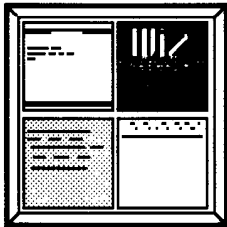


\$79.95

WIZTM PROFESSIONAL

*The OS9 Level II Terminal
Program*



Copyright © 1988
by
William L. Brady

October 20th 1988

Dear Wiz Pro User;

Wiz Professional uses most of the OS-9 Level II windowing features. Wiz **must** use these features in the way that they were designed to be used, and it does. If you have added patches to your OS-9 system modules that alter the functionality of the Operating System, they must be removed in order for you to use Wiz Pro.

Some problem patches have been identified:

A patch to /W to cause a pause or no pause condition.

A patch to 'boot-up' in an 80 column window. (boot-80?)

A patch to 'boot-up' in the RGB mode.

The problem these patches create show up when Pro is first opening up the main-on-line screen(s), and when you run a shell from Pro. (ALT-S). (Pro runs the shell in the 'base' or default, window.)

The objective of all of these patches can be accomplished in 'legal' ways. First, do not modify /W. Wmode your standard window descriptors to suit your taste and make a new boot if you like. Booting into an 80-col window can be accomplished by the procedure make80 placed in your startup file. A 'montype r' placed in the startup file will accomplish the final patch.

I suggest that if you do make patches, that you always install them at boot time by placing a 'modpatch <patchfile' line in your startup file. Never make a new boot with patched files unless recommended by Microware, Tandy, or the OS-9 Users Group. This way, you can more easily nullify patches by changing your startup file. But more importantly, you can always find out what patches you are actually running by simply listing startup.

Note that some IPATCH files are not patches at all, but replacement modules. Also, except for /W, changes to device descriptors are not patches either. These comments don't apply to these.

If you have downloaded the abridged version of Wiz Pro, just replace the modules in your system with the ones on the disk that are new. I suggest Carl Kreiders ARC utility if you have it. You don't need to make a new boot.

Good luck, and thanks for all of the good words to myself and and about Wiz Pro.

-Bill Brady

Wiz would not exist but for the invaluable help we have received from a number of people, thanks:

To Ray Mayeux, Steve Pursely, Jerry Murphy, Keven Pittsinger, Greg Morse WayneDay,

To Kevin Darling; for his OS9 Systems expertise, which as far as I know is nonpariel; and both Kevin and Marsha for tolerance and patience with my ravings, and

To Dale Puckett for his aid as Beta Tester & critic, and to Esther for putting up with the two of us, and

To Paul Ward , Dan Robins and Frank Hogg, for their advice,

And for the Wiz Pro Manual, again, we have Jane Larivee to thank that it is as readable as it is.

The information contained within this document is subject to change without notice. It is against the law to copy this document or Wiz Pro on any any media including magnetic disk for any purpose other than your own personal use.

© Copyright William L. Brady, 1988

The Author does not guarantee the suitability, sufficiency or accuracy of Wiz. There are no warranties, expressed or implied, including those of merchantability and fitness for a particular application, concerning the software or other materials delivered herein.

TRS-80 Color Computer is a Registered Trademark of Tandy Radio Shack Corporation. Kermit is a product of Columbia University. Dynacalc is a Registered Trademark of Computer Systems Center. OS-9 is Copyrighted (1983) by Microware Systems Corporation.

Should you use this Program?

Wiz Pro is a very powerful program. It gets its power from Basic09 and OS-9 Level 2. To do this, it must be fully integrated into the system. There are two reasons why this is important to you. First, you will have to make a new boot. Tandy & Microware do not allow third party s/w writers to provide bootable OS-9 software for the Color Computer. Making a new boot can be tough, but once you master it, you are in control of your own computer. You are the 'system manager'. The second factor is that you may need to 'clean-up' your system. If you have installed patches to system modules, you may find that they interfere with the interrupts that are vital for the rs-232 pak to function. You will need to remove them.

If you have questions or problems, or if you have never used any terminal program, or just want to chat, call me at 301-952-1761 after 6PM eastern time.

-Bill Brady

"It is against the law to copy this document or Wiz Pro on any any media including magnetic disk for any purpose other than your own personal use."

Table Of Contents

Introduction.....	3
What's New.....	3
Components of WizPro.....	4
What do you do now.....	4
Wiz Pro Block Diagram.....	5
Other Wiz Features.....	6
Sect 1 Installing WizPro on your computer.....	7
Making a New Boot.....	7
Using ProInstall and NewHost.....	9
Before You Run Wiz Pro.....	10
Sect 2 WizPro Operation.....	11
On Line Keys.....	12
Clipper.....	14
Conference.....	16
Wait.....	16
Menu.....	17
Stop!.....	17
Pause.....	18
Fkeys.....	18
WizPro File Menu.....	19
Sect 3 Standard XPROCs Summary.....	20
Autolog.....	20
Break.....	22
Configuration.....	23
Download.....	24
Enables.....	25
Full/Half Duplex.....	25
Help/Info.....	26
Print.....	26
Quit.....	26
Reset.....	27
Shell.....	27
Upload.....	27
Working Directory.....	28
XModem.....	28
Operating Examples.....	30
Sect 4 Customizing WizPro.....	32
Installing a XPROC or XPROG.....	32
Quick Reference Chart.....	33
Sect 5 Building WizPro.....	34
Technical Discussion.....	35
How Wiz Pro invokes an extension.....	36
How to run an XPROC before WizAuto.....	37
How to Run an extension directly from autolog.....	37
Automatic extension operations.....	37
WizPro Parameters.....	38
ProAcia Reference.....	44
Sect 8 If you have problems.....	46
INDEX.....	48
Appendix A Programming Examples.....	51
Appendix B Other Pro Extensions.....	52

Introduction

WizPro is like a mini-operating system for telecommunications. It provides a serial I/O platform and the tools and resources for keyboard, serial port, screen, buffer, and file management. Built upon this platform are a collection of functions which may be changed, deleted or augmented.

Wiz Pro is also a complete telecommunications program. But I like to describe it as a user interface to the telecommunications world. Here are some of the most unique of its features:

Wiz Pro runs in a multi-tasking environment, in windows on the Color Computer 3; it does not 'take over' your computer. You can do other things while Wiz is running, even while it is up or downloading files.

Wiz Pro is fully re-definable. Almost every parameter that it uses is contained in external files that are modifiable. For example the user can configure Wiz Pro to run in either a text or a graphics screen. (Although everyone so far prefers the text screen for the faster speeds and more colors.)

Wiz Pro is fully extendable. Many procedures are external, and can be replaced by the user. Extra external procedures may be added, again by the user. These extensions can easily make use of Wiz Pro resources. Installation is straightforward, yet extends into the Wiz Pro Menu.

Wiz Pro extensions may be created by anyone. There are no fees or agreements required.

Extensions could include: Autopilot programs, protocol programs, procedure creation programs, configuration extensions AND EVEN OTHER TERMINAL PROGRAMS. Extensions may be placed in the public domain, and some will be supported in the public domain by the author.

WHAT ARE THE DIFFERENCES BETWEEN WIZ AND OTHER OS-9 TERMINAL PROGRAMS?

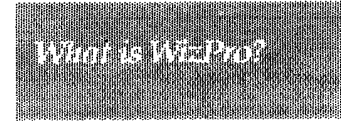
Wiz is designed to be easy to use, fast, and changeable to meet future requirements. Other programs are designed to be used in a limited set of circumstances, with certain hosts, at limited baud rates and with a fixed set of capabilities.

WHAT ARE THE DIFFERENCES BETWEEN WIZ AND WIZPROFESSIONAL?

Wiz Pro is an entirely new program. It has grown out of concepts pioneered by Wiz, and discoveries made during the development and servicing of Wiz. Wiz Pro does not replace Wiz, but gives the advanced user, and novice alike, a growth path for the future. Wiz Pro is designed to resist obsolescence. The original Wiz retains uniqueness in its VT-52 Emulation, Graphics input modes, and simpler design.

Some of the New "neat" stuff.

Mouse Paws! (pause). Clicking the mouse brings the host to a screeching halt, no more messages getting by. Click again to resume.



Multi-Tasking

User Defineable

User Extendable XPROCS and XPROGS

Auto-Pilots Protocols

Why Wiz?

What's New?

Wiz, the original, retains the VT52 mode, and the Graphics input mode as unique operations.

Wiz Pro File Menu. Allows you to move up and down your disk directory tree quickly and easily using arrow keys, and quick "finds".

Clipper works on the last 10k bytes that you have received. You can use the arrow keys to mark lines which you can send to the printer, disk or modem. Clipper also lets you go back on-line with host while the buffer is still on the screen... and more. You can also load the buffer from a file.



