!	Sends a short modem break signal
^@	Sends a long modem break signal
^#	Inverts the DTR serial port line for 1 second
^[Sends an ESCAPE (ASCII code 27) character

These sequences don't have to appear alone, they can be embedded within a STR_EXP's characters.

Example:

(Type the string HELLO followed by a carriage return (CTRL-M))

TYPE HELLO^M

(And send a short modem break signal)

TYPE ^!

WATCH ON_OR_OFF_EXP

Description: The "**WATCH**" command is used to turn the mouse cursor into a watch cursor, which is the standard Macintosh indicator that a certain function will require the user to wait. Because other parts of White Knight can take the watch cursor "out from under you" in changing it

back to other things (like the I-beam cursor in the Buffered Keyboard or the normal arrow cursor if the mouse is moved over the Data Area of the Terminal Window), it is recommended that the "WATCH" command be used only when your Procedure is in a tight loop, as during a locked condition (see the "LOCK" command below) or during a compilation (see the "COMPILE LOUD" and "COMPILE QUIET" commands below).

Example:

(Turn the cursor into a watch during a compilation)

WATCH ON

COMPILE QUIET MYDISK:MYPROCEDURE

(When we get to the next instruction, we know that the compile has)

(finished, so turn the cursor back into an arrow)

WATCH OFF

Variable Manipulation

ADD NUM_VAR,NUM_EXP

Description: The "ADD" command is used to add the value NUM_EXP to the value held in NUM_VAR. Only NUM_VAR is affected by this command. If NUM_EXP is a numeric variable, that variable is not changed.

Example:

(Add 15 to the numeric variable Z%)

ADD Z%,15

AND NUM_VAR,NUM_EXP

Description: The "AND" command performs a binary AND operation on NUM_VAR and NUM_EXP, and places the result in NUM_VAR. Only NUM_VAR is affected by this command.

Example:

(AND in the value 256 to the numeric variable M%)

AND M%,256

CONCAT STR_VAR,STR_EXP

Description: This command appends (concatenates) the string of characters in STR_EXP to the end of STR_VAR. Only STR_VAR is affected by this command. If the resulting concatenation would make STR_VAR longer than 132 characters, any characters past the 132nd are lopped off into oblivion.

Example:

COPYINTO A\$,HELLO

(A\$ now contains the word "HELLO")

CONCAT A\$, THERE

(Notice the space between the comma and THERE - A\$ now contains)
(the string "HELLO THERE")

CONTAINS STR_VAR,STR_EXP

Description: This command tests to see whether STR_VAR contains the characters in STR_EXP. If it does, it sets the YES/NO flag to "YES", otherwise it sets the YES/NO flag to "NO". It does not change either STR_VAR or STR_EXP. Upper and lowercase letters are considered different.

Example:

(Does the string variable B\$ contain the word HELLO?)

CONTAINS B\$,HELLO

(If it does, ring the bell)

IF YES BELL

CONVUP STR_VAR

Description: This command converts any lowercase letters in STR_VAR to their uppercase equivalent.

Example:

COPYINTO A\$,Hello There

(The string variable A\$ now contains the string "Hello There". Let's)
(convert the lowercase letters to uppercase)

CONVUP A\$

(A\$ now contains the string "HELLO THERE")

COPYINTO STR_VAR,STR_EXP

Description: This command replaces the current contents of STR_VAR with the contents of STR_EXP. Only STR_VAR is changed by this command.

Example:

(Put the string of characters "HELLO" into the string variable A\$)

COPYINTO A\$,HELLO

(And copy that string into the string variable B\$)

COPYINTO B\$,A\$

DIVIDE NUM_VAR,NUM_EXP

Description: This command divides NUM_VAR by NUM_EXP and places the result in NUM_VAR. Only NUM_VAR is changed by this command. If a division by zero would result, no division is done and the ERROR flag is set to "ERROR". Otherwise, the division is done and the ERROR flag is set to "NO ERROR".

Example:

(Put the value 15 into the numeric variable A%)

LET EQUAL A%,15

(And divide A% by 3)

DIVIDE A%,3

(A% contains the value 5)

ELAPSED NUM_VAR

Description: This command calculates the number of seconds that have elapsed since the last "SAVETIME" command was executed, and places that number in NUM_VAR.

Example:

(Save the current time internally)

SAVETIME

(Now introduce a pause of 6 seconds)

PAUSE 360

(And calculate the elapsed time)

ELAPSED A%

(The numeric variable A% should now hold 6)

EMPTY STR_VAR

Description: This command tests to see if STR_VAR is empty (contains no characters). If it is, it sets the YES/NO flag to "YES". If STR_VAR contains characters, the YES/NO command is set to "NO". STR_VAR is unaffected by this command.

Example:

(Erase all characters from the string variable A\$)

ERASE A\$

(Trust me, it is - but let's go ahead and test that it's really empty).

EMPTY A\$

IF YES BELL

IF NO TYPE He lied! He lied!

ERASE STR_VAR

ERASE ALL

Description: The command "**ERASE ALL**" will destroy the contents and leave empty all string variables (including the @ variables). To erase individual string or @ variables, use the "**ERASE**" command in conjunction with the appropriate STR_VAR parameter.

Example:

(Erase all string variables)

ERASE ALL

(Erase only the M\$ string variable)

ERASE M\$

FILL\$ STR_VAR,NUM_EXP,STR_EXP

Description: The "**FILL\$**" command replaces the contents of STR_VAR with NUM_EXP number of the first character in STR_EXP. Only the contents of STR_VAR is changed by the "**FILL\$**" command.

Example:

(Put the string "HELLO" into the string variable A\$)

COPYINTO A\$,HELLO

(Erase the contents of B\$)

ERASE B\$

(Now put 15 of the first character of A\$ into B\$)

FILL\$ B\$,15,A\$

(A\$ now contains the string "HHHHHHHHHHHHHHHH")

GETGLOBAL NUM_VAR, NUM_EXP

Description: The "GETGLOBAL" command is used to obtain the value of certain special internal variables not accessible in any other manner. These variables are called Global Variables, and are accessed by placing the desired Global Variable number (from 0 to 11) into NUM_EXP. The value of that Global Variable is then copied into NUM_VAR. The Global Variables and values they return are:

<u>Global #</u>	<u>Function</u>
0	Current emulation (0 = TTY, 1 = VT100, 2 = VT52).
1	Current cursor row (1 - 24).
2	Current cursor column (1 - 132).
3	Current duplex (0 = full, 1 = half, 2 = echo, 3 = null).
4	Number of 60'ths of a second elapsed since Macintosh was started up.
5	Current month (1 - 12).
6	Current day (1 - 31).
7	Current year (1904 - ???).
8	Current hour (0 - 23).
9	Current minute (0 - 59).
10	Current second (0 - 59).
11	Current day of week (1-7: Sunday = 1).

Example:

(Get the day of the week global)

GETGLOBAL A%,11

(Find out if today is Monday -> Is A% = 2?)

TEST A% = 2

(If it is, bang a gong)

IF YES BELL

GETCOST NUM_VAR

Description: This value gets the number of cents currently displayed in the billing clocks (in the General and Macros Status Bars) and returns that value in NUM_VAR.

Example:

(How much have we spent?)

GETCOST A%

(A% now holds the amount of money spent in cents)

INSTR NUM_VAR, STR_VAR, NUM_EXP, STR_EXP

Description: This command returns in NUM_VAR the character position (from 1 to the length of STR_VAR) that the string STR_EXP is first found in STR_VAR. The search for STR_EXP begins at position NUM_EXP in STR_VAR. If the string isn't found, NUM_VAR is set to zero. Upper and lower case are considered different in the search.

Example:

COPYINTO A\$, THE QUICK BROWN FOX IS EVEN QUICKER TODAY

(Find the first position of the string "QUICK" in A\$)

INSTR A%, A\$, 1, QUICK

(A% now equals 5)

(Get past first QUICK by choosing a starting character as 6, and)

(find position of the second QUICK string)

INSTR A%, A\$, 6, QUICK

(A% now equals 29)

LEFT\$ STR_VAR, NUM_EXP, STR_EXP

Description: This command copies the first (leftmost) NUM_EXP characters in STR_EXP into STR_VAR. Only STR_VAR is affected by this command.

Example:

(Copy the string "HELLO" into A\$)

COPYINTO A\$, HELLO

(Get the first 3 characters in A\$ and put them in B\$)

LEFT\$ B\$, 3, A\$

(B\$ now contains the string "HEL")

LENGTH NUM_VAR, STR_EXP

Description: The "LENGTH" command calculates the number of characters in STR_EXP and returns that number in NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Copy the string "HELLO" into A\$)

```
COPYINTO A$,HELLO
```

(Now put the size of A\$ into A%)

```
LENGTH A%,A$
```

(A% now equals 5)

```
LET EQUAL NUM_VAR,NUM_EXP
```

Description: This command takes the value NUM_EXP and copies it into the numeric variable NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Put the value 535 into the numeric variable Z%)

```
LET EQUAL Z%,535
```

```
LOADVAR STR_EXP
```

Description: This command loads from disk a set of saved variables that were saved with a previous "SAVEVAR" command. The contents of all string and numeric variables are destroyed in favor of the new values. The @ variables, however, are unchanged by this command. STR_EXP is the filename of the file created with the "SAVEVAR" command. WARNING: Do not use the "LOADVAR" command with any other file than one created with the "SAVEVAR" command!

```
MID$ STR_VAR,NUM_EXP1,NUM_EXP2,STR_EXP
```

Description: This command copies NUM_EXP2 number of characters from STR_EXP starting at position NUM_EXP1 into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the string "HELLO" into A\$)

```
COPYINTO A$,HELLO
```

(Copy the first 2 characters starting at position 3 of A\$ into B\$)

```
MID$ B$,3,2,A$
```

(B\$ now contains the string "LL")

MULTIPLY NUM_VAR, NUM_EXP

Description: This command multiplies NUM_VAR by NUM_EXP and places the result in NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Put the value 5 into the numeric variable L%)

LET EQUAL L%, 5

(And multiply L% by 3)

MULTIPLY L%, 3

(L% now equals 15)

NUMTOSTRING NUM_VAR, STR_VAR

Description: This command converts the number NUM_VAR to a string of characters and copies that string into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the value -53 into the numeric variable M%)

LET EQUAL M%, -53

(Convert M% to a string and put that in the string variable A\$)

NUMTOSTRING M%, A\$

(A\$ now contains the string "-53")

OR NUM_VAR, NUM_EXP

Description: The "OR" command performs a binary OR operation on NUM_VAR and NUM_EXP, and places the result in NUM_VAR. Only NUM_VAR is affected by this command.

Example:

(OR in the value 256 to the numeric variable M%)

OR M%, 256

REPLACES STR_VAR, NUM_EXP1, NUM_EXP2, STR_EXP

Description: This command replaces NUM_EXP2 number of characters in STR_VAR starting at position NUM_EXP1 with the contents of STR_EXP.

Only STR_VAR is changed by this command. If the replacement causes STR_VAR to be longer than 132 characters, all characters past the 132nd in STR_VAR are lopped off into oblivion.

Example:

(Put the string "HELLO" into the string variable A\$)

COPYINTO A\$,HELLO

(Replace 2 characters in A\$ starting at character #3 with)

(the string "FOOBAR")

REPLACE\$ A\$,3,2,FOOBAR

(A\$ now contains the string "HEFOOBARO")

RIGHT\$ STR_VAR,NUM_EXP,STR_EXP

Description: This command copies NUM_EXP number of the last (rightmost) characters in STR_EXP into STR_VAR. Only STR_VAR is changed by this command.

Example:

(Put the string HELLO into the string variable A\$)

COPYINTO A\$,HELLO

(Copy the last 3 characters in A\$ to B\$)

RIGHT\$ B\$,3,A\$

(B\$ now contains the string "LLO")

SAVETIME

Description: This command saves the current time in an internal variable. It is used in conjunction with a later "**ELAPSED**" command, to calculate how many seconds have elapsed between two points in time.

Example:

(Save the current time)

SAVETIME

(Now wait until 6:00:00 AM)

WAIT 06:00:00

(And put the number of seconds that elapsed between the SAVETIME)

(command and 6:00:00 AM in the numeric variable A%)

ELAPSED A%

SAVEVAR STR_EXP

Description: This command saves the contents of all numeric and string variables (except for the three @ variables) into a disk file designated by the filename contained in STR_EXP. The variables can be later restored via a "LOADVAR" command. This command does not change the current state of any variables.

Example:

(EEK! I've used up all my variables and need one more numeric variable)
(for a temporary calculation. Better save off my variables, first)

SAVEVAR MYDISK:THEVARS

(Now I can do some temporary work with whatever variables I want)
(without having to worry about destroying anything important)

LET EQUAL A%,5

(And so forth...)

(Now let's bring back my original variables)

LOADVAR MYDISK:THEVARS

(And get rid of the temporary file)

DELETE MYDISK:THEVARS

STRINGTONUM STR_VAR,NUM_VAR

Description: This command converts the string of characters in STR_VAR to a numeric equivalent and places that number into NUM_VAR. Only NUM_VAR is changed by this command. If STR_VAR contains characters of a non-numeric nature, NUM_VAR will be set to zero.

Example:

(Copy the string "12345" into the string variable A\$)

COPYINTO A\$,12345

(Convert A\$ to a number and put that number in the numeric variable A%)

STRINGTONUM A\$,A%

(A% now equals 12345)

SUBTRACT NUM_VAR,NUM_EXP

Description: This command subtracts NUM_EXP from NUM_VAR and puts the result in NUM_VAR. Only NUM_VAR is changed by this command.

Example:

(Put the value 15 into the numeric variable A%)

```
LET EQUAL A%,15
```

(And subtract 10 from A%)

```
SUBTRACT A%,10
```

(A% now equals 5)

TEST NUM_VAR,NUM_TEST_OP,NUM_EXP

Description: This command does an arithmetic test between NUM_VAR and NUM_EXP (the kind of test is designated by NUM_TEST_OP) and sets the YES/NO flag based on the result of the test. If the test is true, the YES/NO flag is set to "YES", otherwise, the YES/NO flag is set to "NO". Neither NUM_VAR nor NUM_EXP is changed by this command. NOTE: you can replace the commas before and after NUM_TEST_OP with spaces for readability purposes as I've done in all examples in this chapter.

Example:

(Put the value 15 into A%)

```
LET EQUAL A%,15
```

(Is 15 less than 25?)

```
TEST A% < 25
```

(We know it is, so...)

```
IF YES BELL
```

TIMEDATE STR_VAR

Description: This command takes the current time and date, converts both into a string of characters, and copies that string into STR_VAR. The string is in proper international format that your System software has been configured for (don't expect any digit or character to be in any given position!).

Example:

(Get the current time and date into the string variable A\$)

```
TIMEDATE A$
```

(A\$ might now hold something like "06/16/87 23:49:26")

INBUFFER

Description: This command will set the YES/NO flag to YES if there is at least one byte in the serial port buffer waiting to be read. If the serial port buffer is empty, the YES/NO flag is set to NO. This command is typically used in conjunction with the "FETCHBYTE" command, and it should only be used after a "LOCK ON" command to prevent White Knight's terminal emulator from "stealing" the incoming data out from under your Procedure.

Example:

LOCK ON

(Is there a byte waiting in the serial port buffer?)

:WAIT

INBUFFER

(If there is, fall through to my routine that reads the byte and reacts) (to it. Otherwise, wait for it to arrive. A tight loop like this is) (dangerous, by the way, since the it totally locks up the Mac, but) (sometimes there's no way around that.)

IF NO JUMPTO WAIT

(Read the byte here into A\$

FETCHBYTE A\$

FETCHBYTE STR_VAR

Description: This command sets the YES/NO flag like the **INBUFFER** command, but if a byte is available, one byte is read and is copied (erasing any old contents) to the string variable specified in "STR_VAR".

Example:

See example under "**INBUFFER**"

BYTEVAL STR_VAR,NUM_EXP,NUM_VAR

Description: This command gets the ASCII value of byte number "NUM_EXP" contained in "STR_VAR" and places that value in "NUM_VAR". If "NUM_EXP" is larger than the actual length of "STR_VAR", the returned value is garbage and shouldn't be used.

Example:

(Put the string "HELLO" into A\$)

COPYINTO A\$,HELLO

(Tell me what the ASCII value is of the second byte in A\$)

BYTEVAL A\$,2,A%

(A% now equals 69, which is the ASCII code for "E")

BYTEADD NUM_EXP,STR_VAR

Description: This command concatenates a byte with the ASCII value specified in "NUM_EXP" to the end of the string variable specified in "STR_VAR".

Example:

(Copy the string "HELL" into A\$)

COPYINTO A\$,HELL

(Now, add the byte with the ASCII code 79 to the end of A\$)

BYTEADD 79,A\$

(A\$ now contains "HELLO")

File Transfers

All file transfer commands set the Error flag to either "ERROR" or "NO ERROR" to indicate the failure or success of the operation performed.

CLOSE

Description: This command is used to terminate the file received started with a **RECA** or **RECN** command. If a **CLOSE** command is executed when a file capture is not happening, this command does nothing.

Example:

(Append all incoming data to the file MAIL on the disk MYDISK)

RECA MYDISK:MAIL

(And keep capturing until we receive the string "END OF MAIL")

PROMPT END OF MAIL

(Now close the capture file)

CLOSE

RECN STR_EXP

Description: This command is equivalent to choosing **File->File**

Capture->New. To close the file, the **CLOSE** Procedure command is used. **STR_EXP** is the filename that data is captured to. If a file by that name exists, it is destroyed.

RECA STR_EXP

Description: This command is equivalent to choosing **File->File Capture->Append To**. To close the file, the **CLOSE** Procedure command is used. **STR_EXP** is the filename that data is appended to the end of. If the file doesn't exist, it is created.

SUSPEND ON_OR_OFF_EXP

Description: This command is equivalent to choosing **File->File Capture->Pause**. "**SUSPEND ON**" is equivalent to checkmarking that choice, and "**SUSPEND OFF**" is equivalent to uncheckmarking it.

Example:

(I want to capture everything between the strings "BEGIN" and "END")
(I'll do a LOCK ON and LOCK OFF to sandwich the RECA and)
(SUSPEND ON so I don't get any unwanted data.)

LOCK ON

RECA Capture File

SUSPEND ON

LOCK OFF

PROMPT BEGIN

SUSPEND OFF

PROMPT END

CLOSE

RECK STR_EXP

Description: This command is equivalent to choosing **File->Receive File Using->Kermit Protocol**. **STR_EXP** is the filename that the data is received to using the Kermit protocol.

Example:

(Receive the file JUNK to the disk MYDISK using Kermit protocol)

RECK MYDISK:JUNK

RECX STR_EXP

Description: This command is equivalent to choosing **File->Receive File Using->XMODEM Protocol**. STR_EXP is the filename that data is received to using the XMODEM protocol.

Example:

(Receive the file JUNK to the disk MYDISK using XMODEM protocol)

RECX MYDISK:JUNK

RECY STR_EXP

Description: This command is equivalent to choosing **File->Receive File Using->YMODEM Protocol**. STR_EXP must be present but is ultimately ignored (since the YMODEM protocol itself will supply a filename).

Example:

(Receive a batch of files - there doesn't have to be a disk)

(or folder named Junk!)

RECY Junk

RECZ STR_EXP

Description: This command is equivalent to choosing **File->Receive File Using->ZMODEM Protocol**. STR_EXP must be present but is ultimately ignored (since the ZMODEM protocol itself will supply a filename).

Example:

(Receive a batch of files - there doesn't have to be a disk)

(or folder named Junk!)

RECZ Junk

RECF STR_EXP

Description: This command is equivalent to choosing **File->Receive File Using->Flash Protocol**. STR_EXP must be present but is ultimately ignored (since the Flash protocol itself will supply a

filename).

Example:

(Receive a batch of files - there doesn't have to be a disk)
(or folder named Junk!)

RECF Junk

SENDA STR_EXP

Description: This command is equivalent to choosing **File->Send TEXT File**. STR_EXP is the filename of the file to send.

Example:

(Send the text file named junk)

SENDA Junk

SENDK STR_EXP

Description: This command is equivalent to choosing **File->Send File Using->Kermit Protocol**. STR_EXP is the filename of the file to send using the Kermit protocol.

Example:

(Send the file named Junk using Kermit)

SENDK Junk

SENDX STR_EXP

Description: This command is equivalent to choosing **File->Send File Using->XMODEM Protocol**. STR_EXP is the filename of the file to send using the XMODEM protocol.

Example:

(Send the file named Junk using XMODEM)

SENDX Junk

SENDY STR_EXP

Description: This command is equivalent to choosing **File->Send File Using->YMODEM Protocol**. STR_EXP is the name of the file (or batch file) to send using the YMODEM protocol.

Example:

(Send the file named Junk using YMODEM)

SENDY Junk

SENDZ STR_EXP

Description: This command is equivalent to choosing **File->Send File Using->ZMODEM Protocol**. STR_EXP is the name of the file (or batch file) to send using the ZMODEM protocol.

Example:

(Send the file named Junk using ZMODEM)

SENDZ Junk

SENDF STR_EXP

Description: This command is equivalent to choosing **File->Send File Using->Flash Protocol**. STR_EXP is the name of the file (or batch file) to send using the Flash protocol.

Example:

(Send the file named Junk using Flash protocol)

SENDF Junk

SENDCLIP

Description: This command is equivalent to choosing **Edit->Paste**. It sends the contents of the Clipboard to the serial port.

Disk File Manipulation

COMPILE LOUD STR_EXP

COMPILE QUIET STR_EXP

Description: The "**COMPILE LOUD**" and "**COMPILE QUIET**" commands are used to compile a Procedure source code file into an executable Procedure File. "**COMPILE LOUD**" shows the Compiler Window during the compilation, while "**COMPILE QUIET**" does not. It is suggested that you consider using the "**WATCH**" command to change the mouse cursor into a watch (and later back to the normal arrow) before doing a "**COMPILE**

QUIET" in order to notify the user of a brief delay. The compiled Procedure File is given the same filename as the source code file, with the suffix ".PROC" appended. Both "**COMPILE LOUD**" and "**COMPILE QUIET**" set the error flag to "ERROR" if an error occurs during the compilation, otherwise, the error flag is set to "NO ERROR". If an error occurs, the variable E% will contain the line number in which the error occurred.

DELETE STR_EXP

Description: This command is used to delete a disk file. STR_EXP is the filename of the file to delete. No confirmation of the deletion is done, so please be sure that STR_EXP is a file that you are sure you wish to destroy.

Example:

(Delete the file "JUNK" in the same folder as White Knight)

DELETE :Junk

GETALL

GETSELECT FILETYPE_EXP

Description: The "**GETALL**" and "**GETSELECT**" commands are used to specify what kinds of files the "**GETFILE**" command will allow the user to select from in the standard file selection dialog box. "**GETALL**" (the default state White Knight starts up with) means that all file types are acceptable. "**GETSELECT**" allows you to limit the files that may be chosen to those of type FILETYPE_EXP.

Example:

(Allow the user to choose any file on the disk)

GETALL

GETFILE A\$

(Limit him to choosing only files of type "TEXT")

GETSELECT TEXT

GETFILE A\$

GETFILE STR_VAR

Description: This command puts up the standard file selection dialog box for the user to choose a file. The filename is returned in STR_VAR. Since the filename must fit in a string variable, the dialog won't let the user choose a file which has a filename of more than 132 characters. If the user clicks on the "Select" button, the YES/NO flag is set to "YES". If the user clicks on the "Cancel" button, the YES/NO flag is set to "NO".

Example:

(We want the user to choose a TEXT file, so...)

GETSELECT TEXT

(Now put up the standard file selection dialog box)

GETFILE A\$

(If he clicks on the "Cancel" button instead of choosing a file, the YES/NO flag is set to "NO", and we'll just quit)

IF NO END

(Otherwise, the full filename he chose is now in the A\$ string variable)

PUTFILE STR_VAR

Description: This command puts up the standard file definition box, which prompts the user to type a filename to create and specify where to create it at. The file is not actually created with this command, to do that, follow up with a "USEROPENO" command. The full filename is returned in STR_VAR. If the user clicks on the "Save" button, the YES/NO flag is set to "YES". If the user clicks on the "Cancel" button, the YES/NO flag is set to "NO".

Example:

(Let's find out what to name a file and where to put it)

PUTFILE A\$

(If he clicked on the "Cancel" button, the YES/NO flag is set to "NO", and we'll just quit there)

IF NO END

(Otherwise, the filename to create is contained in A\$)

GETVOL STR_VAR

Description: This command prompts the user to select a volume/path via the standard dialog box that is used with such commands as the "Received File Destination" command under the File menu. The Yes/No flag is set to "YES" if the user selects a volume, or "NO" if the user instead clicks on the "Cancel" button. The volume name is returned in STR_VAR. WARNING: The volume name returned does not have a trailing colon (if you need it to, simply use a "CONCAT" command to put one there).

Example:

(Let the user choose where to put the file I'm about to create, which)
(I'll call "Foobar")

GETVOL A\$

(Don't do anything if the Cancel button was clicked)

IF NO END

(Add the filename to the end of the volume name in A\$)

(Notice the leading colon before the filename)

CONCAT A\$, :Foobar

(I could now create the file contained in A\$)

RENAME STR_VAR, STR_EXP

Description: This command allows you to rename a disk file to a different name. STR_VAR contains the original filename, and STR_EXP contains the filename you wish that file to be changed to. This command cannot be used to move a file from one disk (or HFS folder) to another location, but just to change its name at its present location.

SCREENDISK

Description: This command is equivalent to clicking in the "Display Screen To Archive File" button" in the General Status Bar.

Example:

(Take a "snapshot" of the text in the Data Area of the Terminal Window)

(and put it in the "Archived Text" text file)

SCREENDISK**MAKEFILE** FILETYPE_EXP,STR_EXP

Description: This command is used to change the existing file named in STR_EXP (it will not create a file and will set the Error Flag to "ERROR" if the named file does not exist) to the filetype contained in FILETYPE_EXP. This is useful for giving data files you create their own filetype, which can be specified later via a "GETSELECT" command. The "USEROPENO" and "USEROPENA" commands use the default filetype "TEXT" when creating a new file. WARNING: Never use the "MAKEFILE" command with an open file - always do a "USERCLOSE" first!

Example:

(Create a file called "Foobar")

USEROPENO 1,Foobar

(We could write out any data we want in it now, but for now,)

(we'll just close it.)

USERCLOSE 1

(Now change its filetype to "JUNK")

MAKEFILE JUNK,Foobar

GETRECPATH STR_VAR

Description:

This command puts the received file destination pathname into the string variable specified in "STR_VAR". The pathname does not conclude with a colon character. This command is useful for constructing full filenames (by concatenating a colon character and then the desired filename to this pathname) for use with the "RECX", "RECY", "RECK", "RECZ", and "RECF" commands so that the file is saved in the folder expected by the user.

Example:

(Receive a file with a unique filename using XMODEM and put it into)
(the received file destination folder.)

(First, get the received file destination folder in A\$)

GETRECPATH A\$

(Concatenate a colon character to the end of that)

CONCAT A\$, :

(Get a unique filename in B\$)

UNIQUE B\$

(Concatenate the unique filename to the end of the pathname)

CONCAT A\$, B\$

(Receive the file using XMODEM)

RECX A\$

SETRECPATH STR_EXP

Description:

This command sets the received file destination folder to the pathname specified in "STR_EXP". If the path is not valid, the root level of the volume containing White Knight will be used. The pathname should not conclude with a colon character.

Example:

(Set the received file destination folder to "HD:Downloads Folder")

SETRECPATH HD:Downloads Folder

UNIQUE STR_VAR

Description:

This command will create a filename based on the current date and time and returns it in the string variable specified in "STR_VAR". Because it is based on the time, to insure uniqueness you should put a "PAUSE 120" command between any two "UNIQUE" commands.

Example:

See example above for "GETRECPATH".

Disk File Input And Output

There are two user "paths" that may be open at one time for use with the following commands. They are known as path 1 and path 2. When you open a file for reading or writing, you assign it one of the unused paths. When you close a file, that path then becomes available for use with another file. In other words, you can have up to two file open for reading and writing at the same time. All paths are closed at the termination of a Procedure File. For an example of how these commands are used in an actual Procedure File, see Example #3 in the chapter "Procedure Examples".

USERCLOSE NUM_EXP

Description: Closes path number NUM_EXP and makes that path available for a "USEROPENI", "USEROPENO", or "USEROPENA" command. If the path is not currently open, this command does nothing - so use it if you're at a place in your Procedure where you're not sure if a path is in use or not.

USEROPENA NUM_EXP,STR_EXP

Description: Opens the filename STR_EXP using path number NUM_EXP for Append. Append means that if the file doesn't exist, it will be created as a new and empty file. If it does exist, any further "USERWRITE" or "USERWRRCR" commands will be done at the end of the file, so previous data is not destroyed. If the file can't be created, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USEROPENI NUM_EXP,STR_EXP

Description: Opens the filename in STR_EXP using path number NUM_EXP for Input. Input means that you'll be using only "USERREAD" commands with that path, not "USERWRITE" or "USERWRRCR" commands. If the file doesn't exist, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USEROPENO NUM_EXP,STR_EXP

Description: Opens filename in STR_EXP using path number NUM_EXP for Output. Output means that you'll be using only USERWRITE and USERWRRCR commands with that path. If "filename" already exists, it will be destroyed and recreated as an empty file by this command. If the file can't be created, the Error flag is set to "ERROR", otherwise, it's set to "NO ERROR".

USERREAD NUM_EXP,STR_VAR

Description: The procedure disk Input/Output commands are meant to be used with text files that have lines that are a maximum of 132 characters long each and end with a carriage return. This command reads data from path number NUM_EXP (a file opened with the "USEROPENI" command) either up to 132 characters or the first carriage return (the carriage return is discarded). The data is copied into the specified string variable STR_VAR. If the read is unsuccessful (most likely because you've reached the end of the file) the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERWRCR NUM_EXP

Description: Writes a single carriage return to path number NUM_EXP (a file opened with either the "USEROPENA" or "USEROPENO" commands), effectively terminating a line for later input by the "USERREAD" command.

USERWRITE NUM_EXP,STR_EXP

Description: Writes the data in STR_EXP to path number NUM_EXP (a file opened with either the "USEROPENA" or "USEROPENO" commands). It does not write a carriage return at the end of that data, so be sure and use "USERWRCR" commands where necessary. If the write was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERPUT% NUM_EXP,NUM_VAR

Description:

Used to write the contents of a numeric variable to a file opened with the "USEROPENO" or "USEROPENA" command. NUM_EXP is the path number (1 or 2) that was specified in the "USEROPENO" or "USEROPENA" command and NUM_VAR is the numeric variable to write to that file. Variables written with a "USERPUT%" command MUST be read back in with a

"**USERGET%**" command! If the write was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERGET% NUM_EXP, NUM_VAR

Description:

Used to read a value into a numeric variable that was previously written with a "**USERPUT%**" command from a file opened with a "**USEROPENI**" command. NUM_EXP contains the path number (1 or 2) specified in the "**USEROPENI**" command, and NUM_VAR is the numeric variable to read the value into. "**USERGET%**" should only be used to read in data that was saved with a "**USERPUT%**" command. If the write was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERPUT\$ NUM_EXP, STR_VAR

Description:

Used to write the contents of a string variable to a file opened with the "**USEROPENO**" or "**USEROPENA**" command. NUM_EXP is the path number (1 or 2) that was specified in the "**USEROPENO**" or "**USEROPENA**" command and STR_VAR is the string variable to write to that file. Variables written with a "**USERPUT\$**" command MUST be read back in with a "**USERGET\$**" command! If the write was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

USERGET\$ NUM_EXP, STR_VAR

Description:

Used to read a string of characters into a string variable that was previously written with a "**USERPUT\$**" command from a file opened with a "**USEROPENI**" command. NUM_EXP contains the path number (1 or 2) specified in the "**USEROPENI**" command, and STR_VAR is the string variable to read the data into. "**USERGET\$**" should only be used to read in data that was saved with a "**USERPUT\$**" command. If the write

was unsuccessful (path hasn't been opened, disk full, etc.), the Error flag is set to "ERROR", otherwise it's set to "NO ERROR".

Printer Manipulation

ECHO ON_OR_OFF_EXP

Description: This command is equivalent to choosing **Local->Printer Echo**. ON_OR_OFF_EXP controls whether or not echoing should be turned on or off.

Example:

(Start sending everything that comes over the serial port to the)
(printer now)

ECHO ON

SCREENPRINT

Description: This command is equivalent to clicking the "**Display Screen To Printer**" button in the General Status Bar.

Example:

(Send a "snapshot" of the text in the Data Area of the Terminal Window)
(to the printer)

SCREENPRINT

WRITE STR_EXP

Description: This command is equivalent to choosing **Local->Print TEXT File**. The filename of the file to print is supplied in STR_EXP.

Example:

(Print the file named "Stock Quotes Capture")

WRITE Stock Quotes Capture

Settings Manipulation

*

Description: Any line in a Procedure file that begins with the asterisk character is treated like a comment (the rest of the characters in the line are ignored), but unlike those lines that start with a left parenthesis character, these lines will actually be included in the executable Procedure file. These lines are mainly for use with the

"TRACE" command.

TRACE ON_OR_OFF_EXP

Description: "TRACE ON" turns on special trace mode. "TRACE OFF" turns off special trace mode. The special trace mode is a bit different than the normal trace mode provided by the **Service->**

Procedure->Monitor Execution choice and the "LOUD" Procedure command. Instead of showing each Procedure command as it executes, the special trace mode shows the last line in the Procedure file encountered which begins with a "*" character. This can be used as an aid in debugging to display messages such as:

```
"* I'm now in the 'Dial A Number' routine. *"
```

ANSWERBACK STR_EXP

Description: This command changes the VT100 Answerback Message (which is displayed in the dialog box brought up by choosing **Customize->VT**

Modes) to STR_EXP. STR_EXP may contain control characters.

Example:

(Change the answerback message to "Hello" followed by a carriage return)

```
ANSWERBACK Hello^M
```

BBAR

Description: This command is equivalent to choosing **Local->Status Bar->Buffered Keyboard**.

Example:

(Activate the Buffered Keyboard)

```
BBAR
```

COMM STR_EXP

Description: This command is used to change the serial port settings for baud rate, parity, data bits, stopbits, and duplex. STR_EXP is in the form:

BAUD-PARITY-DATABITS-STOPBITS-DUPLEX

where:

BAUD may equal 300, 450, 1200, 2400, 4800, 9600, 19200, or 57600.

PARITY may equal N, O, E, K, S, or I (meaning NO, ODD, EVEN, MARK, SPACE, or IGNORE)

DATABITS may equal 5, 6, 7, or 8

STOPBITS may equal 1, 1.5, or 2

DUPLEX may equal FULL, HALF, ECHO, or NULL

Example:

(Set the serial port to 1200 baud, NO parity, 8 data bits, 1 stopbit,
(and FULL duplex)

COMM 1200-N-8-1-FULL

CONTROL1 ^CHAR_CON

CONTROL2 ^CHAR_CON

CONTROL3 ^CHAR_CON

Description: These three commands modify the Control Character Buttons in the General Status Bar. The buttons are designated #1, #2, and #3 from left to right. CHAR_CON should be an uppercase letter from A to Z.

Example:

(Set up the three Control Character Buttons to be CTRL-A, CTRL-B,
(and CTRL-C)

CONTROL1 ^A

CONTROL2 ^B

CONTROL3 ^C

CRC ON_OR_OFF_EXP

Description: This command is analogous to selecting the "XMODEM-CRC" option in the dialog box brought up by choosing **Customize->Options->X-Y-ZMODEM**. A "CRC ON" command would checkmark that option, and a "CRC OFF" command would select "XMODEM-Classic".

Example:

(Turn on CRC error checking)

CRC ON

DELKEY ON_OR_OFF_EXP

Description: This command is analogous to the "Backspace key sends DEL" option in the dialog box brought up by choosing **Customize->**

Options->Key Mapping. A "DELKEY ON" command would checkmark that option, and a "DELKEY OFF" command would checkmark the "Backspace key sends Backspace" option.

Example:

(Make the backspace key a DELETE key (sends ASCII code 127))

DELKEY ON

(Make the backspace key normal (sends ASCII code 8))

DELKEY OFF

DISPLAY NUM_EXP

Description: This command is analogous to the "Display columns" option in the dialog box brought up by choosing **Customize->**

Options->Emulation. NUM_EXP must be a number between 20 and 132.

Example:

(Give me an 80 column display)

DISPLAY 80

ALT ON_OR_OFF_EXP

Description: This command is analogous to the "` key sends ASCII code: X" option in the dialog box brought up by choosing **Customize->**

Options->Key Mapping. A "ALT ON" command would checkmark that option, and a "ALT OFF" command would uncheckmark it. This command does not change the ASCII code number portion of that option.

Example:

(I've got 27 in the ASCII code number portion of that option,)

(so turn on the ` key so that it works like an ESCAPE key)

ALT ON

FONT NUM_EXP1,NUM_EXP2

Description: The **"FONT"** command is used to change the font and/or point size currently used in the Data Area of the Terminal Window. NUM_EXP1 is the font number to use, and NUM_EXP2 is the point size to use. If the font number or point size isn't available, this command has no affect. How does one find out the font number? It is displayed at the top of the point size selection dialog box when you choose **Local->Window->Font**, choose a font from under the **Font** menu, and then choose **Other Point Size** from under the **Size** menu. For instance, if I choose the Monaco font, the top of the Other Point Size selection dialog box would display the message:

Select the point size for use with the font
"Monaco" (font number 4)

Example:

(Use Monaco (font number 4) font, 9 point size)

FONT 4,9

GBAR

Description: This command is equivalent to choosing **Local->Status Bar->General**.

Example:

(Display the General Status Bar)

GBAR

NOBAR ON_OR_OFF_EXP

Description: This command is used to turn on or off the Status Bar display. It will override all Status Bars, including those brought up by **"TRACE"**, **"LOUD"**, and **"@ ON"**. **"NOBAR ON"** will turn off the Status Bar Display, and **"NOBAR ON"** will turn it on.

Example:

(Turn off the Status Bar Display)

NOBAR ON

LF ON OR OFF EXP

Description: This command is analogous to the "Return key sends:" option in the dialog box brought up by choosing **Customize->Options->Key Mapping**. A "LF ON" command is equivalent to clicking the "carriage return and linefeed" radio button, and a "LF OFF" command is equivalent to clicking the "carriage return only" radio button.

Example:

(Return key sends a carriage return followed with a linefeed)

LF ON

LOADSET STR EXP

Description: This command is equivalent to choosing **Customize->Settings->Load**. STR_EXP is the filename of the Settings File to load.

Example:

(Load the Settings File "BBS Settings" which is on the disk "MYDISK")

LOADSET MYDISK:BBS Settings

LOUD

Description: This command is equivalent to checkmarking the **Service->Procedure->Monitor Execution** choice.

Example:

(Start monitoring the Procedure command execution)

LOUD

MACRO STR EXP

Description: This command is equivalent to choosing **Customize->Macro Keys->Load**. STR_EXP is the filename of the Macro Key file (created by choosing **Customize->Macro Keys->Save**) to load.

Example:

(Load in the Macro Key file "BBS Macros" on the disk "MYDISK")

MACRO MYDISK:BBS Macros

FILTERT STR_EXP
FILTERP STR_EXP
FILTERF STR_EXP

Description: These commands are used to load the Filter File named in "STR_EXP". **FILTERT** loads the Filter File to the Terminal Filter. **FILTERP** loads the Filter File to the File Transfer Filter. **FILTERF** loads the Filter File to the File Capture Filter.

Example:

(Load in the Filter File "Show Controls" to the Terminal Filter)

FILTERT Show Controls

MBAR [NUM_EXP]

Description: This command is equivalent to choosing **Local->Status Bar->Macro Keys**. NUM_EXP is optional. If it is not supplied, the current active set is displayed. Otherwise, NUM_EXP must be either 1, 2, or 3, which changes the current active set to that number.

Example:

(Display the Macros Status Bar, set #2)

MBAR 2

MODEM

Description: This command instructs White Knight to communicate through the modem serial port.

Example:

(Use the modem port)

MODEM

NOZAP ON_OR_OFF_EXP

Description: This command is supplied to support Procedures written in previous versions of White Knight. **"NOZAP ON"** switches to IGNORE Parity, and **"NOZAP OFF"** switches to NO Parity.

Example:

(Don't zap the high bits of incoming characters by switching to)
(IGNORE parity.)

NOZAP ON.

PRINTER

Description: This command instructs White Knight to communicate through the printer serial port.

Example:

(Use the printer port to communicate through)

PRINTER

QUIET

Description: This command is equivalent to uncheckmarking the **Service->Procedure->Monitor Execution** choice.

Example:

(Stop monitoring Procedure command execution)

QUIET

REDIAL LIMIT NUM_EXP

Description: This command sets the upper limit of attempts a "REDIAL" command will try. NUM_EXP should be a number between 1 and 255, or zero if you wish there to be no limit for redialing tries.

Example:

(Set the redial limit to 15 tries)

REDIAL LIMIT 15

RESET

Description: This command sets the Elapsed Time Clock to 00:00:00 and the Billing Timer to \$0.00.

Example:

(First dial our number)

DIAL ATDT 555-1212

(Wait for a connection)

PROMPT CONNECT

(Now synchronize the elapsed time and billing clocks)

RESET

SCREEN ON_OR_OFF_EXP

Description: This command is analogous to the **Local->Window->Hide** menu choice. A "**SCREEN OFF**" command will hide the Terminal Window (make it invisible), and a "**SCREEN ON**" command will make the Terminal Window visible.

Example:

(Make sure the Terminal Window is visible)

SCREEN ON

WINDOW ON_OR_OFF_EXP

Description: This command shows ("**WINDOW ON**") or hides ("**WINDOW OFF**") the Macros Window.

Example:

(Show the Macros Window)

WINDOW ON

BUTTON NUM_EXP1,NUM_EXP2,ON_OR_OFF_EXP1,ON_OR_OFF_EXP2

Description: This command controls how a Macro Key is displayed in the Macros Status Bar and Macros Window. "**NUM_EXP1**" is the Macro Key Set Number (1, 2, or 3) containing the desired key, and "**NUM_EXP2**" is the Macro Key Number (0 through 9). "**ON_OR_OFF_EXP1**" should be "**ON**" if you want the Macro Key to be visible in the Macros Status Bar, or "**OFF**" if you want it to be invisible. "**ON_OR_OFF_EXP2**" should be "**ON**" if you want the Macro Key to be visible in the Macros Window or "**OFF**" if you want it to be invisible. In order to accommodate groups of Macro Keys being turned on or off, this command doesn't actually show or hide any Macro Keys. To do that, use a "**REDRAW**" command after the last "**BUTTON**" command in the group.