Wiz Pro uses more than 64K. Wiz overlays procedures in and out of its process space, but the switch is transparent to the user.

Fkeys. Almost every ALT-Key does something, or can be made to do something via Fkeys.

COMPONENTS OF WIZ PRO

ProAcia. A port driver designed for supporting telecommunications not with a terminal, but to another computer (the "Host").

WizPro© .The main procedure.

WizStart© The startup program

WizUtils©. Internal extensions to Wiz Pro.

WizClipper©. The input buffer editor.

WizAuto. Use this to call and log on to the host computer.

WizInfo. Mainly a Diagnostic, tells you what's happening.

WPxmod. Wiz's xmodem protocol.

WizConfig. Port, colors, and fonts configuration program.*

WizHelp. Interactive help program. User source provided.

WizSend. Procedure for sending text files.*

WizQuit. Shutdown procedure/session logger.*

WizFkeys. Single-key operations such as macro send.

WizShell. A simple XPROC to run a shell.

NewHost. An extension to help you create autolog files.

ProInstall. Quick way to install/edit/remove extensions and other default settings.

ProStuff. All the things Wiz Pro wants to remember about you.

WizCsum/WizCRC. Xmodem machine language modules.

* source provided to Registered Wiz Pro Developers. Available to users.

ProAcia as furnished with Wiz Pro is designed to support a standard rs-232 port such as the Tandy RS-232 pack, the Disto SuperController, and the PBJ 2SP Pak.

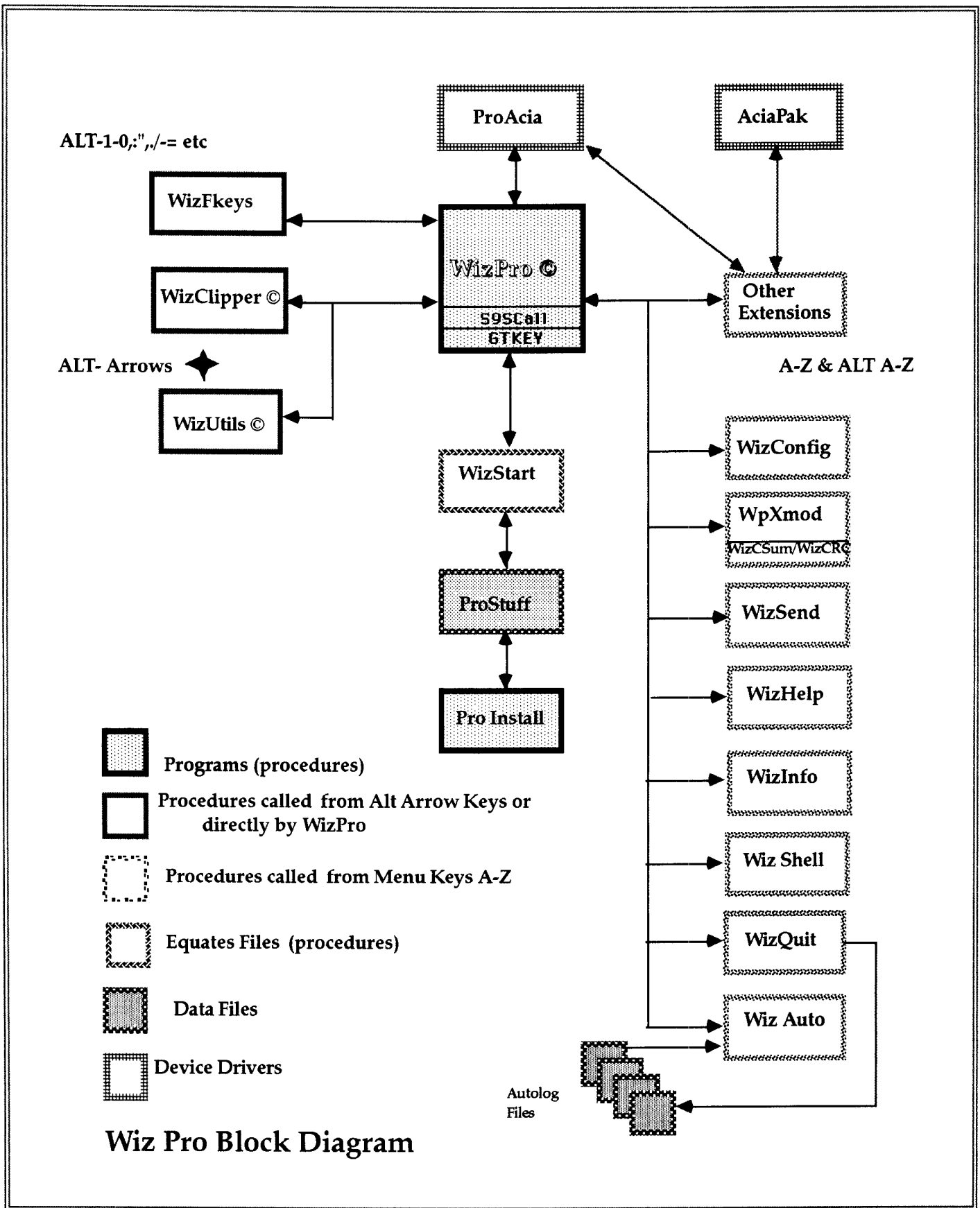
There may be more procedures than you see here by the time you read this manual. Remember, Wiz Pro can be extended for almost any purpose.



What do you Do Now ?

1. Install Wiz Pro Section 1
2. Run Wiz Pro..... Section 2
3. Make some autolog files..... Section 3
4. Install some extensions..... Section 4
5. Customize Wiz Pro..... Section 4,5
7. Write you own extensions..... Section 5,6 and beyond

NOTE: The term "Wiz" as used here to describe a capability, generally refers to a capability present in both Wiz and Wiz Pro. The term Wiz Pro or Pro generally refers to Wiz Professional, and WizPro is the primary Wiz Professional Program Module.



Other Wiz Pro features Include:

A dedicated **DEVICE DRIVER** which is optimized for the CoCo when it is a terminal, rather than talking to a terminal.

INTERRUPT DRIVEN: Wiz operates at up to 19.2k baud.

SIGNAL DRIVEN: Detects control bytes, blocks, etc.

PRINT THROUGH: You can print files or messages from the host while on-line.

CONFERENCE MODE: A one-line device window at the bottom of the screen is used for line composition while the main screen is updated by the Host.

MODEM RESET, HOST, TIMER RESET: You don't have to exit Wiz, you can log your session with one host, then proceed to autolog with the next, even reset your modem if it is AT command compatible or responds to **DTR toggle**.

DISK DOWNLOAD (TEXT): You can store everything to disk while on-line - NO LIMITS to filesize, no 'capture buffer', no lost characters. You can even upload and download at the same time!

AUTOLOGGING: Uses the simple but effective Tandy Model 100 system.

DISK UPLOAD (TEXT): Allows you to send files to a host while on-line; no buffer "pre-load" required. Handshake by specified character, CR, or timeout. Will break long lines based on a value that you supply for each host.

On-line HELP: Holding down ALT while you tap H will bring up the Help screens. Help also available in XPROCS.

CHANGE DATA DIRECTORY: Just about any time, including when you autolog.

Wiz uses /DD, the OS9 default device. It always looks in /dd/com/pro for it's autolog files. You can run Wiz from any directory.

COLORS: Wiz will change the screen colors at any time and for each autolog session. You can have different colors for each host/service/BBS that you use.

FONTS: Works like the color changes.

BOLDFACE and PROPORTIONAL: OS9 does it, Wiz supports it.

XMODEM: Will send or receive any size/type file, includes CoCoBin, 1k XMODEM XMODEM CRC, and Ymodem (Xmodem batch) download.

YMODEM: Pro does true YMODEM download. Ymodem is a batch file transfer method.

X-ON/X-OFF: will work with hosts that "dribble", or continue to send a few characters after they receive the x-off. Even works with hosts that sometimes ignore an x-off!

BREAK KEY: User definable. <ALT-BREAK> sends TRUE LINE BREAK

MACROS: Two 80 character macros can be composed and sent while you are on-line. Good for use in conferences.

CONTROL BYTES: Sent directly from the keyboard. (ALL).

USAGE LOG: Keeps track of your on-line time for each host. Estimates your **Billing** too.

7 OR 8 BIT: User definable. Auto Changeover for Xmodem.

PARITY: Mark, space, odd, even, none.

STOP BITS: One or two.

SHELL: Interface to the OS9 shell while on-line, or while autologging.

WAIT STATE: The Wiz will go to sleep instantly from the on-line state, disconnecting itself from the port hardware so that you may run a program that uses a different driver.

unlimited EXTENSIONS: Install another terminal program in the WizPro menu, run it any time; but let WizPro do your autologging, conferencing, and time keeping.

Function Keys. (**FKeys**). Send macros, etc; can be easily extended, re-defined.

File menu. Wiz Pro relieves you from the drudgery of typing filenames when they are needed. It shows you what's available, and can find and select files and directories from one key stroke!

On-screen Date/time clock and Session timer.

MANY other features are possible with Wiz Pro.

Section 1. INSTALLING WIZ PRO ON YOUR COMPUTER

What do you need other than this Program?

The RS-232 pak must be in slot 1. If you have an old style Multi-pak and are using a CoCo-3 you should have the PAL upgrade. You can use a straight one-to-one modem cable between the RS-232 Pak and the modem. If your modem is set to assert carrier detect all the time, you will see your characters on the screen even if you are not connected to a host computer. If you are NOT configured this way, be sure to use the "<" - wait for carrier symbol, in your autolog file.

Wiz Pro does not use "capture buffers" for downloading or saving data from the host. Instead it uses the X-OFF/X-ON technique to cause the host to pause while it writes to disk, prints or is otherwise busy. Almost every host accommodates this "handshake" technique. Some cannot handle X-OFF while you are logging on, (usually a limitation of a communications "node" rather than a host), but this is not a problem as Wiz Pro does not use x-off unless it's driver (ProAcia) buffer is nearly full or you are actually downloading to disk or printing on-line.

If you want to simulate a capture buffer on a CoCo 3, use a ramdisk. Set the "D" option from the menu with a filename of: /ramdisk/filename. You can leave this on the entire session, even while uploading or using xmodem.

There is, however, buffering of the most recent 10240,(10k), incoming characters, and the ProAcia ACIA driver also contains a buffer. Both are automatically rotational.

Making a new Boot

The first step is to make a new boot. This must be done because of the way OS-9 level 2 works. Level 2 requires device drivers to be in system space. Loaded drivers each use one block (8k) of system space. There are ways to get around this problem, and if you know how, by all means do so. But we shall proceed here on the basis that you will be making a new boot.

I recommend the use of OS9Gen. The simplest procedure is to make a MODS directory, and CHD to there. Then type MDIR to see what modules you currently have in your system space. For each one that you want to be in your new boot, type 'SAVE NAME' (one file per line). For example, 'save os9p2'. (Start with os9p2, skip os9p1 and those before it, skip CC3Go, and STOP with whatever is just before GRFDRV.). As soon as this is done, do an mdir again, then a dir, and make sure you have them all saved. Then copy ProAcia and the appropriate device descriptor from the Wiz Pro disk into this directory also (mwp.rs232 is the 'standard' descriptor).

Now if you have the ls, or fls, utility just type 'ls >bootlist'. Ls does the same thing as the dir command, but puts one filename on a line like:

```
os9p2
init
SCF
...
```

Requirements:

A 512k Color Computer 3

A rs-232 Pak or other s6551 ACIA hardware

Basic09(runb)/OS9 level II version 2.0

If you use the rs232 pak, You must also have a Radio Shack Multi-Pak adapter. WizPro will also work with other products that are functionally the same as the RS-232 Pak.

You need the standard utility 'Rename' in your CMDS directory.

OS9Gen

You will get an error 209 if you try to save cc3go. Copy it over from your OS-9 Release disk instead.

Kevin Darling's Save command is included on the Wiz disk. To use save, type "save name1 name2 ", where name1 is the filename to save to, and name2 is the module to save.

Instead of :

```
os9p2    init    SCF    ...
```

The idea is to make a file called 'bootlist' that contains a list of the file names that will end up in your os9boot file on the new bootable disk. If you don't have ls or fls, then use any text editor, or build, to make your bootlist.

Use the same order that you started with in your original boot, the order in which you found the files with mdir. Put ProAcia just before aciapak in your file, or you may want to replace aciapak and t2 with ProAcia and mwp. If you have double sided 40-t floppies you may also want to replace your d0 d1 etc with d0_40t.dd etc from the OS-9 Master Boot/Config disk. If you have Flight Sim II, put FT FTDD and the FSim INIT in there also. (from your fsm master disk: boot up on FSim & save them as you did the other modules, [alt-cntrl-left arrow breaks you out of fsm to a shell]).

Once you have a MODS directory with all of the modules in it, and a bootlist, then format a fresh disk and type 'os9gen #40k /d0 <bootlist'. OS9-Gen will read the 'list' and create a new OS9boot file on /d0.

After formatting and os9genning, the new disk will need at least one file: /d0/cmds/grfdrv. If you do not have a hard disk, you will also need to dsave or arc all the files over from your old opsys or boot disk. Don't forget Shell, you can't boot without it!

Now you must copy the WizPro Files to your system disk cmds directory. You can simply copy all of the files from the cmds directory on the Wiz Disk, or you can merge some of them together for efficiency (see "building Pro").

You must also create a '/dd/com/PRO' directory. This will contain your autolog files. For now, just copy over the files found in the Wiz disk 'com' directory.

Note:

I do not show any extensions like d0.dd. The extension is not important, but you should know that d0_40t.dd is d0 in memory.

The MODS Directory

OS9P2	INIT	IOMan	RBF	CCDisk
d0	d1	SCF	CC3IO	windint
vdgint	term	w	w1	w2
w3	w4	w5	w6	w7
w8	w9	w10	w11	w12
w13	w14	w15	Proacia	MWP
aciapak	t2	printer	p	ft
ftdd	clock	cc3go	Bootlist	

A Bootlist:

```
OS9P2
INIT
IOMan
RBF
CCDisk
d0
d1
SCF
CC3IO
windint
vdgint
term
w
w1
w2
w3
w4
w5
w6
w7
w8
w9
w10
w11
w12
w13
w14
w15
Proacia
MWP
aciapak
t2
printer
p
ft
ftdd
clock
cc3go
```

What Can You leave OUT?

SIO and S1 are the serial port driver and descriptor. On a Color Computer 3 they are not useable. If you never use pipes, you can also leave out pipeman, piper, and pipe. Note that WindInt comes with Multi-View, and replaces GRFInt. VDGInt can be kept in your bootfile for VDG Screen compatibility. (Pre LII type screens.) Acia-pak and T2 can be omitted if you are not going to use a remote terminal, or another terminal program that requires it.

Summary of steps:

1. Copy SAVE to your CMDS dir.
2. Make a dir called MODS.
3. Copy or save the modules you will need from:
 - a. the Wiz disk
 - b. the OS-9 boot/config disk
 - c. Flight Sim (if desired)
4. Make a new bootlist.
5. Format a new disk.
6. OS-9gen the new disk.
7. Copy over the Wiz files to the new disk, plus CMDS - including "Rename" and "Runb". (Mkdir /dd/com/pro)

Using ProInstall

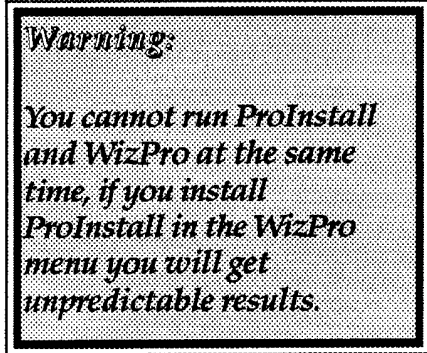
ProInstall allows you to quickly install extensions in WizPro, and to modify a number of other WizPro default settings. You do not need to run Pro Install in order to use WizPro.

ProInstall knows about most of the WizPro variables and can read the ProStuff file created by WizPro and modify it.

You may want to use ProInstall to add something to the menu, or to change the type of screen used by Wiz from text to graphics. (Running in graphics screens allows you to use special fonts and things like boldface, the text screen is faster and provides more colors.)

You don't always need ProInstall to change things. This may also be done with completely separate programs that modify the WizPro resource file 'ProStuff' directly. As new extensions become available for Pro, they will likely be installed this way.

ProInstall provides both an expert and non-expert mode. If you are adding a simple extension, choose the non-expert mode. (answer 'n' at the prompt.) To run ProInstall, go to any window and type proinstall. ProInstall will not run, however, unless it finds the file /DD/COM/PRO/ProStuff. So you must run Wiz Pro at least once so it can create this file.



Using NewHost

NewHost is a special sort of extension. It comes installed in the WizPro menu, but is probably the first one you will remove. It helps you to make up your autolog files, or at least the part that contains the commands that you send to your modem and to the host system during the login process.

You will use NewHost right from within WizPro. That way you can combine it with the actual WizPro communications settings that you are using. Be sure to set all of your serial port values, the break key, and your working directory before running NewHost. (although you can do it after if you like.)

NewHost includes a lot of on-line help, and prompts you through the process, but you may want to refer to the section on making autolog files as you run it.