Example:

(Make Macro Key Set #2, Key #3 visible in the Macros Status Bar and)

(invisible in the Macros Window. Make Macro Key Set #1, Key #5)
invisible in the Macros Status Bar and visible in the Macros Window.)

BUTTON 2,3,ON,OFF

BUTTON 1,5,OFF,ON

(Now redraw both the Macros Status Bar and Macros Window)

REDRAW

REDRAW

Description: The "**BUTTON**" command does not redraw either the Macros Status Bar or the Macros Window. This is to provide a "cleaner" display when changing the visibility and/or invisibility of groups of Macro Key buttons. Therefore, the "**REDRAW**" command simply redraws the Macros Status Bar and Macros Window.

Example:

See example under "**BUTTON**" command.

SETCOST NUM_EXP

Description: This command is analogous to the **Local->Timer->Billing Cost** choice. **NUM_EXP** is the amount to increment the billing clock each minute in 100ths of a cent.

Example:

(Set billing cost to \$5.00/hour)

SETCOST 833

SLOW ON_OR_OFF_EXP

Description: This command is analogous to the "XMODEM, YMODEM, ZMODEM block timeout" option in the dialog box brought up by choosing **Customize->Options->X-Y-ZMODEM**. A "**SLOW ON**" command sets the timeout value to 20 seconds. A "**SLOW OFF**" command sets the timeout value to 5 seconds.

Example:

(Set XMODEM timeout to 5 seconds)

SLOW OFF

STRIP ON_OR_OFF_EXP

Description: This command is analogous to the "Use filter" option in the dialog box brought up by choosing **Customize->Options->Text Transfer**. A "STRIP ON" command checkmarks this option, and a "STRIP OFF" command uncheckmarks this option.

Example:

(Use the filter for further file captures)

STRIP ON

TTY**VT52****VT100****VT102**

Description: These three commands tell White Knight what kind of terminal it is to emulate, a TTY (generic) terminal, a DEC VT52 terminal, a DEC VT100 terminal, or a DEC VT102 terminal.

Example:

(Emulate a 80 column VT52 terminal)

DISPLAY 80

VT52

VBAR

Description: This command is equivalent to choosing **Local->Status Bar->VT100 Keys**.

Example:

(Display the VT100 Status Bar)

VBAR

XKSTRIP ON_OR_OFF_EXP

Description: This command is analogous to the "Use filter" option in the dialog box brought up by choosing **Customize->Options->File Transfer**. An "XKSTRIP ON" command checkmarks this option, and an "XKSTRIP OFF" command uncheckmarks this option.

Example:

(Use the filter for text file protocol receives)

XKSTRIP ON

Procedure Execution Modifiers

ATTACH FILETYPE_EXP,STR_EXP

Description: This command is used in conjunction with the "RUN" command in order to launch a Macintosh application specifying a document for that application to open once it has begun execution. For instance, if you wanted to leave White Knight and run a BASIC program, you would use the "ATTACH" command to specify the name of the BASIC program, and then the "RUN" command to execute the BASIC interpreter application. Or, you could use it to cause a spreadsheet program to execute and automatically open up a desired spreadsheet data file.

This command is provided for use by those with above average Macintosh experience, as it requires that you know the filetype, as well as the file name of the document you wish to attach to the application about to be executed. FILETYPE_EXP is a four character Finder filetype, and STR_EXP holds the file name of the document to be attached.

Example:

(I want to start up MacWrite and have it open a MacWrite file)
(named "Diary". I happen to know that MacWrite documents have a)
(filetype of "MACA" Both MacWrite and the document reside on a disk)
(named "MYDISK")

ATTACH MACA,MYDISK:Diary

RUN MYDISK:MacWrite

DO STR_EXP

Description: This command causes the current Procedure File to stop execution, and the Procedure File contained in STR_EXP to be executed.

Example:

(Chain off to the Procedure File "Second Proc" on the disk "MYDISK")

DO MYDISK:Second Proc

CRAWL ON OR OFF EXP

Description: Because White Knight uses a compiler, the speed at which a Procedure File executes may make the command tracing feature activated by the **Service->Procedure->Monitor Execution** (and the "LOUD" Procedure command) useless for keeping track of what's happening at any given moment. A "CRAWL ON" command slows down the execution of the Procedure File to a rate of one command executed every two seconds. A "CRAWL OFF" command makes the Procedure File execute at its normal speed. The delay caused by a "CRAWL ON" command is always cancelled when the Procedure File is terminated, so that future Procedure Files are not affected by it.

Example:

(Yikes! Things are running much too fast for me to see, slow it down!)

CRAWL ON

END

Description: The "END" command terminates the execution of a Procedure. While White Knight will always terminate the Procedure after it has executed the last physical Procedure command in a Procedure File (whether or not that command is an "END" command), the "END" command is useful to terminate the Procedure File at some other place than at the physical end of the File.

Example:

(If the Yes/No flag is set to "YES", branch to the routine GETMAIL)

(Otherwise, end the Procedure File immediately)

IF YES JUMPTO GETMAIL

END

:GETMAIL

(If I get here, collect my waiting mail)

GOSUB LABEL

Description: The "GOSUB" command is used with small collections of commands (called subroutines) that are used by various routines in a

Procedure. When this command is executed, an immediate branch to LABEL is performed, where commands are executed in order until the next "RETURN" command. At that point, execution is branched back to the command following the original "GOSUB" command. Subroutines may "GOSUB" to other subroutines to a level of twenty deep, with subsequent "RETURN" commands returning to the command following the most recently executed "GOSUB" command.

Example:

(This routine will branch execution to the subroutine FIRST)
(FIRST will ring the bell once and branch to the subroutine SECOND)
(SECOND will ring the bell once and return back to FIRST. FIRST then)
(returns back to the command following the original GOSUB command,)
(which halts execution of the Procedure.)

```
GOSUB FIRST
END
:FIRST
BELL
GOSUB SECOND
RETURN
:SECOND
BELL
RETURN
```

HOST

Description: The "HOST" command terminates execution of the Procedure File and puts White Knight in "Host Mode", which is equivalent to choosing **Service->Cancel Procedure**, and then choosing **Local->Host Mode->Become Host**. The Host Mode command "DO" may be used to exit the Host Mode and execute a Procedure File.

Example:

(Quit the Procedure File here, and immediately go into Host Mode)
HOST

IF ERROR PROC_CMD

Description: If the Error flag is set to "ERROR", the Procedure Command

PROC_CMD is executed. If the Error flag is set to "NO ERROR", the Procedure Command PROC_CMD is not executed.

Example:

(If there's an error, ring the bell and quit the Procedure)

```
IF ERROR GOSUB ERROR ROUTINE
```

(Otherwise, just end the Procedure here)

```
END
```

```
:ERROR ROUTINE
```

```
BELL
```

```
RETURN
```

IF NO ERROR PROC_CMD

Description: If the Error flag is set to "NO ERROR", the Procedure Command PROC_CMD is executed. If the Error flag is set to "ERROR", the Procedure Command PROC_CMD is not executed.

Example:

(If the Error flag is set to "NO ERROR" end the Procedure, otherwise)

(ring the bell once and then end the Procedure)

```
IF NO ERROR END
```

```
BELL
```

```
END
```

IF NO PROC_CMD

Description: If the Yes/No flag is set to "NO", the Procedure Command PROC_CMD is executed. If the Yes/No flag is set to "YES", the Procedure Command PROC_CMD is not executed.

Example

(If the numeric variable A% is not less than 15, ring the bell)

```
TEST A% < 15
```

```
IF NO BELL
```

IF YES PROC_CMD

Description: If the Yes/No flag is set to "YES", the Procedure Command PROC_CMD is executed. If the Yes/No flag is set to "NO", the Procedure Command PROC_CMD is not executed.

Example

(If the numeric variable A% is less than 15, ring the bell)

```
TEST A% < 15
IF YES BELL
```

JUMPTO LABEL

Description: The "JUMPTO" command is used to branch immediately to a different place in the Procedure File, marked by LABEL.

Example:

(Branch immediately to the label LABEL1)

```
JUMPTO LABEL1
```

(The following BELL command will never be executed!)

```
BELL
```

```
:LABEL1
```

(Ring the bell once and end the Procedure File)

```
BELL
```

```
END
```

LOCK ON OR OFF EXP

Description: The "LOCK" command is included for very special circumstances, and should not be used indiscriminantly. The normal policy of execution is for White Knight to perform one Procedure instruction, look for and process any incoming data over the serial port or handle user input, execute the next Procedure command, process incoming data and handle user input, and so forth. A "LOCK ON" command places the Procedure file in a closed loop, where Procedure commands will be executed continuously without interruption until a "LOCK OFF" command is executed, which puts White Knight back into a normal mode of operation.

Be very careful with this command, because incoming data, menu selections, keyboard presses, mouse activity, etc. are completely locked out. If your Procedure goes into an endless loop, the Macintosh would actually have to be turned off and then back on in order to regain control. It is most useful when you have a small number of commands that must be executed in a burst before any waiting incoming data is

collected and processed.

Example:

(Look for the string of characters "CONNECT" to come over the)
(serial port)

PROMPT CONNECT

(Now open up the file MYDATA on the disk MYDISK, and then look for the)
string "PASSWORD?" to come over the modem. If the string came and went)
before the OPEN command was finished, I would miss it completely, so)
(let's put the Procedure in a locked condition during the open and use)
(an ALERT command to do the looking)

ALERT1 PASSWORD?/JUMPTO GOT IT

LOCK ON

USEROPENI 1,MYDISK:MYDATA

(OK, open the floodgate, if the string was received during the)
(USEROPENI command, the active ALERT command will see it)

LOCK OFF

(If we got here, the string was not in the waiting data, so we'll just)
(use a normal PROMPT command to look for the string)

PROMPT PASSWORD?

(In either case, we ought to get to here when "PASSWORD?" is received)

:GOT IT

(Just ring the bell and end the Procedure here)

BELL

END

NEST STR_EXP

Description: This command is used to execute a second Procedure file from within a Procedure file, and then later return to the same point in the first Procedure file. The Procedure file to execute is specified in "STR_EXP". In this way, separate Procedure files can be called as subroutines similar to the way the "GOSUB" and "RETURN" Procedure commands work with labels. Procedures can be nested 6 levels deep, but since a Modem Driver is essentially a nested Procedure file that can be called at any given moment, you should take care not to nest your Procedure files more than 5 levels deep. Although the Procedure Compiler will allow you to, you should never put a "NEST" command in an "ALERT" statement, since the return point will be the command following the "ALERT" statement, which is probably not what you want. The "RETURN" equivalent for a "NEST" command is "NESTEND"

Example:

```
(This is the first Procedure file)
(Call the Procedure file named "Ring Bell")
NEST Ring Bell
END
```

```
(This is a second Procedure file named "Ring Bell". It will ring)
(one bell and then return to the calling Procedure.)
```

```
BELL
NESTEND
```

NESTEND

Description: This command is used to return to the calling Procedure file in a file executed with a "NEST" command. If the Procedure file was not executed with a "NEST" command, this command functions exactly like the "END" command. The "NEST" and "NESTEND" commands are useful for developing a library of functions, rather than having to copy blocks of code from one Procedure to another.

Example:

See example under "NEST" command.

ONPANIC PROC_CMD

Description: This command places the Procedure Command PROC_CMD in a temporary holding space, if the "PANIC" condition occurs while a "PROMPT" or "ALERT" command is active, PROC_CMD is retrieved and immediately executed. The "PANICAFTER" command is used to describe when the "PANIC" condition exists.

Example:

```
(If the string "CONNECT" does not come over the serial port in 30)
(seconds, set the "PANIC" condition and log off immediately)
```

```
PANICAFTER 30
ONPANIC JUMPTO GET OUT
PROMPT CONNECT
```

```
(If we get here, everything went fine, we'll just end the Procedure)
(File here, though)
```

```
END
```

:GET OUT

(Gadzooks! If we got here the "CONNECT" string didn't arrive, so we)
(would follow these comments with a routine to hang up the phone.)

TYPE +++

PROMPT OK

PAUSE

TYPE ATH^M

END

PANICAFTER NUM_EXP

Description: This command is used in conjunction with the "ONPANIC" command to specify how long to wait for a desired string of characters to arrive before giving up and setting a "PANIC" condition. NUM_EXP is the number of seconds to wait once a "PROMPT" or "ALERT" command is executed before executing the Procedure Command specified in the most recent "ONPANIC" command. Whenever a "PROMPT" or "ALERT" command is satisfied by an incoming string, the "ONPANIC" command is cancelled and must be followed by another "ONPANIC" command if desired to work for subsequent "PROMPT" or "ALERT" commands.

Example:

(Wait 30 seconds for "CONNECT", branch to "GET OUT" if it doesn't come)

ONPANIC JUMPTO GET OUT

PANICAFTER 30

PROMPT CONNECT

(Type a CTRL-C, then wait for the string "PASSWORD?")

PAUSE

TYPE ^C

(The PANICAFTER command was turned off if I got here, so I need to do)
(another one)

PANICAFTER 30

PROMPT PASSWORD?

(Everything worked fine, so type my password and end the Procedure)

PAUSE

TYPE MyPassword^M

END

:GET OUT

(Whoops, one of the PROMPT's never arrived, so I'll hang up the phone)
(and alert the user with a couple of bells.)

TYPE +++

```
PROMPT OK
TYPE ATH^M
BELL
BELL
END
```

PAUSE [NUM_EXP]

Description: This command stops execution of the Procedure File for a specified period of time, and then allows the Procedure File to continue execution as normal. NUM_EXP is a number of 1/60ths of a second to delay. If you wished to pause the Procedure for one second, you would use a "PAUSE 60" command, for six seconds "PAUSE 360". NUM_EXP is optional - if it isn't supplied, a value of 60 (one second) is used.

Example:

(Wait of the string "CONNECT" to come over the modem)

```
PROMPT CONNECT
```

(Always pause for a second before typing something to make sure the remote machine is ready to handle your input. The following command will pause for 1 second)

```
PAUSE
```

(Now, type in my secret password and a carriage return)

```
TYPE MyPassword^M
```

QUIT

Description: This command causes the Procedure File to immediately terminate execution, and then it will cause White Knight to immediately terminate execution (equivalent to choosing **File->Quit**) and return to the Finder.

Example:

(I'm all done with White Knight, quit back to the Finder)

```
QUIT
```

RETURN

Description: This command is used in conjunction with the "GOSUB"

command to end execution of a subroutine and branch back to the Procedure command following the most recently executed "GOSUB" command.

Example:

```
(Ring the bell five times)
GOSUB RING5
(Ring it five more times)
GOSUB RING5
(What the hell, ring it five more times)
GOSUB RING5
(And end the procedure)
END
:RING5
BELL
BELL
BELL
BELL
BELL
(Enough ringy-dingy's, return back to the GOSUB that called "RING5")
RETURN
```

RUN STR_EXP

Description: This command is used to transfer from White Knight to a second Macintosh application, bypassing the Finder. STR_EXP is the filename of the application to execute. If filename is wrong or doesn't exist, an error message is generated to the user and the Procedure File is terminated, but White Knight is not terminated.



The "RUN" command is not guaranteed to work properly in all Macintosh environments, such as those running under a file sharing or multitasking system. Rigid testing is indicated before trusting this command. If you are running White Knight under MultiFinder, you must have the "Using MultiFinder" option checkmarked in the dialog box presented by selecting **Customize->Options->White Knight**.

Example:

(Here's a handy little Procedure I use to branch directly to another)

(Macintosh application. It first restricts the GETFILE command to) (files of only type "APPL", which is the filetype of all Macintosh) (applications, has the user select the file, and the executes it.)

GETSELECT APPL

GETFILE A\$

(Make sure he didn't click the Cancel button)

IF NO END

IF YES RUN A\$

WAIT STR_EXP

Description: The "WAIT" command is used to temporarily stop execution of a Procedure File until a certain time of day. STR_EXP is the time of day to wait until in 24 hour format (10:00:00 PM would be given as 22:00:00). The time must be in the form:

HH:MM:SS

where:

HH = hours (00 to 24)

MM = minutes (00 to 59)

SS = seconds (00 to 59)

leading zeros are necessary for values less than 10. The string:

5:30:8

must be supplied with leading zeros, as in:

05:30:08

to function correctly. When the specified time is reached, execution resumes normally with the command following the "WAIT" command.

Example:

(Wait until 10:30 PM)

WAIT 22:30:00

(And ring a bell)

BELL

SHUTDOWN

Description: Closes White Knight similar to the "QUIT" command and then executes a system Shut Down similar to if you chose "Shut Down" from under Finder's "Special" menu.

Example:

(Ring a bell to tell them we're about to shut down)

BELL

(Shut down the Macintosh)

SHUTDOWN

User Defined Menus

White Knight gives you the opportunity to create your own customized pull down menu that can be used to either send a string of characters or execute a Procedure File. You have complete control of how your menu is named and it's contents, and your menu can contain up to 20 menu choices. Here is an example Procedure File that will create a menu called "Scott" which has five choices. The first four choices will just send a string telling us which menu choice sent it. The last menu choice will execute a Procedure File named "PROCFILE" which is on a disk named "HD20".

(Make sure there aren't any user defined menus left around by some other Procedure File)

MENUOFF

(Create a menu called "Scott" in Memory)

DEFINE MENU Scott

(Add the five menu choices I want in order)

ADD TO MENU This is menu choice #1

ADD TO MENU This is menu choice #2

ADD TO MENU This is menu choice #3

ADD TO MENU This is menu choice #4

ADD TO MENU This is menu choice #5

(Tell White Knight what each macro key is supposed to do)

MENUDOES 1,I just chose menu choice #1^M

MENUDOES 2,I just chose menu choice #2^M

MENUDOES 3,I just chose menu choice #3^M

MENUDOES 4,I just chose menu choice #4^M

MENUDOES 5,\HD20:PROCFILE

(Show the menu in the menu bar)

INSERT MENU**END**

ADD TO MENU STR EXP

Description: This adds a menu choice to the menu you've defined. The new choice is added to the end of any existing commands. There is no way

to insert a command before others or delete individual commands, so choose the order of your menu choices with forethought. STR_EXP contains the text that will appear in the menu choice with one notable exception. Certain special characters in STR_EXP are used to affect the appearance of the menu choice, so be careful about using non-alphanumeric characters in STR_EXP.

The following is provided for very experienced Macintosh users only. You will probably never use hardly any of the following information (except maybe the dividing line thingy), and some of it is pretty obtuse, but I'll document it here so you can play with it and show people all kinds of "betcha didn't know" magic later.

The three characters ^ ! and / should never be used in menu choice strings. Trust me on that. The three characters < (and - do have special functions which are of some use. Here's what they do:

<u>Character</u>	<u>Function</u>
<	The menu item has a special character style. The style to use is designated by the character following the < character and is one of the following: <B - boldfaced <I - italicized <U - underlined <O - outlined <S - shadowed I don't know that there's any reason more than one style couldn't be used, but let's keep this thing under control, eh?
-	Menu is filled with a line of hyphen characters. This should always be used with the (character so that the choice is not selectable by the user.
(Choice is disabled. Many Mac menus have a dividing line of disabled hyphen characters. To put one of those in your menu, you would use the special command: ADD TO MENU (- As another example, the command: ADD TO MENU This is a choice<O<U Would make the words "This is a choice" appear as both outlined and shadowed in the menu.

DEFINE MENU STR_EXP

Description: This gives your menu a name. I highly recommend you do a "MENUOFF" command before executing this command to be sure any previously defined menu is disposed of before the new one is defined. Note that this just creates the menu in memory - it does not display the menu in the menu bar. Once you've executed this command, you'll do some "ADD TO MENU" and "MENU DOES" commands to define the contents, appearance, and functions of the menu choices, and then execute an "ENABLE MENU" command to display the menu in the menubar and make it choosable by the user. STR_EXP holds the menu's title that will appear in the menu bar when the menu is enabled, and it may not be more than 20 characters long.

DISABLE MENU NUM_EXP

Description: This command can be used to disable (make gray and unselectable) any menu item (if NUM_EXP is from 1 to 20) or the entire menu if NUM_EXP is zero.

Example:

(Disable the third choice in my menu)

DISABLE MENU 3

ENABLE MENU NUM_EXP

Description: After any "DISABLE MENU" command, this command can be used to enable any menu item (if NUM_EXP is from 1 to 20) or the entire menu if NUM_EXP is zero.

Example:

(Re-enable the third choice in my menu)

ENABLE MENU 3

INSERT MENU

Description: This command is done after all of the menu items have been added to the menu, it simply draws the menu in the menu bar in an

enabled state.

MENUDOES NUM_EXP,STR_EXP

Description: This command defines exactly what your menu choices do when they are selected. NUM_EXP is a number from 1 to 20 that corresponds to a menu choice (1 being the topmost menu choice and 20 being the bottommost menu choice). If the first character of STR_EXP is a backslash (\), the rest of STR_EXP is considered to be a filename of a Procedure File to execute immediately. If the first character is not a backslash, STR_EXP it is considered to be a string of characters to send through the serial port immediately. In other words, STR_EXP corresponds exactly to what you would put into a Macro Key string (including special and control characters). **STR_EXP may not exceed 68 characters.**

MENUOFF

Description: This disables the defined menu, removes it from the menu bar, and wipes it out of memory. If there is no defined menu, this command does nothing. User-defined menus are not removed or disabled when a Procedure terminates - they stay on the menubar until a "MENUOFF" command is executed. Therefore, it's a real good idea to put a "MENUOFF" command before any "DEFINE MENU" commands to wipe out any old user-defined menus left behind by other Procedure Files.

Example:

(Get rid of my user defined menu)

MENUOFF

User Defined Dialog Box Commands

White Knight's Procedure language allows you to use one of five preprogrammed dialog boxes for displaying information and collecting information from the user. Each of the five different dialog boxes has a different appearance, and is useful in a specific kind of user notification or input request.

QUERY0 STR_VAR

Description: The dialog box brought up by this command (that's a zero on the end, not the letter "O") contains three lines of prompting text, one editable text item, and an "OK" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom). On exit, STR_VAR will contain the text the user typed into the editable text item. What the user types is echoed back to them as asterisks, giving a "secure input mode" for such things as passwords. The maximum number of characters that can be typed is 20.

Example:

```
COPYINTO X$,Please type in your password now.  
COPYINTO Y$,Each characters will be shown as an asterisk character  
COPYINTO Z$,for your protection.  
QUERY0 A$  
(A$ now contains their password)
```

QUERY1 STR_VAR

Description: The dialog box brought up by this command contains three lines of prompting text, one editable text item, and an "OK" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom). On exit, STR_VAR will contain the text the user typed into the editable text item.

Example:

```
(Put my prompting message in X$, Y$, and Z$)  
COPYINTO X$,Please type the name of your favorite music group,  
COPYINTO Y$,"Hack And Slash" movie, or television evangelist.  
COPYINTO Z$,If you have no favorite, just click the "OK" button.  
(Bring up the dialog box and get the user's input)  
QUERY1 A$  
(His response is now in A$)
```

QUERY2

Description: The dialog box brought up by this command contains three lines of prompting text and an "OK" button. On entry, you should pass

the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom).

Example:

(This just tells him what we're about to do, it doesn't get any info)
(from the user)

COPYINTO X\$,This is going to take me a while.

COPYINTO Y\$,Go grab yourself some dinner, OK?

(We don't need three lines, so just erase Z\$ so it shows up blank)

ERASE Z\$

(And put up the dialog)

QUERY2

QUERY3

Description: The dialog box brought up by this command contains three lines of prompting text, a "Yes" button, and a "No" button. On entry, you should pass the three lines of text to prompt the user with in the string variables X\$, Y\$, and Z\$ (from top to bottom). On exit, the Yes/No flag will be set to "YES" if the user clicked on the "Yes" button, or "NO" if the user clicked on the "No" button.

Example:

(Ask him if he really, really, etc. wants to do something)

COPYINTO X\$,Do you really, really, really, really,

COPYINTO Y\$,really, really, really, really, really,

COPYINTO Z\$,REALLY, REALLY want to hear the bell?

(Put up the dialog box)

QUERY3

(If he clicked on the "No" button, quit the Procedure)

IF NO END

(Otherwise, we know he clicked on the "Yes" button)

BELL

QUERY4

Description: The dialog box brought up by this command contains one line of prompting text, a "Yes" button, and a "No" button. On entry, you should pass the line of text to prompt the user with in the string variable X\$. On exit, the Yes/No flag will be set to "YES" if the user

clicked on the "Yes" button, or "NO" if the user clicked on the "No" button.

Example:

(Get a brief confirmation)

COPYINTO X\$,Wanna hear the bell?

(Put up the dialog box)

QUERY4

(And succumb to the user's command)

IF YES BELL

QUERY5

Description: The dialog box brought up by this command contains one line of prompting text and an "OK" button. On entry, you should pass the line of text to prompt the user with in the string variable X\$.

Example:

(Let the user know that everything is fine)

COPYINTO X\$,The Procedure was successful!

(Put up the dialog box)

QUERY5

User Defined Windows

The next group of Procedure commands deals with a new part of the White Knight interface, the User Window. Often times, it is desirable to hide what's happening in the Terminal Window (via a "SCREEN OFF" Procedure command) from the neophyte user, yet provide some sort of feedback that progress is being made. As you know, the various "QUERY" command dialog boxes are all modal, which means that all action behind the dialog box stops until the dialog box has been clicked closed by the user.

The User Window is modeless, which means it will stay around until you specifically get rid of it. Additionally, it is updated automatically when other windows overlap it and it does not affect the continued execution of your Procedure file. Finally, you can even control the size and location of the User Window on the display screen.

An example usage of a User Defined Window is shown at the end of this section.

UWOPEN0 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

CUWOPEN0 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

Description: These commands display a draggable window with a Title Bar, but no Close Box. That's a zero, not the letter "O", at the end of both of these commands. For both commands, "NUM_EXP1" is the top coordinate, "NUM_EXP2" is the left coordinate, "NUM_EXP3" is the bottom coordinate, and "NUM_EXP4" is the right coordinate of the window. "UWOPEN0" uses these coordinates exactly to size and position the window, so they should be given in global QuickDraw screen coordinates. "CUWOPEN0" uses the coordinates in a relative manner to size the window, the actual position of the window is centered on the screen. The window title is passed in the string variable W\$. The text to appear in the window is passed in X\$, Y\$, and Z\$. These strings are jammed right up against each other, so you'll want to take care to include necessary spacing. You can pass up to 132 characters in each string variable (taking care that the window size is large enough to display all of the characters).

UWOPEN1 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

CUWOPEN1 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

Description: These commands are similar to "UWOPEN0" and "CUWOPEN0" (respectively), except they produce a non-dragable window without a Title Bar and no Close Box. Therefore, W\$ is not used by these commands.

UWOPEN2 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

CUWOPEN2 NUM_EXP1,NUM_EXP2,NUM_EXP3,NUM_EXP4

These commands are similar to "UWOPEN0" and "CUWOPEN0" (respectively), except they include a Close Box in the Title Bar.

IF CLOSEUW PROC_CMD

Description: This command is used with dialog boxes created with "UWOPEN2" and "CUWOPEN2" to monitor whether or not the user has clicked in the Close Box. When the user does this, the window is not closed, it's up to you to do this explicitly (using a "UWCLOSE"

command). If the Close Box has been clicked in before this command has been executed, the Procedure command "PROC_CMD", which is a valid Procedure Command is executed. Otherwise, this command does nothing.

UWCLOSE

Description: This command removes the User Window created by any of the "UWOPEN" and "CUWOPEN" commands. Please note that you can have only one User Window open at any time. If the User Window is not open when this command is executed, it does nothing. Please note that an open User Window will survive even after a Procedure has ended or has been cancelled unless it is explicitly closed using this command. Therefore, if you want to give the user the ability to abort the Procedure cleanly, you should use a "UWOPEN2" or "CUWOPEN2", and when a click in the Close Box box is detected, branch to a closing routine.

UWUPDATE

Description: If you want to change the contents of an open User Window, this command is used to redisplay the text passed in the string variables X\$, Y\$, and Z\$. For User Windows created with "UWOPEN0", "CUWOPEN0", "UWOPEN2", and "CUWOPEN2", it will also set the window title to the string passed in W\$.

TOP NUM_VAR

Description: This command is useful when used before a "UWOPEN0", "UWOPEN1", or "UWOPEN2" command. It returns the vertical global coordinate in the numeric variable "NUM_VAR" of the bottom of the menu bar (which can be a different size on different machines - don't take it for granted). To fit your window flush below the menu bar, you would add 19 to the number returned by this command for "UWOPEN0" and "UWOPEN2" commands or add 8 to the number returned by this command for "UWOPEN1" commands.

Example Usage Of User Window Commands

(Set up the user window centered on the screen)

COPYINTO W\$, Sample User Window

COPYINTO X\$, Click in the Close Box to stop this noise.

ERASE Y\$

ERASE Z\$

(The window will be 200 pixels wide by 50 pixels high.)

CUWOPEN2 0,0,50,200

SAVETIME

:LOOP

IF CLOSEUW JUMPTO GETOUT

ELAPSED A%

TEST A% > 3

IF YES GOSUB RINGBELL

JUMPTO LOOP

:RINGBELL

BELL

SAVETIME

RETURN

:GETOUT

UWCLOSE

Procedure Examples

EXAMPLE #1:

Description: Shows how the "@ ON", "@ OFF", and "SHOW@" commands are used together to display the Special Status Bar. Also used: **PROMPT**, **ERASE**, and **COPYINTO**.

(Load the the three @ variables with our message to display)

```
COPYINTO @0,Waiting for the string "CONNECT" to arrive.
```

```
COPYINTO @1,Please be patient!
```

(We only need two lines, so we'll ERASE the third)

```
ERASE @2,
```

(Turn on the Special Status Bar)

```
@ ON
```

(And show the contents of the three @ variables)

```
SHOW@
```

(Now wait for the "CONNECT" string to arrive)

```
PROMPT CONNECT
```

(And turn off the Special Status Bar)

```
@ OFF
```

EXAMPLE #2:

Description: Shows how the "ALERT" and "PROMPT" commands are used together to look for two different possible strings to be received, and react by branching to the correct routine to handle the one that is received. Also used: **BELL**, **END**, and **JUMPTO**.

(Wait for either "you have mail" or the string "Command?" and branch to) (appropriate routine. This allows me to be looking for two things at) (once)

```
ALERT1 you have mail/JUMPTO HAVEMAIL
```

```
PROMPT Command?
```

(I got the "Command?" string so I JUMPTO somewhere else)

```
JUMPTO DOCOMMAND
```

```
:HAVEMAIL
```

(If I get here, I got the "you have mail" string and need to)

(insert the commands here to handle that)

(But for this example, we'll just string 'em both together)

```
:DOCOMMAND
```

```
BELL
```

```
END
```

EXAMPLE #3:

Description: Shows how the "USEROPENO", "USEROPENA", "USEROPENI", "USERREAD", "USERWRITE", "USERWRCR", "USERGET%", "USERPUT%", "USERGET\$", "USERPUT\$", and "USERCLOSE" commands are used to create, read, and write data with disk files. The Procedure creates a file, writes two lines of text, writes a numeric variable, writes a string variable, closes the file, reopens the file, writes one more line of text, closes the file, reopens the file for input, reads the lines of text and variables back in, and finally closes the file again.

(Set up A4\$ to contain "This is A4\$")

```
COPYINTO A4$,This is A4$
```

(Set up M3% to contain the value "53")

```
LET EQUAL M3%,53
```

(Create the file named "DATAFILE" for output using path number 1)

(in the same folder that White Knight resides.)

```
USEROPENO 1, :DATAFILE
```

(Write a line of text to the file)

```
USERWRITE 1,This is line #1
```

(Terminate it with a carriage return)

```
USERWRCR 1
```

(And do the same thing again for the second line)

```
USERWRITE 1,This is line #2
```

(Terminate the line with a carriage return so USERREAD can read)

(it back in.)

```
USERWRCR 1
```

(Write the numeric variable M3%)

```
USERPUT% 1,M3%
```

(Write the string variable A4\$)

```
USERPUT$ 1,A4$
```

(Now close the file)

```
USERCLOSE 1
```

(Let's reopen it now for append)

```
USEROPENA 1, :DATAFILE
```

(And write one more line of text to it)

```
USERWRITE 1,This is line #3
```

```
USERWRCR 1
```

(And close it up again)

```
USERCLOSE 1
```

(OK, Let's open it up for input this time)

```
USEROPENI 1, :DATAFILE
```

(And read the first two lines of text into the string)

(variables A\$ and B\$)

USERREAD 1,A\$

USERREAD 1,B\$

(Read in a numeric variable to M3%)

USERGET% 1,M3%

(Read in a string variable to A4\$)

USERGET\$ 1,A4\$

(Read in the last line of text to C\$)

USERREAD 1,C\$

(Close up the file to finish things off)

USERCLOSE 1

Example #4:

A very simple Procedure to dial a number using 1200 Baud, NO Parity, 8 Data Bits, 1 Stop Bit, and FULL Duplex. The number is (000)555-1212. I'm using a Hayes compatible modem. Please note that this is a bogus phone number.

(Set the serial port settings)

COMM 1200-N-8-1-FULL

(Tell the modem to dial the number)

DIAL ATDT 1 000 555 1212

Example #5:

Here's how I log onto CompuServe (using a CIS node - not Telenet!). I have a Macro Keys file named CIS MACROS on the disk named MYDISK that I want to load in, also.

(Set the communications parameters)

COMM 1200-N-8-1-FULL

(Load in my Macro Keys file - you may or may not want to do this)

MACRO MYDISK:CIS MACROS

(Now dial my local access number)

DIAL ATDT 555-1212

(Wait for the last part of the message CONNECT from the modem)

PROMPT NECT

(reset my elapsed time and billing clocks so I know how long)

(I've been on and how much I've spent)

RESET

(hang loose a second for the connection to get established solidly)

PAUSE

(need to send them a CONTROL-C to get started)

TYPE ^C

(CompuServe asks me for "User ID:")

PROMPT ID

(Always use a PAUSE after a PROMPT!)

PAUSE

(Type in my user ID with a carriage return at the end)

TYPE 73176,61^M

(Now CIS asks me "Password:")

PROMPT word

(Give the line a chance to settle down)

PAUSE

(I type in my password now, followed by a carriage return)

TYPE MY/SECRET^M

(The ! character is what CIS uses for a command prompt)

PROMPT !

(Getting the hang of this PROMPT/PAUSE routine, now?)

PAUSE

(Onward to MAUG's MAC USER forum!)

TYPE GO MACUS^M

(Hey human, ol' White Knight did all the work! Wake up! Wake up!)

BELL

BELL

BELL

BELL

Example #6:

I've just downloaded a neat public domain program called Space Potatoes, put it on a disk called GAMES and three of my buddies want a copy. I just instruct my modem to answer the phone, tell them to give me a call, type in the secret password GOOBER, and my Mac will send the file XMODEM. I'm going to put this Procedure on that same disk and call it BUDDIES PROCEDURE. Of course, while one guy's online, the other two have their White Knight's set up for redialing.... **Remember that in order for this to work, all of your buddies have to call in using 1200-N-8-1-FULL parameters.**

(Set up the serial port settings)

COMM 1200-N-8-1-FULL

(Turn on modem autoanswer)

AUTOANSWER ON

(I can ignore everything else until my secret password comes in!)

PROMPT GOOBER

(Tell 'em what we're about to do - notice the leading carriage)
(return to get past the password they typed and make things look clean)

TYPE ^MStarting XMODEM transfer^M

(Send that file, White Knight!)

SENDX GAMES:SPACE POTATOES

(Now hang up the phone)

HANGUP

(Start this procedure over again for whoever's left.)

DO GAMES:BUDDIES PROCEDURE

Example #7:

Redialing one service is fine, but let's get real clever and redial 4 services (or more if you wanted) in a daisy-chain fashion. When you get connected, White Knight rings like crazy till you choose **Cancel Procedure** from under the **Service** menu. This procedure assumes that the communications parameters are the same for all four services, and have been set before this Procedure is executed. The two **ALERT** commands are used to trap the modem's "NO CARRIER" and "BUSY" messages, and the **PROMPT** is used to trap the "CONNECT" message. Suggestion: you could dress this up by inserting the name of the service you are currently dialing into a string variable, and then use the **CONCAT** command to attach the name of the connected service to the displayed message. This Procedure works pretty much like the Phonebook Gang Dial feature.

:FIRST

DIAL ATDT 555-1111

ALERT1 CARR/JUMPTO SECOND

ALERT2 BUSY/JUMPTO SECOND

PROMPT NECT

JUMPTO ONLINE!

:SECOND

PAUSE

DIAL ATDT 555-2222

ALERT1 CARR/JUMPTO THIRD

ALERT2 BUSY/JUMPTO THIRD

PROMPT NECT

JUMPTO ONLINE!

:THIRD

PAUSE

DIAL ATDT 555-3333

ALERT1 CARR/JUMPTO FOURTH

```
ALERT2 BUSY/JUMPTO FOURTH
PROMPT NECT
JUMPTO ONLINE!
PAUSE
:FOURTH
PAUSE
DIAL ATDT 555-4444
ALERT1 CARR/JUMPTO FIRST
ALERT2 BUSY/JUMPTO FIRST
PROMPT NECT
:ONLINE!
COPYINTO @0,Connected to remote service!
COPYINTO @1,Please select "Cancel Procedure" from under the Service
COPYINTO @2,menu to go online.
@ ON
SHOW@
:LOOP
BELL
PAUSE
JUMPTO LOOP
```

Example #8:

The following code fragment can be added to the beginning of a Procedure File to make sure that only those who know the password (in this case, the word "SECRET") can execute it. If the Procedure is compiled with encryption, a routine like this can make the file impervious to compromise.

(Present a dialog box asking for a password)

```
COPYINTO X$,This procedure is limited to authorized users
```

```
ERASE Y$
```

```
COPYINTO Z$,Please type in your password
```

```
QUERY1 A$
```

(See if the user actually typed something)

```
EMPTY A$
```

```
IF YES JUMPTO ABORT
```

(Convert what he typed to all uppercase)

```
CONVUP A$
```

(See if it contains the password "SECRET")

```
CONTAINS SECRET
```

```
IF NO JUMPTO ABORT
```

```
JUMPTO CONTINUE
```

:ABORT

(This is what happens if he fails to type the correct password)

END

:CONTINUE

(Rest of Procedure begins here)

Example #9:

You may be wondering how to set the YES/NO or ERROR flags to a known state. Here's how to do it:

(To set the YES/NO Flag to "YES":)

LET EQUAL A%,1

TEST A% = 1

(To set the YES/NO Flag to "NO":)

LET EQUAL A%,1

TEST A% = 2

(To set the ERROR Flag to "ERROR":)

LET EQUAL A%,1

DIVIDE A%,0

(To set the ERROR Flag to "NO ERROR":)

LET EQUAL A%,1

DIVIDE A%,1

Example #10:

NULL duplex sends text to the screen but not out the serial port. We take advantage of this in our Procedures to print out information to the user without it being sent out.

(Let's say we we're using FULL duplex. Let's print out the current)

(time and date for the user.)

COMM -NULL

TIMEDATE A\$

TYPE ^MThe current time and date:

TYPE A\$

TYPE ^M

COMM -FULL

Example #11:

Now let's get really sneaky with what we did in Example #10 and use that strategy to send a line of information to the printer.

(Wait for a connection)

PROMPT CONNECT

(Save the current time)

TIMEDATE A\$

(Don't send data out the port)

COMM -NULL

(Turn on the printer echo - sneaky!)

ECHO ON

(Type the message - there's a trailing space at the end.)

TYPE Logged into CompuServe at:

TYPE A\$

TYPE ^M

(Hate to waste a whole sheet of paper on one line, but you get)

(the idea. Now turn off printer echo and restore the duplex.)

ECHO OFF

COMM -FULL

Example #12:

Here's an example of how to react to the flag settings after a DIAL or REDIAL command, of if the Procedure was executed by a Phonebook entry. In this example, I'll use a REDIAL command.

(Redial the number)

REDIAL LIMIT 15

REDIAL ATDT 555-1212

IF YES JUMPTO YESON

IF ERROR JUMPTO ERRON

(If I get here, the YES/NO flag is set to "NO" and the ERROR flag)

(is set to "NO ERROR", meaning the REDIAL was cancelled by the user)

END

:ERRON

(If I get here, the YES/NO flag is set to "NO" and the ERROR flag)

(is set to "ERROR", meaning we exceeded the redial limit without)

(a connection.)

END

:YESON

IF NO ERROR JUMPTO ERROFF

(If I get here, the YES/NO flag is set to "YES" and the ERROR flag)

(is set to "ERROR", meaning the modem is not responding to the

(dialing command properly.)

END

:ERROFF

(If I get here, the YES/NO flag is set to "YES" and the ERROR flag)

(is set to "NO ERROR", meaning a connection has been established.)

Example #13:

Sometimes during Procedure development, it's useful to get a listing of the contents of string and/or numeric variables. For string variables, I usually bracket the string with '<' and '>' characters to show any leading or trailing spaces. In this example, I'll show how to list the contents of the string variable A\$ and the numeric variable Z%.

(Set up variables with sample values)

```
COPYINTO A$,This is the string variable A$
```

```
LET EQUAL Z%,53
```

(Turn on NULL duplex so output goes only to my screen and not)
(out the serial port)

```
COMM -NULL
```

(Print out contents of A\$)

```
TYPE The variable A$ contains <
```

```
TYPE A$
```

```
TYPE >^M
```

(Must convert Z% to a string variable so I can print it out)

```
NUMTOSTRING Z%,B$
```

(Value is now in B\$, print it out in same fashion)

```
TYPE The variable Z% contains the number <
```

```
TYPE B$
```

```
TYPE >^M
```

Memory Considerations

Because White Knight is an expandable program, there are no cut and dry rules as to how much free memory it requires for execution. Although White Knight is roughly 360K in size, there are several things which require additional memory:

- The Display Buffer. Each screen (24 lines) saved in the Display Buffer requires an additional 6480 bytes.
- RCMD's. An RCMD is executed entirely from memory. Therefore, there needs to be at least enough free memory to hold the RCMD itself, plus enough free memory to suit additional needs of the RCMD.
- The Serial Port Buffer. There must be enough free memory to hold the Serial Port Buffer, the size of which is configurable by choosing **Local->Serial Port**.
- Approximately 100K of program overhead. This is a wide variety of internal settings, plus things like filters and Procedure variables.

When running under MultiFinder, you can adjust the "chunk" of memory allocated to White Knight by clicking once on White Knight's icon, then choosing **Get Info** from under Finder's **File** menu. Type in the desired amount next to the prompt "Application memory size (K)".



Do not give White Knight less than 500K of free memory under any circumstance.

The default application memory size is 1000K, which is enough to hold 35-40 screens in the Display Buffer plus leave a comfortable amount of free memory for other needs.

About Filenames

In various places throughout this manual, the word **filename** is used to indicate a group of characters that describe the location of a file (either a Macintosh application or document) on a disk. There are different ways of doing this, depending on the kind of filing system you are using.

The original Macintosh filing system was called MFS (Macintosh Filing System), which worked well for floppy disks. Folders that are displayed in a Finder window are for cosmetic purposes only, and they do not play a part in constructing a filename for White Knight. An MFS filename consists of the disk volume name (which appears under the disk icon in the Finder), a colon character, and the name of the file (which appears under the file's icon in the Finder).

Under MFS, a file with the name of "Junk" that resides on a disk named "Stuff" would be given the filename "Stuff:Junk"

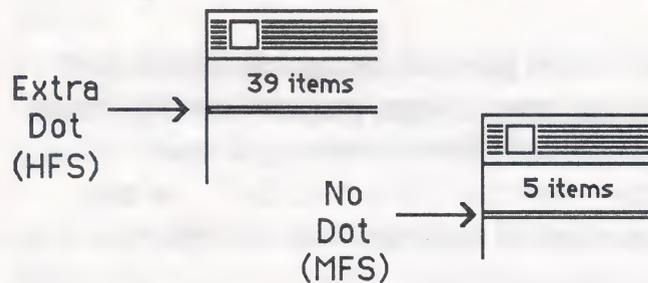
When hard disks became common in the Macintosh world, it became necessary for Apple to devise a more efficient filing system, which they named HFS (Hierarchical Filing System). Under HFS, the names of folders which lie between the desktop (the top level window in the Finder) and the file are significant and must be included in the filename in the order they appear and separated by colon characters. An example should clarify this.

Under HFS, a file named "Junk" is inside of a folder named "Garbage". The "Garbage" folder is inside of another folder named "Trash". The "Trash" folder lies on the desktop of a disk named "Stuff". The proper filename would then be: "Stuff:Trash:Garbage:Junk"

How do you know if a disk uses HFS or MFS? If you are using a hard disk on a Mac Plus or above, it is almost certainly HFS (I've never seen an exception).

If you are using an 800K (double-sided) or larger floppy disk, it is also almost certainly HFS. To find out the format of a 400K (single-sided) floppy disk, open the disk icon in the Finder and look at the upper left hand corner of the window. There is a small division between the two lines that appear just below the phrase that tells you the number of items on the disk. A HFS disk will have an extra dot at the far left of this division, and a MFS disk will have no dots in this

division. The illustration below shows an example of both kinds of disks.



It's important to know what kind of disks you are working with so that you don't construct a faulty filename and get the dreaded "File Not Found" error message.

One last tip. If you are sure that you've spelled everything correctly and are positive a filename should work but doesn't, you should be aware that the Finder allows you to (inadvertently or otherwise) put leading or trailing spaces in disk, folder, and file names. If these are present, they must be included in the filename you present to White Knight. So, if problems persist, try renaming the entities in a filename to make sure none of these "ghost" spaces exist.



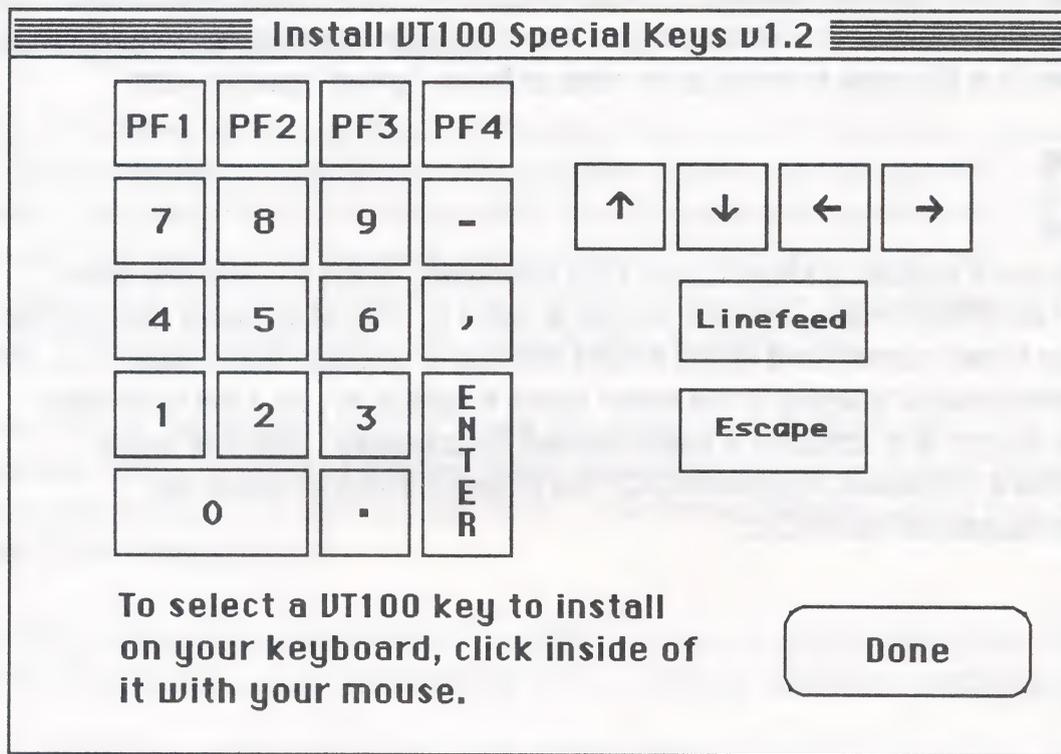
There are a couple of shortcuts for HFS filenames. If the file is in the same folder as White Knight, you can just put a colon in front of its name (without the disk or folder names) and White Knight will find it. A colon at the beginning of a filename means "starting at the folder White Knight is in", so if the folder that White Knight is in contains a folder named "Procedures", and that folder contains a file called "GENie.PROC", the shortcut filename would be ":Procedures:GENie.PROC".

The "Install Special Keys" Utility

If you are using a keyboard that White Knight does not recognize, or just don't like the way White Knight maps the VT100 special keys, you can use the "Install Special Keys" utility on the White Knight master disk to define your own keyboard mapping. This should be done after the "WK's 11.0 Stuff" file has been created, but before you create any further Settings Files.

To begin, start up White Knight and choose **Customize->Options->Key Mapping**. In the dialog box that appears, click on the "Other" choice for the "Keyboard" item. Click on the "OK" button and then choose **File->Quit**.

Start up the "Install Special Keys" utility, and then select the "WK's 11.0 Stuff" file, which will be in the same folder as White Knight. Then choose **File->Install VT100 Keys**. The following dialog box will be displayed:

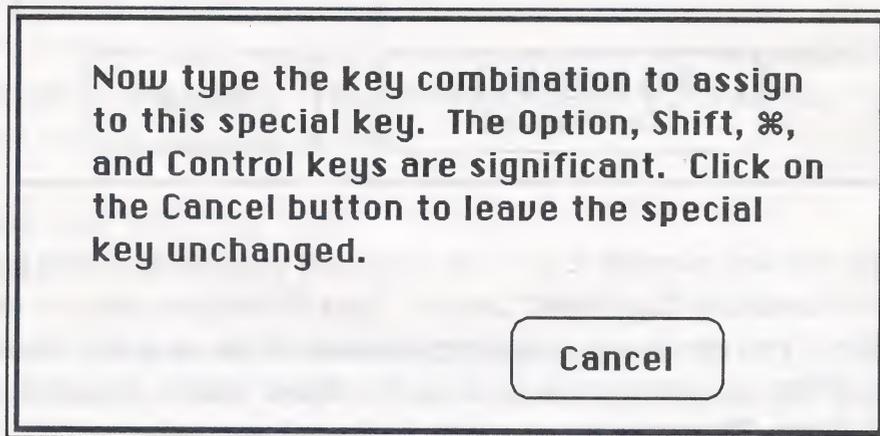


The keys in the picture correspond to those on an actual VT100/102 keyboard. What we need to do is tell White Knight which keys on your keyboard correspond to which keys in the picture.



It's very important that you install all of the keys in the picture, not just a few. Those keys not installed will be undefined and therefore unavailable. I've found it useful to use Shift-Return for the "Linefeed" key.

Let's start with the up-arrow key. Position your mouse inside the square containing the up-arrow in the picture and click. The following dialog box will be displayed:



Now, type the key on your keyboard that you wish to assign the up-arrow function. If you have an up-arrow key on your keyboard, you'll probably want to use that. If you change your mind, just click in the "Cancel" button before typing any keys. As the dialog box mentions, the Option, Shift, Control, and ⌘ keys can also be used. You could use Shift-Tab for instance, and the unshifted Tab key would not be affected.

Once the key is typed, the dialog box disappears and the up-arrow key in the picture is greyed out (which tells you it has been defined. Go through now and click and type for the rest of the keys in the picture.

Once you're finished, choose **File->Quit - Save Changes** and the installation is complete. If you change your mind, select **File->Quit - Don't Save Changes** and everything you did will be ignored.

The "Install Special Keys" utility can also be used to set up keyboard equivalents for Macro Keys, just like in the "Macro Key Selection" dialog box in White Knight. To do this, select **File->Install Macro Keys**. The following

dialog box is shown:

Select Macro Key To Edit	
Set Number:	Macro Key Number:
<input checked="" type="radio"/> 1	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
<input type="radio"/> 2	<input type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 <input type="radio"/> 8 <input type="radio"/> 9
<input type="radio"/> 3	
<input type="button" value="Set Keystroke Equivalent"/> <input type="button" value="Done"/>	

Select the set and key number to give the keyboard equivalent to and then click on the "Set Keystroke Equivalent" button. Type the key you wish to assign to that Macro Key. You can assign keyboard equivalents for as many Macro Keys as you wish. When you're finished, click on the "Done" button and then choose **File->Quit - Save Changes** to save the keyboard equivalents.

The GETPARAM And PUTPARAM Procedure Commands



The information in this chapter is meant only for very experienced White Knight users!

The **GETPARAM** and **PUTPARAM** Procedure commands allow you to get at the "guts" of White Knight to perform operations normally not possible through other Procedure commands. The **GETPARAM** and **PUTPARAM** Procedure commands use the following syntax:

```
GETPARAM NUM_EXP,NUM_VAR  
PUTPARAM NUM_EXP1,NUM_EXP2
```

These commands are used to examine or modify one of many internal memory parameters used by White Knight. These internal parameters are referenced by a number which corresponds to their relative offset from the beginning of the default Settings File. The **GETPARAM** command is supplied with the internal parameter number in NUM_EXP, and the contents of that internal parameter (which is always numeric) is placed in the numeric variable designated by NUM_VAR. In other words, to get the current value in the internal parameter #100 and put it into the numeric variable A%, you would use the Procedure command:

```
GETPARAM 100,A%
```

The **PUTPARAM** command is used to modify the contents of an internal parameter (designated by NUM_EXP1) with the contents of NUM_EXP2. To change the contents of internal parameter #100 to a value of 15, you would use the Procedure command:

```
PUTPARAM 100,15
```

Because the **PUTPARAM** command modifies things that were never originally meant to be diddled by the user (but I have yielded for the benefit of those who understand how to handle digital dynamite), it is inherently dangerous. Never

change an internal parameter not listed below, and modify those only with the values designated in the list. Don't use a **PUTPARAM** to modify any parameters that are not listed below with the words "Use **PUTPARAM** to modify."

Perhaps this warning is making a mountain out of a molehill, but the problem is that these internal parameters are saved in Settings Files, and a wrong value can therefore propagate to other Settings Files without warning. If you plan to distribute Procedures that use the **PUTPARAM** instruction, please give thought to a beginner and display a proper warning message to the user before modifications are done (and please give him a chance to bail out if he gets scared!).

Think of it this way. Using the **PUTPARAM** command is like using ResEdit on a System file. If you understand that and the implications of making a mistake, don't be afraid to plunge ahead.

When there is an equivalent Procedure command, use that in favor of a **PUTPARAM** command, as the equivalent is guaranteed to be safer. For some things, like finding out what baud rate or duplex is being used at a given moment, there are no equivalent Procedure commands and a **GETPARAM** command must be used. Some operations can only be done with the **PUTPARAM** command. You can use the **GETPARAM** command on any of the internal parameters listed below, but, again, unless a listing below specifically states "Use **PUTPARAM** to modify", don't use the **PUTPARAM** command with it. By the way, you will find "holes" in the list below. Some internal parameters are of absolutely no interest to you, or are reserved for future expansion.

You'll probably never need any of this stuff. However, I have found a few things very useful, and if I've learned anything in the last 10 versions of White Knight, it's not to second guess your hunger for information and clever ways to use it. The following is presented with the forewarning that you need to know White Knight inside and out to make sense of it, and programming experience (when high and low bytes are discussed) may be necessary to use a few of these.

Parameter #0: Current baud rate. Use **COMM** command to modify.

Meaning: 0 = 300, 1 = 450, 2 = 1200, 3 = 2400, 4 = 4800,
 5 = 9600, 6 = 19200, 7 = 38400, 8 = 57600

Parameter #1: Current parity. Use **COMM** command to modify.

Meaning: 0 = None, 1 = Odd, 2 = Even, 3 = Mark, 4 = Space

5 = Ignore

Parameter #2: Current Databits. Use **COMM** command to modify.

Meaning: 0 = 7 bits, 1 = 8 bits, 2 = 5 bits, 3 = 6 bits

Parameter #3: Current stopbits. Use **COMM** command to modify.

Meaning: 0 = 1 bit, 1 = 2 bits, 2 = 1.5 bits

Parameter #4: Current duplex. Use **COMM** command to modify.

Meaning: 0 = Full, 1 = Half, 2 = Echo, 3 = Null

Parameter #6: XMODEM, YMODEM, ZMODEM timeout in seconds. Use **PUTPARAM** to modify.

Meaning: 1 - 255 (seconds)

Parameter #7: ASCII char for control button #1 on General Status Bar.

Parameter #8: ASCII char for control button #2 on general status bar

Parameter #9: ASCII char for control button #3 on general status bar

Use **CONTROL1**, **CONTROL2**, or **CONTROL3** commands to modify.

Meaning: (0-255) ASCII value of character button sends.

Parameter #10: Current number of screen columns. Use **DISPLAY** command to modify.

Meaning: (20-132) Number of columns

Parameter #11: Current terminal emulation type. Use **TTY**, **VT52**, **VT100**, or **VT102** commands to modify.

Meaning: 0 = TTY, 1 = VT100, 2 = VT52, 3 = VT102

Parameter #13: Current Status Bar. Use **GBAR**, **BBAR**, **MBAR**, **VBAR**, **LOUD**, or **QUIET** commands to modify.

Meaning: 1 = General, 2 = Buffered Keyboard, 3 = Macros,
4 = VT100, 5 = Monitoring procedure

Parameter #14: Should cursor flash? Use **PUTPARAM** to modify.

Meaning: 0 = no flash, 1 = flashing

Parameter #15: Cursor shape. Use **PUTPARAM** to modify.

Meaning: 0 = underline, 1 = block

Parameter #16: Amount of delay between each character (in 60ths of a second) sent with a **SEND** Procedure command or **Send TEXT File** menu command if Parameter #27 is nonzero and Parameter #32 = 1. Use **PUTPARAM** to modify.

Meaning: (0 - 255) number of 60th's of a second to delay.

Parameter #18: Backspace key mapping. Use **DELKEY** command to modify.

Meaning: 0 = backspace (ASCII 8), 1 = DElete (ASCII 127)

Parameter #19: Active set when macros status bar is showing, or when Macro Key is executed from keyboard (using COMMAND-number). Use **MBAR** command to modify.

Meaning: 1, 2, or 3 (active set number)

Parameters #20 through 23: Four ASCII characters to use as a file type (default MACA) for File Captures, Archive File, and protocol non-MacBinary file receives. Use **PUTPARAM** to modify.

Meaning: Each parameter contains ASCII value of character.

Parameter #24: Send and respond to auto-receive sequence? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #25: What to do if received MacBinary file exists? Use **PUTPARAM** to modify.

Meaning: 0 = destroy old file, 1 = create unique filename

Parameter #27: Delay after each character or line (depending on Parameter #32) sent with a **SEND** Procedure command or **Send TEXT File** menu command. Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #28: Should file sent with **SEND** Procedure command or **Send TEXT File** menu command be forcefully wrapped at a certain column? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #29: Column number to wrap line if Parameter #28 is nonzero. Use **PUTPARAM** to modify.

Meaning: (20-132) column number

Parameter #31: Using MultiFinder? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #32: Kind of delay for **SEND** procedure command or **Send TEXT File** menu command if Parameter #27 is nonzero. Use **PUTPARAM** to modify.

Meaning: 0 = delay between lines sent, 1 = delay between characters sent

Parameter #33: Enable CompuServe 'B' protocol recognition? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #34: Supercharged XMODEM? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #36: Enable RLE graphics recognition? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #38: Kermit uses special ^Q line turnaround handshake? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #39: Should VTMouse delay? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #40: Should ` key send the ` character or the ASCII code in Parameter #145? Use **ALT** command to modify.

Meaning: 0 = send ` character, 1 = send ASCII code in Parameter #145.

Parameter #41: Should Procedures be monitored in status bar? Use **LOUD** and **QUIET** commands to modify.

Meaning: 0 = no, 1 = yes

Parameter #42: Should date/time heading be printed on the printer? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #43: If Parameter #42 is nonzero, should date/time be printed on every page or first page only. Use **PUTPARAM** to modify.

Meaning: 0 = every page, 1 = first page only

Parameter #44: Should printer form feed when the number of lines in Parameter #45 has been printed? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #45: If Parameter #44 is nonzero, do form feed after this number of lines. Use **PUTPARAM** to modify.

Meaning: (1-255) Number of lines before form feed

Parameter #46: Wait for character in Parameter #488 after each line sent by **SENDA** procedure command or **Send TEXT File** menu command? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #48: Is data currently being echoed to the printer? Use **ECHO** command to modify.

Meaning: 0 = no, 1 = yes

Parameter #49

Parameter #50: These two bytes contain the cost (in 100ths of a cent) for the billing clock to increment after each elapsed minute. It is a two byte integer value, so Parameter #49 holds the high byte value, and Parameter #50 holds the low byte value. Use **SETCOST** to modify.

Parameter #54: Should protocol non-MacBinary receives use the File Transfer Filter? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #55: At end of file transfer, ring bell this number of times. Use **PUTPARAM** to modify.

Meaning: 0-255 times

Parameter #56: XMODEM style. Use **PUTPARAM** to modify.

Meaning: 0 = classic, 1 = CRC, 2 = 1K blocks

Parameter #57: YMODEM style. Use **PUTPARAM** to modify.

Meaning: 0 = classic, 1 = 1K blocks, 2 = YMODEM-G

Parameter #60 = Should the following functions include blank lines in their output? Use **PUTPARAM** to modify.

- 1) **Local->Window->Archive** menu command
- 2) **Local->Buffer->Archive** menu command
- 3) "Display Screen To Archive File" button
- 4) "Display Screen To Printer" button
- 5) SCREENDISK Procedure command.
- 6) SCREENPRINT Procedure command.

Meaning: 0 = yes, 1 = no

Parameter #61: Should the **Delete A File** menu choice skip the confirming dialog after a file is chosen? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #62: Should the Delete A File menu choice continually cycle through the select and delete process until the user clicks on the file selection dialog's "Cancel" button? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #63: Should the **Edit->Locate Text** menu choice use a zooming rectangle when the window scrolls? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #65: Display title screen on program startup? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #66: Should "Display Screen To Printer" button on General Status Bar confirm with a dialog box that the user actually wants to do that? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #67: Should "Display Screen To Archive File" button on General Status Bar confirm with a dialog box that the user actually wants to do that? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #68: Should Close Box in Terminal Window confirm with a dialog box that the user actually wants quit? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #69: Should "Reset Elapsed time clock and billing timer" button on General Status Bar confirm with a dialog box that the user actually wants to do that? Use **PUTPARAM** to modify.

Meaning: 0 = yes, 1 = no

Parameter #70: If a Procedure File is executed by a macro key while another Procedure File is executing, should we put up a dialog telling the user to cancel the executing Procedure, or should we cancel it automatically and execute the new Procedure? Use **PUTPARAM** to modify.

Meaning: 0 = dialog, 1 = auto-execute

Parameter #71: At end of MacBinary file transfer. Use **PUTPARAM** to modify.

Meaning: 0 = don't rename, 1 = rename

Parameter #72: Use paragraph format for non-MacBinary file receives. Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #73: Maximum packet size classic Kermit wants. Use **PUTPARAM** to modify.

Meaning: (1 - 94) maximum packet size in bytes.

Parameter #74: Kermit's start of header character. Use **PUTPARAM** to modify.

Meaning: (0-255) ASCII code for start of header character.

Parameter #75: Kermit style. Use **PUTPARAM** to modify.

Meaning: 0 = classic, 1 = long packet, 2 = sliding windows.

Parameter #76

Parameter #77: These two bytes contain the maximum packet size for long packet Kermit. It is a two byte integer value, so Parameter #76 holds the high byte value, and Parameter #77 holds the low byte value. Use **PUTPARAM** to modify.

Parameter #78: Number of windows for Sliding Windows Kermit. Use **PUTPARAM** to modify.

Meaning: 1-33 windows.

Parameter #79: Treat outgoing Kermit files as. Use **PUTPARAM** to modify.

Meaning: 0 = text, 1 = binary

- Parameter #80: Kermit end of packet character. Use **PUTPARAM** to modify.
Meaning: (0 - 255) ASCII code for end of packet character.
- Parameter #81: Terminal Emulator uses Terminal Filter? Use **PUTPARAM** to modify.
Meaning: 0 = none, 1 = pre-emulation, 2 = post-emulation, 3 = both
- Parameter #82: File captures use File Capture Filter? Use **PUTPARAM** to modify.
Meaning: 0 = no, 1 = yes
- Parameter #83: Number of seconds White Knight's Kermit wants remote Kermit to wait before timing out (default is 3). Use **PUTPARAM** to modify.
Meaning: (1 - 255) seconds to wait
- Parameter #84: **Edit->Locate Text** is case-insensitive? Use **PUTPARAM** to modify.
Meaning: 0 = no, 1 = yes
- Parameter #85: Do form feed after printing finishes? Use **PUTPARAM** to modify.
Meaning: 0 = no, 1 = yes
- Parameter #88: Current font number. Use **FONT** command to modify.
Meaning: (0 to 255) font number
- Parameter #89: Current font size. Use **FONT** command to modify.
Meaning: (1 to 255) font point size
- Parameter #90: High byte of maximum symbol definitions.
Parameter #91: Low byte of maximum symbol definitions. Use **PUTPARAM** to modify either of these.
- Parameter #92: High byte of maximum symbol references.
Parameter #93: Low byte of maximum symbol references. Use **PUTPARAM** to modify either of these.
- Parameter #94: Does the compiler encrypt Procedure Files? Use **PUTPARAM** to modify.
Meaning: 0 = no, 1 = yes
- Parameter #96: Show compiler window during compilation? Overridden by **COMPILE LOUD** and **COMPILE QUIET** commands. Use **PUTPARAM** to modify.
Meaning: 0 = no, 1 = yes
- Parameter #97: Should the DTR line be dropped when exiting White Knight?

Use **PUTPARAM** to modify.

Meaning: 0 = drop DTR, 1 = don't drop DTR

Parameter #142: Do we have an additional VT100 font? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #143: If Parameter #142 is nonzero, additional VT100 font high byte. Use **PUTPARAM** to modify.

Parameter #144: If Parameter #142 is nonzero, additional VT100 font low byte. Use **PUTPARAM** to modify.

Parameter #145: If Parameter #40 is nonzero, this contains the ASCII code that that the ` key will send. Use **PUTPARAM** to modify.

Meaning: (0 - 255) ASCII code of character to send

Parameter #146: Shift-backspace sends. Use **PUTPARAM** to modify.

Meaning: 0 = DEL, 1 = Short Break, 2 = Backspace

Parameter #424: Does Option key send control characters? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #466: VT emulation ignores bells (ASCII code 7)? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #467: VT emulation scrolls up lines before full screen clear? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #468: TTY emulation scrolls up lines before full screen clear? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #469: TTY emulation ignores bells (ASCII code 7)? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #470: TTY emulation ignores full screen clears? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #473: Dial phone using touch tone or pulse? Use **PUTPARAM** to modify.

Meaning: 0 = tone, 1 = pulse

Parameter #475: How long to wait after dialing for a connection. Use **PUTPARAM** to modify.

Meaning: 1-255 seconds

Parameter #476: Modem command intercharacter delay. Use **PUTPARAM** to modify.

Meaning: 0-255 sixtieth seconds

Parameter #488: ASCII character to wait for after each line sent with a **SEND A** Procedure command or **Send TEXT File** menu command if Parameter #46 is nonzero. Use **PUTPARAM** to modify.

Meaning: (0 - 255) character to wait for

Parameter #489: Amount of delay after each line (in seconds) sent with a **SEND A** Procedure command or **Send TEXT File** menu command if Parameter #27 is nonzero and Parameter #32 is zero. Use **PUTPARAM** to modify.

Meaning: (1 - 255) delay time in seconds

Parameter #490: Do final report after file transfer? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #492: Current serial port in use. Use **MODEM** and **PRINTER** commands to modify.

Meaning: 0 = printer, 1 = modem

Parameter #493: Should the Return key send a carriage return only or a carriage return followed with a linefeed? Use **LF** command to modify.

Meaning: 0 = carriage return only, 1 = CR/LF

Parameter #495: Use MacBinary format for. Use **PUTPARAM** to modify.

Meaning: 0 = all files except TEXT, 1 = all files, 2 = no files

Parameter #500: Number of seconds to delay after each soft carriage return (Shift-Return) in the Buffered Keyboard is sent.

Meaning (0 - 255) seconds delay.

Parameter #504: Is a Macintosh Plus keyboard (with built-in numeric keypad) currently in use? DO NOT MODIFY!

Meaning: 0 = no, 1 = yes

Parameter #505: Redial attempt limit. Use **REDIAL LIMIT** command to modify.

Meaning: (0 - 255) attempts

Parameter #507: VT100 relative origin mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #508: VT100 smooth scroll mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #509: VTmouse waits for host? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #585: VT100 Wraparound mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #586: VT100 Autorepeat mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #587: VT100 Newline mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #588: VT100 Cursor key mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #589: VT100 Keypad mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #590: Remember Data Buffer between sessions? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #591: VT102 Insert mode. Use **PUTPARAM** to modify.

Meaning: 0 = reset, 1 = set

Parameter #592: Allow resumption of interrupted ZMODEM transfers? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #597: Use paragraph format conversion for file captures? Use **PUTPARAM** to modify.

Meaning: 0 = no, 1 = yes

Parameter #658: ZMODEM sends this sequence before sending a file. Use **PUTPARAM** to modify.

Meaning: 0 = ESC-Z (White Knight auto-receive),
1 = rz^M (Unix), 2 = "uz^M" (Opus)

Using International Or Special Fonts With VT Emulation

White Knight can use any font installed in your Macintosh in any point size when using TTY emulation. However, if you need to use international, diacritical, or special characters (those typed in conjunction with the Option or Shift-Option keys) with VT52, VT100, or VT102 emulation, some minor surgery needs to be done on White Knight. It will not require rubber gloves, just a simple short Procedure File. Therefore, if you're not familiar how to create, compile, and execute a Procedure File, please refer to the chapters covering this before proceeding.

We will use the **PUTPARAM** Procedure command to change 3 bytes in the Settings File. Parameter number 142, when non-zero, tells White Knight that an additional font is to be listed under the **Font** menu when the **Local->Window->Font** choice is selected when VT emulation is being used. The next two bytes (143 and 144) contain the high and low values of the font number to display in the menu. You can get the proper font number for these values by selecting the font while using TTY emulation, and then choosing **Other Point Size** under the **Size** menu - the font number is displayed at the top of the Other Point Size selection dialog box.

To get the high and low values to put into bytes 143 and 144, take the font number and divide it by 256. This is the high value. Next, take the high value, multiply it by 256, and subtract that number from the font number. The remainder is the low value. Unless you're a programmer, this is all more than likely gobbledegook (which I dare any spelling checker to recognize), which is alright, because unless you're a hacker-type into creating your own fonts, I would highly suggest you use the Monaco font, which is monospaced, comes in 9 and 12 point sizes, and contains all of the international, diacritical, and special characters you're likely to ever need.

If you are a font hacker, the only thing you need to know is that your font must be in one of the sizes that are available for the TTY-VT52-VT100 font, and that your font must be monospaced (every character has the same width) to the same width as the TTY-VT52-VT100 font in the desired size.

To install the Monaco font, the following Procedure is compiled and executed:

(Tell White Knight we've got an additional font for VT emulation)

PUTPARAM 142,1

(Monaco font is number 4, so the high value is zero)

PUTPARAM 143, 0

(The low value is 4)

PUTPARAM 144,4

If you decide later on to get rid of this, you can compile the following Procedure:

(Tell White Knight not to use any additional VT52/VT100 font)

PUTPARAM 142,0

When an additional font is installed, White Knight will use that font for all characters except:

- 1) Boldfaced ASCII characters
- 2) Special VT Graphics Characters or U.K. character set characters

in which case it will use the correct characters contained in the TTY-VT52-VT100 font.

RCMD's: Extending White Knight

Unlike most software that you'll purchase for your Macintosh, White Knight has the ability to be internally expanded to offer new features. Extensible software means fewer radical rewrites to add new features - it is a much simpler process to add a small module that implements a new feature. A module of this type is known as an "RCMD" (pronounced "are-command").

RCMD's are to White Knight what XCMD's are to Hypercard. In fact, an RCMD could be written (and probably will be) to execute Hypercard XCMD's!

The possibilities of RCMD modules are limitless. Their function can range from something very simple to something very complex. You've already see one RCMD module - the procedure editor ProcEdit. As we mentioned, ProcEdit is not built into White Knight, it is an entirely separate entity.

When I started playing around with the possibilities of RCMD modules, I constructed a bunch of sample RCMD's (including ProcEdit) that did a lot of different things. Seven of these are included on your White Knight master disk, and more will be released through the FreeSoft RoundTable on GENie in the months to come. I am convinced now that this feature may prove to be one of White Knight's strongest. In the months to come, I think you'll see a flood of third-party RCMD's distributed that will add more power to White Knight 11 than I ever dreamed of.

Using an RCMD module is easy. You'll notice that the RCMD modules included on your Master Disk have the same icon as a compiled Procedure File. This is because a part of the module is a Procedure command that executes the module. Therefore, an RCMD module is executed just like a compiled Procedure File, by selecting **Service->Initiate Procedure** and then selecting the desired RCMD module.

Writing an RCMD module is not easy. You would need to be a competent Macintosh programmer who can use a language like C, Pascal, or assembly language (actually, any language that can create "CODE" resources - if you're a Mac programmer, you know what that means). The documentation that describes what you need to know in order to write an RCMD module is included in the "White Knight Developer's Toolkit", which is discussed in the chapter "Even More White Knight Features".

The sample RCMD modules supplied with White Knight 11 are really just that - samples. You may find any or all of them useful or worthless, depending on your needs.

AddLF

The RCMD module "AddLF.PROC" is used to add linefeeds after carriage returns in a text file. Although many Macintosh word processors will correctly put carriage returns at the end of every line when saving a file as plain text, few will insert carriage return and linefeed pairs, which is what many non-Macintosh computers need.

When you execute AddLF, you will first be prompted for the source text file that you wish to add linefeeds to. After selecting this file, you will then be asked to name the resulting file. AddLF does not change the original file, but creates a second file with the linefeeds added. Once you've named the target file, a dialog box with a progress ruler (showing the percentage of the conversion completed) will be displayed and AddLF will go to work.

The only limitation to AddLF is that each line in the source file must be 199 characters or less.

FastDump

The RCMD module "FastDump.PROC" replaces the FASTDUMP Procedure command in Red Ryder version 10.3. Essentially, this is a "fastest possible" data capturing tool for those times when the sending machine does not understand XON/XOFF handshaking. FastDump does not do any handshaking, and does not strip parity bits (therefore, it's only useful with NONE or IGNORE parity).

After executing FastDump, you will be prompted to name the file that incoming data will be captured to. After you've done this, all data received over the serial port will be saved in that file. You won't see the data coming over, but the mouse cursor will change from an arrow to a watch when data is being received.

To stop the capture and close the file, simply click your mouse button.

FolderBatch

The RCMD module "FolderBatch.PROC" is used to quickly construct a Batch

File (for sending with YMODEM, ZMODEM, Kermit, or Flash protocols) containing all of the files residing in a selected folder. After executing FolderBatch, you will be first prompted to select the desired folder. Next, you will be asked to name the resulting batch file. FolderBatch will then create that batch file containing all of the filenames of the files in the selected folder.



FolderBatch will not include any files in folders (sub-folders) that are contained in the target folder.

QuickB

This RCMD Module is used to provide support for CompuServe's QuickB file transfer protocol. It uses the same interface as White Knight's built in file transfer protocol, and supports all of the settings in the dialog box brought up by selecting **Customize->Options->File Transfer**. It supports both uploading and downloading automatically.

To use the "QuickB.PROC" module, it should be linked into a Macro Key (you can choose **Service->Initiate Procedure** to execute it, but that's a bit time consuming). When you have told CompuServe to send you a file, it will prompt you to select the protocol to use. Choose "QuickB" protocol. Then, CompuServe will prompt you for "Filename for your computer:". Type in a filename you wish the file to be created with, and then press your Return key. Immediately after pressing the Return key, execute the QuickB module.

Uploading is similar, except that when you get the "Filename for your computer:" prompt, you should type in the full pathname of the file to send (including the disk volume and folder names - see the chapter "About Filenames" if you're not familiar with this process).

SetFile

This RCMD module is used to change the Finder Information for a file. This includes the file's type and creator signatures, as well as all of the Finder Flags.



Changing Finder Information without knowing what you are doing is the equivalent to performing brain surgery with a butter knife. If you don't know

what a particular item does, leave it alone or you could cause irreparable damage to the file.

Once you've executed SetFile.PROC, you will be prompted to select the file you wish to view or modify. Then, all of the Finder Information will be displayed. You can click on the "OK" button to save any changes you've made, or click on the "Cancel" button to bail out without saving any changes.

Speak

This RCMD module is used to interface White Knight with Macintalk, Apple's unsupported speech synthesis software.



To use "Speak.PROC", you must have the "Macintalk" file in your System folder. Many Bulletin Board Services and Macintosh User Groups can provide you with this file. Macintalk does not perform exceptionally well on the newer Macintosh models (Mac II, for instance), and Apple has not given any indication that they will release any further versions.

Speak.PROC is never executed directly. You must first set up a few variables and then call it using a Procedure NEST command. The variables that need to be set up are:

T\$ - this contains the text to be spoken.

P% - this contains the pitch value. If it is zero, a default pitch value of 150 is used.

R% - this contains the speed rate value. If it is zero, a default rate of 125 is used.

M% - this should be either 0 for natural speech or 1 for robotic speech.

F\$ - this contains the filename of the Macintalk "Exceptions File" to use. If it is empty, no Exceptions File will be used.

Here is a sample Procedure which illustrates how to use Speak.PROC:

```
(Speak the text in T$)
```

```
COPYINTO T$,This is a test of speech in White Knight.
```

```
(No exceptions file)
```

```
ERASE F$
```

```
(Use default pitch and rate)
```

```
LET EQUAL P%,0
LET EQUAL R%,0
(Speech mode is "natural")
LET EQUAL M%,0
(Call the Speech RCMD - below filename works if Speak.PROC is in)
(the same folder as White Knight)
NEST :Speak.PROC
END
```

TabFile

This RCMD is used to do to a text file what **Edit->Copy Table** does to selected text in the Data Area of the Terminal Window. It replaces all occurrences of more than once space in a row with a tab character. "Tab delimited" data is much easier to import to a spreadsheet or database program.

To use "TabFile.PROC", execute it and select the source text file. Next, you will be prompted to name the destination file. The source file is not changed by TabFile, the destination file contains the tab delimited data.

While TabFile is working, a progress ruler is shown which displays the percentage of the conversion completed.

Troubleshooting And Technical Support

The FreeSoft Company is pleased to offer technical support to our customers. However, because of the volume of calls that we get, we must ask that you adhere to the following before contacting us:

- 1) Read the manual. This sounds trite, but we just can't afford the time to read the manual to you. If you have read the manual and still can't solve your problem:
- 2) Check the index for possible references to the area of your problem.
- 3) Read the rest of this chapter to see if your problem is discussed here.

If all of the above fails, you're ready to give us a call. Here's what you'll need to do before you call:

- 1) You'll need to have your Macintosh next to the phone and turned on.
- 2) You'll need to start up White Knight (if possible - if it's crashing when you try to start it up, just stay in the Finder).
- 3) Have this manual and your White Knight master disk next to you.

I don't want to discourage anyone who needs it from taking advantage of Technical Support, and we're happy to help you. But please remember that our resources are limited, so we really need you to be all set up like we mentioned in the last paragraph. If you are in a word processor when you call, you will probably hear the sound of my teeth grinding as I sit waiting for you to get into White Knight. Remember that there are probably other people waiting for help, so please stick to the nature of the problem and don't ask what we're working on or when the next version will be out. Fair enough?

Technical Support can:

- 1) Find the cause of a duplicatable problem in most instances.
- 2) Answer specific questions about features.
- 3) Accept suggestions for future enhancements (we can't get enough of these).
- 4) Point out something that obviously should have been in this manual but for some reason didn't make it.

Technical Support cannot:

- 1) Debug your Procedures for you. If you really hit a brick wall with one, give us

a shot and we'll do what we can - this is a last resort.

- 2) Answer general questions that are covered in depth in the manual (like "How do I download" or "How do I make the modem dial?"). In this case, we'll just point you to the chapter that covers it.
- 3) Comment on products not yet released.
- 4) Chat about the weather.

Technical Support can be reached at (412)846-2700. Our hours will probably be expanded in the near future, but as of press time Technical Support is available from 2 PM to 5 PM (Eastern Time Zone - same as New York City), Monday through Friday. A phone call is highly preferable to a letter because it takes us much longer to reply to letters than to give an answer over the phone (and sometimes our reply letter will read "We need to talk about this - please call"). Please do not fax us with Technical Support questions unless you are outside of the United States and Canada (we understand the time zone differences and don't expect you to stay up till the middle of the night). Our fax machine number is (412)847-4436.

This rest of this chapter discusses the most common problems we get on our Technical Support line and their solutions.

Problem: When I insert my White Knight master disk, I get a dialog box saying "This disk is unreadable" or "This is not a Macintosh disk".

Probable Causes:

- The disk has been wiped out, probably due to exposure to a strong magnetic field during shipping. Send it back to us (the address is on the disk label) with a note that it is defective and we'll send a new one out to you the same day.
- Your dog or child has taken a liking to diskettes. See above.

Problem: I get a dialog box saying "Gadzooks! The requested serial port couldn't be opened.", or White Knight crashes immediately after the Title Screen is cleared.

Probable Causes:

- Only one Macintosh application can have a serial port open at any given time. There may be another application (or desk accessory, or INIT, or Control Panel device, etc.) that already has the requested serial port open.
- AppleTalk always uses the printer port, so don't try to use White Knight with the printer port with AppleTalk active (you can deactivate AppleTalk via the Chooser desk accessory).
- I have seen some graphics tablet drivers that use the modem port, so these must be removed from the System Folder, and the Macintosh restarted, before

White Knight can open that port. These drivers are typically "INIT" files, so use Apple's Virus Rx program (available for free from your local Apple dealer) to get a list of all "INIT" files in your System Folder.

- It's also possible that your hardware is damaged (even if everything else on the Mac seems to function properly). If you can't figure out why this dialog box keeps appearing, have your system checked out by your local dealer.

Problem: Whenever I choose **Service->Modem->Initialize**, White Knight seems to get hung up and then finally it comes back with a dialog box telling me that the modem isn't responding, or when I choose **Service->Dial Or Redial**, the modem doesn't dial and I see the word "ERROR" printed in the Data Area.

Probable Causes:

- Is the modem turned on and cabled to the proper serial port?
- Is it a Hayes compatible modem (call the manufacturer)?
- If the modem has lights labeled "MR" and "TR", are they both lit? If the MR light is not lit, the modem is probably damaged. If the "TR" light is not lit, contact the manufacturer and ask them how to set up the modem so that it "ignores the DTR signal from the computer".
- On occasion, I've seen modems get "brain damaged" for no apparent reason. Unplug it, wait a couple of moments, then plug it back in.
- You're not using "NULL" duplex, are you?
- Try choosing **Local->Serial Port** and increasing the "Modem command inter-character delay" value.
- Are you using a correct serial port setting? Try setting it to 300-N-8-1-FULL. If it works then, you were using something bogus (like 7 data bits with NO or IGNORE parity).
- If none of the above, and if White Knight's Elapsed Time Clock in the General Status Bar keeps running, you've got a hardware problem. It's either the modem, the cable, or the serial port in your Macintosh. Most likely it's the cable. Next likely is the modem. Least likely is the serial port.

Problem: A window isn't getting drawn in the right place, or isn't getting drawn at all. White Knight crashes spontaneously and inconsistently.

Probable Causes:

- Something got corrupted in the default Settings file - to return a window to the default size and position, hold down the Option key before the window is drawn and continue to hold it down until the window is fully drawn.
- You moved White Knight from a Mac with a large display monitor to one with a small monitor. Follow the advice above to return the windows to their default size and position.

- Do you have **Local->Window->Hide** checkmarked?
- Throw away the "WK's 11.0 Stuff" file (in the same folder as White Knight) and let White Knight recreate it.
- You have an "INIT" in your System Folder that has chosen White Knight as its enemy. Use Apple's "Virus Rx" program (your local Macintosh dealer can provide you with this program if it didn't come on one of the disks that came with your machine) to list the "INIT" files in your System Folder, and then remove them one at a time from the System Folder (just move them out of the System Folder onto the desktop, and then restart your Macintosh) and see if the problem disappears.
- You might have a corrupted or not up-to-date System file. Contact your local Macintosh dealer to have your System software replaced or updated.

Problem: When I load a text file into my word processor, the text is formatted all wrong and/or columns of data in a table don't line up properly.

Probable Causes:

- The font and point size your word processor uses as a default doesn't match what was used by the originator of the document. First, select all text in the document and change it to Monaco font - 9 point size. Experiment with the left and right margin settings until the paragraphs look reasonable. If the file contains a table, try putting in tab stops a half inch (or more) apart.

Problem: When I load a text file created by White Knight into my word processor, there are garbage characters or rectangle characters strewn throughout the text.

Probable Causes:

- These are control characters. See the chapter "Filters".

Problem: When I download an IBM-PC program and then either transfer it to a second IBM-PC or try to run it on my Mac (with a IBM-PC coprocessor board or emulation software), the program bombs.