Once you use NewHost to create your modem and host dial and login strings, and Configuration to set your port, colors, etc, you <Q>uit Wiz Pro - selecting the log and <S>tart new autolog file options. Your new autolog file will be created in the directory /DD/COM/PRO.

Making autolog files is tricky, look at the samples provided on the Wiz Pro disk. Note tricks like only waiting for 'U' and '!' in 'User Name:' Be generous with pauses, "=", they don't slow you down much.

A basic 'autolog' is the process when Wiz Pro:

Tells the modem to dial a phone number.

Waits for the host computer to answer.

Sends control characters and strings to the host and waits for responses.

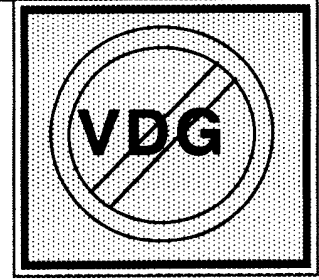
To tell the host who you are.

And give the host your password.

Before You Run WizPro:

Verify the following:

1. You have made a new boot with ProAcia & MWP and you are running from that boot:
Type MDIR, you should see ProAcia & MWP in memory.
2. You are running in a window and NOT a VDG screen.
3. All of the modules on page 4 are in your CMDS directory.
- use copy to copy them from the Wiz disk
4. Runb, Rename, and Tmode are in memory or your commands directory.
- use MDIR or Dir -x
5. You have created a directory /DD/COM/PRO.
- use makdir
6. If you are running Wiz in a graphics configuration, be sure that you have merged STDFONTS and any custom fonts that you want to use. (you must have stdfonts in memory)
7. You have an RS-232 pak or equivalent in Slot 1 of your MPI.



If you have downloaded Wiz Pro abridged version, make sure that the 'e' attribute bit is set on all the files.

Type 'attr filename e'

XPROCS and XPROGS.

For ease of identification, extensions to Wiz Pro fall into two categories. The first are extensions that know about and can use Wiz Pro parameters. These are called XPROCS.

The second type are programs that can run by themselves without Wiz Pro. These are called XPROGS. An example would be the Dir command, although it would be a bit wasteful, you could install it in the menu as an XPROG. More likely, the XPROGS you install will be other terminal programs, and desk accessories like gcal.

Remember, that both XPROCS and XPROGS can either be on your disk, or in memory. If in memory, they run quicker, and once you know which ones you want to keep in memory, you can use ProInstall to set up Wiz Pro to load and unload them for you automatically.

Mouse Pause

Wiz Pro uses the SS.MsSig System call to detect the mouse button down state. Which joystick port is active, and selection of HiRes adapter is not within the power of Wiz Pro, but is a system wide selection. See the OS-9 Level II documentation. The instructions that follow assume that you have set your system parameters to whichever port you are using.

Section 2. Wiz Pro Operation

This may be a little technical, but I feel that its better to say too much than too little, and I always like to know what a program is doing, and I think you do too.

Wiz Pro is a Basic09 program. It is always run in packed form. That means that it runs from Basic09's runb. When the shell sees that WizPro belongs to runb, it first loads runb then WizPro.

Once WizPro is running, it checks to find out if it has been given a filename parameter, and if not, it then tries to get one by placing the 'enter filename' screen up. This is accomplished by the WizStart proc.

WizStart attempts to read the file '/dd/com/pro/prostuff'. If it cannot find it, it creates it. (note that other programs can read and modify prostuff also, that's why WizStart reads it each time, to see if there are changes.)

Then the startup screen is drawn. This 'flexes' OS-9's muscles for things to come.

Next come the 'WizLoads'. WizStart contains the names of files that you want Wiz & OS-9 to keep in memory, but not in Wiz's process space. This way they can be executed right away when you want them. WizLoads can be changed using ProInstall. Note that all Wiz procs/extensions can be run from either memory or disk. (with the possible exception of WizPro itself, at the time of this writing there are bugs in GShell and Shell plus that prevent proper memory allocation under certain conditions.

At this point, if Wiz discovers that all is GO for doing an autolog, (calling another computer), it runs the proc WizAuto, or an alternative proc installed by ProInstall. WizAuto reads the autolog file and, using the information found there, dials the phone and 'logs on' to the host computer.

Whenever Pro returns from an extension, (including WizAuto), it checks to see if it should run another. This way extensions may call each other, or even themselves.

Once on-line with the host, what happens is up to you. Any number of other procs/extensions can be run before you quit Wiz. (if you actually quit, that is; many users just let it run all the time).

If you start the exit process, WizQuit is called. Here you are given the opportunity to log the session, start a new autolog file, return to Wiz, or to just exit.

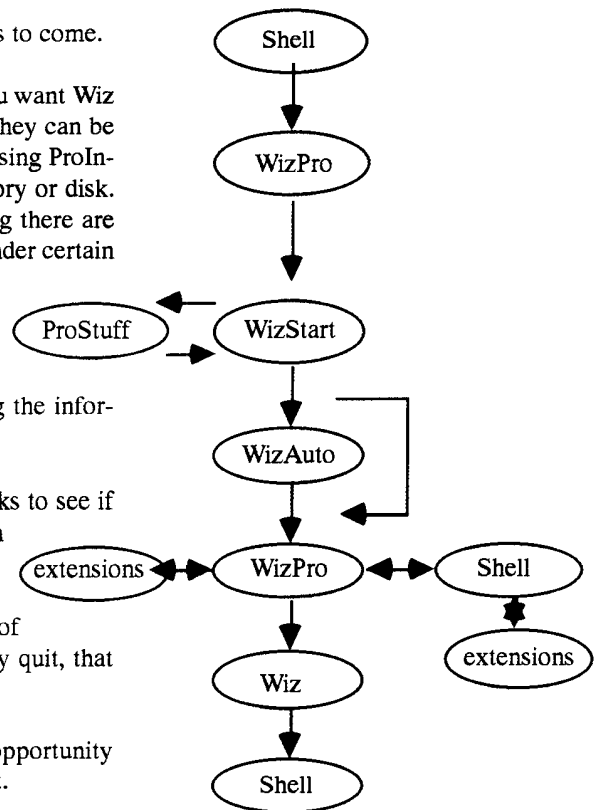
Before it exits, Wiz removes, (unlinks), the procs loaded from WizLoads, and sets up an 80 col text window. (the latter for Multi-View compatibility.. MV doesn't like closing back to the window descriptor.)

Each procedure on page 5 has its own operating instructions, of course, but some operations appear in several places. An example is the Wiz File menu, which will appear whenever you must select an existing file, usually when that file is to be sent to the host.

OK, let's run WizPro!



What happens when you run Wiz Pro:



From the Shell: type

```
'WIZPRO'
"WIZPRO ("hostname")
"WIZPRO ("fullpathname")
```

If you have shellplus, the ("") is not necessary.

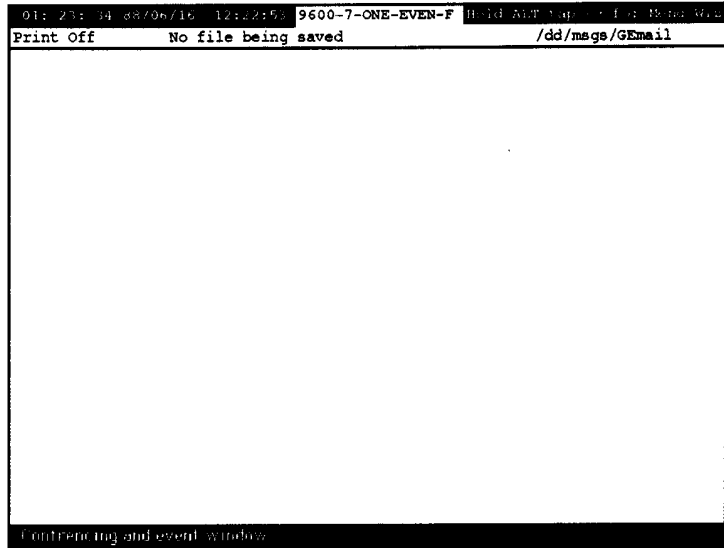
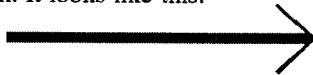
From Shell+: type

```
'WIZPRO -OR-
'WIZPRO hostname' -OR-
'WIZPRO fullpathname'
```

From Multi-View: Click on the WizPro Icon, or the Autolog file Icon.

Wiz Pro On-Line!

If you have NOT autologged, that is, not supplied Wiz with a host filename either from the command line or when Wiz requests it, the next thing you will see is the Wiz Startup screen. I won't show you a picture of that one, it's better in color anyway. Then you will see the Wiz Pro 'main' screen. It looks like this.