Probable Causes:

- You've got a Filter turned on for a non-text file and binary data is getting stripped out. Turn off the File Transfer Filter.

Problem: When I try to download/upload a file, the file transfer is not starting like it should.

Probable Causes:

- You're telling White Knight to start the file transfer too soon. Wait until the remote machine specifically tells you to "start your transfer now" (or something to that effect).

- The remote machine doesn't support the file transfer protocol you have selected.
- You've selected a file transfer protocol that is not appropriate for your serial port settings. All of the protocols except Kermit require 8 data bits, NO (or IGNORE) parity, and 1 stop bit. Kermit can be used with any serial port setting.

Problem: During a file transfer, I seem to get a lot of errors. The file transfer finally quits or takes forever to complete.

Probable Causes:

- You've got a lot of static and noise in the telephone connection. Try disconnecting and calling back to see if you get a cleaner line. This can be especially troublesome for international calls or when using a "cheapie" long distance service.
- You've got your timeout value set too high or too low. Try reducing it (in the dialog boxes brought up by **Customize->Options->X-Y-ZMODEM** or **Customize->Options->Kermit** to 5 seconds. If you're communicating with a commercial service like CompuServe, try increasing it to 20-30 seconds.
- You're using a 9600 (or higher) baud modem that has "modem level software handshaking" implemented. The binary data that is part of a protocol transfer is being mistaken by the modems as handshaking data and is being stripped out. Consult your modem's user manual (or the modem manufacturer's technical support) to find out how to turn off "modem level software handshaking".

Problem: After downloading or capturing a file, when I try to open it I get a dialog box saying "Application busy or missing."

Probable Causes:

- If it is a text file, read the section "Working With A Received Text File" in the chapter "Receiving Text Files (File Captures)". These instructions are also valid for text files received with a file transfer protocol.
- It may be a compressed file. If the file was listed on the service with the letters ".SIT" or ".PIT" at the end of the filename, this is certainly so. You'll need to download the decompression program and decompress the file before it can be used. The System Operator of the service you downloaded the file from should be able to help you get the decompression program.
- You downloaded a document that is for an application that you either don't own or is on a disk volume not present in your Macintosh.
- It's even possible that the file is for some other computer than a Macintosh. The System Operator (more commonly abbreviated "SYSOP", which is pronounced "SIS-OP", not "SEYES -OP") of the system you downloaded the file from should be able to instruct you how to use the file in any case.

Problem: The cursor keys or numeric keypad keys no longer work like they should in VT emulation.

Probable Causes:

- Try flip-flopping the "Cursor Key Mode" or "Keypad Mode" items (whichever is applicable) in the dialog box brought up by selecting **Customize->VT Modes**.
- Select **Customize->Options->Key Mapping** and make sure the "Keyboard" item matches the keyboard you have installed. If you have a keyboard not listed, or it still won't work for some reason, click on the "Other" item and follow the instructions in the chapter "The 'Install Special Keys' Utility".

Problem: My Procedure is not responding to a prompt from the remote system.

Probable Causes:

- What's coming over doesn't match what's in your PROMPT command.
- Checkmark the **Service->Procedure->Monitor Execution** choice and see if the prompt is being received before the PROMPT command is executed.

Problem: My Procedure worked fine on my old machine, but I've upgraded to a faster machine and now it doesn't work.

Probable Causes:

- You wrote your Procedure based on timing rather than responding to prompts. The faster machine executes the Procedure instructions faster, so the timing is no longer valid. Rewrite the Procedure to be based on responding to prompts, or if timing is absolutely necessary, use PAUSE commands for timing (PAUSE commands have consistent delays for all Macintosh machines).

Problem: When a Procedure is executing, I get a dialog box saying "File Not Found". What does this mean?

Probable Causes:

- One (or more) of your Procedure commands loads in a file of some sort, and your filename is wrong. Read the chapter "About Filenames".

Problem: When I download an application, it is shown in the Finder with a document icon rather than an application icon.

Probable Causes:

- You do not have the "Use MacBinary format for:" item set to "All files except type "TEXT" checkmarked in the dialog box brought up by selecting **Customize->Options->File Transfer**
- If you are running under MultiFinder, make sure that the window corresponding to the folder that you are downloading to (this is set by choosing

File->Received File Destination) is closed on Finder's desktop. If you forget to do this, I have found that often times I can close the window and reopen it and the icon will be displayed properly.

Problem: Everything seems to run fine until I choose a menu choice that would normally bring up a file selection or file definition dialog box. Then, I get a system crash.

Probable Causes:

- You are running an INIT that is not compatible with White Knight. See the Troubleshooting item earlier in this chapter for instructions on how to list the INIT's that are installed in your System Folder.

Problem: I can't get a caret character (Shift-6 on the keyboard) into a Macro Key string or Procedure TYPE command. It sends something goofy.

Probable Causes:

- The caret character is telling White Knight to treat the character that follows it as a control character. To send an actual caret character, type 2 caret characters in a row.

Even More White Knight Features

We've already discussed RCMD's in a previous chapter. RCMD's give a great amount of flexibility for customizing White Knight for your unique needs, and to provide future features without hardcoding them into White Knight. Because of their advanced nature, writing RCMD's is not a subject I chose to address in this manual. In addition to RCMD's, there are two other advanced features, Modem Drivers and Shell Resources, that are beyond the scope of this manual.

So, I've decided to put together a special "White Knight Developer's Toolkit", which will cover these advanced subjects. I'll discuss this in detail at the end of this chapter.

RCMD's

The sample RCMD's included with White Knight 11 should give you a taste for the capabilities of RCMD's, but they by no means demonstrate the limits. The White Knight Developer's Toolkit contains full documentation on how to write an RCMD. You will need to own and be proficient with a language compiler that can create Macintosh "CODE" resources (like C or Pascal). Your RCMD will have access to much of White Knight's internal variables, as well as to the Procedure variables and flags. Sample source code for an RCMD written in Lightspeed C is included.

Modem Drivers

White Knight's built-in modem handling routines are all geared to the "AT" command set made popular by the Hayes™ brand modems. We've found that there are few modems (besides those made by Hayes) that are truly Hayes compatible. There have been numerous liberties taken with this de facto standard, and manufacturers have extended the basic command set to support brand specific features. For this reason, White Knight's built-in routines were by necessity geared towards lowest common denominator modem support.

It seemed a shame to me not to provide some sort of way to take advantage of the numerous features that give the individual modems their personality (and value), so I devised a special kind of Procedure, called a "Modem Driver" expressly for the purpose of extending the built-in modem handling routines.

By writing your own Modem Driver, you can:

- Support modems or devices other than "Hayes compatible".

- Modify the built-in routines to work with any sort of device.
- Add support for up to 10 additional modem routines, each of which will appear with your description under the **Service->Modem** menu beneath the built-in functions. These can be brand-specific features, or features that I didn't implement but you might find desirable for your application. In addition, these additional routines can also be called from a Procedure, in the same manner as the built-in routines.

The White Knight Developer's Toolkit contains full documentation on how a Modem Driver is written, and how one functions. Sample source code (in White Knight's Procedure language) of a Modem Driver is included.

Shell Resources

If the QUERY dialog boxes in White Knight's Procedure language don't seem to have enough ammunition for your application, you'll be glad to hear that White Knight supports customized dialog boxes, called Shell Resources. You can call dialog boxes from your Procedures that include groups of radio buttons, checkboxes, static text items, editable text items, pushbuttons, icons, and pictures. All of the user's interaction with the dialog is brought back to your Procedure through the Procedure variables. By using Shell Resources, you can create a totally customized environment inside of White Knight. In fact, I have seen some terrific ones done by several third party companies that had me blinking my eyes wondering if it was really White Knight I was looking at.

Creating Shell Resources is not terribly difficult, but you will need Apple's ResEdit program (or some other resource creation program) and know how to create dialog boxes. Full documentation and sample shell resources are included in the White Knight Developer's Toolkit, along with a special Shell Resource Compiler application.

Getting The White Knight Developer's Toolkit

I suspect that the White Knight Developer's Toolkit will be released in the FreeSoft RoundTable on GEnie (at least in preliminary form) at the same time this product is released. If you are not on that network, we will make it available at a modest price by mail. Please contact us for further information.

The ASCII Character Set

The ASCII character set comprises codes 0 through 127. The characters for codes 128 through 255 are undefined in ASCII, and will differ not only from machine to machine, but even from font to font on the Macintosh. For that reason, codes 128 through 255 are not listed in the table on the next page. To see what characters the font you are using in the Data Area of the Terminal Window implement for these codes, choose **Customize->Filter->Edit**, and select the Terminal Filter. Scroll down to the code numbers you wish to view and click on the "Characters" radio button to see the actual characters.

CONTROL CHARACTERS (Codes 0-31 And 127):

A "*" next to a control character means that most Macintosh word processor programs understand and can react properly to that code. A "#" next to a control character means that it is used in TTY emulation. An "&" next to a control character means that it is important to VT emulation.

<u>Code-Character</u>	<u>Code-Character</u>
0-NULL	16-DATA LINK ESCAPE
1-START OF HEADING	17-DEVICE CONTROL 1 (XON)
2-START OF TEXT	18-DEVICE CONTROL 2
3-END OF TEXT	19-DEVICE CONTROL 3 (XOFF)
4-END OF TRANSMISSION	20-DEVICE CONTROL 4
&5-ENQUIRE	21-NEGATIVE ACKNOWLEDGE
6-ACKNOWLEDGE	22-SYNCHRONOUS IDLE
-BELL	23-END OF TRANSMISSION BLOCK
-BACKSPACE	&24-CANCEL
&#*9-HORIZONTAL TAB	25-END OF MEDIUM
&10-LINEFEED	&26-SUBSTITUTE
&11-VERTICAL TAB	&27-ESCAPE
-FORM FEED	28-FILE SEPARATOR
&#*13-CARRIAGE RETURN	29-GROUP SEPARATOR
&14-SHIFT OUT	30-RECORD SEPARATOR
&15-SHIFT IN	31-UNIT SEPARATOR
&127 - DELETE	

Alphanumeric Characters (Codes 32-126)

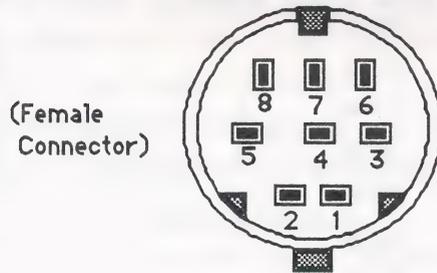
These codes are consistent for all computers that support the ASCII character set standard.

<u>Code-Character</u>	<u>Code-Character</u>	<u>Code-Character</u>
32-SPACE	65-A	98-b
33-!	66-B	99-c
34-"	67-C	100-d
35-#	68-D	101-e
36-\$	69-E	102-f
37-%	70-F	103-g
38-&	71-G	104-h
39-'	72-H	105-i
40-(73-I	106-j
41-)	74-J	107-k
42-*	75-K	108-l
43-+	76-L	109-m
44-,	77-M	110-n
45--	78-N	111-o
46-.	79-O	112-p
47-/	80-P	113-q
48-0	81-Q	114-r
49-1	82-R	115-s
50-2	83-S	116-t
51-3	84-T	117-u
52-4	85-U	118-v
53-5	86-V	119-w
54-6	87-W	120-x
55-7	88-X	121-y
56-8	89-Y	122-z
57-9	90-Z	123-{
58-:	91-[124-
59-;	92-\	125-}
60-<	93-]	126-~
61-=	94-^	
62->	95- _	
63-?	96-`	
64-@	97-a	

Macintosh Serial Port Pinouts

The three major Macintosh models as of the printing of this manual: Mac Plus, Mac II, and Mac SE (and all of the variants for each model, such as the Mac IIcx or Mac SE/30) all use the same serial port connector, illustrated below. This information will be primarily useful only for those who will need to construct a custom cable for a particular application. Note that depending on whether you are looking at the socket or the plug, the pinouts might be a mirror image of what's shown in this illustration. To get your correct orientation, notice that there is a wider gap between pin #5 and pin #4 than there is between pin #4 and pin #3.

Macintosh Serial Port Pinouts: DIN-8



- 1 - Output Handshake
- 2 - Input Handshake
- 3 - RS-232 Transmit data
- 4 - Ground
- 5 - RS-232 Receive data
- 6,7,8 - Not used for RS-232 connections

Keyword Index

ADDF.PROC 252

ASCII 42, 44, 45, 58-62, 68, 87, 88, 99, 129, 165, 177, 178, 194, 240-242, 244-247, 250, 265

ATDP 53, 162

ATDT 53, 67, 161, 162, 165, 198, 225, 227, 228, 230

auto-answer 46, 47, 107, 119, 138, 163

batch file 91, 94, 100, 101, 138, 181, 182, 253

baud 19, 20, 24-27, 30, 45, 50, 83, 95, 107, 118, 136, 142, 145, 192, 193, 225, 239, 260

BBS 26, 112, 118, 122, 125, 196

Buffered Keyboard 56, 57, 116, 117, 140, 141, 166, 192, 240, 247

cable 7, 21, 23, 29, 30, 258, 267

capture 59, 61, 62, 74, 80, 85, 87, 88, 110, 120, 138, 139, 178, 179, 191, 197, 245, 252

carriage return 36, 37, 44, 45, 56, 57, 58, 61, 62, 67, 79, 83, 84, 88, 105, 111, 113, 114, 119, 125, 135, 142, 144, 148, 150, 157, 162, 164, 165, 189, 192, 196, 210, 224, 226, 247, 252

Clipboard 16, 17, 56, 71, 182

Close Box 10, 52, 73, 93, 117, 132, 220-222, 244

compiler 128, 129, 131, 146, 149, 150, 152, 157, 158, 182, 203, 207, 245, 263, 264

CompuServe 28, 33, 56, 68, 90, 103, 108, 110, 142, 144, 145, 225, 226, 230, 241, 253, 260

control character 18-20, 27, 39, 41, 42, 56, 58, 61, 62, 67, 76, 79, 81, 135, 143, 161, 162, 164, 165, 192, 193, 216, 246, 259, 262, 265

Convert Settings 8

Customize->Control Buttons 20

Customize->Filter->Edit 59, 87, 99, 265

Customize->Filter->Load 62

Customize->Filter->Save 61

Customize->Macro Keys->Edit 65

Customize->Macro Keys->Load 63, 72, 196

Customize->Macro Keys->Save 63, 196

Customize->Macro Keys->Window 69, 73

Customize->Options->Compiler 149, 150, 157

Customize->Options->Emulation 32, 33, 39, 62, 194

Customize->Options->File Transfer 62, 96, 99, 103, 201, 253, 261

Customize->Options->Kermit 105, 260

Customize->Options->Key Mapping 41, 84, 194, 196, 235, 261
Customize->Options->Printer 75, 77
Customize->Options->White Knight 57, 115, 211
Customize->Options->TEXT Transfer 17, 61, 62, 81, 86, 87, 201
Customize->Options->X-Y-ZMODEM 102, 193, 200, 260
Customize->Settings->Load 7, 196
Customize->Settings->Save 7
Customize->TTY Modes 33
Customize->VT Modes 34, 192, 261
Data Area 9-11, 14-16, 18, 19, 28, 37, 39, 40, 46, 58, 59, 74, 77, 81,
100, 153, 161, 164, 166, 185, 191, 195, 255, 258, 265
Data Buffer 11, 12, 15-17, 34, 248
databits 19, 25-27, 45, 136, 193, 240
delete 43-45, 53, 57, 80, 103, 104, 112, 113, 115-117, 125, 133, 138,
140, 175, 183, 194, 214, 240, 243, 265
Delphi 56, 90
desk accessories 12, 18, 58, 71, 93, 95, 257
Developer's Toolkit 251, 263, 264
dial 21, 47-49, 53-55, 67, 107, 125, 126, 128, 138, 142-144, 161, 162,
165, 198, 225, 227, 228, 230, 246, 257, 258
Display Screen To Archive File 18, 19, 115, 117, 185, 243
Display Screen To Printer 18, 19, 115, 117, 191, 243
duplex 19, 24-27, 29, 45, 81, 107, 118, 136, 142, 170, 192, 193, 225,
229, 230, 239, 240, 258
Edit->Append To->Archive 17
Edit->Append To->Existing File 17
Edit->Append To->New File 17
Edit->Clear 17
Edit->Copy 1, 255
Edit->Copy Table 255
Edit->Copy, Then Paste 16
Edit->Cut 17
Edit->Locate Next 15
Edit->Locate Text 15, 243, 245
Edit->Paste 2, 182
Edit->Print Selected Text 17, 74
Edit->Undo 17
Elapsed Time Clock 18, 19, 198, 244, 258
escape 36, 37, 44, 68, 77, 165, 194, 265
FastDump.PROC 252
File Capture Filter 61, 62, 87, 88, 197, 245

file creator 87, 96, 100

File Transfer Filter 62, 99, 106, 197, 259

File Transfer Status Window 92-95

File->Archive File Destination 17

File->Create Batch File 101

File->Delete A File 115, 116

File->File Capture->Append To 85, 179

File->File Capture->Close 85

File->File Capture->New 61, 85, 178

File->File Capture->Pause 85, 139, 179

File->Quit 132, 210, 235-237

File->Receive File Using->Flash Protocol 180

File->Receive File Using->Kermit Protocol 114, 179

File->Receive File Using->XMODEM Protocol 107, 110, 180

File->Receive File Using->YMODEM Protocol 180

File->Receive File Using->ZMODEM Protocol 113, 180

File->Received File Destination 99, 262

File->Send File Using->Flash Protocol 182

File->Send File Using->Kermit Protocol 114, 181

File->Send File Using->XMODEM Protocol 107, 181

File->Send File Using->YMODEM Protocol 181

File->Send File Using->ZMODEM Protocol 182

File->Send Text File 17, 80, 85, 181

Filter File 62, 128, 197

Flash 15, 30, 33, 34, 89, 92, 95, 116, 120, 138, 139, 180, 182, 240, 253

FolderBatch.PROC 252

font 10, 11, 14, 15, 39, 40, 57, 58, 76-78, 80, 195, 245, 246, 249, 250, 259, 265

FreeSoft RoundTable 111, 112, 251, 264

function keys 69

General Status Bar 18, 19, 25, 64, 74, 140, 185, 191, 193, 195, 240, 244, 258

GENie 56, 90, 91, 111-113, 234, 251, 264

hang up 3, 24, 29, 30, 46, 47, 107, 121, 123, 138, 162, 163, 209, 227

hierarchical 1-3, 233

high level printing 74, 75, 77

Horizontal Scroll Bar 9-11, 15, 57, 132

Host Mode 100, 118-124, 147, 204

Host Mode Help 122

icon 4-7, 12, 51, 62, 63, 69-71, 96, 100, 101, 104, 110, 129, 131, 147,
232, 233, 251, 261, 262

Install Special Keys 235, 236

Kermit 89-91, 95, 105, 106, 110, 113, 114, 120, 138, 139, 179, 181,
242, 244, 245, 253, 260

linefeed 36, 44, 45, 58, 67, 84, 87, 88, 99, 105, 106, 196, 236, 247,
252

local echo 24, 118

Local->Buffer->Archive 17, 115, 243

Local->Buffer->Erase 16

Local->Buffer->Remember 11

Local->Buffer->Size 11

Local->Host Mode->Become Host 119, 204

Local->Host Mode->Passwords 122

Local->Print TEXT File 74, 191

Local->Printer Echo 74, 191

Local->Serial Port 19, 25, 49, 107, 232, 258

Local->Status Bar->Buffered Keyboard 56, 192

Local->Status Bar->General 18, 195

Local->Status Bar->Hide Status Bar 18

Local->Status Bar->Macro Keys 64, 197

Local->Status Bar->VT100 Keys 39, 201

Local->Timer->Billing Cost 19, 200

Local->Timer->Reset To Zero 19

Local->Window->Archive 19, 115, 243

Local->Window->Bring Others To Top 12

Local->Window->Colors 12

Local->Window->Erase 16

Local->Window->Font 14, 195, 249

Local->Window->Hide 12, 199, 259

Local->Window->Tab Stops 14

low level printing 74-77

MacBinary 93-99, 103, 241, 244, 247, 261

Macro Key 45, 56, 63-72, 77, 116, 126, 129, 128, 135, 139, 147, 162,
165, 196, 197, 199, 200, 213, 216, 225, 236, 237, 241, 244, 253, 262

Macro Key File 63, 128, 196

Macros Window 64-66, 69-72, 126, 199, 200

modem 7, 16, 21-23, 26, 29, 30, 42, 45-47, 49-51, 53, 54, 67, 68, 79,
80, 107-109, 119, 128, 130, 136-138, 142, 143, 161-163, 165, 197,
207, 210, 225, 226, 230, 247, 257, 258, 260, 263, 264

Modem Driver 50, 137, 163, 207, 263, 264

MultiFinder 12, 49, 93, 95, 116, 119, 211, 232, 241, 261

parity 19, 24-29, 45, 106, 107, 118, 136, 142, 192, 193, 197, 198, 225, 239, 252, 258, 260

Pause Remote 19, 20

Phonebook 51-55, 128, 147, 156, 227, 230

picture 70-72, 126

ProcEdit 5, 131, 132, 134, 251

Procedure 6, 53-57, 63, 68, 69, 72, 73, 115, 116, 121, 123, 125-137, 140, 142-161, 164-166, 177, 179, 182, 183, 187-189, 191, 192, 196, 198, 202-213, 216, 218, 219, 221, 223-230, 232, 238-245, 247, 249-254, 261-264

QuickB.PROC 253

RCMD 101, 232, 251-255, 263

redial 47-49, 54, 55, 107, 128, 138, 143, 161, 162, 198, 227, 230, 247, 258

Return key 36, 44, 56, 57, 67, 84, 109, 110, 112, 117, 134, 196, 247, 253

RLE graphics 33, 241

serial port 7, 17-19, 21, 23-27, 29, 30, 39, 45, 46, 49, 56, 68, 80, 81, 89, 106, 107, 118, 119, 127, 128, 136, 142, 162, 164, 165, 177, 182, 191-193, 197, 198, 206-208, 216, 225, 226, 229, 232, 247, 252, 257, 258, 260, 267

Service->Dial Or Redial 47, 107, 128, 138, 160, 258

Service->Initiate Procedure 129, 131, 147, 251, 253

Service->Modem->Auto-Answer 46, 107, 119, 163

Service->Modem->Hang Up 47, 107, 162

Service->Modem->Initialize 46, 162, 258

Service->Modem->Turn Off Auto-Answer 46, 163

Service->Modem->Use Driver 50

Service->Phonebook->Close 52

Service->Phonebook->Gang Dial 55

Service->Phonebook->New 51

Service->Phonebook->Open 52

Service->Phonebook->Open Default Phonebook 52

Service->Procedure->Compile TEXT File 128, 148

Service->Procedure->Edit Procedure 131

Service->Procedure->Monitor Execution 148, 157, 158, 192, 196, 198, 203, 261

Service->Procedure->Syntax Check 149

Service->Procedure->Write For Me 127

SetFile.PROC 253, 254

Settings File 6-8, 53-55, 128, 139, 147, 196, 238, 249, 258

Shell Resources 263, 264

Show Control Characters 62

Speak.PROC 254, 255

startbits 25

Status Bar 9, 10, 18, 19, 25, 38, 39, 45, 56, 64-66, 69, 72, 74, 116, 119, 140, 141, 148, 150, 153, 160, 185, 191, 193, 195, 197, 199-201, 223, 240-244, 258

stopbits 19, 25, 29, 136, 192, 193, 240

SYSOP 109, 260

TabFile.PROC 255

Terminal Filter 59, 62, 197, 245, 265

Terminal Window 8-12, 15, 16, 18, 28, 37, 39, 46, 49, 57-59, 64, 74, 78, 81, 92, 100, 116, 117, 119, 148, 153, 161, 164, 166, 185, 191, 195, 199, 219, 244, 255, 265

Title Bar 9, 10, 18, 53, 72, 73, 92, 93, 132, 148, 220

TTY 14, 31-35, 38, 39, 118, 170, 201, 240, 246, 249, 265

Vertical Scroll Bar 9, 11, 12, 53, 60, 70, 132

VT100 14, 31, 32, 34, 37-40, 43, 56, 129, 140, 141, 170, 201, 235, 240, 246-250

VT100 Keys 39, 140, 141, 201, 235

VT102 14, 31, 32, 34, 36-38, 43, 201, 240, 248, 249

VT52 14, 31-34, 37-40, 43, 170, 201, 240, 249, 250

Word Processor 15-17, 61, 80, 82, 83, 86-88, 98-100, 126, 148, 256, 259, 265

word-wrapping 83

XMODEM 2, 89-91, 93, 95, 98, 101-103, 107, 108, 110, 113, 120, 138, 180, 181, 186, 187, 200, 226, 227, 240, 241, 243

XOFF 19, 20, 26, 82, 252, 265

XON 20, 26, 82, 252, 265

YMODEM 89, 91, 102, 103, 108, 110, 113, 120, 138, 139, 180-182, 200, 240, 243, 253

Zap Control Characters 62

ZMODEM 89, 91, 92, 95, 102-104, 108, 111, 113, 120, 138, 139, 180, 182, 200, 240, 248, 253

Zoom Box 10

Procedure Command Index

* - 191
@ - 160
ADD - 166
ADD TO MENU - 213
ALERT1 - 160
ALERT2 - 160
ALERT3 - 160
ALT - 194
AND - 166
ANSWERBACK - 192
ATTACH - 202
AUTOANSWER - 163
BBAR - 192
BELL - 161
BUTTON - 199
BYTEVAL - 177
BYTEADD - 178
CLEAR - 161
CLOSE - 178
COMPILE LOUD - 182
COMPILE QUIET - 182
COMM - 192
CONCAT - 166
CONTAINS - 167
CONTROL1 ^ - 193
CONTROL2 ^ - 193
CONTROL3 ^ - 193
CONVUP - 167
COPYINTO - 167
CRAWL - 203
CRC - 193
CUWOPENO - 220
CUWOPEN1 - 220
CUWOPEN2 - 220
DEFINE MENU - 215
DELETE - 183
DELKEY - 194
DIAL - 161
DISABLE MENU - 215
DISPLAY - 194
DIVIDE - 168
DO - 202
ECHO - 191
ELAPSED - 168
EMPTY - 168
ENABLE MENU - 215
END - 203
ERASE - 169
ERASE ALL - 169
FETCHBYTE - 177
FILL\$ - 169
FILTERT - 197
FILTERP - 197
FILTERF - 197
FONT - 195
GBAR - 195
GETALL - 183
GETCOST - 170
GETFILE - 184
GETGLOBAL - 170
GETLINE - 164
GETRECPATH - 186
GETSELECT - 183
GETVOL - 185
GOSUB - 203
HANGUP - 162
HOST - 204
IF CLOSEUW - 220
IF ERROR - 204
IF NO - 205
IF NO ERROR - 205
IF YES - 205
INBUFFER - 177
INITMODEM - 162
INSERT MENU - 215
INSTR - 171
JUMPTO - 206
LEFT\$ - 171
LENGTH - 171
LET EQUAL - 172
LF - 196
LOADSET - 196
LOADVAR - 172
LOCK - 206
LONG BREAK - 163
LOUD - 196
MACRO - 196
MAKEFILE - 186
MBAR - 197
MENUDOES - 216
MENUOFF - 216
MID\$ - 172
MODEM - 197
MULTIPLY - 173
NEST - 207
NESTEND - 208
NOBAR - 195
NOZAP - 197
NUMTOSTRING - 173
ONPANIC - 208
OR - 173
PANICAFTER - 209
PAUSE - 210
PRINTER - 198
PROMPT - 164
PROMPT ^ - 164
PUTFILE - 184
QUERY0 - 217
QUERY1 - 217
QUERY2 - 217
QUERY3 - 218
QUERY4 - 218
QUERY5 - 219
QUIET - 198
QUIT - 210
RECN - 178
RECA - 179
RECF - 180
RECK - 179
RECX - 180
RECY - 180
RE CZ - 180
REDIAL - 161
REDIAL LIMIT - 198
REDRAW - 200
REPLACE\$ - 173
RENAME - 185
RESET - 198
RETURN - 210

RIGHT\$ - 174	UWOPEN2 - 220
RUN - 211	UWUPDATE - 221
SAVETIME - 174	VBAR - 201
SAVEVAR - 175	VT52 - 201
SCREEN - 199	VT100 - 201
SCREENDISK - 185	VT102 - 201
SCREENPRINT - 191	WAIT - 212
SENDA - 181	WATCH - 165
SENDCLIP - 182	WINDOW - 199
SENDF - 182	WRITE - 191
SENDK - 181	XKSTRIP - 201
SENDX - 181	
SENDY - 181	
SENDZ - 182	
SETCOST - 200	
SETRECPATH - 187	
SHORT BREAK - 163	
SHOW@ - 160	
SHUTDOWN - 212	
SLOW - 200	
STRINGTONUM - 175	
STRIP - 201	
SUBTRACT - 175	
SUSPEND - 179	
TEST - 176	
TIMEDATE - 176	
TOP - 221	
TRACE - 192	
TTY - 201	
TYPE - 165	
UNIQUE - 187	
USEDRIVER - 163	
USERCLOSE - 188	
USERGET% - 190	
USERGET\$ - 190	
USEROPENA - 188	
USEROPENI - 188	
USEROPENO - 188	
USERPUT% - 189	
USERPUT\$ - 190	
USERREAD - 189	
USERWRRCR - 189	
USERWRITE - 189	
UWCLOSE - 221	
UWOPENO - 220	
UWOPEN1 - 220	





Scott Watson's



OKYTO[®]



An easy to use Macintosh-To-Macintosh file transfer application for use with modems or AppleTalk.

© 1989, The FreeSoft Company. All Rights Reserved.
Version 1.0 - Illegal To Distribute Without The Written
Consent Of The FreeSoft Company. This software and
manual is protected by United States And International
Copyright Laws.

The FreeSoft Company, 105 McKinley Road, Beaver Falls, PA 15010.

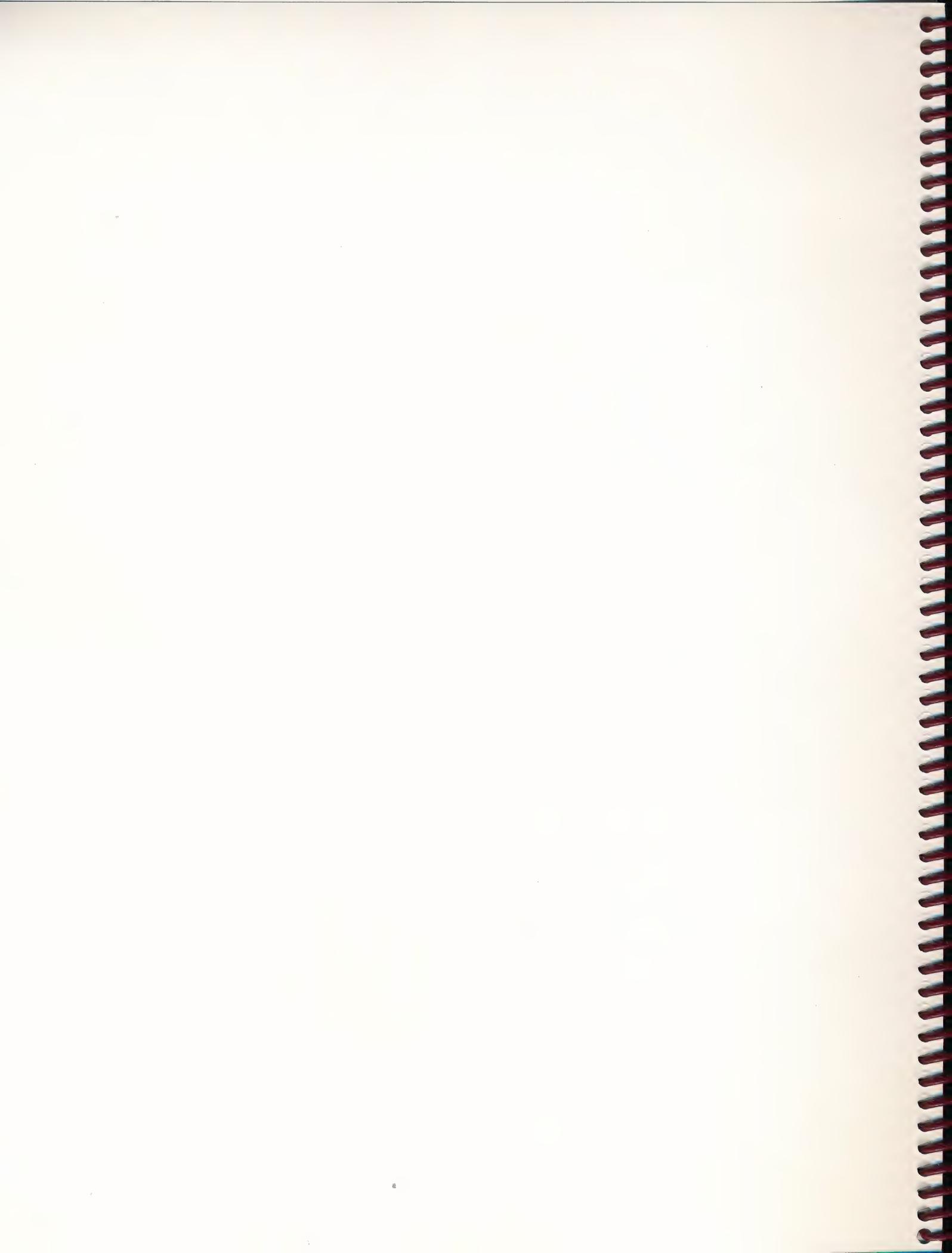


Table Of Contents

End User License Agreement For Okyto.....	i
A Bit Of History About Okito And Okyto.....	1
What Okyto Will (And Won't) Do.....	3
Getting Connected.....	5
Now That You're Connected.....	13
Sending And Receiving Files	17
Sending All Files In A Folder.....	21
Disconnecting.....	22
The Progress Log.....	24
Automated Procedures.....	26
Procedure Commands.....	28
Example Procedures.....	31
Now That You're A Guru.....	33
Designating Files By Full Pathname.....	37
A Special Way Of Connecting.....	39
Closing Thoughts.....	41

End User License Agreement For Okyto

Don't do anything with the Okyto master disks until you've read and agree with the following!

Execution of the Okyto software constitutes your acceptance of the terms and conditions of the End User License Agreement For Okyto. In return for acquiring a license to use the Okyto software contained in this package, you agree to the following terms and conditions. If you do not agree, return the entire contents of this package within 7 days of the date of purchase to the place of purchase for a full refund.

In this license, the words "Okyto" refers to all of the programs and data files recorded on the disks enclosed with this package.

1. The Okyto software and documentation are copyrighted, and you agree to respect all rights and privileges of The FreeSoft Company as the holder of these copyrights. You, as licensee, own the magnetic media on which Okyto is recorded. Title and ownership of Okyto is retained by The FreeSoft Company, including the copy of Okyto recorded on the disks enclosed with this package. The nonexclusive license set forth in this agreement is granted to you by The FreeSoft Company. The license is not a sale of Okyto or any copy. You will take all steps necessary to insure that each licensed copy of Okyto does not reside or execute on more than one Macintosh computer at any given time. You may make and execute as many copies of Okyto as you like on a single Macintosh computer. You may make as many copies of Okyto as you like for archival and backup purposes.
2. You will not use Okyto to support any illegal acts as defined by the laws of your country, state or province, and local governments, including international copyright laws. You will take all steps necessary to insure that Okyto is used only for legal purposes. You will not knowingly allow any copyrighted software to be transmitted by Okyto to another computer without permission from the owner of the copyright.
3. Limited Warranty: The FreeSoft Company warrants to you, the original

licensee of this package, that the disks on which Okyto is recorded and the user manual are free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase. If the disk or manual is defective, The FreeSoft Company will replace the defective item(s) at no charge to you, if you return the defective item(s) to The FreeSoft Company within ninety (90) days from the date of purchase. This paragraph sets forth the only warranty or representation of any kind with respect to Okyto made by The FreeSoft Company or anyone involved in the creation, production, delivery, or licensing of Okyto. Furthermore, any implied warranties are limited in duration to ninety (90) days from the original date or purchase.

4. The FreeSoft Company makes no warranty or representation of any kind, express or implied, with respect to Okyto, or its quality, reliability, performance, or its merchantability or fitness for any particular purpose. Okyto is licensed "as is". You assume the entire risk as to the quality, reliability, and performance of the software and documentation.

5. In no event shall The FreeSoft Company or anyone else involved in the creation, production, delivery, or licensing of Okyto be liable to you for any consequential, special, incidental, direct, or indirect damages of any kind arising out of the use of Okyto or any defect in Okyto, even if The FreeSoft Company or anyone else involved in the creation, production, delivery, or licensing of Okyto has been advised of the possibility of such damages.

6. If you violate any of the terms of this license, you agree to forfeit to The FreeSoft Company, at the request of The FreeSoft Company, all copies of the Okyto software and documentation in your possession.

7. The terms and conditions of this license supersede the terms and conditions of all previous licenses. Use of this software constitutes your agreement to abide by the terms and conditions of this license for all versions in your possession.

Now let's have some fun.

A Bit Of History About Okito And Okyto

(Both Pronounced Oh Kee' Toe)

Theodore Bamberg (1875-1963), is widely regarded by his peers to be one of the finest conjurers who ever lived, despite overcoming an immense professional handicap. Being born into a seven consecutive generation line of the Bamberg dynasty of magicians during the classical period of the magical arts certainly must have helped him choose to become a magician himself, but when he was a young man of 18 years, a swimming accident left him all but deaf. Without the ability to hear himself speak, and without modern medical and therapy methods, his speech became less and less intelligible. To any person, this is a serious disability, but to a magician of his period, when such acts traditionally relied upon the performer to narrate his illusions as well as to interact with the audience, his deafness and inability to speak clearly was a personal disaster.

His novel solution to this dilemma was to make *himself* something of an illusion. He decided to bill himself as an Oriental magician who would therefore not be expected to understand or speak the language of the country in which he performed, and with the aid of heavy makeup and flowing silk robes, the mirage was quite convincing. He adopted the stage name Okito, which was simply Tokyo (in those days spelled as "Tokio") spelled, well...sideways. Without the luxury of double-talk and vocal misdirection, Theo Bamberg's Okito character had to rely upon pure technical and artistic talent. He apparently had a great deal of both, as his legacy lives on through the thousands of present day finger flingers who have studied and frequently perform the feats Bamberg created.

Updated with the modern sideways spelling of Tokyo, this program is dedicated to Theodore Bamberg, the son of David, Court Magician to King William III and Princess Wilhelmina, and the father of David, whom you've probably heard of as Fu Manchu.

So what's all this hoci-poci (the plural, I think, of hocus pocus) have to do with a telecommunications program for the Macintosh? When I

decided to write this program, it was because of the large number of people who had contacted me wanting a simple solution to transferring files between Macintoshes. These folks were not digital gurus; they did not want to get intimate with the telecommunications beast (hmmm...that's a colorful thought). They didn't want to know why the damn thing worked (or, more often than not, why the damn thing wouldn't work), they just wanted to accomplish this simple and noble goal without muss, fuss, or cuss.

Since this program was designed to be used in a closed environment (that is, with one copy of Okyto always talking to another copy of Okyto), there was a lot I could do on the inside to make it easy to use on the outside, and that's just what I've attempted. If I've done my work right, you should be up and transferring files like a pro within an hour of reading this sentence.

It may seem ironic that in the face of much Macintosh software racing towards quality by means of quantity, this package was designed to do one thing extremely well. Andy Rooney said, "Computers make it easier to do a lot of things, but most of the things they make it easier to do don't need to be done." The point of Okyto is to break away from the "features" race and concentrate solely on making something easier that frequently needs to be done.

One of the aspects of the art of magic, after all, is making a complex series of operations seem simple and natural to the observer, and this is the essence of Okyto. What Theodore Bamberg accomplished, in parallel to overcoming the handicap of the traditionally large telecommunications learning curve, is the basis for the namesake of this project. I hope that for you it performs some magic, because making this damn thing work was a real trick.

Scott Watson

September 27, 1989

What Okyto Will (And Won't) Do

Okyto is designed to provide a fast and easy to use medium for transferring files between Apple Macintosh computers. Traditional telecommunications software requires the user to fiddle with a variety of protocol settings, command the modem to dial (or answer) using an obscure and unintuitive command language, select a proper file transfer protocol (with their own variety of often complex options), perform a transfer, repeat this process for each file to be transferred, and then disconnect the telephones. Some of the better programs have taken great strides towards simplifying this process (hint, hint), but even the best have not made it a simple process.

Because Okyto was designed with specific abilities in mind, rather than with a shotgun approach to the general telecommunications market, it can do some things very well that general telecomm software does to only a fair or poor degree, and some things that other packages can't do at all.

By the way, if some of the below reads like Greek, don't be concerned. It will all be described in more detail later.

- Okyto uses a "multiplexed error correcting protocol". Most traditional software uses a single channel scheme. In plain English, "single channel" means that you can either be sending a file or receiving a file, but not doing both at the same time. Okyto can actually broadcast and receive over five discreet channels simultaneously, which results in a significant reduction of the total amount of time spent getting work done. This savings not only translates directly to smaller long distance telephone charges, but also translates directly into more productive time available for you and your computer.
- Because of this multiplexed environment, you could, for instance, be sending a file, receiving a file, typing conversational messages back and forth, and looking through the directory of disk volumes, folders and files on the remote user's disk all at the same time.
- Okyto can be operated attended on both sides by people, attended

by only one person and unattended on the other side, or with both sides unattended.

- Okyto offers a simplified scripting language that allows a novice to set up complicated automated sessions. This language even allows the user to designate a series of tasks to be performed at a specified time of day, such as when phone toll rates are lower. Perhaps you might choose to write a Procedure to automatically have your Okyto call one system, transfer files, disconnect, call another system, transfer files, disconnect, and so on.
- Okyto can be run in the foreground, or as a background application under MultiFinder, meaning that it can perform its duties without needing the full attention of the user or the computer. You could, for instance, be working on a spreadsheet or word processor while Okyto operates silently in the background.
- Okyto offers sophisticated error correction. It is highly tolerant of data transmission delays and telephone line noise. It can be used with high speed (9600 baud and up) modems.
- Okyto offers multi-level password protection, meaning that you can have full control over what individual callers are and aren't allowed to do when connecting to your Okyto. You can even specify that certain folders on your disk volumes are "off limits" so that no one may access your private files. It is very secure and vandal proof.
- Okyto offers an optional Progress Log, which details exactly what operations took place during a connection. This Progress Log information can be sent to a window on the Macintosh screen, a disk file (for a permanent or temporary record), or both.

All of these abilities come with only one limitation. You must use Okyto to connect with another copy of Okyto. This means that it won't do things like VT100 terminal emulation, XMODEM or Kermit protocol file transfers, or many of the other things found in general purpose telecommunications software. Okyto is not meant to compete against or replace such products.

Getting Connected

In this chapter, you're going to learn how to get your Okyto up and running, and connected to another Okyto. It's going to be discussed in great detail, although in actual practice you'll find that to prepare for a connection with another Okyto generally takes about a minute.

Note: There are some things that will need to be discussed verbally before you make the actual connection. Therefore, read all the way through this manual up to the "Disconnecting" chapter at least once before an attempt is made to connect.

To use Okyto, you'll need:

- An Apple Macintosh Plus or greater using the latest System/Finder versions for that machine.
- If you wish to connect machines via AppleTalk, you'll need an AppleTalk adapter and cable for each machine (your local Apple dealer can set you up with all you'll need for this).
- If you wish to connect machines via modem, you'll need a Hayes compatible modem and the correct modem cable to fit your Macintosh. "Hayes compatibility" is defined here as being able to use the same "AT" command set as the Hayes Smartmodem™ line of products. The modem manufacturer or dealer can verify this.

OK, let's get started!

Step 1: Pull down the "Okyto" menu and choose "Perform Connection Now..."

A dialog box will appear that looks like this:

Make connection to another Okyto.
 Wait for connection from another Okyto.

Connect via: **AppleTalk** **Modem**

My name is:

Password:

Okyto™
By Scott Watson
© 1989, The FreeSoft Company
All Rights Reserved

Version 1.0

NOTE: In Okyto, all dialog boxes will appear centered in the middle of your display screen, no matter what size screen you have installed with your Macintosh.

Step 3: Select either the "Make connection via:" or "Wait for connection via:" item depending on what you want to do. Remember this rule: to connect two Okytos, one Okyto will be **making the connection**, and the other will be **waiting for a connection**. Which will do what must be discussed and agreed upon beforehand. This usually comes down to one person saying "OK, my Okyto will call yours." which translates to "My Okyto will be making the connection, so you should set yours up to wait for a connection."

Step 4: Choose how you wish to connect, by selecting either the "AppleTalk", or "Modem" item.

This option, if not obvious to both parties, should also be discussed and agreed upon. Both parties have to select the same item here.

Step 5: Position your mouse cursor inside the rectangle

directly to the right of the words "My name is:", click your mouse button, and then type in your name. This is just to let the other party know who is connected to their Okyto. If you want to be informal, you can just type in your first name. In any case, what you type has to be 40 characters or fewer.

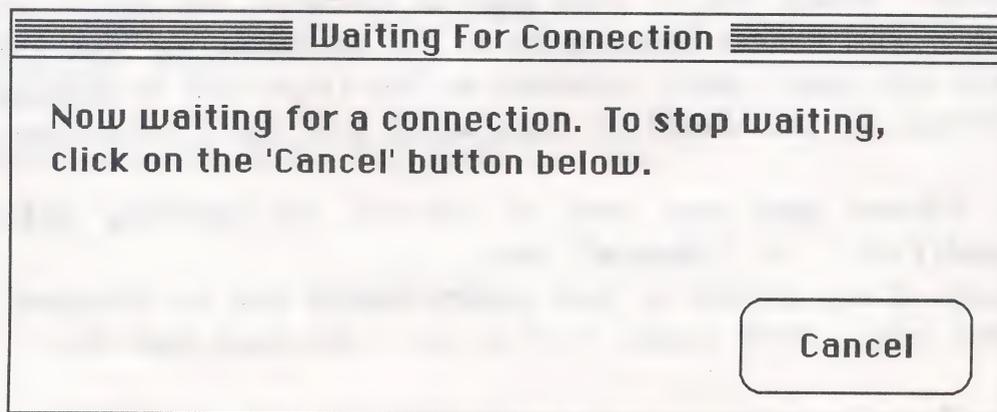
NOTE: We will skip the "Password:" item for now - just leave that box empty. We'll come back to what passwords are later on.

Step 6: Click your mouse inside the button labeled "OK", or just press your "Return" key on the keyboard. If for any reason you aren't happy with the way things are going, you can just click your mouse inside the button marked "Cancel". You'll find that you can cancel out of just about everything in Okyto, so don't worry about any untimely explosions happening should you make a mistake and get someplace you had no intention of being. Whenever there's more than one button in a dialog box, you'll notice that one of the buttons has around its perimeter a darker outline. This means that pressing the "Return" key on your keyboard is a shortcut for clicking inside this button.

If you're connecting via modem skip ahead in this chapter to the section titled "If you're connecting via modem..."

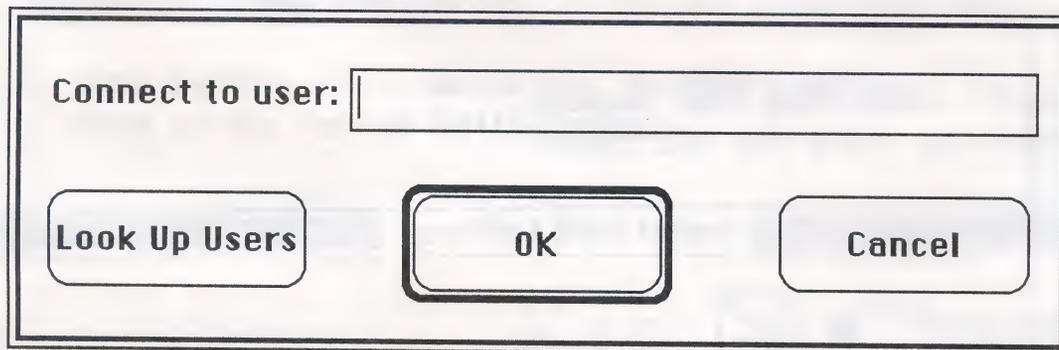
If you're connecting via AppleTalk...

You'll first see a dialog box telling you that Okyto is logging into the AppleTalk network. This will disappear in a few moments. At this point, one of two things will happen. If you're waiting for a connection, the following dialog box will appear:

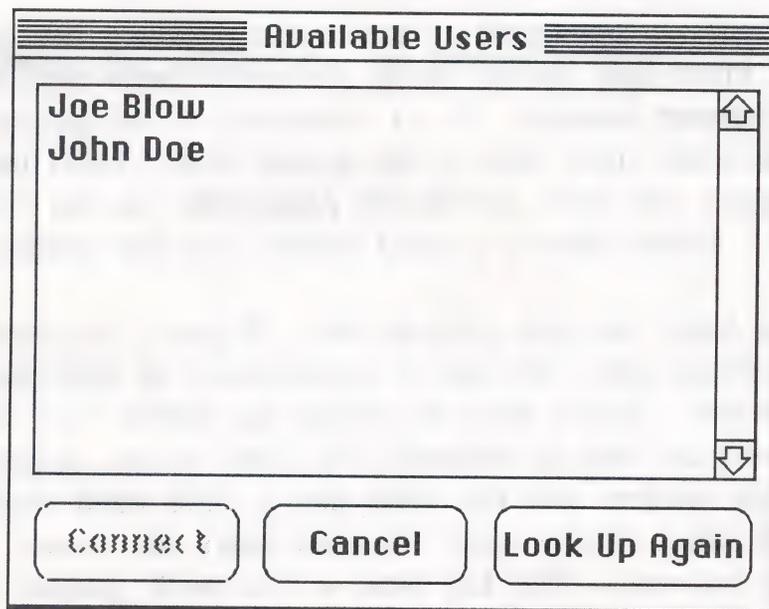


This dialog box lets you know that Okyto is now patiently awaiting the other person (who is making the connection) to arrive. If you get tired of waiting, you can click your mouse inside the "Cancel" button. Otherwise, the connection will occur when the other Okyto user arrives. You can skip forward now to the chapter "Now That You're Connected."

If you're making the connection, the following dialog box will appear:



Next to the words "Connect to user", type in the name of the person you wish to connect to, then click inside the "OK" button. If you're not sure how the other person entered their name, you should click inside the button titled "Look Up Users." Okyto will search the AppleTalk network for other users waiting for connection, and display them in the following dialog box:

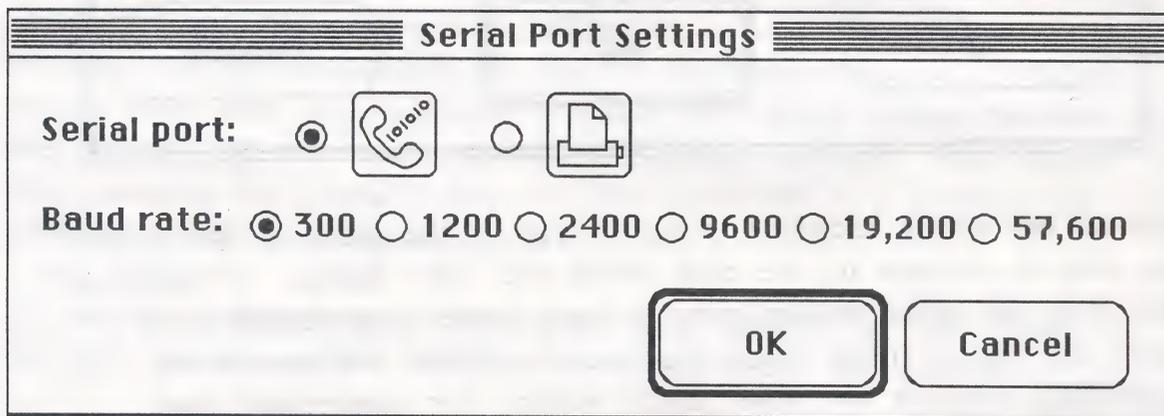


To select a user, click your mouse directly on their name (the name should become highlighted when you do this), and then click on the "Connect" button. A shortcut is to double-click on their name. It may be that the desired person isn't yet waiting for connection. If this is the case, wait a bit and then click on the "Look Up Again" button to see if they appear.

You can skip forward now to the chapter "Now That You're Connected."

If you're connecting via modem...

The following dialog box will appear:



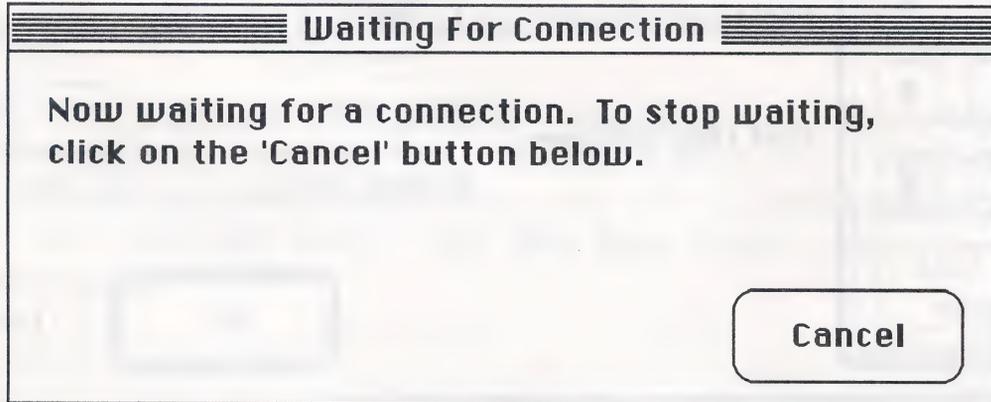
If the modem cable is connected to the modem port on the back of the Macintosh, click your mouse inside the small circle next to the icon of the telephone handset. If it's connected to the printer port, click inside the small circle next to the printer icon. Don't use the printer port unless you have turned off AppleTalk via the "Chooser" desk accessory. Unless there's a good reason, use the modem port.

Next, select the baud rate you wish to use. If you're not familiar with the term "baud rate", it's just a measurement of how fast you can send characters. You'll want to choose the fastest baud rate at which both users are able to transmit. In other words, if one user has a 1200 baud modem and the other has a 2400 baud modem, both users will select "1200 baud" because that's the fastest rate that both users can transmit. The key here is that **both people have to choose the same baud rate** in order for a connection to succeed,

so you'll need to discuss and agree upon this beforehand.

If you're waiting for a connection, Okyto will first send the appropriate commands to your modem to cause it to automatically answer and attempt to connect to the next incoming call (which, we hope, will be the Okyto making the connection).

The following dialog box will now be displayed:



This dialog box lets you know that Okyto is now patiently awaiting the other person (who is making the connection) to arrive. If you get tired of waiting, you can click your mouse inside the "Cancel" button. Otherwise, the connection will occur when the other Okyto user arrives. You can skip forward now to the chapter "Now That You're Connected."

If you're making the connection, the following dialog box will be displayed:

Dial A Number																
Number to dial:	<input style="width: 100%;" type="text"/>															
<table border="1" style="margin: auto;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>*</td><td>0</td><td>#</td></tr> <tr><td colspan="3" style="text-align: center;">Pause Two Seconds</td></tr> </table>	1	2	3	4	5	6	7	8	9	*	0	#	Pause Two Seconds			Telephone line is: <input checked="" type="radio"/> tone. <input type="radio"/> pulse.
	1	2	3													
	4	5	6													
	7	8	9													
*	0	#														
Pause Two Seconds																
Wait for answer up to <input style="width: 50px;" type="text" value="30"/> seconds.																
Dial this number: <input type="radio"/> until connected or <input style="width: 50px;" type="text" value="15"/> times. <input checked="" type="radio"/> only once.																
<input type="button" value="OK"/> <input type="button" value="Cancel"/>																

You should first select either the "tone" or "pulse" item, depending on what type of telephone line you're using. The "Wait for answer up to x seconds" item tells your modem how long to listen (after dialing) for another modem to answer the phone before deciding there's no connection. If you select the "Dial this number only once" choice, Okyto will do just that. However, if you wish to automatically redial the number until there is a connection (in case it's busy, for instance, or you arrive before the other person is ready), you choose the "until connected" choice. Note: to comply with the laws of the United States, you should not instruct Okyto to redial more than 15 times. If however, you are using Okyto in a country that does not regulate mechanical automatic redialing, you can enter the number zero in the box and it will redial forever (or until connected).

To enter the phone number to dial, just type it in the box next to the words "Number to dial:" or click your mouse inside the appropriate buttons inside the picture of the telephone touchpad. The "Pause Two Seconds" button is useful, for instance, for when you need to dial a series of digits (like a long distance service access number or a code to get through a switchboard to an "outside line") pause a calculated number of seconds, and then continue dialing. These pauses will

show up as commas in the number to dial.

For instance "99,,555-1212" means dial 99, wait 4 seconds (there's two two-second pauses there), and then dial 555-1212. By the way, there's nothing wrong with typing hyphens, parenthesis, or spaces between the numbers to dial. Such characters are ignored.

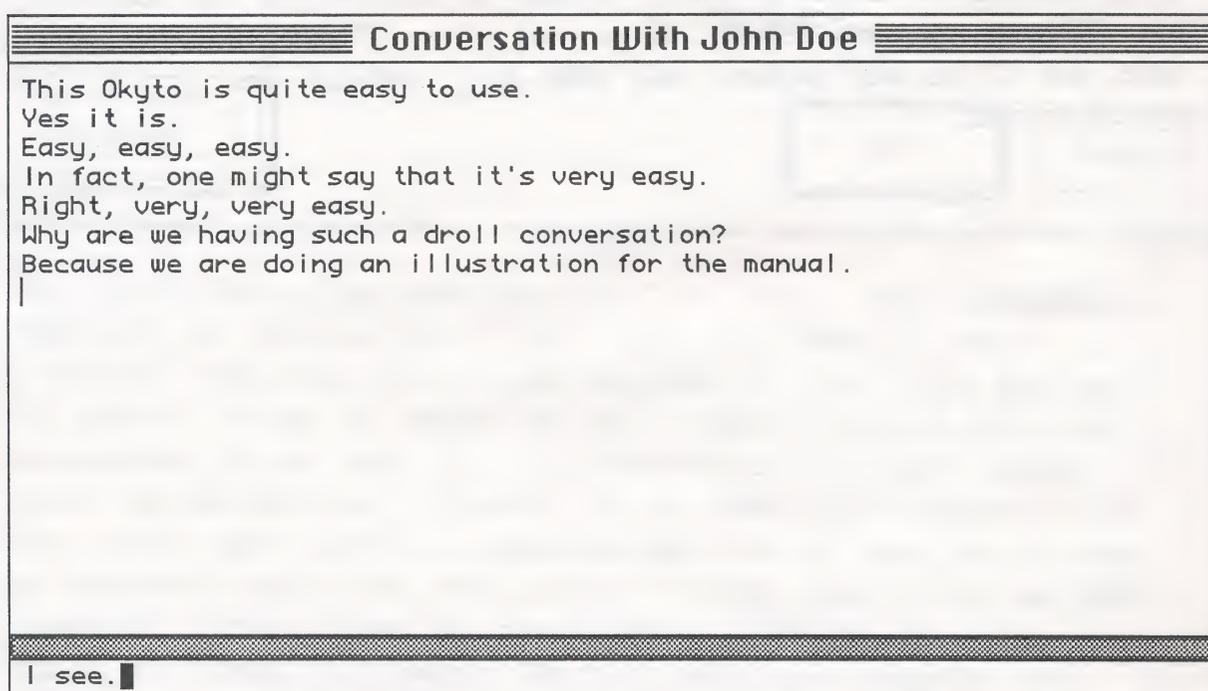
Finally, click on the "OK" button to get things going. If everything goes well (the other user is properly set up to be waiting for a connection) you will soon be connected with another Okyto!

A final note, after the connection is made, don't worry if it takes up to 20 seconds for Okyto to react. Usually it takes only a few seconds, but different modems report connections after different amounts of time, and Okyto will respond only after both modems are synchronized.

Now That You're Connected

At the moment of connection, three large windows will appear, one on top of the other in a staggered fashion. They're staggered so that you can click your mouse inside of any window to bring it to the top quickly. If you're connected with a person named Joe Blow, for instance, the windows will be titled (from top to bottom in original order) "Conversation With Joe Blow", "Outgoing Files To Joe Blow", and "Incoming Files From Joe Blow".

The Conversation Window



The Conversation Window is used to type messages back and forth with the other person. You can type a message at any time you're connected, even during file transfers. The letters you type appear at the bottom of the window below the horizontal dividing line. This text is not editable with your mouse - to get rid of something you've typed, just backspace over the mistake and retype it.

To send what you've typed, make sure the Conversation Window is the topmost window, and press the "Return" key on your keyboard. You'll see the mouse cursor (which is normally shaped like an arrow)

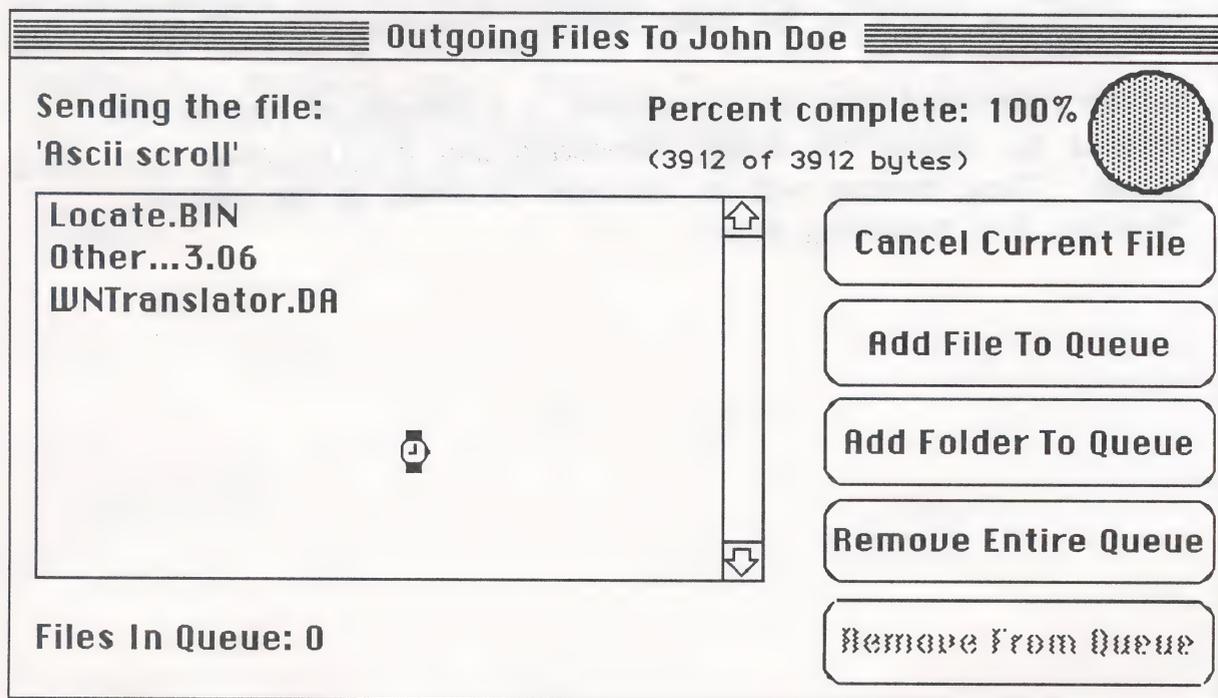
turn momentarily into a wristwatch shape. This tells you that Okyto is busy doing something. Whenever the cursor is a wristwatch, Okyto is telling you to hold off doing anything for a moment.

When the cursor returns to an arrow shape, you'll see the text you sent appear in the portion of the Conversation Window above the horizontal dividing line. This confirms that the text you sent arrived safely and is displayed in the other person's Conversation Window as well. Don't try sending another line of text until the last line you sent is shown in the top area of this window.

Note that the text above the horizontal dividing line is not editable, but you can select all or a portion of the text using the mouse (just as you would with a word processor) and then choose **Copy** from under the **Edit** menu to copy that text to the Macintosh Clipboard.

It's customary to type a greeting to the other person when you first connect. No need for anything lengthy, a simple "Howdy, Bub." is considered polite and to the point.

The Outgoing Files Window



The Outgoing Files Window is used to select and tell you about files

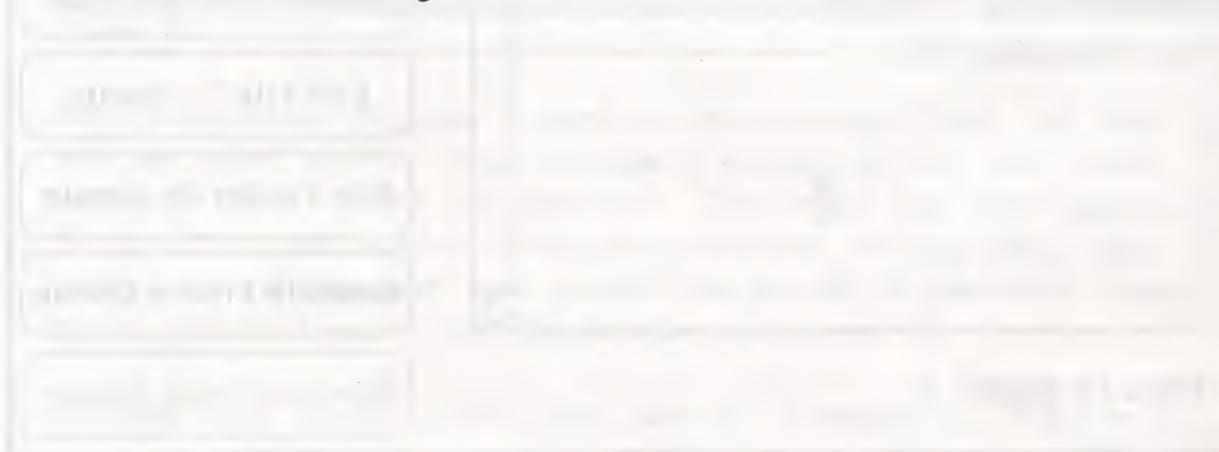
to be sent from your Okyto to the other person's. This window is divided into four different areas. At the top of the window is the status display. When no files are being sent from your Okyto, this area will display the message "Currently idle". When a file is being sent, this area will show the status of the file transfer, including the name of the file being sent, and how much of the file has been sent so far, both graphically, percentage wise, and in terms of actual bytes of data.

The middle left portion of this window is a rectangle with a vertical scroll bar. This is called the **Outgoing Files Queue**. With Okyto, you can specify that a whole batch of files are to be sent, rather than just one at a time. While one file is being sent, the rest of the files can be thought of as "standing in line" in this queue. When a file transfer is finished, a file is taken out of the queue and then sent.

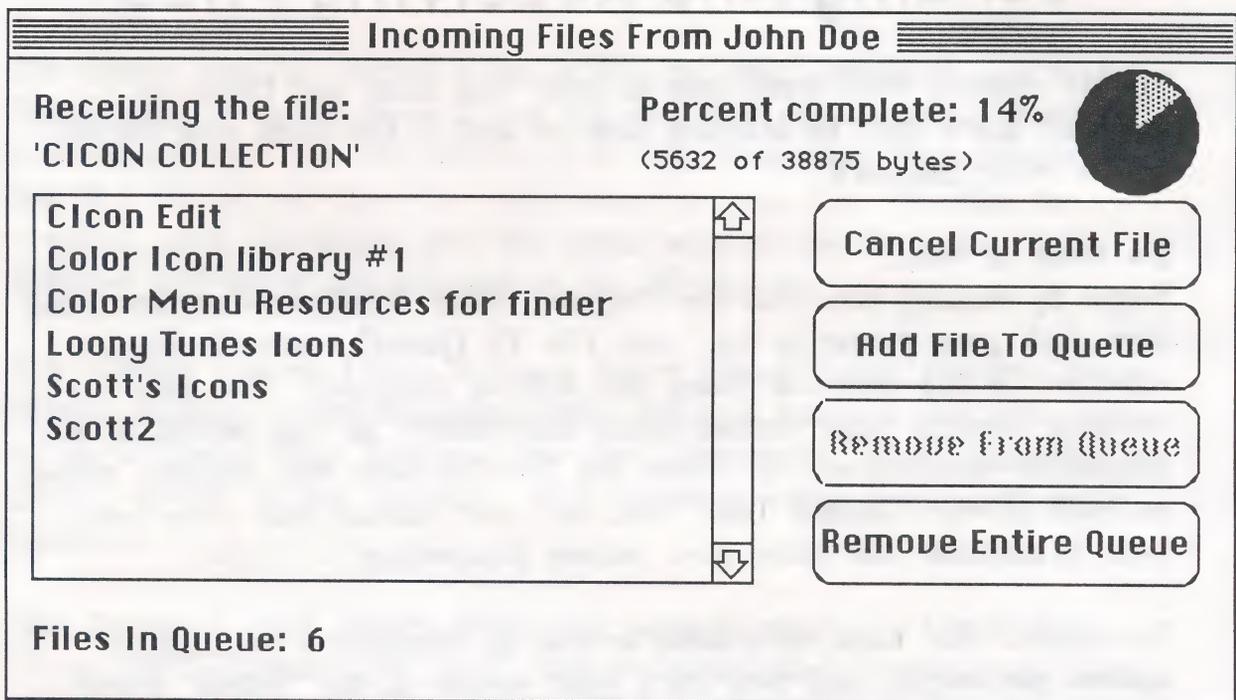
Below the queue is a line of text telling you the number of files in the queue. If a file is currently being sent, this number does not include the file being sent. You don't have to wait for a file to be completely sent before adding another file to the Outgoing Files Queue!

NOTE: When you add a file to your Outgoing Files Queue, it is automatically added to the other person's Incoming Files Queue!

On the right hand side of this window is a column of buttons that controls the current file being transferred and the Outgoing Files Queue. These buttons will be discussed in detail in the chapter "Sending And Receiving Files".



The Incoming Files Window



The Incoming Files Window is used to select and tell you about files coming into your Okyto from the other person's. Its layout is exactly like that in the Outgoing Files Window, described previously. We'll discuss the function of the buttons in this window in the chapter "Sending And Receiving Files". When a file is received, it is placed in the same folder as Okyto. I'll show you how to change the destination of received files later.

Sending And Receiving Files

In this chapter, we'll learn how to send files from one Okyto to another. Let's start by learning how to send a file from your Okyto to the other person's.

To send a file...

Begin by making sure that the Outgoing Files Window is topmost, and then click your mouse in the "Add File To Queue" button in that window. A file selection dialog box will be displayed. You can navigate through your various disks and folders as you would in any Macintosh application. If you're not familiar with this process, please go back to the "Guided Tour" disk and user manual that came with your Macintosh and learn how before proceeding.

To select a file, click on it once so that its filename is highlighted against the others, and then click your mouse in the "Select" button - or you can just double-click on the file name.

Now (if you can follow quickly enough, otherwise, just take my word for it), you'll see several things happen. First, the file will be shown in the Outgoing Files Queue. Next, it will be removed from this queue and then you will see the status display change to show you that the file is now in the process of being transferred. You'll also notice that the "Cancel Current File" button now becomes selectable. Clicking in this button cancels the current file transfer, deleting any portion that was received from the receiver's disk.

If you click on the "Add File To Queue" button a couple of more times and select more files to be sent, you'll see that they are added (in the order selected) to the Outgoing Files Queue. As the current file transfer ends, the topmost file in the Outgoing Files Queue is removed and made the current file transfer. To get rid of all of the files in the Outgoing Files Queue, just click your button in the "Remove Entire Queue" button. To get rid of an individual file in the Outgoing Files Queue, click your mouse on the file's name in the Outgoing Files Queue so that it becomes highlighted against the others, and then click on the "Remove From Queue" button. Neither the "Remove Entire Queue" nor the "Remove From Queue" functions

have any effect on the current file transfer.

To receive a file...

The easiest way to receive a file is for the other person to add that file to their Outgoing Files Queue (as described previously). You'll immediately see that file displayed in your Incoming Files Queue, and if a file is not currently being received, that file will then be removed from the queue and the status area of the Incoming Files Window will show the progress of the file receive.

The "Cancel Current File", "Remove Entire Queue", and "Remove From Queue" buttons in the Incoming Files Window function just like those in the Outgoing Files Window, described previously. The "Add File To Queue" button in the Incoming Files Window works a bit differently, however.

This button is used mainly when you want to receive a file from an Okyto that is unattended (we'll discuss unattended operation in more detail later). In this case, there's nobody around on the other end to add a file to their Outgoing Files Queue, so you have to do it for them.

When you click in the "Add File To Queue" button, you'll see the following dialog box appear:

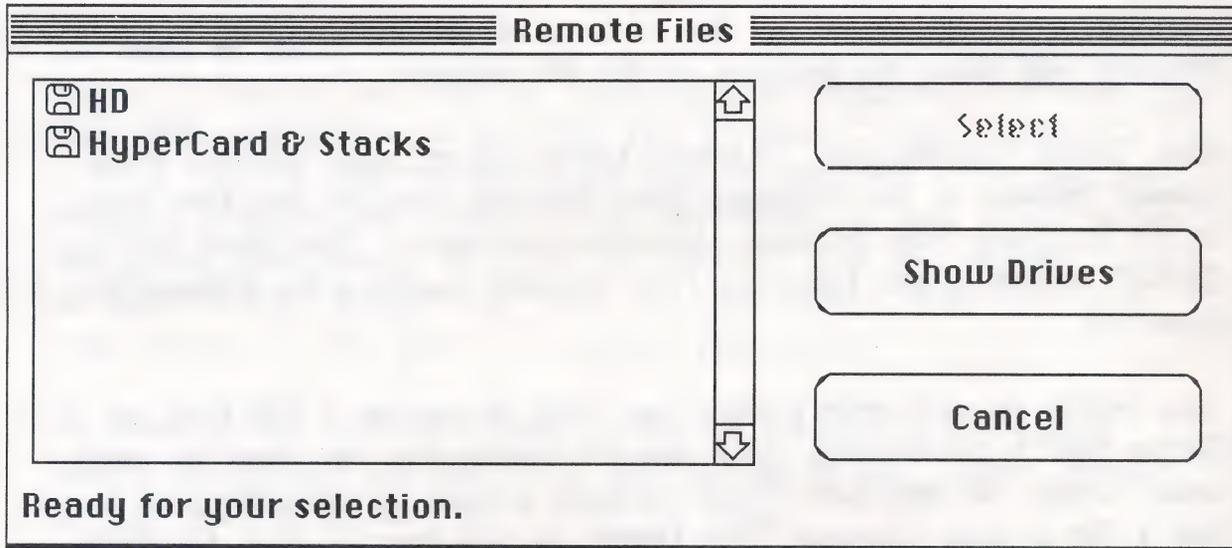
Please type the full pathname of the file you wish to receive from the remote machine:

Lookup Files **OK** **Cancel**

Now, you can either type in the full pathname of the file you wish to receive and then click on the "OK" button, or you can use a more Macintosh-like dialog box to display the available files and select the

one you want. We're not going to get into "full pathnames" here, if you're interested in finding out about them, please read the chapter "Designating Files By Full Pathname".

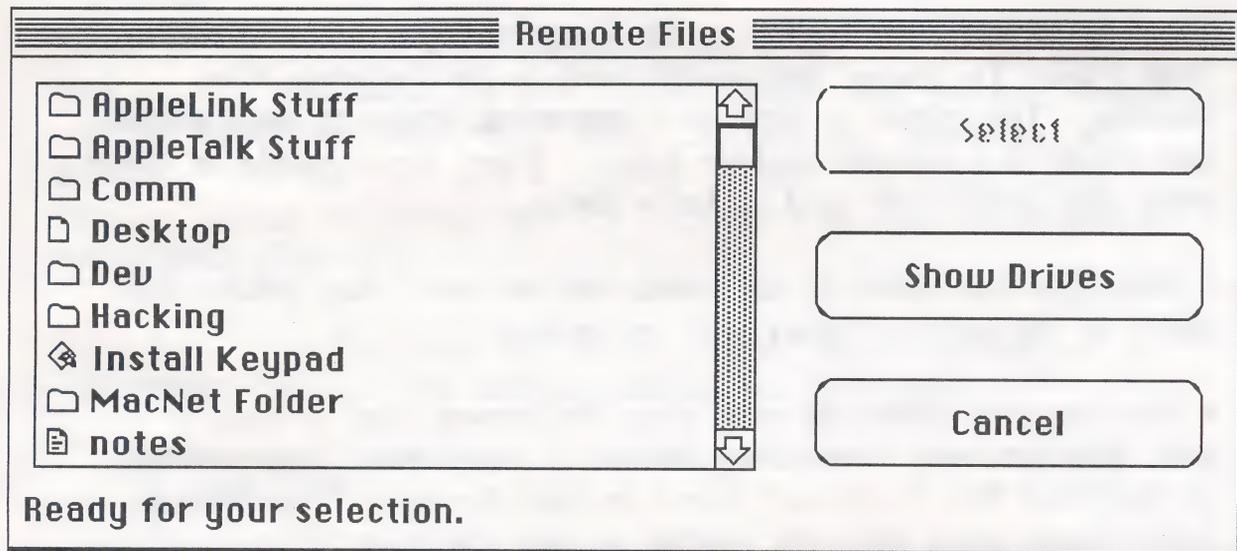
Let's keep it simple now. Just click in the "Lookup Files" button and the following dialog box will appear:



The first display shows a list of the mounted disks on the remote machine. To "open" a disk so that you can look at the files and folders on it, click on the desired disk name and then on the "Open Drive" button. Yes, I know there's no "Open Drive" button in the above picture. The "Select" button changes its title to reflect the type of thing (like "Open Drive" or "Receive Text File") that you've highlighted.

Once you've done this, the display area of the window will clear and the "Ready for your selection" text at the bottom of the window will change to "Waiting for directory information..." When that text is displayed, it means "Wait a minute while the other machine tells me what's there." Just wait until that text is replaced with "Ready for your selection." You are forewarned that at the slower baud rate speeds or with disks that contain a large number of files or folders, this can take a while. However, this process has been optimized to be as fast as possible. Be patient and let Okyto do its thing. You don't have to wait for it to finish listing. Once the thing you're after shows up, you can press your Return key to cancel the rest of the list.

The contents of the selected disk will be displayed. On one of my machines, it looks like this:



If you need to go into a folder, just click on the folder to highlight it and then click on the "Open Folder" button.

Unlike the regular Macintosh file selection dialog boxes, you can't "back up a level" once you've opened a drive or folder. To back up, you need to go back to the start, so click on the "Show Drives" button to do this.

Finally, you're going to get to the file you wish to receive. Click on the file to highlight it and then click on the "Receive" button. The file selection window will disappear, and then you will see that file immediately appear in your Incoming Files Queue. Ta da!

By the way, the little icons next to the items in the file display tell you what each item is. To wit:

-  - a disk drive (hard, floppy, or other degrees of rigidity)
-  - a folder
-  - an application file
-  - a text file
-  - a document (a file other than text)

Sending All Files In A Folder

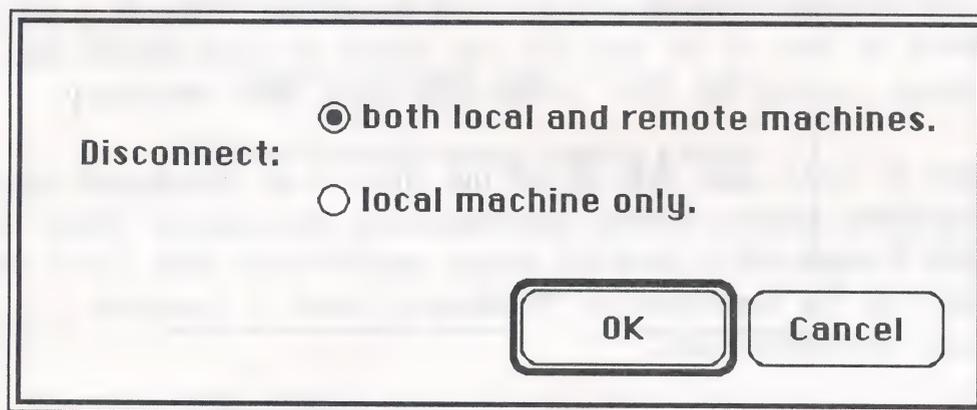
There is an additional button in the Outgoing Files Window titled "Add Folder To Queue" that is not found in the Incoming Files Window. This button is used as a timesaving device to send all files that reside in a selected desktop folder. There are a couple of things about this goody you need to know about.

- Only files that reside in the folder will be sent. Any folders that reside in the selected folder will be ignored.
- This function cannot be used when the remote user is looking at your disk directory (when they choose "Lookup Files" after clicking on the "Add File To Queue" button in their Incoming Files Window. You'll know when they are looking at your directory because an icon of a diskette will be displayed in the bottom of your Outgoing Files Window, and the "Add Folder To Queue" button will be dimmed.
- Likewise, the remote machine cannot look at your disk directory while you are in the process of adding files from a folder to their Incoming Files Queue. They will know that you are sending the contents of a folder because an icon of a folder will be displayed at the bottom of their Incoming Files Window, and the "Lookup Files" button will be dimmed.

Disconnecting

When you're finished with everything you've set out to do, there's two ways to disconnect the two Okytos. **Only one person should do the disconnecting when both Okytos are attended by people.** You can just send a message through the Conversation Window saying something like "I'll disconnect us now." to establish who's going to do it.

Let's look at the first way. Go to the "Okyto" menu and choose "Disconnect Now..." The following dialog box will appear:



As this dialog box always appears with "both local and remote machines" selected, you can just go ahead and click on the "OK" button. Your Okyto will send a message to the other Okyto to disconnect, and after the other Okyto agrees to do this, your Okyto will disconnect itself (hanging up the phone first if you've connected via modem).

The "local machine only" choice in this dialog box is there as a sort of "panic button" when things go sour. For instance, (and this is rather rare, so don't get paranoid) you may be connected via modem using a phone line that is so full of static that the modems may just plain give up the ghost and hang up on you. Unfortunately, neither Okyto will be aware this has happened. The status of the file transfers will appear to freeze. At this point, you'll want to use this selection to force Okyto to realize that it's lost contact.

It's important not to get in a panic just because the file transfers look frozen. It may be that the other person is selecting a file to be sent to you, or he may be in the middle of making a choice from a menu (in which case, his Okyto will not respond to yours during these periods). Don't fret; when they're through, everything will resume normally.

The second way to disconnect is to choose "Disconnect When All Tasks Are Completed" from under the "Okyto" menu. When this choice is checkmarked (you can uncheckmark it by selecting it again if you change your mind), Okyto will keep a close watch on the Incoming and Outgoing File Queues. When all file transfers are completed, and both Incoming and Outgoing File Queues are empty, Okyto will automatically disconnect both machines. This is a nifty little option to turn on so that you can attend to other chores and not worry about running up your phone bill more than necessary.

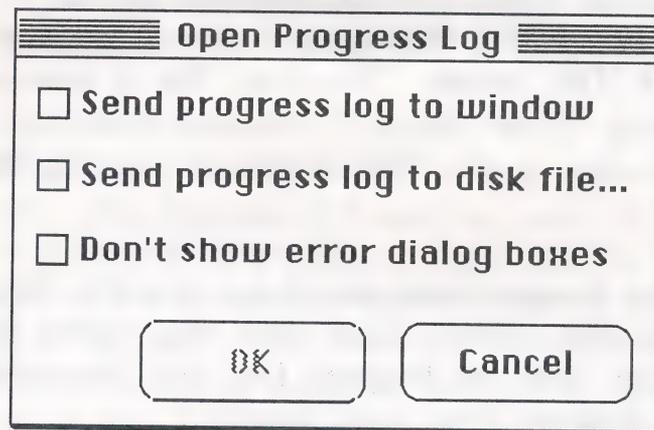
Remember to make sure that all of the files to be transferred are in their respective queues before checkmarking "Disconnect When All Tasks Are Completed". A little verbal confirmation that you're about to do this via the Conversation Window is worth a thousand premature disconnections.

NOTE: The "Quit" command under the "Okyto" menu should only be used when you are not connected to another Okyto. It does not perform any disconnection services.

The Progress Log

The Progress Log is used to keep tabs on the progress of a session, or to confirm the success or failure of a previous unattended session. It's also used to suppress error dialog boxes from appearing on an unattended machine.

To bring up the Progress Log, choose "Open Progress Log" from under the "Okyto" menu. A dialog box will appear that looks like this:



If you checkmark the "Send progress log to window" item, a window will appear on your screen titled "Progress Log". This window displays the most current 17 lines of progress information. As the window becomes full of lines, the topmost line is scrolled off into oblivion to make room for the new line at the bottom. You can, by the way, use your mouse to copy information to the Macintosh Clipboard just like you would in a word processing program.