When you see the clock at the top left start updating, you are in the **on-line** mode. At this time the alpha-numeric keys, and control keys that you tap will be sent to the modem, and whatever comes from the modem will appear on the screen. The a-z keys that you tap while holding down the ALT key, and other special keys defined on the following pages, are all commands to Wiz Pro.

On-Line Keys

Once on-line, some keys talk to WizPro and its extensions, others go to the host computer.

WizPro itself has only 7 command keys:

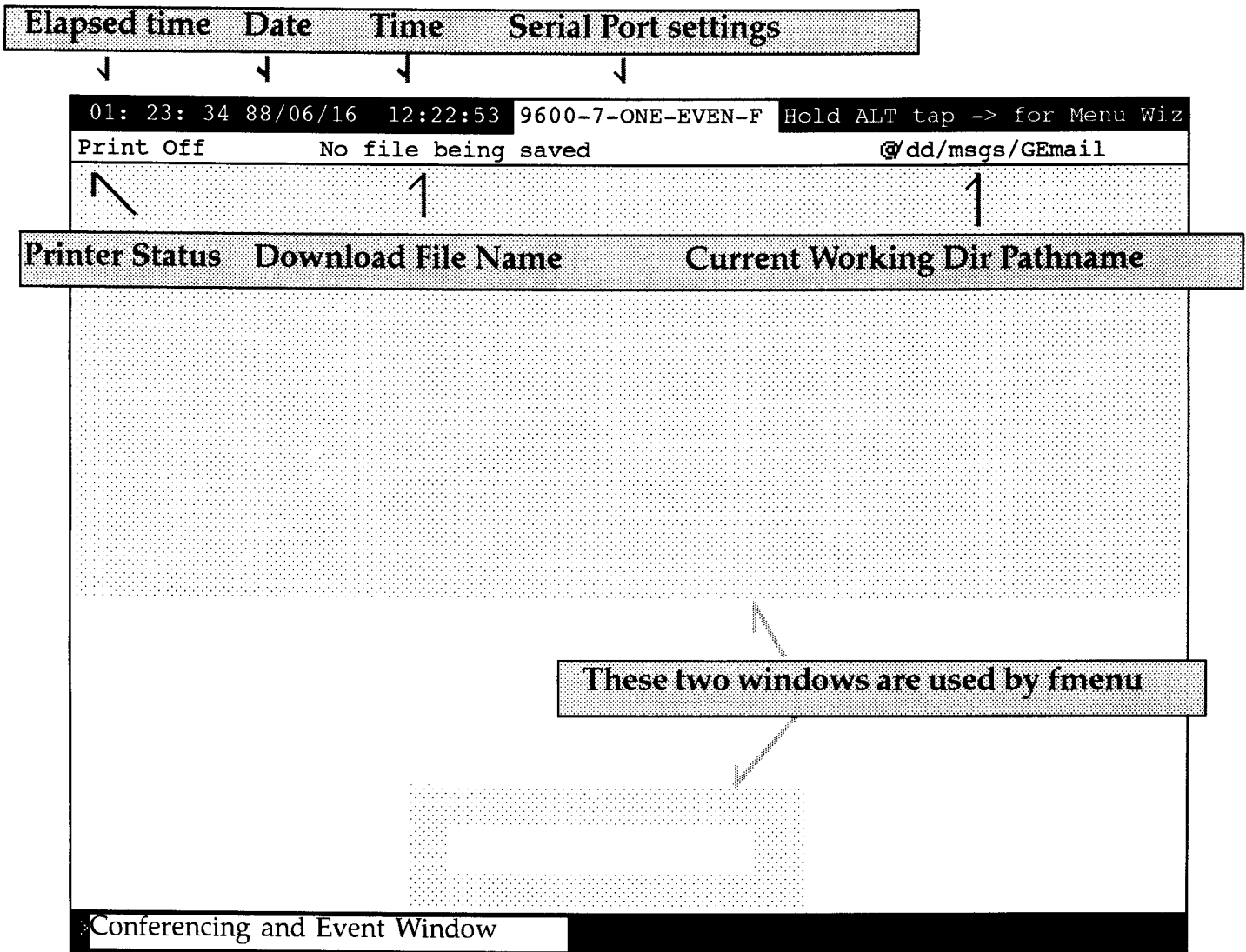
- MousePaws - Pushing the mouse button stops the host (& Wiz), until clicked again.
- Up Arrow - Call Clipper.
- Down " - Conference Mode (Toggle.)
- Left " - Enter Wait State.
- Right " - Put Menu on screen.
- Break Key - Will send whatever byte you have defined. Usually cntrl-O (15).
- Alt-Break - Sends, (creates), a true line break.

IMPORTANT!

There is a concept that you need to grasp to fully understand Wiz Pro. There are seven 'built-in' Wiz Pro functions that are basic parts of any good terminal program. But from this point Pro departs very markedly from other terminal programs, and many other types of computer programs as well. Here is the difference: None of the other capabilities are part of Pro itself. They are all 'extensions'. This concept is important because extensions can be removed and installed at any time. You can do this yourself using the program ProInstall. Now, an extension can be anything, you could add a text editor as an extension if you wish. However, there are certain extensions that Pro is a little more attached to than others. In the coming months you will see extensions to Pro that are not covered in this document.

The Wiz Professional main on-line screen

Below is a close up view of the main screen. It shows the three device windows: 'status bar', main screen, and the event window at the bottom. Also shown are the two overlay windows used by fmenu. (light shading) The large area at the top is where the directory of filenames is shown, and the bottom is the 'select' window.



If you have autologged with a file containing billing information, (#\$.00nn), the serial port settings display on the status bar will be replaced by the estimated running cost. (If you have autologged, your port has already been set). Note that the number shown here has the minimum billing increment included, but is not rounded to the next increment. This 'final estimate' is calculated when you <Q>uit.

The two lightly shaded windows are not part of the main on-line screen, but this is where they will show up.

Everything not on this page is defined external to WizPro, including the functions performed by the ALT A-Z keys in the menu.

Menu is an overlay screen, not a separate mode. ALT right arrow does two things:

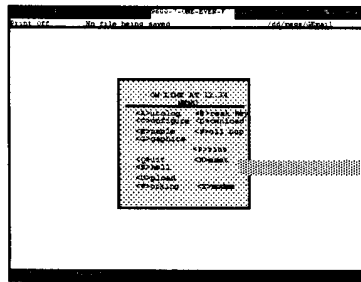
- 1) Shows those menu items that are currently installed.
- 2) Bypasses the need to use the alt key with A-Z keys.

The menu itself will disappear whenever any key is hit or a character comes in from the outside world. (This feature is great for when you are 'idling' or waiting in Conference for someone to show up. When something happens the menu will disappear!)

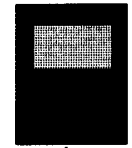
If the menu is not on the screen, you may use the alt-arrow keys or alt a-z directly.

So you have a choice: use ALT-(A-Z) or call the menu and tap the A-Z key by itself.

In the pages that follow, a brief description of each of these seven functions is presented. See "Operating Examples" for more information.



7 Keys:



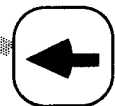
Pause



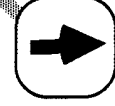
Clipper



Conf



Wait



Menu



STOP!

Clipper: (XPROC WizClipper)

The Clipper lets you back up and mark lines of previously received text which you can then save to disk, the printer, or send back to the modem. Clipper has some "special" features also.

While you are on-line, holding down the ALT key while you tap up arrow will run the clipper. Clipper has three modes "Edit, On-line, and Snap". Each mode is explained below:

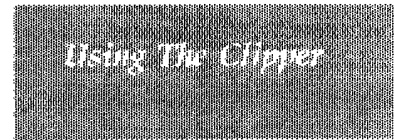
.....
Edit Mode: (no text marked)

- l= Load buffer from disk file
- t= Mark top
- b= Mark bottom
- o= Go On-Line, but stay in clipper
- q= Quit the Clipper
- g= Goto
- f= find
- shift up arrow = back one screen
- " dn " fwd " "
- alt up " bottom of buffer (oldest data)
- alt dn " top of buffer (newest data)

Snap Mode: (once a block of text has been marked)

- e= Eject printer page
- m= Snap to modem: send clip to host
- p= Print Clip
- s= Snap clip to disk
- u= Unmark text, return to edit mode

S



On-Line Mode: (you have typed "o" while in edit)

When you are in the Clipper On-Line mode, whatever you type on the keyboard goes to the host. What comes in from the host is placed in the buffer and on the screen below the buffer display. But you don't see any change in the buffer portion of the screen until you return to the edit mode.

When you run clipper, you always start out in the edit mode. From edit, you can go to the on-line mode by typing 'O', or to the snap mode by marking both the top and bottom of a block.

From on-line you can only return to edit. From snap, you can either return to edit or quit.

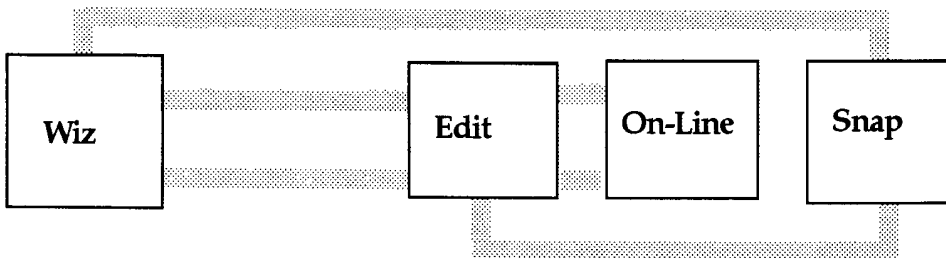
Clipper works on lines, (not characters), of 'straight text' only, such as created by Wiz Pro. If you load a file that has formatting information such as tabs or line feeds, Clipper will not be able to display the data correctly. This is why Clipper gives you the option to process loaded files before placing them in the buffer. If lines seem to be displayed on top of each other on the screen, re-load the file using the filter.

O

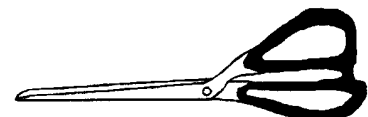
The Break Key will exit the on-line mode

Loading files:

If lines seem to be displayed on top of each other on the screen, re-load the file using the filter.



See the section on Operating Examples.



Conference

(also called buffered keyboard) (XPROC WizU-tils)



ALT-Down Arrow puts the Wiz in the Conference (Co) mode. A ">" will appear at the left hand side of the conference and event window. Anything you type will go into that space, but will not actually be sent to the host until you tap <enter>. ALT-Down Arrow will also take you out of the conference mode.

If you enter Clipper or Upload while in the Co mode, those extensions automatically use a "timed" send as though you are in the half-duplex mode. This is done because most host computers will "gag" a talkative conferee.

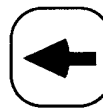
Unlike Wiz, Wiz Pro allows you to send control characters while in the Conference mode. Just hold down the CTRL key and tap the desired A-Z key. Ctrl-X is reserved, it does the same thing as OS-9 does on the command line: deletes whatever you have typed without sending it.

By the way, although you may only be able to 'see' 80 characters at a time in the Co window, you can actually enter and send up to 256 bytes.

You can use conference when you are in either duplex, or half-duplex mode. Conference can also be used as an alternative to half-duplex, and is very useful when you are communication with another terminal program vs a host. (like when you are chatting with a friends computer over the phone line.)

Because GENie don't like type-ahead, I use the Co mode there even though I run full-duplex.

Wait (XPROC WizUtils)



The wait mode does two things: it closes the serial path, thus freeing the ACIA, and then the Wiz executes a sleep 60 system call. Each second Wiz will wake and update the clock display at the bottom of the screen. It then checks to see if you have typed anything on the keyboard. If you have, Wiz exits the wait state. The wait mode is there for use with the level 2 windows. The usage is to place Wiz in WAIT and then use the clear key to go to another window. In this other window you could, for example, run a kermi.

Many users leave Wiz running all of the time.. and in the wait state when not being used.

What is Conferencing?

Most commercial computer services and some BBS systems allow you to join a meeting of other computer users. When you are in a host conference, all messages from each person in the same 'room' or 'channel' are sent to all other members there. It's sort of like a 'party line' telephone, or a CB radio.

Conferencing is not only enjoyable, but can be a way for you to get valuable information. But you must be able to generate clear concise messages and insert them into the conversation at the proper time. With most terminals and terminal programs, conferencing can be very trying because other 'people's messages tend to appear right in the middle of the line that you are carefully composing on your screen. The Wiz co mode does away with this problem by giving you a "private" window to type in. This happens in the event window at the bottom of your screen.

There is a special rule for the co mode. CTRL-X works just as it does with the OS9 shell. It will clear the event window, and reset the cursor at the left hand edge. (delete the line before you send it.)

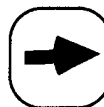
While conferencing it is nice to see your messages only once. If you are connected to the host in full duplex you may see everything twice, once when you hit enter, and again at the same time everyone else sees it. Try setting your host mode to no echo. The best result is when you only see your messages, (on the main screen), at the same time everyone else does.

A good use for your default macros, (the ones that you put into your autolog file), is to set one to the no echo command string for that host, and the other to the command string that sets the "nickname" or "handle" that you will use on that host. When you enter the co, just tap f1, <enter>, f2, <enter>.

Different conferences have different on-line etiquette, please follow it. One thing that is almost universally courteous: when entering a co, type a real quick Hi!, then wait and watch a while before typing anything else. And never exit without telling everyone that you are leaving, and giving them a chance to respond with a "bye" or "wait a minute, I have a question for you...".

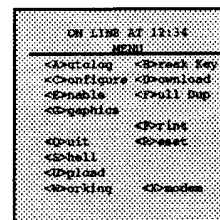
Also, be careful & courteous in what you say. Conferences are often recorded. The author has gotten himself in hot water more than once for becoming too "chatty". Remember, once you hit enter, it's too late to recall the message! Do not type in all uppercase; it is considered by many to be the equivalent of shouting.

Menu (WizPro)



To call up the menu overlay, tap RIGHT ARROW while holding down the ALT key.

To select a menu option, simply tap the key which you see in brackets, for example: <E>. Upper or lower case is OK. If you tap a letter not on the menu, hit the break key, or if something comes in from the host, the menu will simply disappear. While the menu is on the screen, characters will still be received if they come in from the host, so you can go to the menu at any time you wish.



Download and Print are Toggle options. That is, if you enter "P" and the print function is not active, it will be activated, if it is already active, entering "P" will cause it to become inactive. Note also that you can select both Print and Download options. (a strange mode but I know someone who does it every day!).

Stop! (WizPro)



Any time you tap the break key while in the on-line mode a special character is sent to the host. This character can be any value between 0 and 255. This means that you can send any value that can be put in an ASCII byte. (Many hosts accept a 15, Control-O, as an 'output off' command.) It is sent instantly, even when the data is coming in from the host. Note that this is a different operation from 'pause', which is a control-S. A control-O tells the host 'I don't want to see any more of this file', and a control-S says 'wait a while'. Control-S must be followed by a control-Q for the host to re-start sending where it left off. The control-O works by itself to simply abort the current operation.

Wiz itself also responds to the break key, when appropriate, and has its own 'pause', called Wait.

Note that ALT-BREAK sends true line break from the ACIA. What is a true line break? Well, in the old teletype days, it meant that the operator temporarily 'broke' the circuit, so that no current flowed. This got the attention of the operator at the other end because his teletype could not 'click'. The true line break is a hardware, vs a software function, and where implemented, gets the immediate attention of the host. Whereas the control-O or its equivalent says 'please stop', the true line break 'pulls the plug'.

When NOT on-line, the break key can be used to abort or cancel operations at many points. So the break key sorta switches gears depending on if you are talking to Wiz or the host at the time, but the effect is the same: If you get stuck, try the break key!

Note that either ALT-BREAK or ALT-SPACEBAR sends true line break from the ACIA.

Pause (WizPro)

Sometimes the host will scroll data off your screen before you can read it. To prevent this, you may tap the button on your mouse or joystick and the host will be paused immediately. Tapping the mouse button, or any other key, will continue. Pause sends a CTRL-S/CTRL-Q.



FKeys (XPROC WizFkeys)

Fkeys are extensions to WizPro. The first two Fkeys are F1 and F2. (ALT-1 and ALT-2) F1 and F2 send macros 1 & 2. Other FKeys could be made to send other macros.

FKeys are called by holding down the ALT key and tapping a non-letter key. (1-9,0,-,=,/, etc).

FKeys are contained in the procedure WizFkeys.

To Change a macro: First go to the Configuration (ALT-C), menu and type <1>. The current macro #1 will be displayed and you will be asked if you want to replace it. Type <Y>. A question mark will come up on the left hand of your screen. At this point, type your message. It must fit on the screen, or at most wrap around just to the question mark. When you are done, hit <ENTER>. You have two macros, so type 2 to do the second.

When you are on-line tapping F1 or F2 will send the text part of the macro, but NO CR IS GENERATED by Wiz. At this point just hit <ENTER> to finish off the macro.