Because the Progress Log Window only keeps track of the most current information that has been sent to it, you might want a more permanent record of everything that was posted to the Progress Log. To do this, checkmark the "Send progress log to disk file..." item. This will bring up a standard file creation dialog box where you can give the desired disk file a name and location. This disk file will be saved as a plain text file. Do not try to open this file by double-clicking on it in the Finder (you'll get an error message of the order "Application is busy or can't be found"). Instead, you'll need to open your favorite word processor/text editing application (like MacWrite, Word,

WriteNow, or QUED for instance), and then open the text file by choosing "Open" from under the "File" menu. Some word processors may require you to use a "Translator" type program on the text file before it can be opened by the word processor. If in doubt, ask the manufacturer of the word processor how to open a "plain text" file. Don't call me, because I don't keep track of the myriad of these things.

The last item, "Don't show error dialog boxes" can be selected whenever you've got the Progress Log going to either a disk file or a window. This choice, when selected, redirects any error messages to the Progress Log, rather than displaying a dialog box and waiting for you to click on the "OK" button. Therefore, this is especially useful when you are setting up an Okyto to run unattended, as there won't be anyone there to click on the "OK" button of a dialog box if an error should occur.

When you open the Progress Log, you'll see that the "Open Progress Log..." choice under the "Okyto" menu turns into "Close Progress Log". Select this choice to close the Progress Log disk file, and/or remove the Progress Log Window from your screen.

Automated Procedures

Okyto has a built in language that you can use to write an automated sequence of tasks, called a **Procedure**. Unlike most traditional "scripting" languages, Okyto's is appropriate for rank beginner and guru alike.

In fact, there are only 12 commands in this language, and you already know how they work! So please don't shy away from this powerful feature because of previous exposures you may have had with telecommunications software.

Procedures are not only useful for creating fully automatic and unattended sessions, they can be useful in any session as a timesaving device. For instance, if I need to send 10 files to the other machine, I'll need to first connect, then click on the "Add File To Queue" button in the Outgoing Files Window 10 times, each time selecting the file to be sent. This can waste a lot of time in an actual connection. A better solution would be for me to create a Procedure containing 10 "Send The File" commands. Then, I make the actual connection and execute that Procedure. The first time you try this you'll appreciate how much time (and phone tolls) you've saved by using a Procedure.

Remember, a Procedure replaces your interaction while it's executing. Don't try clicking on any buttons in the Incoming or Outgoing Files Windows while a Procedure is executing (you can, however, continue to send text via the Conversation Window). A Procedure, however, has no effect on the other Okyto - the other person is free to do whatever they like while a Procedure is executing on your machine.

Creating A Procedure...

To create a Procedure, select "New" under the "Procedure" menu. An empty window will appear, with a black dot in the upper left corner. This black dot is the **insertion point**. It tells you where the next command you add will be inserted. Once you've added a few lines, you'll see that you can move the insertion point to any command line simply by clicking your mouse on that line. If you add a command

when the insertion point is on the same line as a command, Okyto will move all lines from the insertion point forward one line, and place your new command where the insertion point was. To append new commands at the end of the Procedure, move the insertion point to the line following the last command. The insertion point will automatically move forward as new commands are added.

You can't modify a command once it has been entered. The only way to change a command is to first remove it by placing the insertion point on that line and selecting "Delete Command" under the "Procedure" menu, and then inserting a new command in its place.

Saving A Procedure...

Once you're satisfied with a Procedure, you can choose "Save" or "Save As..." from under the "Procedure" menu to save the Procedure to disk. "Save As..." differs from "Save" in that it saves a copy of the Procedure under a different name than it was previously saved with.

If you choose the "Revert" command under the "Procedure" menu, all changes you've made since the Procedure was last saved are thrown out, and the Procedure is reloaded from disk.

Opening A Saved Procedure...

To open a Procedure that you've previously saved, choose "Open..." from under the "Procedure" menu.

Executing A Saved Procedure...

To execute a Procedure, it must be saved first. If you've opened a saved Procedure, make some changes, and then execute the Procedure before saving it with the changes, the version on the disk (without the changes) is actually executed. To execute a Procedure, choose "Execute..." under the "Procedure" menu. When a Procedure is running, you'll see that the "Execute..." choice becomes "Cancel Procedure". Select this to stop the execution of the Procedure.

Procedure Commands

Each of the 12 Procedure commands will be discussed in detail in this chapter. To add a Procedure command, simply select that command from under the "Command" menu. Depending on the type of command, you may need to satisfy one or more dialog boxes before the command is actually added.

Send The File...

This command is analogous to clicking on the "Add File To Queue" button in the Outgoing Files Window. It prompts you for the file to be sent.

Receive The File...

This command is analogous to clicking on the "Add File To Queue" button in the Incoming Files Window. You cannot however, use the "Lookup Files" button in the file definition dialog box that appears, so you'll need to type in the full pathname of the file to be received. For more information on full pathnames, see the chapter "Designating Files By Full Pathname".

Wait until...

This command allows you to make the Procedure wait until a certain event occurs. There are four variants of this command, as the dialog box will prompt you to select from:

- 1) Wait until time of day. You enter the time in 24 hour format (10:05:00 P.M. would be entered as 22:05:00 - leading zeroes are necessary!). The Procedure will wait until that time of day before proceeding with the next Procedure command.
- 2) Wait until x seconds have elapsed. You enter the number of seconds. Okyto will wait this number of seconds before proceeding with the next Procedure command.
- 3) Wait until disconnected. Okyto will wait until it is not connected to another Okyto before proceeding with the next Procedure command.
- 4) Wait until all tasks are completed. Okyto will wait until all file transfers are finished, and the Incoming and Outgoing File Queues are empty before proceeding with the next Procedure command.

Connect...

This command is equivalent to choosing "Perform Connection Now..." from under the "Okyto" menu. You'll be taken through all of the dialog boxes you would if you chose the "Perform Connection Now..." menu choice all the way up to the point where an actual connection would occur.

If Not Connected...

There are two variants of this command, as the dialog box will prompt you to select from:

- 1) If Not Connected, Stop. If Okyto is not currently connected to another Okyto when this command is executed, the current Procedure is cancelled.
- 2) If Not Connected, Execute The Procedure File... If Okyto is not currently connected to another Okyto when this command is executed, the current Procedure is cancelled, and a second Procedure (which you'll be prompted to designate when selecting this command) will be immediately executed.

Both of these commands are useful after a Connect... command to verify that you really did make a connection.

Disconnect When All Tasks Are Completed

This command is equivalent to checkmarking the "Disconnect When All Tasks Are Completed" choice under the "Okyto" menu.

Execute The Procedure File...

This command is equivalent to choosing "Execute..." under the "Procedure" menu. The current Procedure is closed, and the designated Procedure is immediately executed.

Launch The Application

This command is equivalent to choosing "Quit" from under the "Okyto" menu, and then executing another Macintosh application from the Finder. Make sure you are disconnected before using this command, as it performs no disconnection services.

Quit To Finder

This command is equivalent to choosing "Quit" from under the "Okyto" menu. See the note at the end of the chapter "Disconnecting" before using this command.

Sound A Bell

This command sounds a brief tone on your Macintosh's speaker (on some Macintosh models, it plays the "Alert Sound Setting" you specify in the Control Panel). It's useful to let you know that a certain point in your Procedure has been reached while you're paying attention to something else.

Open Progress Log...

This command is equivalent to choosing "Open Progress Log..." from under the "Okyto" menu. See the chapter "The Progress Log" for more details about what this means.

Close Progress Log

This command is equivalent to choosing "Close Progress Log" from under the "Okyto" menu. See the chapter "The Progress Log" for more details about what this means. If the Progress Log is not open when this command is executed, it has no effect. Therefore, it's a good idea to use this command before any "Open Progress Log..." commands just to be on the safe side.

Send The Contents Of The Folder...

This command is equivalent to clicking on the "Add Folder To Queue" button in the Outgoing Files Window. Note that this command cannot be queued - it can't be used when Okyto is already sending the contents of a folder. Therefore, if you want to send more than one folder, do it in this way:

Send The Contents Of The Folder...

Wait Until All Tasks Are Completed

Send The Contents Of The Folder...

Wait Until All Tasks Are Completed

Send The Contents Of The Folder...

and so on.

Example Procedures

Because your needs will be unique, it's difficult to provide an example Procedure that would be meaningful to your application of Okyto. However, here's one that I've found useful, and it will also teach you how Procedures are created.

A Continuous Wait For Connect Procedure

- 1) Choose "New" under the "Procedure" menu.
- 2) Choose "Close Progress Log" under "Command".
- 3) Choose "Open Progress Log..." under command.
- 4) Checkmark both the "Send progress log to window" and "Don't show error dialog boxes" items, and click on the "OK" button.
- 5) Select "Connect..." under the "Command" menu.
- 6) Fill out the dialog boxes to make your Okyto wait for a connection.
- 7) Select "Wait Until.." under the "Command" menu.
- 8) Select the radio button next to the words "until disconnected." and then click on the "OK" button.
- 9) Choose "Save" under the "Procedure" menu and give this Procedure a name.
- 10) Choose "Execute The Procedure..." under the "Command" menu. When prompted, select the Procedure file you just saved.
- 11) Finally, Choose "Save" under the "Procedure" menu. This automatically saves the Procedure (remember, we added a line since our last "Save") with the same name as before.

Voila! When this Procedure is executed, it will immediately go into the "waiting for a connection" mode. Once a connection is established, it patiently waits for the connection to end. When that happens, the Procedure runs itself over again to wait for the next connection.

The reason for opening the Progress Log at the beginning of this Procedure is to be able to suppress any error dialog boxes that might occur. Such a dialog box would "freeze up" Okyto if it were unattended.

On the other end, a person might create a Procedure to connect with yours that looks like the one below. The below is just how the

Procedure editing window would look after entering each command. Note, however, that the comments in parentheses are mine and would not show up in the actual Procedure window.

(Just in case a Progress Log is already open...)

Close Progress Log

(Now open a progress log, send it to a window or a disk file if you wish, and select the "Don't show error dialog boxes" item so errors won't "freeze up" my system.)

Open Progress Log

(Wait until three o'clock in the morning)

Wait Until 03:00:00 Before Proceeding

(Now, make a connection to the waiting Okyto. By the way, don't worry about the "ATS7=30DT" junk before the number, that's the actual modem command Okyto sends to cause the modem to dial.)

Make 1200 Baud Connection Via Modem, Dialing 'ATS7=30DT 555-1212'

(If the connection fails, get outta town)

If Not Connected, Stop

(Otherwise, send the three files)

Send The File 'Hard Disk:First File'

Send The File 'Hard Disk:Second File'

Send The File 'Hard Disk:Third File'

(Now, we'll just sit back and relax until the file transfers are finished, and then disconnect both Okytos.)

Disconnect When All Tasks Are Completed

(And let me know when things are done with a couple of bells. What am I doing up at three o'clock in the morning, anyhow?)

Sound A Bell

Sound A Bell

As you can see from the above examples, it's easy to construct very complex (yet easily understood) sequences with just the 12 Procedure commands that Okyto offers.

Now That You're A Guru...

The title of this chapter may seem a bit facetious, but the truth is that if you've made it this far, you are performing things with Okyto which has caused other people to lose perfectly good heads of hair.

Take a moment to decide whether or not you'd like to challenge your good luck by going a bit farther.

Still with me? Good.

Under the "Okyto" menu, choose "Settings..." You'll see a dialog box like the following:

The dialog box is titled "Modem initialization command:". It features a text field containing "ATEOQ0V1". Below this are four buttons: "Set Timing/Retry Values...", "Set File Transfer Block Size...", "Set Received File Destination...", and "Set Security Passwords..". Underneath these buttons is another text field labeled "Current received file destination:" containing "Same volume/folder as Okyto.". At the bottom, there are three checked checkboxes: "Use color", "Echo password entry with *'s", and "Using MultiFinder". Finally, there are "OK" and "Cancel" buttons at the bottom right.

Modem initialization command:

ATEOQ0V1

Set Timing/Retry Values... Set File Transfer Block Size...

Set Received File Destination... Set Security Passwords..

Current received file destination:

Same volume/folder as Okyto.

Use color Echo password entry with *'s

Using MultiFinder

OK Cancel

NOTE: Danger! The "Set Timeout/Retry Value...", "Modem initialization command:", and "Set File Transfer Block Size..."

items should be fiddled with only under the direction of FreeSoft! If you can't resist playing with these, do yourself a favor and write down the original values so you can restore them later if things go "boom".

There are a number of things in this dialog box that you may want to play with.

Use color

This item is selectable only on a Macintosh that supports Color QuickDraw. If it is checkmarked, color will be used to enhance the various displays in Okyto.

Using MultiFinder

This item must be checkmarked if you are running Okyto under MultiFinder. If it is not checkmarked under MultiFinder, the "Launch Application" Procedure command will not function properly.

Echo password entry with *'s

If this item is checkmarked, anything typed into the "Password:" item in the "Perform Connection" dialog box will be displayed as asterisks. This feature is useful if someone is watching over your shoulder and you don't want them to find out your password.

Set Received File Destination

You'll recall that when Okyto receives a file, it places it in the same folder that Okyto resides. A quick look at the bottom of the above dialog box explains why - the default "received file destination" is the same volume/folder as Okyto. You can change the destination of received files by clicking the "Set Received File Destination..." button. This button brings up a dialog box allowing you to select drives and open folders on those drives until you are inside the folder that you wish received files to go. When you get there, just click on the "Select" button.

Set Security Passwords

Okyto allows you to use up to three levels of password protection to control what functions are available to those who connect with you. Recall that there's a "Password:" item just below the "My name is:" item in the first dialog box you see after selecting "Perform

Connection Now..." under the "Okyto" menu. This is where the password is entered.

The default state of Okyto is that no passwords are used. This means that anyone who connects with you may perform any function, including looking at the contents of your disks and extracting any files they choose. You may, for instance, wish to limit that capability only to yourself (calling from a remote Okyto), or to a few select others who have been given the appropriate password.

To install entry passwords, click on the "Set Security Passwords" button. You'll see a dialog box like the one below:

Set Passwords

Full clearance:

Password:

Clearance to add files to my Outgoing Files Queue only:

Password:

Clearance to add files to my Incoming Files Queue only:

Password:

Don't replace existing files

A password may be up to eight characters long, and upper and lower case letters are considered the same (Bobby and BOBBY are the same password). If two passwords are alike, the connecting person is given the higher of the two levels with like passwords (the priority is from top to bottom as they are listed in the above dialog box).

For example, let's say you install the password "FULL" for full clearance, "OUTFILES" for clearance to add files to your outgoing files

queue only, and "INFILES" for clearance to add files to your incoming files queue only. If a person were to connect to your Okyto, they would have to know one of these passwords to do anything. If they attempt to do something they don't have clearance for, they will get the error message "You don't have clearance for that function."

If a password is left blank for a particular level, that means that anyone will be assigned that clearance level that calls in without supplying a password. So, if you're going to use passwords, use them for all three levels.

The strategy here is to choose a password that cannot be easily guessed by an unwelcome visitor. Using passwords requires a bit of concentration, because if a person types the password incorrectly, there's no way to re-enter it except by disconnecting and connecting once again. We want to make it as troublesome as possible for an unwelcome visitor who happens to know your Okyto is waiting for a connection.

The button "Add Off Limits Folder" is used to declare a folder off limits to outside callers. When clicked, you are asked to select a folder from a dialog box. You can declare as many folders as you like as off limits. Remember that only that folder is declared off limits - the folders contained in that folder are not off limits unless they are specifically declared as such. The names of these off limits folders are kept in a file called "Okyto Off Limits Folders". Don't throw away that file or move it out of the folder containing Okyto. You can add a file that resides in an off limits folder to your Outgoing Files Queue, but remote callers can not request a file from such a folder. It's a real good idea to make your "System Folder" off limits.

If a request is made for Okyto to receive a file, and a file by that name already exists in the destination folder, Okyto will normally destroy the existing file in favor of the one to be received. If the checkbox "Don't replace existing files" is checkmarked, Okyto will not allow existing files to be deleted.

One more thing. For your protection, Okyto will not allow a file to be received with a name of "Okyto" or "Okyto Off Limits Folders". You must also never change Okyto's filename or it will not work.

Designating Files By Full Pathname

In various places throughout this manual, the words **full pathname** is used to indicate a group of characters that describe the location of a file (either a Macintosh application or document) on a disk. There are different ways of doing this, depending on the kind of disk filing software your Macintosh is using.

The original Macintosh filing system was called MFS (Macintosh Filing System), which worked well for floppy disks. Folders that are displayed in a Finder window are for cosmetic purposes only, and they do not play a part in constructing a full pathname for Okyto. An MFS full pathname consists of the disk volume name (which appears under the disk icon in the Finder), a colon character, and the name of the file (which appears under the file's icon in the Finder).

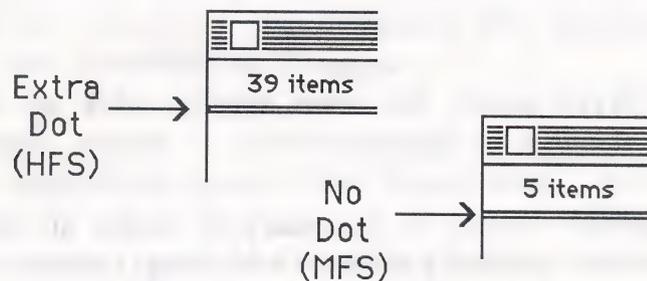
Under MFS, a file with the name of "Junk" that resides on a disk named "Stuff" would be given the full pathname "Stuff:Junk".

When hard disks became common in the Macintosh world, it became necessary for Apple to devise a more efficient filing system, which they named HFS (Hierarchical Filing System). Under HFS, the names of folders which lie between the desktop (the top level window in the Finder) and the file are significant and must be included in the full pathname in the order they appear and separated by colon characters. An example should clarify this.

Under HFS, a file named "Junk" is inside of a folder named "Garbage". The "Garbage" folder is inside of another folder named "Trash". The "Trash" folder lies on the desktop of a disk named "Stuff". The proper full pathname would then be:
"Stuff:Trash:Garbage:Junk" - the order is important!

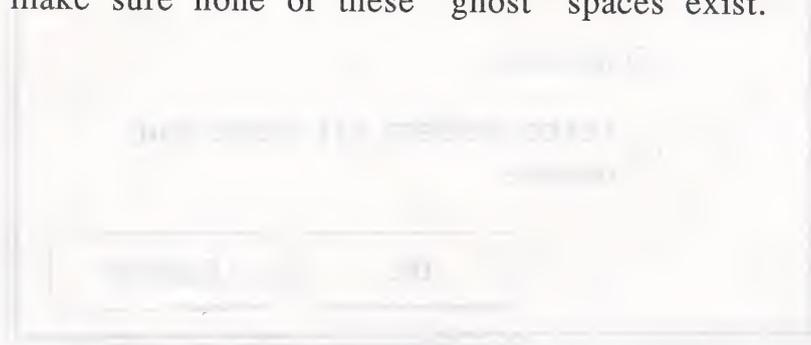
How do you know if a disk uses HFS or MFS? One sure fire way is to make sure the disk has at least one folder, and then look at that disk by creating a junk Procedure and choosing "Save" from under the "Procedure" menu. If the file creation dialog box displays the folder, you've got an HFS disk, otherwise it's MFS. Another way is to open the disk icon in the Finder and look at the upper left hand corner of the window. There is a small division between the two lines that

appear just below the phrase that tells you the number of items on the disk. A HFS disk will have an extra dot at the far left of this division, and a MFS disk will have no dots in this division. The illustration below shows an example of both kinds of disks.



It's important to know what kind of disks you are working with so that you don't construct a faulty full pathname and get an error message to the tune that "no such file exists."

One last tip. If you are sure that you've spelled everything correctly and are positive a full pathname should work but doesn't, you should be aware that the Finder allows you to (inadvertently or otherwise) put leading or trailing spaces in disk, folder, and file names. If these are present, they must be included in the full pathname you present to Okyto. So, if problems persist, try renaming the entities in a full pathname to make sure none of these "ghost" spaces exist.

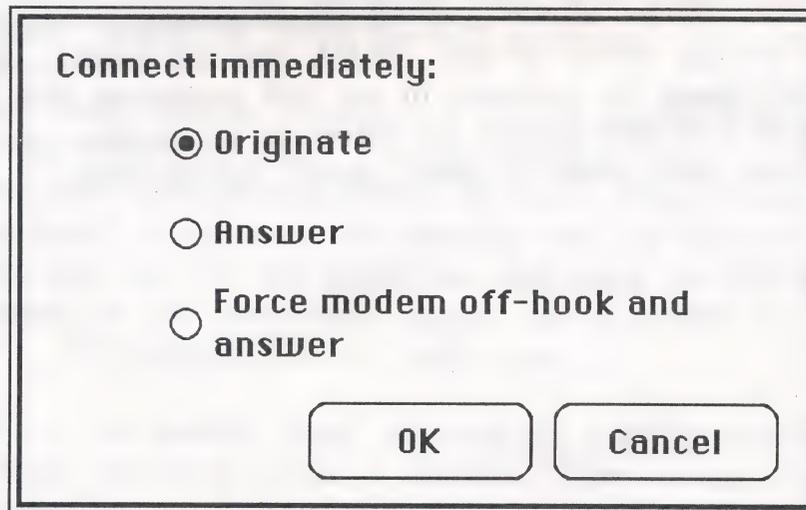


A Special Way Of Connecting

You'll notice that under the "Okyto" menu, there is a choice "Force Immediate Serial Port Connection...". This choice is used for the following reasons:

- 1) The answering Okyto won't, for some reason, pick up the phone when it rings and attempt to connect.
- 2) Some sort of special routine is necessary to make an outgoing call, and this must be done manually (Okyto's dialing routine won't work).
- 3) You wish to quit Okyto, do some other work, and then resume the connection without having to hang up and call again.

When you choose "Force Immediate Serial Port Connection", you'll see the following dialog box:



Connect immediately:

Originate

Answer

Force modem off-hook and answer

OK **Cancel**

The "Originate" choice should be selected if you are **making the connection** and the modems are already connected.

The "Answer" choice should be chosen if you are **waiting for connection** and the modems are already connected.

The "Force modem off-hook and answer" choice should be chosen if you are **waiting for connection**, and you can hear the phone ringing but your modem won't take the phone line off hook for some

reason (we've seen this happen on some European phone lines).

If you wish to quit Okyto, do something, and then later reconnect without having to hang up and call back, here's how to do it:

- 1) Make sure the Incoming and Outgoing File Queues are empty and there are no file transfers in progress.
- 2) Tell the other person to choose "Quit", re-run Okyto, and then select "Force Immediate Serial Port Connection", and select "Answer".
- 3) You now choose "Quit", do whatever needs to be done, then run Okyto again, choose "Force Immediate Serial Port Connection" and select "Originate".
- 4) Both Okyos should now reconnect.
- 5) It isn't possible to reconnect under AppleTalk, so if you choose "Quit" during an AppleTalk connection, the local machine will also disconnect from AppleTalk. Therefore, it's important in an AppleTalk connection to always disconnect both sides before quitting.

Closing Thoughts

The decision to include Okyto with White Knight 11 (for a limited amount of time) was one of the most difficult I've ever had to make. I literally turned my back on the advice of everyone I asked. This is not a simple program on the inside. It is extremely complex and took months of debugging and rewriting to get satisfactory performance and ease of use, and yet still offer a high degree of flexibility and security. I could have (and probably will) easily charged anywhere from \$49 to \$79 a copy and made a tidy sum. However, I've decided to do it this way for a couple of good reasons:

- By bundling it, I am causing it to be distributed much more widely than if I sold it separately. As a separate product, it would have ended up mainly in the hands of novices. In this way, I'll get it into the hands of both novices and gurus alike, and people of all levels of experience will have a lowest common denominator to fall back on. None of my customers will ever need to feel intimidated when they are asked to participate in a file transfer.
- It's my way of saying "thanks". My company has been an unexpected and tremendous success, and I'd like to show this token of appreciation for your support. There's no better way to say thank you than with a gift.

The great danger of doing it this way is that people tend to think that something given away has little value. If I get into a situation where I recognize that Okyto is being given away freely to those who haven't purchased or updated to White Knight 11, I'm going to feel pretty stupid for doing this much work and getting cheated for it. Please honor the long hours that went into this, and the risk I am taking by bundling it. Don't give it away.

There will be situations where folks will decide that Okyto is the cat's meow for their application, and will need to order multiple copies to satisfy a group demand. It would be unreasonable to expect you to purchase multiple White Knight's just to get the "free" Okytos, so I've decided to offer a special price where you can get legal copies at a painless price. Standalone copies of Okyto may be purchased through The FreeSoft Company directly (not through dealers or distributors) at a price of \$39.95 per copy, with substantial discounts for quantity orders available. Please contact us for details.





PRODUCT REGISTRATION

Thank you for purchasing White Knight. To receive technical support, update information for major version releases and new product announcements, you *must* return this card. We will notify you of the next major version release of White Knight and/or Okyto by mail.

**KEEP YOUR MASTER DISK IN A SAFE PLACE -
YOU'LL NEED IT TO UPDATE!**

*** PLEASE TYPE OR PRINT CLEARLY *** (WE CAN'T NOTIFY YOU IF WE CAN'T READ IT)

Name _____

Address _____

City _____ State _____

Zip _____ Country _____

Company _____

Title _____ Home Phone () _____

Work Phone () _____ Macintosh Model _____

Please advise us of any name or address changes

WE MUST HAVE YOUR DATED SIGNATURE ON THE LINE BELOW!!!

I acknowledge that I am the legal purchaser of White Knight and Okyto, and agree to the terms in the End User License Agreements contained in the White Knight and Okyto user manuals.

Signature _____ Date _____

If you wish, please answer the following questions:

Where did you purchase White Knight? _____

How did you hear about White Knight? _____

Was White Knight purchased for use at home or business? _____

What is your level of computer expertise? _____

Do you have suggestions for improvements, new features or just general comments? _____

PLEASE CONTINUE COMMENTS/SUGGESTIONS ON A SEPARATE PAGE IF NECESSARY

THANK YOU!

RETURN ADDRESS

PLACE
STAMP
HERE

The FreeSoft Company

105 McKinley Road

Beaver Falls, PA 15010-1001

U.S.A.

INTRODUCING ...

SCOTT WATSON'S

SECOND SLIGHT[®]

THE WORLD'S MOST POPULAR BULLETIN BOARD CONSTRUCTION SET FOR THE APPLE MACINTOSH!!

We would like the opportunity to introduce to you Scott Watson's Second Sight. Second Sight, formerly known as Red Ryder Host, is a full featured Bulletin Board System (BBS) which runs on all 1 megabyte of memory and up equipped Macintosh models. When running under MultiFinder, Second Sight uses a 500K memory partition. Second Sight supports international character sets (any Macintosh font may be used), and will operate entirely in the background under MultiFinder.

Second Sight services a single telephone line, connected to a Hayes compatible modem. It was written to be as user-customized as possible, and can be thought in terms of a Bulletin Board Service construction set. The user sets up their bulletin board exactly the way they need it - there are no "built-in" menus to limit how your service will appear to a caller. We have spoken with many people who have told us that without the benefit of any previous communications experience, they were able to set up a fully functional BBS using Second Sight in only a few days of casual work. A full tutorial in the 100+ page user manual takes the user through setting up a sample BBS, teaching how the various parts of Second Sight fit together. For a free brochure listing Second Sight's most significant features and several authorized dealers, please return the coupon below.

PLEASE TYPE OR PRINT LEGIBLY

NAME _____

COMPANY _____

ADDRESS _____

CITY, STATE, ZIP _____

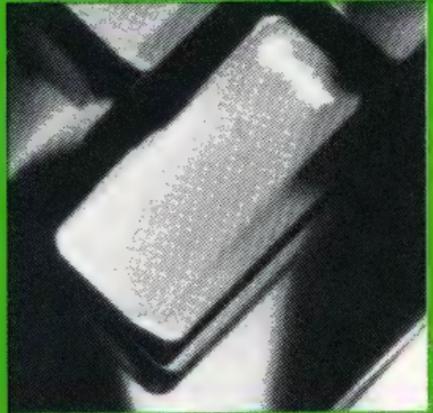
COUNTRY _____

FOR A FREE SECOND SIGHT
BROCHURE, MAIL COUPON

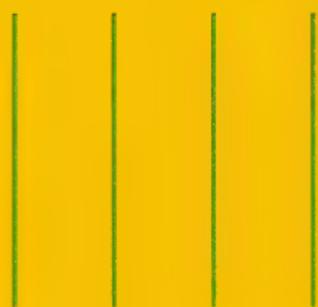
TO:

THE FREESOFT COMPANY
105 MCKINLEY ROAD
BEAVER FALLS, PA 15010
USA

**A Little
Extra Push
Goes a
Long Way
In Business.**



DOW JONES
NEWS/RETRIEVAL®



For Your Most Pressing Decisions, Rely on Dow Jones Exclusives.

News/Retrieval is noticeably different from other online services. It's the only service designed specifically for private investors and business and financial professionals. Much of the vital decision-making information provided by News/Retrieval cannot be found on any other online service.

Call now to subscribe and get: one FREE hour online, plus a waiver of our \$29.95 Start-up Fee. And ask about our low **After Hours/Flat-Fee Pricing Plan for personal investors.**

Let News/Retrieval Point the Way.

You can search high and low for answers. Or you can search quickly and efficiently using News/Retrieval. Designed for private investors, as well as business and financial professionals, News/Retrieval brings you up-to-the-second news about companies, markets, stock prices, industries and investments. All easy to access. And if you'd like to gain a broader perspective, News/Retrieval has the historical data you're after.

When you turn to News/Retrieval, all you need to know is at your fingertips. And there's no higher-quality resource. After all, Dow Jones has been supplying decision makers with timely, trustworthy information in a variety of forms for over one hundred years.



All Yours.

Dow Jones is the *exclusive* online provider of the full text of *The Wall Street Journal*. Authoritative and highly respected, the *Journal* is just one of many sources of information available to you as a user. Get the complete text of selected articles from *Barron's*®, *Business Week*, *Forbes*, *Fortune*, and *The Washington Post*. Plus hun-

dreds of other national and regional business and industry-specific publications --yours in seconds.

Extra Confidence.

Whatever the business day brings, you're ready. Once online, you have more than just the best print publications backing you. You have *exclusive* Dow Jones newswires. Together, the wires deliver the fast, hard business and financial information you need from around the world. And now you can search *all* the wires simultaneously.

- ▶ Get continually updated news about companies and industries with Dow Jones News Service® (the Broadtape).
- ▶ Stay on top of unusual trading patterns with Professional Investor ReportSM.
- ▶ Receive immediate notification of SEC filings and follow-up details with select Federal FilingsSM newswires online.

There are five exclusive Dow Jones newswires plus Business Wire and PR Newswire. These additional sources deliver the full text of news releases from corporations, government agencies, industry associations and more.

Membership Gives You Online Access to:

Breaking Business News
Updates on Competition
Investment Analysts' Reports
Detailed Corporate Profiles
Financial Overviews
P/E Ratios
Current Stock Quotes
Historical Stock Quotes
Company vs. Industry Performance
Instant "Backgrounder" Information
Income Statements/Balance Sheets,
plus other 10K and 10Q Data
SEC Insider Trader Activity
Earnings Reports and Forecasts
An Online Business Library with
over 600 Publications

**Just a sample of approximately
60 business and
financial services available.**

Push Information To the Limit.

In times of crises or relative calm, log on to News/Retrieval. Answers come quickly, along with data that might help you change your thinking and discover new solutions. It only takes seconds to check trends, market conditions, legal decisions and detailed corporate information. The heavy-duty, market-moving news you need.

60 Services in All...

To boost your financial position, keep abreast of stock and bond markets, options and mutual funds. Place buy or sell orders for stocks while online. And get automatic news alerts when the companies you're watching are making the news. Gain greater perspective from extracts of 10Ks and annual reports, earnings estimates and dividends records, and leading analysts' reports.

If you're preparing a presentation, you'll find full-text press releases online *minutes* after release. Then use key word searching to uncover additional data from hundreds of trade journals. And you don't have to be a trained searcher. You can use everyday language to retrieve the articles you need. You can even set up an electronic clipping service customized to your needs...to *find, pull and store* whatever information you designate from the Dow Jones newswires--*24 hours a day*.

To bolster corporate intelligence, get the scoop on insider trading activities, mergers and acquisitions, as well as forecasts on earnings per share and dividends. Analyze historical prices for a competitor's stock. Get critical information on sales and profits by product and service for all public companies. Or focus on the fiscal health of ones privately held.

When you need to spark sales and marketing ideas, check out industry, prospect and client information. Review "hot" news. Rethink the positioning of your products as you keep tabs on new product introductions in the field.

To expedite legal research, turn to us for fast-breaking news of antitrust actions, class-action suits, product liability and recalls. Monitor rumors as well as Justice Department and Supreme Court developments.

When it's time for a change of pace, reserve a flight. Shop at home. Send mail around the world. Read movie and book reviews--or check sports scores. We deliver the services you want for your business *and* leisure hours.

Take the Pulse of The Business World Quickly and Accurately.

With News/Retrieval, business and financial news is pulled directly from Dow Jones newswires and made retrievable by company, industry or category code. This is done *seconds* after it comes across the wires. These stories--many of them Dow Jones exclusives--are held in a searchable file for 90 days.

And News/Retrieval is the only online service that provides the *full text* of *back issues* of *The Wall Street Journal*, *Barron's* and stories from the Broad-tape. Other services just provide abstracts. Here you get the real story--every last word.

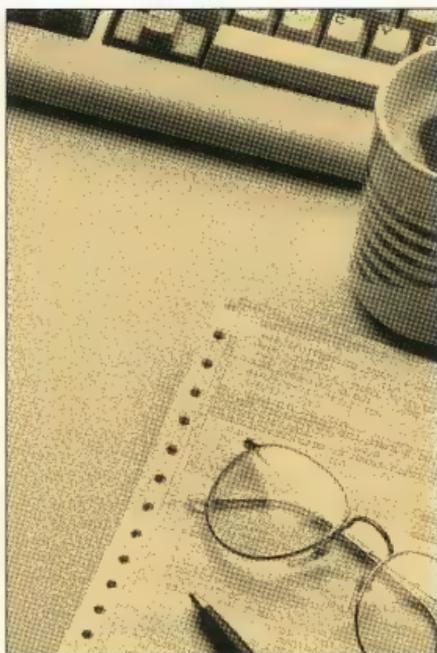
Transform your computer into the most authoritative source possible for everything from business facts and figures... to full-text articles on thousands of topics.

You'll find a little extra push goes a long way in business.

Subscribe today and get...

**One FREE hour online,
a waiver of our \$29.95
Start-up Fee, plus much,
much more.**

**DOW JONES
NEWS/RETRIEVAL®**



You'll Be Hard Pressed To Find a Better Deal...

Pay a low annual service fee and receive:

- ▶ One **FREE*** hour online (per account)
- ▶ Waiver of standard \$29.95 Start-up Fee
- ▶ A comprehensive *User's Guide*
- ▶ An annual subscription to *Dowline*[®], the magazine of Dow Jones News/Retrieval
- ▶ 20% off Dow Jones[®] Software purchased directly from Dow Jones

*Note: This is a one-time offer limited to new U.S. members only. Free time is per account and must be used within 30 days after receipt of password. Certain databases have charges in addition to usage fees, which are excluded from the free-time offer. The usage fees vary depending on which databases you select and the time of day they're used. Offer price is subject to change.

Control No.
K482901

**To Order or for More
Information Call:
1-800-522-3567, Ext. 279.**
New Jersey and overseas:
1-609-520-8349, Ext. 279.

Our Membership Line operators are waiting to take your order. Call Monday through Friday 9:00 a.m.--10:00 p.m. Eastern time. On Saturday: 9:00 a.m.--6:00 p.m. Eastern time.

If you're a personal investor, be sure to ask about our low monthly **After Hours/Flat-Fee Pricing Plan**. It'll bring you even greater predictability in pricing.

Before using the service, it is important that you read the Dow Jones News/Retrieval User Agreement, enclosed with the materials you'll receive as a new member. Use of the service signifies acceptance of the User Agreement's terms and conditions.

DOW JONES NEWS/RETRIEVAL[®]

The lifeblood of business.SM

Another service from Dow Jones Information Services.

P.O. Box 300 . Princeton, NJ . 08543-0300

©1991 Dow Jones & Company, Inc. All Rights Reserved.

Being Online Never Felt So Fine!

FREE Membership

FREE Software

Sign
On Now!



**“For the best value and the most fun,
DELPHI’s the place to be!”**

DELPHI

Join DELPHI for FREE!

Free to Explore

Discover a worldwide community on DELPHI. Download from thousands of programs. Join groups for nearly every type of computer and other special interests. Participate in discussions with other members in conference and message forums. Challenge your skills in lively multiplayer games.

Free to Choose

Choose from two low-cost membership options: \$6 per hour or \$1 per hour with the 20/20 Advantage Plan.*

Free Membership

Join DELPHI's Basic Plan for FREE. The \$5.95 monthly fee includes your first hour online. Additional time is only \$6 per hour.

Free Software

Get around online quickly and easily with the new DELPHI Online Software for PC Compatibles. Pull-down menus and a point and click interface put the world of DELPHI at your fingertips. Order your free disk today!

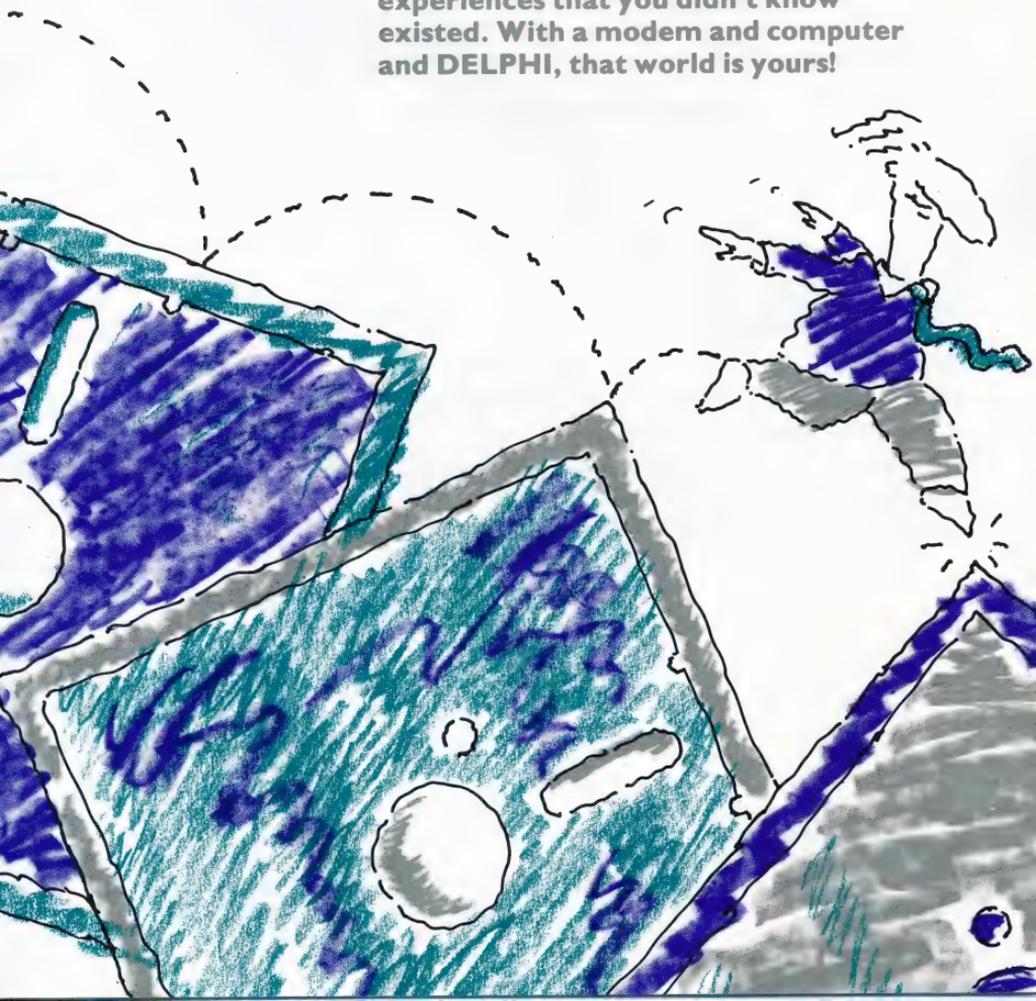
Free to Decide

With DELPHI, there's no risk. Take an hour to explore and if you're not satisfied, simply cancel your account and request a full refund of your first monthly fee.

*Free membership applies for DELPHI's Basic Plan. The 20/20 Advantage Plan carries an entry fee of \$39 which includes the first 20 hours online.

***DELPHI opens doors
for you, your family,
and your business.***

Now that you've explored what your computer can do on its own, you're ready for more. You've heard about the vast online world of information and services — instant telecommunications, across the continent or around the world. It's a rich and varied world that will put you in instant contact with thousands of other computer users, entertain you, and provide services and learning experiences that you didn't know existed. With a modem and computer and DELPHI, that world is yours!



Join a Global Network

DELPHI has members throughout the United States and in over 40 other countries. The people on DELPHI come from all walks of life, with varied interests, knowledge, and opinions. Groups for specific computer types, hobbies, and interests provide ideal meeting places for the interaction that makes DELPHI special.

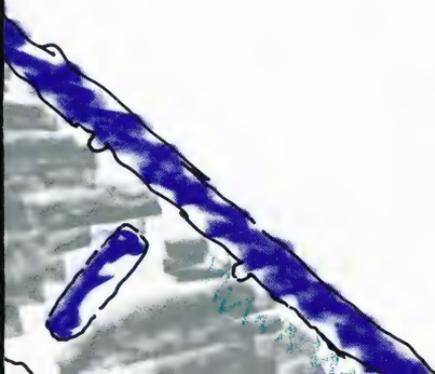


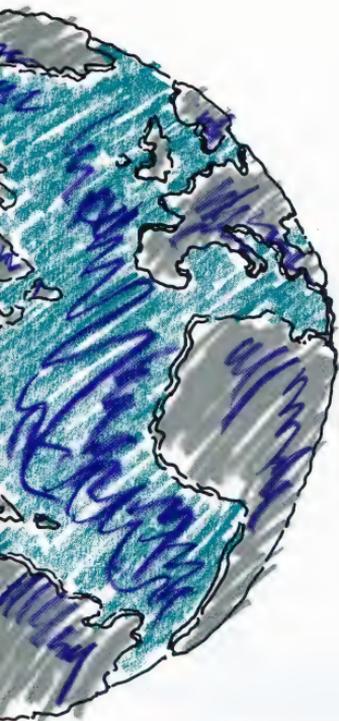
Get to Know Your Computer

Whether you're a novice or a veteran of computing, DELPHI's computer groups will be a valuable resource to you. Learn about hardware, software, and programming from other DELPHI members and from company representatives online to support their products.