In the conference mode you may want to send your messages to a specific person, say his name is Greg. Try this: make your Macro 1 look like this; "-> Greg ", then on-line, compose your message in the event window, but before sending tap F1, THEN <cr>. What will go out will be: "-> Greg message"

Both macros may be loaded from your autolog file. In the autolog file a macro looks like this:

```
#M1 This is Macro 1
#M2 This is Macro 2
```

Note the space between M1 and This; it is required.



You can use your macros as prefaces to the line you've been typing in the conference mode... just hit F1 then CR.

Wiz Pro File Menu.

Since Wiz Pro can run as either text or graphics, it cannot depend on things like file icons. But peering at a directory and trying to type in a filename can be a difficult task, even for experienced users. Wiz Pro has its own style of all text file menu.

When Wiz (or one of the extensions) wants a file name, the following happens:

In an 80x16 dialog window, the files within the current working directory will be displayed, 4 to a line, up to 88 entries. If there are more than 88 Wiz will notify you.

At the same time, a small double dialog window pops up. It is really a window within a window. Wiz talks on the outside, you type on the inside. The outside window is in the menu colors, the inside is in the main screen colors or reversed from the menu. This is the 'select' window, whatever is in it is the current selection.

Inside the select window will appear the first filename: '..'. Your action is to maneuver the filename that you want so that it appears within that inside window... and once that is done you tap <ENTER> and the file or directory is selected.

To help you accomplish this, Wiz provides you with several tools:

First, the **up and down arrow keys** will move you from one file to the next. Up arrow takes you towards the 'top' of the directory and down arrow towards the bottom, or the last entry.

ALT-Up takes to the top, ALT-Down to the bottom. (top and bottom of the portion displayed on the screen only).

Spacebar will take you to the first file that starts with a lowercase letter. If you follow the OS-9 convention of naming directories in upper case and files in lowercase this will allow you to skip the directories. You can also use the space bar instead of the down arrow in the case where the next file(s) are lower case.

If you just start typing a **file name**, Wiz will find the closest fit to that name and place it in the window. For example, if you have two files, 'bill' and 'bob', when you hit 'b' bill will be displayed. But when you type 'o', 'bill' will be exchanged for "bob".

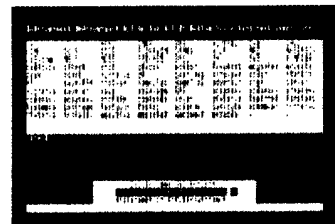
If you have more than 88 entries in that Dir, tapping '+' or ';' will cause more to be displayed. When you get to the end of the actual disk directory, empty slots at the end in the groups of 8 will be filled with "." If you get to the end, and click on one of these dots, you will go back to the top of that directory, even if you are three pages 'down'.

You do not, of course, 'send' a directory/subdirectory. However, you may 'click' on a directory name by tapping a <ENTER> while the directory name is in the window. If this is done, Wiz will go to that directory/sub directory and start the file selection process over again. The first two entries in each file menu will be '..' and '..'. You can 'click' on them in the same way. Thus you can quickly move up and down the directory tree structure of your disk.

You may also type a full pathname, just start it with a '/'. If you type '/dd', which is not a pathname to a file, but a directory, (device), Wiz will change to that directory. Typing /dd/com/pro/cis would select the file 'cis' in the directory /dd/com/pro, without changing your working directory.

Except for the actual reading of the disk directory at the beginning, all of these actions are instantaneous.

Use the break key to cancel a fmenu without selecting a file.



Section 3. Standard XPROCs SUMMARY

Autolog (XPROC WizAuto)

Autolog uses the TANDY Model-100 technique in the autolog file. A synopsis is as follows:

- ? Wait for a character (?P means wait for P)
- = Pause for 2 seconds
- ! Send a specific character (!= means send =)
- ^ Send the next character as control (^M=<CR>)
- < Means wait for carrier.
- > Means "end and go on-line" You MUST have this one.

example: "ATDT5551212^M<?U" means dial 5551212 then wait for carrier then wait for a U.

NOTE

If you have an autodial modem you may not need the "<". You may wait for a message instead, for example "CONNECT". However, this only works if you have your modem set to assert carrier detect all of the time. If you sometimes leave your CoCo on-line, running TSMON, it is better to use CD instead, so that the 'kill' feature of AciaPak works. In this case use '<'.

Making a new autolog file.

The first step is to make a directory on your default device (H0 or D0). You must use "/DD/COM/PRO". In that directory you will make your autolog files. You can use any text editor. You may also copy over the example files from the Wiz Pro disk and edit them. In this case you would simply edit in your local node phone number, and your password. Your first three lines in an autolog file may look something like this:

```
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
#b1#11#p5##
Host : cis Time off : 86/12/01 20:06:50 Elapsed : 01
```

This setup will cause a AT command set compatible modem to:

1. dial the number 3527500,
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
2. then wait for an R, (as RING RESPONSE)
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
3. and a C and a T, (as in CONNECT),
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
4. then wait four seconds (==),
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
5. send a control C,
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>
6. wait for U and :, (as in User:),
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>

<A>

You can autolog and re-autolog at any time. You can also <R>reset and autolog to a different host.

Many people never exit Wiz, they just put it in the "wait" state, and use the <R>reset option to clear the host name and do another autolog

Note the third line: "Host....", if you are making up an autolog file from scratch, you will need this "seed" entry. If you don't have it, you will get an error.

If you don't want to use /DD/COM/PRO, you can also type a full pathname for the autolog file: ("/dd/anydir/cis").

DD is simply another decriptor pointing to your main storage device. If you have a hard disk, then DD and H0 both point to it.

```

7. send 70126,267,
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>

8. wait for P and :, (as in Password:),
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>

9. send password, then exit.
ATDT3527500^M?R?C?T==^C?U?:70126,267^M?P?:password^M>

```

IMPORTANT!

Never ever send your password at any time other than logon. After you logon, if you should see a message requesting your password, either ignore it or hang up and completely re-autolog!

Warning, you must have the > to terminate the autolog sequence, even if there is no actual autolog sequence in the file.

The items following a # or \ in the autolog file will override the Wiz defaults. For example: #b1 means baud rate 1 (set the port to 300 baud).

Here are the settings and their meanings:

```

#B: Baud          #L: Length      #P: Parity
1 300 baud        1 7 bit words   1 NO parity
2 600 baud        2 8 bit words   2 ODD parity
3 1200 baud       3 EVEN parity
4 2400 baud       4 MARK parity
5 4800 baud       5 SPACE parity
6 9600 baud       2 2 stop bits
7 19.2 kilo-baud

#Hp = printer device "/p"
#Innn nnn= max line length before inserting CR in text uploads
#Dn n = secs The half duplex delay in seconds - the delay between
        sending text lines.
#T = If #T mode is TTY
        #V mode is VT
#E = break key value in decimal
#F = Full duplex
#O = Half duplex
#M1<space> = Macro 1
#M2<space> = Macro 2
#Xn = run n (A=1 Z=26) key as next proc after autolog
#$ = Billing rate of host per sencond ex:.000314
#* = Billing increment in seconds ex: CIS= 1 minute = 60
#Rn = Force CR at n characters
\B = boldface      \P = proportional  \N = Normal
\Cn n = palette # for Characters DECIMAL
\S n = palette # for Screen DECIMAL
\Fgb g = GROUP b= BUFFER in DECIMAL std=\F20001 (c801)
\command shell command, execute during autolog

```

If you want to always run the same extension (either an XPROC or an XPROG), with a particular host, use #K. It will be run right after autolog. (WizAuto).

If you want to run an XPROC BEFORE auto, use proinstall to set nextproc.

Note that either ## or \\ ("\" backslash NOT slash "/") terminates the configuration process. In other words, WizAuto reads until it encounters one of these "doublets".

Upper or lower case is allowed, all numbers are decimal.

When Wiz is autologging, if it is waiting for a character, it will continue without actually receiving the character if you tap enter. This is especially useful when the the first message or so gets garbled. Also, sometimes you can start your autolog all over again, without disconnecting: hit <@>, and "tap-enter" your way up to the point in the sequence where autolog and the host "get in sync".

<BREAK>

Break (XPROC WizUtils)

Using ALT-B or selecting from the menu allows you to redefine your break key. You set it to send the character that the host expects for a break or 'output off' command. Note that you enter the number in decimal and Wiz displays it in decimal. The break key value can be loaded from the autolog file.

Almost always the 'standard value' for your break key will be decimal 15, also defined as a CNTRL-O. (^O).

Perhaps a small discussion is needed here: When telecommunications were new, each bit was represented as an electrical current flowing in a wire. One level of current represented a "one", the other a "zero", but there was always current flowing. When an operator wanted to signal something special he would "break" the wire, NO current would flow. Today we use an ACIA , and it operates in a similar fashion to the "current loop", always sending either a "one" or a "zero" voltage level. But it can be made to send "nothing", or zero volts. This is called a "true line break" and is equivalent of "breaking the wire". Today we seldom do this, but use a special character to signal the "something special". The Wiz lets you set that character for whatever the host expects. Wiz can also send a "true line break":

Note that either ALT-BREAK or ALT-SPACEBAR sends true line break from the ACIA.