Download Software

DELPHI's software libraries contain thousands of public domain and shareware files which you can download for use on your computer. Library managers check all submissions to assure high quality programs.





Create your Own Online Network

DELPHI now features "Custom Forums." Having one is like running your own personal network. As the forum host, you'll control who has access to your personal message area. Discussions can be limited to a few members or open to thousands. With special usage incentives and referral credits, you can even make money with a Custom Forum.

The Most Current News

News services from leading international newswires provide up-to-the-minute coverage of sports, politics, national and international events, human interest, and more. Read about it on DELPHI before the newspapers are even printed!

Money Matters

In-depth market analysis and personal portfolio tracking services help you to manage your finances and invest your money wisely. Access current stock and commodity quotes, financial news and business reports, and "inside" information on top companies and financial concerns.

Communication Made Easy

Whether you choose to exchange private electronic mail, post public messages in the Forum, or "talk" in real-time in Conference, interacting with other DELPHI members is both fun and easy. FAX, Telex, and translation services further enhance your communication options.



Carefree Travel

Don't wait in lines or hassle with travel agents when making your business or vacation plans. Find the lowest fares, the hottest attractions, and make all your reservations from the comfort of your home or office with DELPHI's extensive Travel services.

Fun & Games

DELPHI's multi-player games combine fast-paced competition with social exchange to create a real party atmosphere. Play Poker Showdown, TQ Trivia, Stellar Conquest, and other exciting games while "talking" with your opponents.

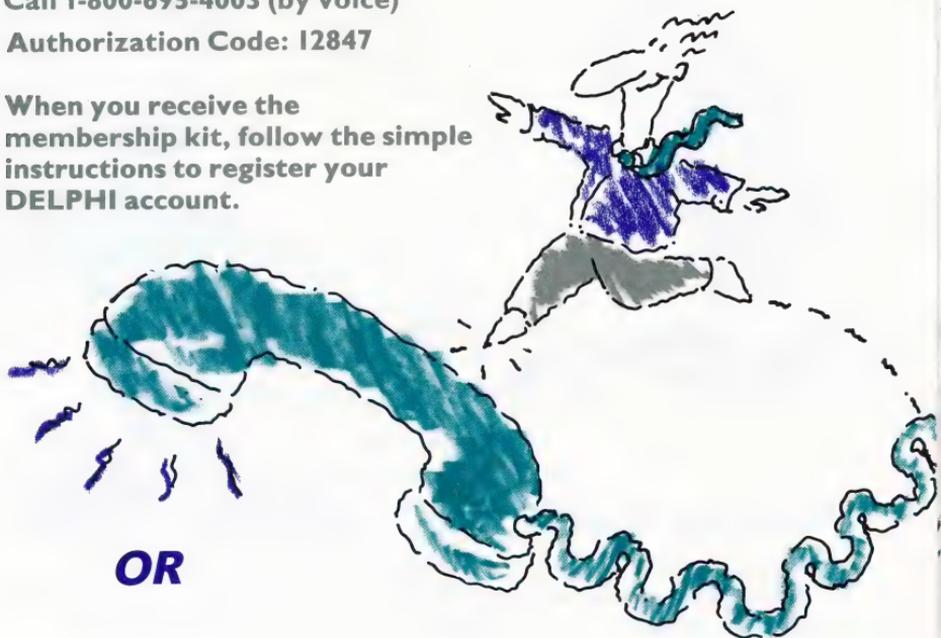
Join DELPHI Now!

PC Owners Free Software!

Order your free copy of the
DELPHI Online Software for
PC Compatibles:

Call 1-800-695-4003 (by voice)
Authorization Code: 12847

When you receive the
membership kit, follow the simple
instructions to register your
DELPHI account.



With Any Computer, Join Immediately Online

Via modem, dial 1-800-365-4636
At password, type PC43

DELPHI is a service of General Videotex Corporation,
1030 Massachusetts Avenue, Cambridge, MA 02138. 1-800-695-4005

Basic Membership Plan

**\$5.95 monthly fee includes
first hour of use each month**

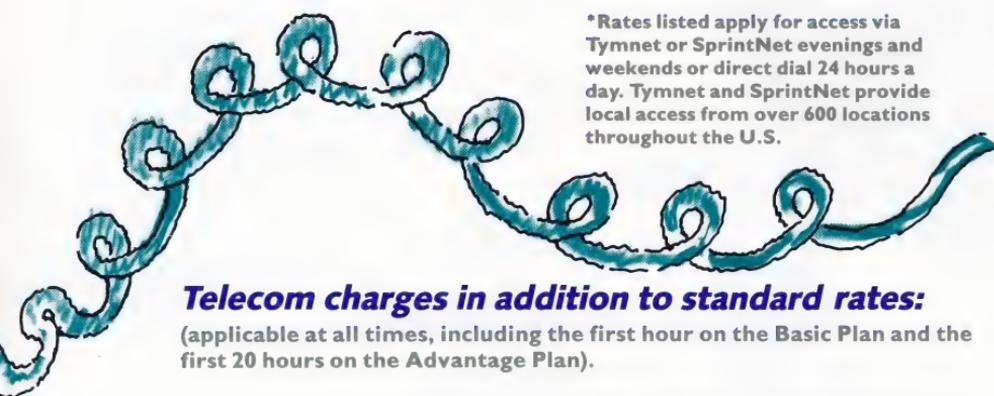
**\$6 per hour after the
first hour each month.***

20/20 Advantage Plan

20 Hours for \$20*

**\$1.20 per hour after the
first 20 hours each month.***

*Rates listed apply for access via Tymnet or SprintNet evenings and weekends or direct dial 24 hours a day. Tymnet and SprintNet provide local access from over 600 locations throughout the U.S.



Telecom charges in addition to standard rates:

(applicable at all times, including the first hour on the Basic Plan and the first 20 hours on the Advantage Plan).

	Home Time	Office Time
Mainland U.S.		
via Tymnet	no charge	\$ 9.00/hr
via SprintNet (Telenet)	no charge	\$ 9.00/hr
via PC Pursuit	no charge	no charge
Canada		
via Tymnet	\$ 1.80/hr	\$12.00/hr
via DataPac	\$25.80/hr	\$25.80/hr
Hawaii	\$ 5.40/hr	\$12.00/hr
Alaska, Puerto Rico	\$12.00/hr	\$12.00/hr
Prepaid International	\$ 1.80/hr	\$ 1.80/hr

Office Time is from 7 am to 7 pm weekdays. Home Time is 7 pm to 7 am weekdays, all day on weekends and some holidays. Use of each member name is limited to one person. Rates subject to change with notice posted online. Storage fees and other premium service charges also apply. See *Using DELPHI* online for more details.

Join DELPHI Now!

Free Membership—Free PC Software

To order your free copy of the DELPHI Online Software for PC Compatibles, call 1-800-695-4003. (by voice)

Authorization Code: 12847

See inside flap for complete sign-up instructions.

GESMnie

GE Mail • Dow Jones

News/Retrieval® • IBM®

PC RoundTable • POP-MED • Microsoft®

RoundTable • Grolier's Electronic Encyclopedia

• EAASY SABRE • Photography RoundTable •

Chat Lines • Charles Schwab Brokerage

Services® • Amiga RoundTable • Air Warrior

• QuikNews Clipping Service • WordPerfect®

RoundTable • CALC — The Computer Assisted

Learning Center • Apple® II RoundTable •

Gemstone III • The Official Airline

Guides Electronic Edition®

Travel Service • Atari®

RoundTable • Diplomacy® •

Medical RoundTable •

Investment ANALY\$TSM •

Newsbytes News Network •

**Online.
In budget.
A world of
fun and
information
from
\$
4.95
a month.**



**Start with
GENie*Basic
Services
at \$4.95
a month.**

That's it. Just \$4.95 a month gives you unlimited, non-prime-time usage of over a hundred GENie*Basic services almost anywhere.†

And when you're ready for more, you can take advantage of hundreds more GENie Value services for just \$6.00 an hour in non-prime time.† Or move up to GENie\$Professional, our premium business services.

When you're online with GENie, your computer becomes your ticket to learn, grow, compete and just have fun. And not only can we open the door to a sophisticated universe of information and great people — we've also made it affordable!

Here's just a sample of the GENie*Basic services you can enjoy for \$4.95 a month:†

Pursue your personal interests — professional or leisure. From Japan to Photography, Religion to Real Estate, GENie's Bulletin Boards let you leave questions, post messages, and share your tips and ideas on dozens of topics. (See next page for a sample listing.)

Research questions? Ask GENie. Grolier's Electronic Encyclopedia has a 10-million-word database that's revised quarterly — so it's never out of date. And it's just one of our quality research services.

Unlimited electronic mail. Another reason to choose GENie. Our GE Mail lets you send all the notes and letters you want to anyone on the GENie system during non-prime time.

Shop and save a bundle. There are over 30 stores to shop online in the GENie Mall — from JC Penney's online catalog to chocolates from Godiva. And Comp-u-store® OnLine†† offers discounts from 10% to 50% on over 250,000 brand-name items!

Money matters. GENie*Basic gives you the latest on stocks and finance. Get the day's stock and mutual fund closings, check the Investors' Bulletin Board, and pick up valuable information on personal finance.

Today's news, weather and sports. Stay up to date on the top daily stories, current weather conditions anywhere in the world, and how your favorite team did today.

Just the ticket for traveling. EAASY SABRE from American Airlines lets you examine schedules and find the lowest fares for more than 650 airlines worldwide. Reserve hotel rooms and rental cars, too. Check the Bulletin Boards for the news from Florida, and tips on travel.

Games and entertainment. Battle the Black Dragon, try your hand at Adventure 550, Dor Sageth, Castle Quest, and more. GENie also keeps you current with movie reviews and soap opera summaries, plus the latest jokes.

Aladdin Support. Get online help with Aladdin, GENie's own communications package, which provides IBM-compatible PC owners and Atari ST owners with an easy way to access GENie's services. (Front ends for Mac and Amiga computers are also on the way.)

That's just the beginning of GENie*Basic.

There are over a hundred services in all, including these Bulletin Board topics:

Science Fiction • Current Affairs • Radio/Electronics • Military • Genealogy • Writers • Medical • Home Office/Small Business • Education • disABILITIES • Automotive • Air and Space Technology.

For every Bulletin Board in GENie*Basic there is also a RoundTable available in our GENie Value services, with software libraries and real-time conferences.

† U.S. prices. There are a few areas with a remote access charge. See page 7 of this brochure for more details.

†† Requires a membership with a nominal fee.

Ready for more? GENie Value services at \$6/hour expand your universe.

Get the most out of your PC. Download more than 100,000 files from our software libraries. Discuss your favorite topics with the experts in "real time." And discover the thrill of competing live against other GENie users — with our multi-player games.

The list of GENie Value services goes on and on — and you can stay on and on, too. Because the rate for non-prime-time usage is just \$6 an hour — for 300, 1200 or 2400 baud. (That's *much* less than comparable services charge.)

Valuable computer support. Our Computer RoundTables let you talk with the experts at leading companies, and get in-depth answers and updates quickly — so you save time, money and frustration. We also offer dozens of RoundTables for PCs and software. (See a sample on next page.)

Plus, over 100,000 files. GENie Value means easy access to tremendous libraries of shareware and public domain software. (At least three times as many as other leading online services offer.) And every file can be downloaded at little cost.

Go back to school — at home. Our Computer Assisted Learning Center (CALC) has a wide variety of courses for fun or college credits[†] — along with a high school tutoring and homework help center.

The tools to manage your investments. GENie Quotes covers over 67,000 security issues of all types. Create your own portfolio to keep track of your most important investments.

Get the news, now. GENie gets you the news hours before other online services. With NewsGrid®, you get hourly compilations from the largest global wire services. For the latest on the computer newsfront, call up the Newsbytes News Network, with daily reports from reporters around the world.

[†] Requires the payment of additional charges.

Plan your getaway. Find out about ticket discounts, packages, vacation spots, and even free travel with the Traveler's Information RoundTable. Match trips to your personal criteria with Adventure Atlas.

Great games — great players, too. Your opponents are other GENie users, ready to engage you in a dogfight using vintage aircraft with Air Warrior. Or create your character and choose your weapon in Gemstone III: The Shadow World. Try your hand at poker, chess, backgammon and more. GENie offers more multi-player graphic games for more kinds of computers than any other online service.

Chat Lines and conferences. Let's talk about it! You can converse, real time, with experts and enthusiasts — on dozens of topics. Discuss interests from medicine to the military, from genealogy to science fiction. Or simply be social — choose a handle for yourself and converse on our Chat Lines! Whether it's a party line, a discussion group, or a private one-on-one — GENie friends are always ready to exchange views and information.

And there's more. For every Bulletin Board in GENie*Basic, there's a corresponding RoundTable in GENie Value. In addition, here are just a few of our GENie Value software and computer RoundTables: IBM PC • Apple II • Macintosh • Starship Amiga • Microsoft • Borland • Flagship Commodore • Atari ST • Tandy • TSR • Unix • Freesoft.

**GENie\$-
Professional
services.
Now, your
business
is online.**

With GENie\$Professional services, you get the information you need for investments and business — and you get it *now*.†

Everything you need to analyze and trade securities.

The Investment ANALY\$TSM offers current and historical quotes, analytical reports and stock screening and selection. VESTOR offers a sophisticated analysis of current market conditions, complete with buy/sell signals. With Charles Schwab Brokerage Services,†† you can trade, change, cancel, and view the status of your stock, options, mutual fund and bond orders — and save on commissions!

GENie\$Professional also gives you access to Dow Jones News/Retrieval®. Search The Wall Street Journal, Barron's, and five Dow Jones newswires — along with 500 trade publications, newspapers and magazines. Current and historical statistics and information are also online, with everything from quotes and indicators to analysts' reports and SEC filings.

If your investments depend on up-to-the-minute information, you've got it. Get real-time quotes, with Charles Schwab Brokerage Services. Check Dow Jones News/Retrieval for current quotes and exclusive coverage of trading activity (15-minute delay) for over 5,000 stocks.

Want to send information? Fax it on GENie. GENie\$Professional users can send messages to facsimile machines anywhere in the world via GE Mail to Fax. You can communicate with virtually any standard fax machine, regardless of whether the recipient is a GENie user.

If you need to be there, we'll help — with The Official Airline Guides Electronic Edition® Travel Service. It's your private travel agency. Call up and book airline schedules, fares and seats. Access more than 42,000 first-class and deluxe hotels. Check essential destination details from climate to visa requirements. And shop from over 90,000 listings of vacation adventures, tours and cruises.

†GENie\$Professional services are premium online services, and carry additional charges.

††Not available in Canada.

And for health matters, there's POP-MED.

Telebase's POP-MED provides useful summaries of thousands of health-related articles — geared both to medical professionals and consumers — from hundreds of magazines, newspapers and journals. Easy menus and simple key word searches let you quickly search POP-MED to review headings, display abstracts or order reprints of the full text.

GENie gives you everything you're looking for — and the time to enjoy it.

GENie's a remarkable value — offering so much more than “shopping services” and offering a much better price than comparably sophisticated online services.

What's more, there's also a money-back guarantee with GENie*Basic services! If you're not completely satisfied after using GENie*Basic during the first month, we'll refund your initial \$4.95 subscription fee.

So why wait? To sign up, simply follow the instructions on the back of this brochure.

GENie Services Pricing

	U.S. (U.S. \$)		Canada (Canadian \$)	
Monthly Subscription Fee	\$4.95/month		\$5.95/month	
Rates	Non-Prime	Prime	Non-Prime	Prime
GENie*Basic Services	No Hourly Charge ^{†††}	\$18/hour	No Hourly Charge ^{†††}	\$25/hour
GENie Value Services	\$6/hour	\$18/hour	\$8/hour	\$25/hour
GENie\$-Professional Services	Prices vary per individual service. These include Charles Schwab Brokerage Services (not available in Canada), Dow Jones News/Retrieval, Official Airline Guides, QuikNews, etc.			

^{†††}Prices listed as of October 1, 1990. Subscription includes unlimited access during non-prime time to select products at up to 2400 baud, and GENie LiveWire magazine. Non-prime time: 6:00 PM to 8:00 AM local time, Monday through Friday, and all day Saturday, Sunday and selected holidays. Prime-time hours: 8:00 AM to 6:00 PM local time. Some service features may be subject to a surcharge. In addition, a state tax may apply to GENie services. In a few areas, there is a \$2/hour remote access surcharge. Where applicable, this charge applies to all services, including the use of GENie*Basic services in non-prime time.

Try us on for size — it's risk-free!

GENie couldn't be easier to try. No sign-up fee. The monthly subscription fee of just \$4.95 gives you unlimited non-prime-time usage of GENie*Basic services. And there's no risk! In fact, we're so sure you'll be satisfied, we're offering a money-back guarantee.

If you're not completely happy with GENie*Basic services after your first 30 days, we'll refund your initial \$4.95 subscription fee.

Think of it! One month to enjoy the information, education, entertainment and savings GENie can offer, with a satisfaction guarantee. It's like test-driving a new car, free, for a month! And the experience of being online is the only way to discover how fun, helpful and valuable GENie is.

You can start right away — all you need is a PC, a modem and communications software.

To sign up, just follow these simple steps:

- 1.** Set your communications software for half duplex (local echo), at 300, 1200 or 2400 baud.
- 2.** Dial toll free: 1-800-638-8369 (or in Canada, 1-800-387-8330). Upon connection, enter HHH
- 3.** At the U#=prompt, enter XTX99437,GENIE then press <RETURN>.
- 4.** Have a major credit card ready. In the U.S., you may also use your checking account number.

For more information in the United States or Canada, call 1-800-638-9636 or write: GENie, c/o GE Information Services, P.O. Box 6403, Rockville, MD 20850-1785.

For information about GENie services in Japan, Germany, Austria, Switzerland and other countries, please write GENie at the above address. GENie*Basic is not available in all countries. Some services may not be available outside the U.S. In addition, some third-party services may be subject to specific country restrictions or carry additional charges.



U.S.A.

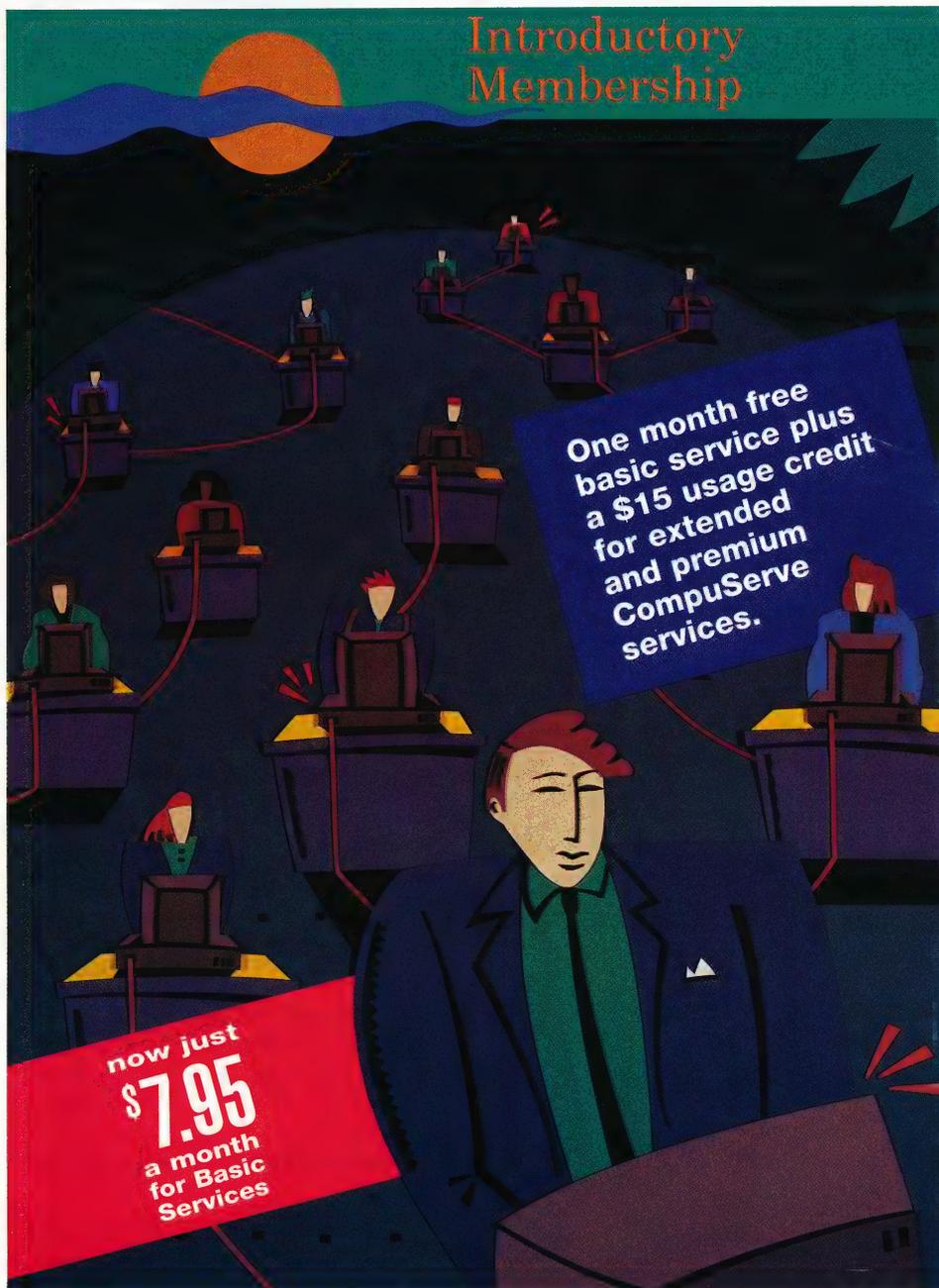
We bring good things to life.

CompuServe®

Introductory
Membership

One month free
basic service plus
a \$15 usage credit
for extended
and premium
CompuServe
services.

now just
\$7.95
a month
for Basic
Services



Welcome To CompuServe Information Service

If you have a personal computer, modem, communications software and telephone, you have everything you need to get connected to the world's largest personal information and communication service. This guidebook provides you with step-by-step instructions and some helpful ways to make the best use of CompuServe. Please keep it as a useful reference.

If you have any questions about setting up your computer equipment to access CompuServe, or about signing on to our service, our Customer Service Department offers immediate help, day or night. Call: **800-848-8990** toll-free in the United States. (Customer Service telephone numbers for offices outside the United States are listed on the back cover of this brochure.)
U.S. hours: Monday-Friday: 8a.m. (0800) - midnight Eastern time
Weekends: Noon-10p.m. (2200) Eastern time
Holiday hours vary.

Please be sure to read Section 2, Getting Online and Becoming a Member, before logging on for the first time.

Section 1. What CompuServe Can Do For You

FREE CompuServe basic services for one month
CompuServe services
CompuServe makes it easy to ...

Section 2. Getting Online And Becoming A Member

Your first time on CompuServe
Becoming a member
Accessing the information service

Section 3. Online With CompuServe Information Service

Simple commands
Free Membership support services

Section 4. For Your Information

An index to online destinations
Troubleshooting
Rates
Service Agreement Terms

FREE CompuServe basic services for one month

This Introductory Membership to CompuServe is your opportunity to connect with a wide range of valuable CompuServe resources. Resources that can help you advance your career, enrich your personal time, and enhance your knowledge. We know you'll enjoy the many benefits of CompuServe and we sincerely hope you'll utilize our rich information and communication resource as often as you can.

Once you sign up, you're ready to enjoy one month of CompuServe basic services absolutely free of connect time charges and with no obligation to continue¹. You'll discover an exciting world of up-to-the-minute news, weather, sports, reference materials, games, current stock market quotes, shopping services, movie reviews, consumer information, and online communications. Additionally, you'll be entitled to send about 60 three-page messages each month by CompuServe Mail. As a CompuServe member, you'll also receive a free subscription to our monthly *CompuServe Magazine* which introduces new products and services and teaches you how to maximize your online productivity.

\$15 usage credit for extended and premium services.

There are over 1700 optional CompuServe services available to you on a pay-as-you-go basis. These extended and premium services include special interest forums, multi-player games, and advanced information resources that enhance the value of our basic package. Your Introductory Membership to CompuServe includes a \$15 usage credit that enables you to explore these services at no charge until the credit is fully used. For additional information, refer to Page 5 in this booklet. We urge you to take advantage of your \$15 usage credit. Discover how CompuServe's extended and premium services increase the value and excitement of your personal computer.

¹ For information on our Alternative Pricing Plan (pay-as-you-go), see page 28.

Remember, you're under no obligation to remain a CompuServe member once you've enjoyed one month of CompuServe basic services. Cancel at any time through Feedback (see Page 20) or, by sending a letter to CompuServe Incorporated, Account Cancellation, P.O. Box 20212, Columbus, OH 43220, U.S.A.

CompuServe services

When you become a member of CompuServe, you have immediate access to an information and communication resource that can enrich your life.

Explore our services and see how your personal computer can become a companion, ready to help you do everything from buying the most economical airline ticket to learning about advancements in quantum physics. To access these services, refer to the Index To Online Destinations on Page 21 of this booklet.

CompuServe basic services membership is \$7.95 per month. These services include unlimited connect time to access:

News, Sports, Weather

Associated Press Online
Hourly News Summaries,
Sports, Entertainment, Business
News, Science & Health
Accu-Weather® Maps and Reports
National Weather Service Reports
UK News, Sports

Reference Library

Grolier's Academic
American Encyclopedia
A 21-volume online
encyclopedia updated quarterly
Peterson's College Database
HealthNet
An online medical reference source

Shopping

The Electronic Mall®
Shopper's Advantage®
A Discount Shopping Club
Consumer Reports
Classified Ads (Read Only)

Financial Information

Current Stock Quotes
Issue/Symbol Reference
Mortgage Calculator

Entertainment & Games

Roger Ebert's Movie Reviews
Science Trivia Quiz
The Grolier Whiz Quiz
ShowBizQuiz
CastleQuest
Black Dragon
Classic Adventure
Enhanced Adventure
Hangman

Electronic Mail

CompuServe Mail
(see Page 5 for details)

Travel and Leisure

EAASY SABRE® and WORLDSPAN
TravelshopperSM Airline, hotel and
rental car information and
reservations
Department of State Advisories
Visa Advisors

Membership Support Services

DOS CIM Support Forum
Mac CIM Support Forum
Navigator Support Forum
Practice Forum
A forum designed to teach you the
fundamentals of real-time online
communication
Directory of CompuServe Members
Ask Customer Service
A helpful resource if you need
online assistance
Free Access to Many Menus

More than 1700 CompuServe extended and premium services provide a direct connection to a world of advanced information and entertainment. Software support forums are an excellent source of shareware and freeware. Special interest forums let you meet and share ideas with people throughout the world; multi-player games take you into a new world of adventure; and valuable information resources help you in your professional and personal life. A fee based on actual time online gives you immediate access to these services, which are available to every member. Your introduction to CompuServe includes a \$15 usage credit, so you can leisurely explore these valuable resources, most of which are available at \$12.80/hour (2400 baud).

A “+” on the online menu indicates you will enter one of the more than 1700 optional CompuServe extended or premium services if you select that choice. Some of these services may instead be marked with a “\$” which indicates additional premium surcharges will apply. To access any extended service, or to determine whether a specific service is included in your basic package, refer to the Index To Online Destinations on Page 21 of this booklet. For Rates, see Page 28.

CompuServe makes it easy to...

S **stay in touch.** CompuServe turns your computer into a communications center. Each month, you're entitled to send about 60 three-page messages with no additional charge. After that, there's only a nominal charge. How does it work? Simple! Just type your message and direct it to any CompuServe member or to any Internet, AT&T Mail, AT&T Easylink or registered MHS address worldwide. Or, for a small additional charge, your message can be transmitted to any MCI Mail®, Telex®, or fax address, anywhere on the globe. So stay in touch the quick and easy way. CompuServe Mail puts the world at your command. Your basic membership includes CompuServe Mail, although some specific functions have additional charges. See Page 29 for details.



See the world. Before you take your next trip, save money and travel time by planning it on CompuServe. CompuServe helps you search airline reservation systems for the lowest fares available on all major domestic and international airlines. You can even check travel times, flight restrictions and all the information you need about connecting flights. Book reservations online and get your tickets by mail, at the airport, or from your travel agent. Wherever you venture, take full control of your travel plans with CompuServe. Your basic membership includes WORLDSPAN TravelshopperSM, EAASY SABRE[®], Department of State

Advisories and Visa Advisors. All other travel services are extended or premium services.

Check the weather report. Is it raining in Seattle? Is Atlanta fogged in? Know what's happening all over the world with CompuServe's weather reports. The National Weather Service provides dependable information and forecasts for travellers. And our Accu-Weather[®] maps give you a clear picture of weather conditions — from Cape Mendocino to Cape Cod... and everywhere in between. With CompuServe, the forecast is always great! Your basic membership includes Accu-Weather maps and reports and National Weather Service reports. All other weather services are extended or premium services.

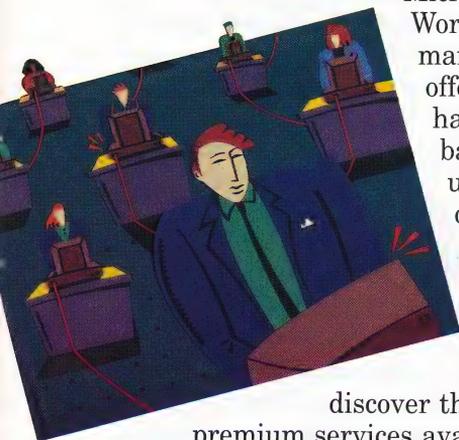
Shop in comfort. When you don't have time to shop, or you'd rather not battle the crowds at the local shopping center, CompuServe's Electronic Mall[®] is open 24 hours a day. Over 100 retail stores, specialty shops, and discount wholesalers are online and ready to take your order and deliver it to your doorstep in no time at all. Nissan, Brooks Brothers, Micro-Warehouse, Hammacher Schlemmer, and Waldenbooks are represented, with new retailers joining our services every month. When you join Shopper's Advantage[®] Club, you can save up to 50% on more than 250,000 brand name products.

Before you buy, you might want to look into our online *Consumer Reports*. From automobiles to appliances, you'll get all the facts you need to make an intelligent purchase decision.

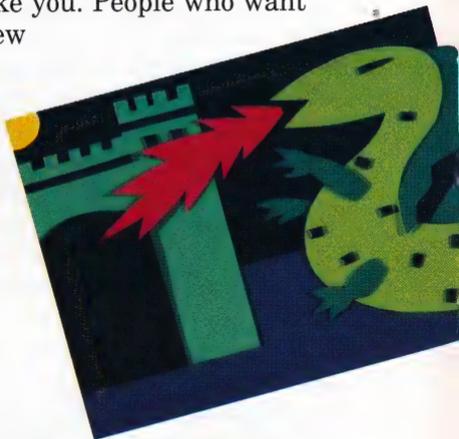
For that hard-to-find item, classic collectible, or if you'd just like to hunt for a bargain, check CompuServe's Classified Ads. Or, if you'd like to sell something, there's only a nominal charge for posting an ad. It's a wonderful service available to CompuServe members worldwide. Whatever you're looking to buy — or sell — it's faster and easier with CompuServe. Your basic membership includes The Electronic Mall®, Shopper's Advantage, *Consumer Reports*, and the reading of Classified Ads. All other shopping services are extended or premium services.

Get support. No matter what kind of computer software or hardware you use, there's a CompuServe personal computing forum for you in our optional extended services. IBM®, Macintosh®, Tandy®, Atari®, Commodore® and Amiga® are just some of the computer systems supported by our service. In total, we provide online support for hardware and software from over 300 companies. If you have a problem with your hardware or software, simply post a message in the appropriate forum. Chances are, someone online has experienced and solved the same problem, or knows someone who has. Within minutes, you may have several helpful responses. Also, you can obtain free software and utility programs through these support forums. Technical representatives for ACIUS, Adobe®, Aldus®, Borland, Lotus®, Microsoft®, Novell, Symantec, WordPerfect®, Central Point, Claris, and many others are online and ready to offer assistance whatever your hardware or software problem. Your basic service membership includes unlimited access to online support for our CompuServe software and to our Practice Forum. Additionally, your basic membership allows you to browse most menus as much as you like at no additional cost.

Learn your way around the system, discover the wide variety of extended and premium services available to you. You can spend all the time you want just getting to know more about CompuServe. All other support forums are extended services.



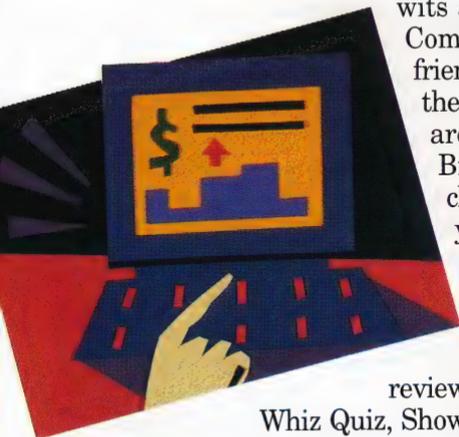
Join a group. CompuServe's special interest forums are dedicated to the interests of people like you. People who want to expand their horizons, explore a new world of possibilities. You'll get to know people with similar occupations, hobbies and interests; interact with experts on topics ranging from literature to laptops. The message board is usually the most active place in a forum, where members stop by to catch up on the latest news and contribute to current discussions. You can talk to others in forum conferences, where members address hot topics and occasionally interview visiting luminaries. Forum libraries include valuable information on everything from software programs to wine lists, depending upon the nature of the group. Special interest forums are extended services.



Slay the beast. Enter a mystical world of wizards, warriors and wonderment. Vanquish a villain in a cold, clammy catacomb. Zap an enemy in a galaxy light years away. CompuServe's games are a thrilling diversion — whether you're attempting a one-player game like the Grolier Whiz Quiz, or a multi-player, multi-level challenge. CompuServe role-playing games are more real, more intriguing...more treacherous than any you've tried before. And much of the fun comes in knowing you're matching wits against other adventurous

CompuServe members. Some are your friends...some unrelenting foes. Among the more popular role-playing games are The Island of Kesmai™ and British LegendsSM. Each one offers a challenging new adventure every time you enter our world of fantasy and fun! And speaking of fun, movie lovers will appreciate Roger Ebert's Movie Reviews. Your basic membership includes Roger Ebert's

reviews, Science Trivia Quiz, The Grolier Whiz Quiz, ShowBizQuiz, CastleQuest, BlackDragon, Enhanced Adventure, Classic Adventure and Hangman. All other games and entertainment services are extended services.



erest forums are
ou. People who want



everything from software
e nature of the group.
es.

d of wizards, warriors
old, clammy catacomb.
ay. CompuServe's
er you're attempting a
quiz, or a multi-player,
playing games are more
than any you've tried
owing you're matching
adventurous
nbers. Some are your
relenting foes. Among
r role-playing games
of Kesmai™ and
sSM. Each one offers a
w adventure every time
world of fantasy and
aking of fun, movie
ppreciate Roger Ebert's
ews. Your basic
p includes Roger Ebert's
ivia Quiz, The Grolier
tleQuest, BlackDragon,
and Hangman. All other
ended services.

Invest wisely. Take control of your investments by using the same electronic tools and information that experts depend upon daily. Our personal investor services help you analyze yesterday's performance, follow today's prices, and tap tomorrow's predictions on thousands of publicly traded securities. You can screen your investment options based on your own criteria. Consult respected market resources, like Standard & Poor's, Value Line and Disclosure. Access up to 12 years of daily, weekly and monthly pricing histories. You can even place electronic buy and sell orders with online discount brokers, 24 hours a day. Let CompuServe's financial services put you in the heart of the financial district. Your basic membership includes Current Stock Quotes, Issue/Symbol Reference and Mortgage Calculator. All other financial services are extended or premium services.

Know what in the world is happening. Turn to

CompuServe daily for news from the Associated Press, The Washington Post®, Reuters, OTC News Alert and United Press International. Get immediate updates as stories break around the world, throughout the day. And be the first to know when acquisitions are approved and mergers are made. CompuServe helps you follow professional developments, back your business decisions with facts, or research contemporary social issues without even leaving your home. And, if you're a sports fan, we really know the score! CompuServe's sports news covers sports around the globe. Your basic membership includes Associated Press Online. All other news services are extended or premium services.



Do your homework. CompuServe helps you find facts fast by connecting your personal computer to the reference libraries of science, medicine, law, literature, and dozens of other disciplines. Access professional journals, consumer magazines, newspapers, specialized newsletters, and published research papers. It's easy to search our databases for available articles and abstracts and to review them online. Use our demographic data to select business sites, analyze local competition and to plan promotional campaigns. Your basic membership includes

Grolier's Academic American Encyclopedia, Peterson's College Database and HealthNet. All other reference services are extended or premium services.

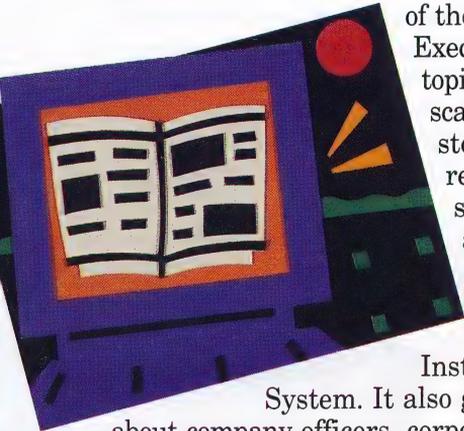
Get all the facts with the Executive Service Option.

CompuServe's Executive Service Option entitles you to special privileges and discounts and gives you access to many additional extended services which are valuable sources of information. One of the most popular features is the Executive News Service. Just specify a topic and our electronic clipping service scans major newswires, clips related stories and files them online for later reading. Our business decision services unlock powerful investment and market research tools, including: Company AnalyzerSM, company screening, return analysis, Disclosure II[®], and the Institutional Brokers' Estimate

System. It also gives you hard-to-find information about company officers, corporate expenditures and dividends for over 10,000 firms. When you're preparing a presentation, the Executive Service Option's SUPERSITE enables you to retrieve superior quality demographic reports for any ZIP code in the nation.

Among the special privileges and discounts you receive with this service are a 25% discount on selected transaction-priced financial databases, a 10% discount on the purchase of most CompuServe products, a 50% increase in your personal file area, and a six-month storage period for personal files — twice the standard amount.

The Executive Service Option is available to you for a monthly minimum charge of \$10. Your monthly \$7.95 CompuServe basic membership is applied to this minimum, making the Executive Service Option an excellent value. To enroll in this service, select it during the sign-up procedure or GO MEMBER. Remember, when you need to know it all, you need CompuServe's Executive Service Option.



Your first time on CompuServe

Some computer users worry that getting online with CompuServe is a long and difficult process. Nothing could be further from the truth. These step-by-step instructions should make it all go smoothly. But, if it doesn't, don't worry. Just call Customer Service toll free at 800-848-8990. For support outside the United States, refer to the back cover of this brochure.

1. Connect your equipment

Please refer to the manufacturer's documentation for installation and operating instructions. To connect to CompuServe, you must have a computer, modem, communications software and telephone line.

Load your communications software. Verify that your communications software is communicating with your modem.

2. Configure your communications software

Most communications software allows you to set certain parameters that influence the software's operation. You can set more parameters according to your personal preferences, but refer to your communication software documentation for advice on how to set the following parameters:

Baud Rate This parameter determines the speed of communication between CompuServe and your computer. CompuServe supports 300, 1200, and 2400 baud, plus 9600 baud in most major cities. The baud rate you select must be supported by both your modem and the CompuServe access number you will be dialing. For the number in your area, refer to Page 12.

Data Transfer and Parity Your communications software must be set to use seven data bits, one stop bit, and EVEN parity (often written "7-1-E"). If you choose not to include a parity bit, set your software to 8 bit, no parity.

Echo If your communications software includes an ECHO parameter, set it to OFF.

Full Duplex If your communications software includes a DUPLEX parameter, set it to FULL DUPLEX.

3. Select your CompuServe network access numbers

The following is a list of major cities in the US and Canada, and their CompuServe local access numbers. Once online, enter GO PHONES at any ! prompt for a comprehensive listing of numbers including 9600 baud. Consult your local telephone company for any applicable toll charges. There are no communications surcharges when using the CompuServe network in the US or Canada. For convenience, record your local access number on Page 17.

<i>300/1200 baud rate</i>	<i>2400 baud rate</i>	<i>300/1200 baud rate</i>	<i>2400 baud rate</i>
Anaheim, CA (714) 520-9724	(714) 520-5231	Montreal, P.Q. (514) 374-8961	(514) 374-5340
Atlanta, GA (404) 266-7014	(404) 266-7060	New Orleans, LA (504) 734-8150	(504) 733-2297
Baltimore, MD (301) 832-0100	(301) 832-0160	New York, NY (212) 422-8820	(212) 888-1020
Boston, MA (617) 542-1796	(617) 482-7061	Newark, NJ (201) 643-0404	(201) 643-6256
Chicago, IL (312) 693-0330	(312) 263-5636	Philadelphia, PA (215) 977-9758	(215) 977-9794
Cincinnati, OH (513) 771-1630	(513) 771-8543	Pittsburgh, PA (412) 391-7732	(412) 261-4192
Cleveland, OH (216) 781-5720	(216) 781-4135	Portland, OR (503) 232-1072	(503) 239-6124
Dallas, TX (214) 953-1971	(214) 953-0436	Sacramento, CA (916) 971-4681	(916) 482-1094
Denver, CO (303) 629-5563	(303) 629-9145	San Diego, CA (619) 283-6091	(619) 280-1896
Detroit, MI (313) 535-1400	(313) 535-1122	San Francisco, CA (415) 956-4191	(415) 398-0905
Houston, TX (713) 462-0202	(713) 462-0923	San Jose, CA (408) 988-8762	(408) 988-5366
Kansas City, MO (816) 474-3770	(816) 472-1283	Seattle, WA (206) 241-9111	(206) 242-5767
Los Angeles, CA (213) 739-8906	(213) 383-9284	St. Louis, MO (314) 241-3102	(314) 241-5337
Memphis, TN (901) 452-1710	(901) 323-0220	Toronto, Ont. (416) 752-4150	(416) 265-8035
Miami, FL (305) 266-0231	(305) 262-1643	Vancouver, B.C. (604) 738-5157	(604) 737-2452
Minneapolis, MN (612) 342-2207	(612) 339-2507	Washington, DC (202) 388-4280	(202) 388-3303

If you do not live in one of these areas, you have two options:

- Configure your software as detailed on Page 11. Dial: 1-800-346-3247 (1-800-FINDCIS) through your modem to access a database that lets you look up your local access number. When you receive a CONNECT

CONNECT
Host Name: PHONES

message, press the 'Enter' key (or a carriage return, ) to receive a HOST NAME: prompt. Refer to your communications software manual for keyboard equivalents. Type PHONES and press a  again. You'll be connected to a menu driven database to help you find the access number nearest you.

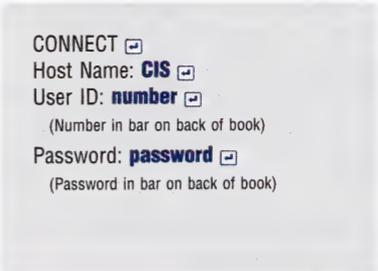
- Call CompuServe direct at 1-800-635-6225 (1-800-63LOCAL) from the United States and Canada. Customer Service can inform you of your local access number, 24 hours a day.

4. Access CompuServe

Before you attempt to go online with CompuServe, fill out the billing information worksheet on Page 15. Then, refer to it when prompted for the information online. To connect to our service, choose your local access number, and initiate the dialing sequence according to your modem software instructions. Once your local access number has been dialed, you will hear a high-pitched tone. For most computer systems, a "Connect" message will appear on the screen.

5. Signal CompuServe that you are ready

When you are successfully connected to CompuServe, a CONNECT message will appear on your screen. You can now signal CompuServe that you are ready.



```
CONNECT   
Host Name: CIS   
User ID: number   
      (Number in bar on back of book)  
Password: password   
      (Password in bar on back of book)
```

Press the "Enter" key (or a carriage return, ) . You will then be prompted HOST NAME: Type CIS and a carriage return. (If you mistype the host name, you'll see a UIC prompt. Hold down the Control key while you press the letter C, and HOST NAME: will appear again. Then, type CIS and a ) .

6. Enter your temporary User ID Number

At the User ID: prompt, enter the temporary User ID Number in the bar on the back cover of this book. Then, press a . This is the only time you'll use this User ID Number. In most cases when you complete the online registration process, a unique, permanent User ID Number appears on your screen. Write it down on Page 17. You will use it in the future, each time you access CompuServe.

7. Enter your temporary Password

At the Password: prompt, enter the temporary Password you'll also find on the back cover of this book. Then, press a . For security reasons, the Password will not appear on the screen as you type it. If you enter the Password incorrectly, you will be given two more opportunities to enter it correctly. You will only use this Password for your initial sign-up. A new temporary password will usually be issued to you at the end of online registration. Ten days later, a permanent password will be mailed to the address given during registration.

Becoming a member

There are no connect-time charges or communications surcharges during the online registration. Your complimentary usage credit for use in the extended services doesn't begin until you log on with your permanent, personal User ID Number provided at the end of the sign-on procedure. Entering EXIT at any point will discontinue the process. You may restart registration from the beginning at any time.

1. Enter your Agreement Number and Serial Number

When prompted, enter the Agreement Number and Serial Number printed in the bar on the back cover of this book.

AGREEMENT NUMBER: **number**

(Agreement Number on back of book)

SERIAL NUMBER: **number**

(Serial Number on back of book)

2. Select your desired billing method

You can choose to be billed by charge card, direct debit to your checking account, or to a business account. Your first one month of CompuServe basic services is free. You will be billed in your second month of membership or as soon as you exceed your \$15 usage credit for extended and premium services.

3. Agree to the Service Agreement Terms

The Service Agreement Terms are printed on Page 31 of this booklet. Confirm your acceptance by entering AGREE at the prompt provided online. Enter a carriage return.

4. Select Options you desire to receive

This Introductory Membership offers you one free month of CompuServe basic services. If you would like to access the extended services available through the Executive Service Option, please indicate so when prompted. There is a monthly minimum of \$10 for the Executive Service Option. Your \$7.95 basic membership is applied to this minimum, making the Executive Service Option an excellent value. The Executive Service Option \$10 monthly minimum is also waived for the first one month of membership. You may remove your Executive Service option at any time by entering GO MEMBER at the ! prompt.

5. Provide name and address information

Provide your full name, address, and telephone numbers when prompted to do so. If you prefer, this information will not be made available to others.

6. Provide billing information

Before you sign on, select a billing option and fill out the information below. Then refer to it when prompted for billing information during registration.

Option 1. Charge Card

Any CompuServe charges can be billed to your VISA, MasterCard, or American Express/Optima. If charges are to be made to a charge card, it is important that the name and address entered during registration correspond exactly to the credit card used. You must also include your card number and expiration date. Each week, charges for the previous week are forwarded to your card issuer. Each month, your card statement lists your CompuServe charges. Please note international cards must be used by members outside the United States.

Billing Information Worksheet

<input type="checkbox"/> VISA	Account #: _____
<input type="checkbox"/> MasterCard	Expiration Date: _____ / _____
<input type="checkbox"/> American Express/ Optima	Bank Name: _____ <small>(VISA and MasterCard only)</small>

Option 2. Direct Debit

Direct debit is a convenient way to manage your account without ever writing a check. If you have an address and a checking account within the United States, this payment method enables you to make payments from your checking account through the Federal Reserve System. Enter your checking account number and bank's name, address and routing transit number during the online registration. You can review your direct debit online by accessing your current charges (GO BILLING). Each month, the bank pays CompuServe on your behalf, and lists that payment on your checking account statement. It may be necessary for CompuServe to call you within 24 hours of your registration to verify your bank account information.

The diagram shows a check with several callouts pointing to specific fields:

- Name on Checking Account:** Points to the top left of the check, containing "MR & MRS JOHN DOE", "123 ANY STREET", and "SOMETOWN, NJ 99999".
- Check Number (Used for acct. number verification only):** Points to the top right of the check, containing "182".
- Routing Transit Number:** Points to the bottom left of the check, containing "01400024".
- Bank Name and Address:** Points to the center of the check, containing "THE COLUMBUS NATIONAL BANK" and "COLUMBUS, OHIO 43213".
- Checking Account Number:** Points to the bottom center of the check, containing "123456789123".

Other visible text on the check includes: "19 ____ 25-2/440", "PAY TO THE ORDER OF _____ \$ _____", and "_____ DOLLARS".

Option 3. Business Account

Business accounts can be created for businesses that have established credit within the United States, Canada or Europe. You may set up separate memberships for individuals in your organization. A single monthly invoice will list the total charges for each individual User ID Number assigned to the account. You can select this option during the online registration process. At this time, you will be asked to supply bank and trade references. Once this information is verified, you'll receive your permanent Password via postal mail. There is a \$10 monthly minimum per User ID Number associated with this billing option. Your CompuServe basic monthly membership of \$7.95 will apply to the \$10 monthly minimum. All business account User ID Numbers automatically include the Executive Service Option.

7. Receive your new, permanent User ID Number

After you've completed the billing information, CompuServe will display your unique, permanent User ID Number. Make a note of it. You will need it the next time you connect to the service. Your Temporary User ID Number will no longer be valid.

Record your User ID Number and telephone access number here for future reference.

CompuServe User ID Number

CompuServe telephone access number
(See Page 12 for your local number.)

8. Receive your new temporary Password

CompuServe will display a new Password, if you opted to pay by credit card or direct debit. Make a note of it and keep it somewhere safe (separate from your User ID Number).

9. Receive a permanent Password

For security reasons, a permanent Password will be sent to you via postal mail and should arrive within 10 days.

Most CompuServe services can be accessed with your permanent User ID Number and your temporary Password. However, there are some services that you may not be able to access until you receive your permanent Password.

You can change your permanent Password at any time by entering GO PASSWORD at any ! prompt.

Remember, your Password is extremely confidential. Never give it to anyone. No one from CompuServe will ever ask for your Password. Please report any such requests to Customer Service using Feedback (GO FEEDBACK).

Accessing the information service

Once you have received your permanent User ID Number, you are an active member of CompuServe and need only follow these steps to access the service:

1. Refer to the network access number you recorded on Page 17.
2. Initiate the dialing sequence for your modem software. You will typically see a CONNECT message on your screen.

3. Signal CompuServe that you're ready to communicate by pressing either a or by holding down the Control key while pressing the letter C and then a . (Remember to enter CIS and a carriage return if you receive a Host Name: prompt.)

```
CONNECT   
Host Name: CIS   
User ID: number   
(Enter new, permanent ID Number)  
Password: password   
(Enter your new temporary password)
```

4. Enter your new, permanent User ID Number provided to you during registration. Enter a carriage return.
5. Enter your new, temporary Password provided to you during the online registration process. Enter a carriage return.
6. The first time you log on with your permanent User ID Number, you'll be prompted for hardware and software information so that a personal profile can be created. You will then be given a list of navigational commands (commands that help you find your way around the service). Subsequent CompuServe sessions will begin at CompuServe's TOP menu after you review our "What's New" information service on CompuServe each week.
7. To get to CompuServe basic services, select option #1 from CompuServe's TOP menu. To access all of CompuServe's services, select any other option off the TOP menu.
8. To discontinue your session, enter OFF or BYE and at any ! prompt. If you're at a : prompt, enter EXIT or QUIT to return to a ! prompt. Then, enter OFF and .

Simple commands

To get around on CompuServe, you'll need to know the three most helpful commands, GO, FIND and HELP. Use them at any ! prompt.

The FIND command lets you locate areas of interest on CompuServe by topic. FIND searches the CompuServe index for a topic you specify, then lists related services and their GO words. Enter FIND, followed by the topic at any ! prompt. Then enter a . For example, enter FIND TRAVEL and a to list travel-related services.

The GO command lets you bypass menus, saving you online time. Use it to take you directly to areas that interest you. For example, enter GO MEMBER and a to access a list of membership services.

To receive a full list of commands for the service, enter GO COMMAND at any ! prompt. The commands listed below can also be entered at any ! prompt. (These commands don't require the word GO prior to entering them.)

T	Returns you to the CompuServe TOP menu
M	Returns you to the previous menu
N	Takes you to the next menu choice
EXIT	Returns you to the CompuServe service from a service provided by another system, such as Travelshopper (also /EXIT)
HELP	Enter this at any ! prompt for immediate instructions
OFF	Disconnects you from CompuServe.

The following "Control" characters can be used at any time:

<input type="checkbox"/> + <input type="checkbox"/>	Interrupts your current display and takes you to an ! prompt
<input type="checkbox"/> + <input type="checkbox"/>	Pauses the display of information on your screen
<input type="checkbox"/> + <input type="checkbox"/>	Resumes the display interrupted by <input type="checkbox"/> + <input type="checkbox"/>

Free membership support services

The following services are in addition to the many special services included in your basic package and are free of connect time charges.

Take the guided tour (GO TOUR)

You are strongly encouraged to take the guided tour of our services. Just enter GO TOUR at any ! prompt.

Practice with Forums (GO PRACTICE)

Forums are a very valuable part of CompuServe extended services. As a CompuServe member, you are invited to visit the Practice Forum where you will receive a general introduction to forum services plus expert assistance with problems such as how to upload and download files. Just enter GO PRACTICE at any ! prompt.

Browsing Questions & Answers (GO QUESTION)

This section contains information on many frequently asked questions. Enter GO QUESTION at any ! prompt.

How to contact Customer Service (GO FEEDBACK)

If you encounter any problems, contact CompuServe Customer Service for friendly, efficient help. Electronic messages can be left for our representatives by entering GO FEEDBACK at any ! prompt. All messages receive expert attention and you should get a response within 24 to 48 hours. You can also telephone us at the numbers listed on the back cover of this brochure.

Other Useful Membership Support Services

GO BILLING	Review monthly bills, change billing address
GO DIRECTORY	Search for CompuServe members by name
GO HELP	Full menu of help and instructions
GO INDEX	Select areas of interest from a full index
GO MEMBER	List all membership support services.
GO PASSWORD	Change password
GO PHONES	Telephone numbers to access CompuServe
GO PROFILE	Choose display and service options

An index to online destinations

CompuServe is the largest single source of online information, with over 1700 different products and services. Many of those services are listed below and on the following pages.

A “+” indicates an extended service for which pay-as-you-go connect-time charges are in effect. A “\$” indicates that premium surcharges apply, in addition to pay-as-you-go connect-time charges. For rates and information, see Page 28 or GO RATES online. An “E” indicates that you must be an Executive Service Option member to use this service. GO EXECUTIVE online for more information. See Page 10 for additional information.

Type of Service	Charge	Symbol	Example
Basic	\$7.95/mo.	none	Academic American Encyclopedia
Extended	hourly rate	+	IBMNEW+
Premium	hourly rate and surcharge	\$	IQuest (\$)
Executive Service Option	hourly rate and surcharge and minimum	E\$	Disclosure (E\$)

Get Support.

Personal computing forums — hardware

Amiga Forums +
 Apple II/III Forums +
 Atari Forums +
 Commodore Users Network +
 DEC Users Network +
 Epson Forum +
 Hewlett-Packard +
 IBM Users Network +
 Intel Corporation +
 Macintosh Forums +
 MIDI Forums +
 Multimedia Forums +
 NeXT Forum +
 Packard Bell Forum +
 Practical Peripherals Forum +
 Tandy Users Network +
 Texas Instruments Forum +
 Zenith Data Systems Forum +

GO COMPUTERS

GO AMIGA
 GO APPLE TWO
 GO ATARI
 GO CBMNET
 GO DECUNET
 GO EPSON
 GO HP
 GO IBMNET
 GO INTEL
 GO MACINTOSH
 GO MIDIFORUM
 GO MULTIMEDIA
 GO NEXTFORUM
 GO PACKARBELL
 GO PPIFORUM
 GO TANDYNET
 GO TIFORUM
 GO ZENITH

Personal computing forums — software/languages

ACIUS Forum +	GO ACIUS
Adobe Forum +	GO ADOBE
Aldus Forum +	GO ALDUS
Ask3Com Forum +	GO ASKFORUM
Autodesk Forum +	GO AUTODESK
Banyan +	GO BANYAN
Borland International Forums +	GO BORLAND
Central Point Software +	GO CENTRAL
CompuServe Information Manager Support Forum	GO CIMSUP
Computer Consultants Forum +	GO CONSULT
Crosstalk Forum +	GO XTALK
Datastorm Forum +	GO DATASTORM
Desktop Publishing Forum +	GO DTPFORUM
Digital Research Forum +	GO DRFORUM
Fox Software Forum +	GO FOXFORUM
Graphic Forums +	GO GRAPHICS
IBM OS/2 Forum +	GO IBMOS2
InfoWorld On-Line +	GO INFOWORLD
Logitech Forum +	GO LOGITECH
Lotus Forums +	GO LOTUS
Microsoft Connection +	GO MICROSOFT
Nantucket Reference Center +	GO NANTUCKET
Novell NetWare +	GO NOVELL
Oracle Forum +	GO ORACLE
<i>Ziff-Net PC Magazine + and PC Week +</i>	GO ZIFFNET
Software Publishing Forum +	GO SPCFORUM
Symantec Forum +	GO SYMANTEC
UNIX Forum +	GO UNIXFORUM
UserLand Software Forum	GO USERLAND
Ventura Software Forum +	GO VENTURA
WordPerfect Support Group +	GO WPSG
WordStar Forum +	GO WORDSTAR

Electronic publications and other interests

AI Expert Forum +	GO AIEXPERT
CompuServe Navigator	GO NAVIGATOR
CompuServe Software	GO CISSOFT
Computer Language Forum +	GO CLMFORUM
Dr. Dobb's Journal +	GO DDJFORUM
LAN Technology +	GO LANTECH
Online Today	GO TODAY

Keep in touch.

Communication

CB Simulator +	GO CB
Information Service Members	GO DIRECTORY
CompuServe Mail	GO MAIL
FAX-Electronic Mail +	GO FAX
Member Recommendation Program	GO FRIEND
HamNet Ham Radio Network +	GO HAMNET
Changing Your Terminal Type	GO TERMINAL
International Access Information	GO PHONES
Network Access Information	GO NETWORK
Practice Forum	GO PRACTICE
Classified Ads (Read only)	GO CLASSIFIEDS

Know what's news.

News, weather, sports

Accu-Weather Maps, Reports
Associated Press Online
Associated Press Sports Wire \$
Associated Press \$E
Business Wire +
Executive News Service \$E
NewsGrid +
CompuServe Magazine Electronic Edition
Reuters \$E
The Washington Post \$E
Weather Reports, Forecasts, Maps
UK News, Sports
UPI US & World \$E
NCAA Collegiate Sports Network +

News and sports forums

Auto Racing Forum +
Journalism Forum +
Sailing Forum +
Scuba Forum +
Sports Forum +

Take a trip.

Travel services

ABC Worldwide Hotel Guide +
Eaasy Sabre
Official Airline Guide ELECTRONIC EDITION®
Travel Services \$
State Department Travel Briefings
TravelshopperSM
West Coast Travel +
Florida Forum +
Travel Forum +
Aviation Services +
Visa Advisors

Shop in comfort.

Shopping services

Classifieds (Read only)
CompuServe Product Ordering
CompuServe SOFTEX Software Sales \$
Consumer Reports
New Car Showroom \$
SHOPPERS ADVANTAGE® Club
The Electronic Mall®
Selected Mall merchants:
AutoQuot-R
Brooks Brothers
CheckFree Corp.
Compact Disc Club
Crabtree & Evelyn
Florida Fruit Shippers
Godiva Chocolates
Hammacher Schlemmer
JCPenney
MicroWarehouse
Nissan
Sears

GO NEWS

GO WEATHER
GO APO
GO SPORTS
GO ENS
GO TBW
GO ENS
GO NEWSGRID
GO ONLINE
GO ENS
GO ENS
GO WEATHER
GO UKNEWS
GO ENS
GO NCAA

GO RACING
GO JFORUM
GO SAILING
GO SCUBA
GO FANS
GO TRAVEL

GO ABC
GO EAASYSABRE

GO OAG
GO STATE
GO PARS
GO WESTCOAST
GO FLORIDA
GO TRAVSIG
GO AVIATION
GO VISA

GO SHOPPING

GO CLASS
GO ORDER
GO SOFTEX
GO CONSUMER
GO NEWCAR
GO SAC
GO MALL

GO AQ
GO BR
GO CF
GO CD
GO CR
GO FFS
GO GC
GO HS
GO JCP
GO MCW
GO NI
GO SR

Sierra Online
Software Discounters International
Time-Life Books, Music, Videos
University of Phoenix
Walden Computer Books
Walter Knoll Florist

GO SI
GO SDI
GO TL
GO UP
GO WB
GO WK

GO MONEY

Invest wisely.

Money matters and markets

Industry Information

Current Market Snapshot Report \$E
Economic Estimates and Market Analysis \$
Market and Industry Index Symbol Lookup +
Stock Market Highlights from Previous Day \$

GO SNAPSHOT
GO MMS
GO INDICATORS
GO MARKET

Security and Company Information

Current Stock Quotes

(Delayed more than 15 minutes)

Analyze the Return on an Investment \$E
Bond Prices & Yields (12 years) \$
Commodity Market News & Analysis \$
Commodity Pricing & Volume Information (10 years) \$
Company Screening (on Investment Criteria) \$E
Detailed Issue Description of Security \$
Disclosure® Financial Statements \$E
Earnings & Growth Estimates on Companies \$E
FundWatch Online by Money Magazine \$

GO BASICQUOTE
GO RETURN
GO BONDS
GO NAT
GO CPRICE
GO COSCREEN
GO EXAMINE
GO DISCLOSURE
GO IBES

(Mutual fund screening and detailed reports)

Highest Federally Insured CD Rates \$
Historical Securities Pricing Services \$
Information on Trading Performance \$
InvesText Brokerage Reports \$
Options Prices & Volumes for Recent Contracts \$
Portfolio Reporting for your Securities \$
Pricing & Dividends Interface for Spreadsheets \$
Researching a Company \$E
Security Prices & News — Current \$
(Delayed over 15 Minutes)
Securities Price & Volume Graph \$
Security Prices & Volumes — 12 years \$
Standard & Poor's® Company Information
& Recommendations \$
Value Line™ Financial Statements & Forecasts \$

GO MONEYMAG
GO RATEGRAM
GO SECURITIES
GO PRISTATS
GO INVTEXT
GO OPRICE
GO PORT
GO MQDATA
GO ANALYZER

GO QQUOTE
GO TREND
GO PRICES

GO S&P
GO VLIN

Online Brokerage Services

E*Trade Securities, Inc. +
Quick & Reilly Securities \$
Spear Rees & Co. \$

GO BROKERAGE
GO ETRADE
GO QWK
GO SPEAR

Financial and Investment Forums +

GO FINFORUM

Get down to business.

Business management and reference

Aviation and Flight Planning +
Business Database Plus \$
Business Dateline \$
Business Demographics \$
Information USA Forum +
Marketing/Management Research Center \$
PaperChase (MEDLINE) \$

GO BUSINESS

GO AVIATION
GO BUSDB
GO BUSDATE
GO BUDEM
GO INFOUSA
GO MKTGRC
GO PAPERCHASE

Rare Disease Database +
SUPERSITE Demographic Information \$E
TRW Business Profiles \$
U.S. Government Publications +
Working From Home Forum +

GO NORD
GO SUPERSITE
GO TRWREPORT
GO GPO
GO WORK

Industry and professional forums

American Medical Informatics Association Forum +
Aviation Forum +
Broadcast Professional Forum +
Computer Consultants Forum +
Computer Training Forum +
Consumer Electronics Forum +
Int'l Entrepreneurs' Forum +
Journalism Forum +
Legal Forum +
Military Forum +
Public Relations and Marketing Forum +
Safetynet Forum +

GO MEDSIG
GO AVSIG
GO BPFORUM
GO CONSULT
GO DPTRAIN
GO CEFORUM
GO USEN
GO JFORUM
GO LAWSIG
GO MILITARY
GO PRSIG
GO SAFETY

Join a group.

Hobbies/lifestyles/education

Aquaria/Fish Forum +
Art Gallery Forum +
Automobile Forum +
Astronomy Forum +
Coins/Stamps/Collectibles Forum +
Comic Book Forum +
Consumer Electronics +
Cooks Online Forum +
Crafts Forum +
Disabilities Forum +
Food/Wine Forums +
Gardening Forum +
Genealogy Forum +
Health & Fitness Forum +
Human Sexuality +
Issues Forum +
Literary Forum +
Mensa Forum +
Military Forum
Model Aviation Forum +
MIDI/Music Forum +
New Age Forum +
Outdoor Forum +
Pets Forum +
Photography Forum +
Religion Forum +
Science Fiction Forum +
Showbiz Forum +
TrainNet +

GO HOME

GO FISHNET
GO ARTGALLERY
GO CARS
GO SPACE
GO COINS
GO COMIC
GO CEFORUM
GO COOKS
GO CRAFTS
GO DISABILITIES
GO FOOD
GO GARDENING
GO ROOTS
GO GOODHEALTH
GO HUMAN
GO ISSUES
GO LITFORUM
GO MENSA
GO MILITARY
GO MODELNET
GO MIDI
GO NEWAGE
GO OUTDOORS
GO PETS
GO PHOTOF
GO RELIGION
GO SCIFI
GO SHOWBIZ
GO TRAINNET

Find facts fast.

Reference

Academic American Encyclopedia
Biz*File \$
Books in Print \$
Newspaper Library \$

GO REFERENCE

GO AAE
GO BIZFILE
GO BOOKS
GO NEWSLIB

Census Bureau Data +
Computer Database Plus \$
Computer Directory \$
Health Database Plus \$
IQuest — Online Reference Resource \$
Magazine Database Plus \$
Marquis Who's Who \$
Neighborhood Demographics \$
Peterson's College Database
UK Newspaper Library \$
HealthNet
Phone*File \$
Business Database Plus \$

GO CENDATA
GO COMPDB
GO COMPDIR
GO HDB
GO IQUEST
GO MAGDB
GO BIOGRAPHY
GO NEIGHBOR
GO PETERSON
GO UKPAPERS
GO HNT
GO PHONEFILE
GO BUSDB

Education forums

Computer Training Forum +
Education Forum +
Foreign Language Education Forum +
IBM/Special Needs Forum +
Science and Math Education Forum +
Students Forum +

GO EDUCATION
GO DPTRAIN
GO EDFORUM
GO FLEFO
GO IBMSPEC
GO SCIENCE
GO STUFO

Have fun.

Games and entertainment

Air Traffic Controller +
BlackDragon
British Legends +
CastleQuest
Chess Forum +
Classic Adventure
Enhanced Adventure
Flight Simulations Forum +
Hangman
Island Of Kesmai +
MegaWars I +
MegaWars III +
Modem-To-Modem Game Support +
Science Trivia Quiz
ShowBizQuiz
SNIPER! +
The Grolier Whiz Quiz
The Multiple Choice +

GO GAMES
GO ATCONTROL
GO BLACKDRAGON
GO LEGENDS
GO CQUEST
GO CHESS
GO CLADVENT
GO ENADVENT
GO FSFORUM
GO HANGMAN
GO ISLAND
GO MEGA1
GO MEGA3
GO MTMGAMES
GO SCITRIVIA
GO SBQ
GO SNIPER
GO WHIZ
GO TMC

Games forums and news

The Gamers' Forum +
Multi-Player Games Forum +
Role-Playing Games Forum +
Play-By-Mail Games Forum +
The Game Publishers Forum +
The Electronic Gamer™ +
The Modem Games Forum +
The MTM Challenge Board +

GO GAMECON
GO GAMERS
GO MPGAMES
GO RPGAMES
GO PBMGAMES
GO GAMPUB
GO TEG
GO MTMFORUM
GO MTMCHAL

Entertainment news

Rober Ebert's Movie Reviews
Magill's Survey of Cinema \$
Showbiz Forum +
Soap Opera Summaries \$
Rocknet +

GO ENTERTAINMENT
GO EBERT
GO MAGILLS
GO SHOWBIZ
GO SOAPS
GO ROCKNET

Troubleshooting

Q *Do I need CompuServe software to use your service?*

A No. GO ORDER to purchase the CompuServe Information Manager software for both PC compatible machines and Macintosh computers, but you can use any communications software that will run on your computer.

Q *I'm getting random characters mixed in with my text. How can I get rid of them?*

A Garbled text after logon can be caused by many factors including noise on your telephone line, faulty equipment connections and software incompatibility. The two most common causes are the easiest ones to correct. Try this:

1. Disconnect your modem and clean the metal contacts which connect to your computer. Make sure that when you reconnect your modem to your computer that the connection is secure.
2. Make sure that all the terminal settings in your communications software match the following:
7 Databits or a wordlength of 7
Even Parity, One Stop Bit, Full Duplex

Q *Sometimes I am prompted for a "UIC" when I am logging on. What should I enter there?*

A Just enter **[Control]+[C]** and the "Host Name" prompt will redisplay. You should enter CIS at the "Host Name" prompt to continue your logon procedure.

By pressing **[Control]+[C]** instead of **[Enter]** or **[Return]** when you connect, you will avoid the "Host Name:" and "UIC:" prompts.

Q *What phone number do I call to connect to CompuServe?*

A Many common numbers are listed on Page 12 of this membership booklet. You may also follow the instructions for 800-FINDCIS on Page 12. Once online, GO PHONES will find all available access phone numbers.

Rates

GO RATES to read current rates. Rates are subject to change without written notice.

Standard Pricing Plan (Flat Rate)

Your CompuServe basic membership of \$7.95 per month includes unlimited connect time to use a wide variety of services. You will not be charged your basic membership of \$7.95 until your second month of membership. All other services are charged at pay-as-you-go prices. Refer to pages 21-26 or to the CompuServe basic services menu (GO BASIC) for a complete listing of the services.

Alternative Pricing Plan (Pay-As-You-Go)

With CompuServe, you have a choice between the Standard Pricing Plan (Flat Rate) and the Alternative Pricing Plan (Pay-As-You-Go). The alternative plan carries a \$2.00 monthly membership support fee which provides unlimited access to the membership support services free of connect time charges. All other services are priced at the hourly connect rates and are subject to communications surcharges and premium surcharges listed below. To change your pricing plan, GO CHOICES at any ! prompt online.

Communication Surcharges (per connect hour)

CompuServe (United States and Canada).....	Free
TYMNET and SprintNet (Contiguous United States only)	
Evenings & Weekends	\$ 1.70/hr.
Daytime	\$11.70/hr.
800 Direct Dial access (United States only)	
All day (1200 and 2400 baud).....	\$ 8.70/hr.
DataPac® through CompuServe gateway	
Available in Canada only	\$10.20/hr.

Network surcharges apply to all usage.

Additional access networks, rates and instructions can be found by entering GO LOGON.

For international rates information, call the office nearest you. Telephone numbers are listed on the back cover of this brochure.

Connect Charges (per connect hour)

Additional extended services are denoted by a "+" on the online menus. They are priced at hourly connect charges based on baud rate and billed in one minute increments. There is a one minute per session minimum. Connect charges do not include communications surcharges.

300 baud	\$ 6.30/hr.
1200,2400 baud	\$12.80/hr.
9600 baud	\$22.80/hr.

Some CompuServe extended services have additional premium surcharges which are indicated by a "\$" on the online menus. For a current listing GO RATES online or, see pages 29 and 30.

Executive Service Option

Monthly billing minimum	\$10.00/month minimum
-------------------------------	-----------------------

Miscellaneous Charges

Account reactivation.....	\$10.00 each
Hardcopy billing detail	\$ 3.50/report
Returned Direct Debit	
payment due to insufficient funds	\$10.00 each
Monthly handling fee for members with a mailing address outside the United States, its territories, Argentina, Australia, Canada, Chile, Europe, Japan, Korea, New Zealand, Taiwan, or Venezuela	\$10.00/month

CompuServe Mail®

If you are on the Standard Pricing Plan, your CompuServe basic membership of \$7.95 per month includes an electronic mail allowance of \$9.00¹. With this allowance you can send up to the equivalent of 60 three-page messages per month with no additional charge. Reading messages is free (except Internet messages). Messages of more than three pages will be counted as more than one message. Messages are priced at \$.15 for the first 7500 characters and \$.05 for each additional 2500 characters. Your \$9.00 message allowance expires at the end of each month. (Note: Each 2,500 characters is about one double-spaced page.)

If you are on the Alternative Pricing Plan, hourly connect rates apply to reading and sending mail.

Mail	see above
receipt-requested feature.....	\$.15/request
multiple-send feature.....	\$.10/additional recipient (applies to Alternative Pricing Plan only)
Mail to MCI® Mail message transfer	\$.45/up to 500 characters; \$ 1.00/501-7500 characters; \$ 1.00/each additional 7500 characters
Mail to U.S. Telex®/TWX® machines	\$ 1.15/300 characters (International Telex/TWX charges are provided online.)
Mail to U.S. Fax	\$.75/1st 1000 characters + .25 for each additional 1000 characters
European.....	\$.90/First 1000 characters + .90/each additional 1000 characters
Postal Messages	
U.S. destinations	\$ 1.50/First page + \$.20/each additional page
Outside U.S. destinations.....	\$ 2.50/First page + \$.20/each additional page

¹ This \$9.00 value does not apply to additional charges for FAX, postal, TELEX, TWX, or MCI electronic mail transmissions.

Money Matters/Markets

MicroQuoteSM

Market and historical quotes (other than Basic Current Quotes):

Rates apply to the following services: Quick Quote SM , Historical Security and Commodity Pricing, Company Analyzer SM , QUOTES Command	
Current and end-of-day stock and options quotes	\$.015/quote
End-of-day commodity quotes.....	\$.02/quote
Historical daily/weekly/monthly quotes	\$.05/quote/day
Dividends, splits, distributions, interest points	\$.15/item reported
Portfolio Valuation.....	\$ 1.00/evaluation + .05/issue
Reports and Charts (see online information)	\$ 1.00 each
Return Analysis (E).....	\$.50/report + .15/issue
Stock Market highlights.....	\$15.00/hour
Company Screening (E).....	\$ 5.00/screen +.50/company reported
Security Screening (E).....	\$ 5.00/screen +.25/security reported
Bonds listing.....	\$.05/bond
Options profile.....	\$ 1.25/report
Current market snapshot (E).....	\$.10/report

Investment Support

Disclosure II SM (E)	\$ 5.00 to \$25.00/report
Futures Focus (Prime/Standard hours).....	\$20.00/\$15.00/hour ¹
Institutional Brokers' Estimate System (I/B/E/S) SM (E).....	\$.50 to \$2.00/report
InvesText.....	\$ 4.00/search (up to 10 items); \$ 4.00/additional report titles (groups of 10); \$11.00/full page
Value Line Data Base II SM	\$.40 to \$1.60/report
Money Market Services	\$ 5.00/report
News-A-Tron Market Reports	\$ 1.25/entry

Investment Support (continued)

S&P Online SM	\$ 1.00/company \$ 2.00/Investment Ideas \$ 2.00/Master List
RateGram.....	\$ 1.00/report
Fund Watch Online by Money Magazine	\$15.00/hour for mutual fund screening/reporting

Online Brokerage

Quick*Way® (Prime/Standard hours).....	\$14.00/\$4.00/hour ²
Spear Rees & Co. (Prime/Standard hours)	\$14.00/\$4.00/hour ²

Games and Entertainment

Soap Opera Summaries	\$ 6.00/hour
----------------------------	--------------

News

Executive News Service.....	\$15.00/hour
AP Sports Wire.....	\$15.00/hour

Reference/Education

Phone*File	\$15.00/hour
Biz*File	\$15.00/hour
IQuest.....	\$ 9.00/search; \$3.00/abstract; \$ 1.00/no hit; \$ 2.00 to \$75.00 surcharge for some databases \$ 5.00/SmartSCAN

ERIC, NTIS, Book Review Digest, Books In Print D&B-Dun's

Electronic Business Directory, Commerce Business Daily,

Magill's Survey of the Cinema	\$ 2.00/search; \$ 2.00/article/listing; \$ 1.00/no hit
-------------------------------------	--

Corporate Affiliations, Media Newsletters, Data Processing Newsletters,

Dissertation Abstracts, PsycINFO, Compendex, PDQ, Thomas Register

Online, D&B - Dun's Market Identifiers (US, Canadian international),

Business Dateline, CCML Aids Articles	\$ 5.00/search; \$ 5.00/article/listing; \$ 1.00/no hit
---	--

Marketing Management Research Center, Patent Research Center, Trademark

Research Center, Legal Research Center.....	\$ 1.00/search; \$ 5.00/database select and retrieval; \$ 5.00/article/listing
---	---

Marquis Who's Who	\$ 4.00/search; \$ 4.00/article/listing; \$ 1.00/no hit
-------------------------	--

Newspaper Library	\$ 3.00/search; \$ 3.00/article; \$ 1.00/no hit
-------------------------	--

TRW Credit Reports.....	\$ 9.00/search; \$29.00/report; \$ 1.00/no hit
-------------------------	---

UK Newspaper Library, UK Trademark Library,

Business Dateline.....	\$ 6.00/search; \$6.00/article; \$ 1.00/no hit
------------------------	---

New Car Showroom	\$ 1.20/comparison search; \$.90/price one model \$.40/Auto selector
------------------------	--

PaperChase (MEDLINE) (Prime/Standard hours).....	\$24.00/\$18.00/hour ²
--	-----------------------------------

Demographic Reports.....	\$10.00 to \$100.00/report
--------------------------	----------------------------

Computer Directory	\$15.00/hour; \$ 1.00/product/manufacturer menu; \$.25/full product/manufacturer listing
--------------------------	---

Computer Database Plus.....	\$15.00/hour; \$ 2.50/full-text article; \$ 1.00/summary
-----------------------------	---

Magazine Database Plus, Business Database Plus,

Health Database Plus.....	\$15.00/hour; \$ 1.50/full-text article; \$ 1.00/summary
---------------------------	---

VISTA Environmental Profiles.....	\$150.00/profile
-----------------------------------	------------------

Travel/Aviation

EMI Flight Planning ³	\$ 2.50 to \$6.00/plan
--	------------------------

Official Airline Guides ELECTRONIC EDITION®

Travel Service (Prime/Standard hours).....	\$28.00/\$10.00/hour ²
--	-----------------------------------

² To determine Standard/Prime hours of operation, Go to RATES.³ Surcharges apply for additional briefings and flight plan registration.

(E) Executive Service Option Feature

All rates quoted in U.S. dollars

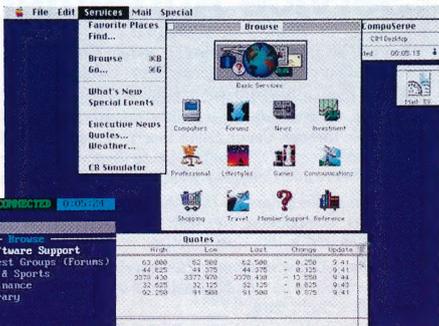
Rates are effective June 1, 1992

Service Agreement Terms

You will be asked to agree to the current CompuServe Information Service Terms during the online registration process. Please read them carefully before becoming a member.

1. The CompuServe Information Service (the "Service") consists of the computing and communications services, software, databases, data, information and all other material (collectively "Information") available through CompuServe Incorporated ("CompuServe"). These terms and any Operating Rules published over the Service constitute the entire agreement (collectively "Agreement") between CompuServe and Customer with respect to the Service and supersede all other communications.
2. Upon notice published over the Service, CompuServe may modify this agreement, the Operating Rules or prices. CompuServe may discontinue or revise any or all other aspects of the Service at its sole discretion and without prior notice.
3. Unless otherwise agreed, Customer's right to use the Service or to designate Users is not transferable and is subject to any limits established by CompuServe, or by Customer's credit card company if billing is through a credit card.
4. Customer agrees to indemnify CompuServe against liability for any and all use of Customer's account.
5. Customer is responsible for and must provide all telephone and other equipment and services necessary to access the Service.
6. Customer shall pay, in accordance with the provisions of the Billing Option selected by Customer, any registration or monthly fees, connect time charges, minimum charges and other charges incurred by Customer or its designated Users at the rates in effect for the billing period in which those charges are incurred, including but not limited to charges for any purchases made through the Service and any surcharges incurred while using any supplemental networks or services other than the Service. The Customer shall pay all applicable sales and use taxes relating to its and the Users' use of the Service. The Customer shall be responsible for all use of the Service accessed through Customer's or its designated Users' password(s).
7. CUSTOMER EXPRESSLY AGREES THAT USE OF THE SERVICE, WHICH INCLUDES THE CONTENTS THEREOF AND ANY STORAGE OR USE OF INFORMATION, IS AT CUSTOMER'S SOLE RISK. NEITHER COMPUSERVE NOR ANY OF ITS INFORMATION PROVIDERS, LICENSORS, EMPLOYEES, OR AGENTS WARRANT THAT THE SERVICE WILL BE UNINTERRUPTED OR ERROR FREE; NOR DOES COMPUSERVE OR ANY OF ITS INFORMATION PROVIDERS, LICENSORS, EMPLOYEES OR AGENTS MAKE ANY WARRANTY AS TO THE RESULTS TO BE OBTAINED FROM USE OF THE SERVICE. THE SERVICE IS DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING BUT NOT LIMITED TO WARRANTIES OF TITLE OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR USE WITH RESPECT TO THE SERVICE OR INFORMATION. NEITHER COMPUSERVE NOR ANYONE ELSE INVOLVED IN CREATING, PRODUCING OR DELIVERING THE SERVICE SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF USE OF THE SERVICE OR INABILITY TO USE THE SERVICE OR OUT OF ANY BREACH OF ANY WARRANTY. THE PROVISIONS OF THIS SECTION 7 WILL SURVIVE ANY TERMINATION OF THIS AGREEMENT.
8. Except as expressly permitted in the Operating Rules, neither Customer nor its designated Users may reproduce, redistribute, retransmit, publish or otherwise transfer, or commercially exploit, any Information which they receive through the Service.
9. The provisions of paragraphs 7 and 8 are for the benefit of CompuServe and its Information Providers, Licensors, Employees, and Agents; and each shall have the right to assert and enforce such provisions directly on its own behalf.
10. This agreement is, and shall be governed by and construed in accordance with the law of the State of Ohio applicable to agreements, made and performed in Ohio. Any cause of action of Customer or its designated Users with respect to the Service must be instituted within one year after the claim or cause of action has arisen or be barred.
11. If Customer's account is a qualified business account and approved by CompuServe for corporate billing, charges for the services provided under this agreement will be accumulated and identified by User ID number and will normally be invoiced following the end of the month in which the service is provided. Terms of payment on all charges are net, ten (10) days in the currency in which billed. If any payment due hereunder is not made by the Customer within thirty (30) days after the invoice date, late charges of one and one-half percent (1 1/2%) per month shall be due and payable with respect to such payment, and CompuServe may, in addition, at its sole discretion and without notice to the Customer, (a) suspend its performance under this agreement and the Customer's and its Users' access to and use of the Service, or (b) terminate this agreement and Customer's and its Users' access to and the use of the Service. For accounts not approved by CompuServe for corporate billing, Customer must provide payment by credit card or electronic funds transfer.
12. This Agreement contains the full understanding of the parties with respect to the subject matter hereof, and no waiver, alteration, or modification of any of the provisions hereof shall be binding on either party unless in writing and signed by duly authorized representatives of the parties. Neither the course of conduct between parties nor trade practice shall act to modify the provisions of this Agreement.

CompuServe introduces an exciting screen star!



The CompuServe Information Manager now appearing on Mac and DOS screens everywhere.

Get the CompuServe Information Manager software for incredible performance.

Get the information and communications power of our global online service and the simple beauty of an easy-to-use format with the CompuServe Information Manager Software, available for Macintosh or DOS computers. Pull-down menus, dialog boxes, and simple commands help you save time and money each time you log on. Convenient features that enable you to perform such tasks as uploading and downloading messages make those savings even greater.

To purchase this software and receive an equivalent value usage credit, GO ORDER online.

CompuServe®

P.O. Box 20212
5000 Arlington Centre Blvd.
Columbus, OH 43220 U.S.A.

Customer Service: Argentina (+54) 1-322-1864 • Australia Toll-free: 008-023-158 or Direct: (+61) 2-410-4555 • Canada Direct (+1) 614-457-8650 • Chile (+56) 2-696-8807 • Germany Toll-free: 0130-86-4643 or Direct: (+49) 89-66-55-0-222 • Japan Toll-free: 0120-22-1200 or Direct: (+81) 3-5471-5806 • Korea Toll-free 080-022-7400 or Direct: (+82) 2-569-5400 • New Zealand Toll-free: 0800-441-082 • Switzerland Toll-free: 155-31-79 • Taiwan Direct: (+886) 2-515-7035 • United Kingdom Toll-free: 0800-289-458 or Direct: (+44) 272-255111 • United States Toll-free: 800-848-8990 • Venezuela Direct: (+58) 2-793-2984 • Elsewhere (+1) 614-457-8650

All names listed in this brochure are proprietary trademarks of their respective corporations.

Printed in USA

CS-147-U-1 (06/92) UNIV-15

Temporary User ID Number: 177000, 1774
Temporary Password: EXPLORE/WORLD
Agreement Number: WC9379
Serial Number: 20007852
Extended Services Usage Credit: \$15.00 Credit