The break key value will be sent any time that you tap the break key while on-line.

It is safe to hit the break key at any time while running Wiz. If it is enabled it will usually abort whatever it is you are doing. If you are at a point where hitting break would not accomplish anything, nothing will happen. The same is also true of Shift Break and CTRL Break.

Confused?

Why is the break key talked about in two places? Page 17 explains how you use it. Here I explain how you change its value.

Configuration

The C will bring up the configuration menu. See section below for a description of the configuration menu, and its sub-menu "colors". After you change the configuration, you may save it for future sessions.

Configuration is rotating menu. If you autolog, you may not use it, since autolog calls configure every time it runs, but takes a "short cut" past the menu. Configure is intuitive, just keep tapping keys until you see the configuration you want.

Configuration Menu:

```

Configuration
*****
<B>AUD = 1200 <L>ENGTH=7 BITS
<P>ARITY=*SPC ONE <S>TOP BITS
<E>stimate Billing Cost
<H>ardcopy Path ..... /P
<D>elay after line.... 2 sec
<I>nsert CR after 255 chars
<?> Help      <1,2>edit macros
                <Q>uit
*****
<C>olors & Fonts menu
*****
C: █
    
```

The hardcopy path option is there for people that have more than one printer, one hooked to the serial port, and one hooked to a parallel port. Often these people name one /P and the other /P1. You can select which path is used by Wiz for clip-print and the print through operations.

The delay after line option is for the half-duplex mode of the <U>pload operation. See that option description under the main menu options above. The delay can be set for from 0-99 seconds. The delay is accomplished via an OS9 sleep call.

Note that XMODEM uses configuration to reset the port for 8 bit binary data, but does so automatically, bypassing the configuration menu. You DO NOT have to use the configuration menu to switch to 8 bits no parity to do xmodem. Xmodem also sets the port in the binary mode. (Turns off the text filter.)

Note that all items in the configuration menu can come from the autolog file, and will be saved to the autolog file if you select "Start new Autolog file" upon exit. In the autolog file all configuration menu entries are prefaced with a '#'.

<C>

A rotating menu frees you from the need to know all the answers. By tapping the letter in <> repeatedly, all of the options for that setting, or 'key', rotate through that spot on the menu. As soon as you see what you want, the selection process is done.

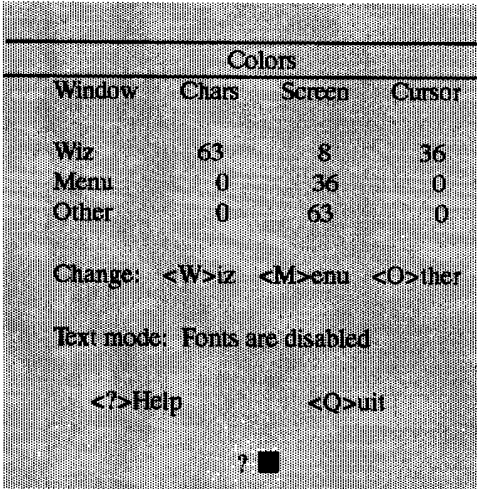
Some host computers limit the length of lines on text uploads. The <I>nsert option allows you to have Pro insert a CR after a specified number of characters.

Configuration changes that apply only to a particular host can be saved to the autolog file if you select "Start new Autolog file" upon exit. Changes that apply to all hosts may be saved by letting Configure re-write the defaults file 'ProStuff' when you quit the configuration process.

Colors and Fonts Menu

The colors and fonts menu lets you customize your screen appearance.

From the configuration menu, select <C>. The following menu will appear:



<C>

If you run in a graphics configuration, you have only two palette registers. Chars and Cursor use the same palette register. Changing either will change both the characters and the cursor. This also applies to the Wiz-Menu-Other settings; they all use the same two PRNS. The net result of this is that changing any changes all, and the last change is the effective one.

This menu doesn't rotate like configuration, except that Tapping the <C> key will cause the Wiz Chars color to advance by one, <S> does the same for screen, this is a left over from Wiz I, handy when you don't know what color number you want.

To change the Wiz Main screen colors tap <W> and enter the new colors for the characters, the screen (background), and the cursor. the change is immediate if the window is visible.

When you type <Q>uit and also <Q>uit configuration, you will be asked if you wish to save the changes to ProStuff. If you do, the colors you have selected become your new defaults. To change colors (main screen only) for a particular host, use the <S>tart new autolog file when <Q>uitting WizPro itself.

If you are running in a graphics screen, you may also enter a new font buffer number and group.

What is a toggle command?

A toggle is like a push-button switch that you press once to turn ON and a second time to turn OFF. This means that you press ALT-D to turn download on and ALT-D again to turn it off.

Download (XPROC WizUtils)

<D>

Selecting option <D> turns on the built-in text recorder. This is a toggle option similar to <P>. After typing <D>, you will notice that a prompt appears: Save Received File as: "?". At this point you may enter the name that you wish to assign to the file. Note that this may be a complete pathname, for example /d1/text/message.from.joe or just a filename. If you just type a file name it will go in the directory shown at the upper right of your screen. The name of the active download file is displayed in the center of the status bar, on the second line.

There is no limit to the size of the file you can save this way, except for the space available on your disk.

Once recording is started, it is transparent to the other Wiz Functions. For example, you do not have to turn off download in order to do an upload. Recording may not take place, however, while you are in an extension.

<E>

Enables (XPROC WizUtils)

Certain Extensions have a special ability to run automatically if a special character or sequence of characters, (trigger), is received from the host. You may have several of these installed in your version of Wiz Pro, but only one at a time may be enabled. If you choose the enable option, you will be given a choice of which of these will be called if a trigger occurs. Note that Enable also arms the trigger for one-shot. The extension can, however, rearm the trigger each time it is called, or arm the trigger on its own account. Presently no extensions to Wiz Pro use triggers directly.

<F>

Full/Half Duplex (XPROC WizUtils)

Most host computers automatically echo back each character as you type it. But if you are typing, and not seeing anything on your screen, try HALF duplex. If you see two characters for each one you type, try FULL. You can also toggle duplex while on-line by using ALT-F.

Holding down the ALT key while you tap F will toggle you back and forth between full and half duplex. What this really means is that you don't have to go to the menu to change this function. ALT-F will also update your status bar.

HALF or FULL duplex can be confusing. Here is a picture that shows how it works:

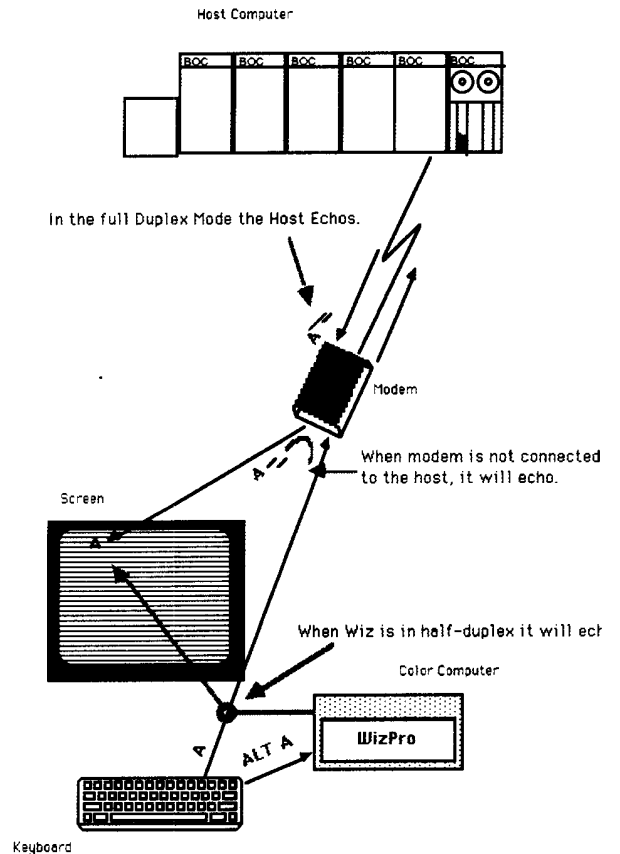
HALF-DUPLEX MODE

The drawing shows where the Wiz HALF duplex mode makes a short circuit and also where the modem and the host echo.

The modem will make the short circuit if you have it set up for the command mode echo.

When you see double characters, it is because HALF duplex is working and either the modem or the host is echoing. You usually won't see triple, because the modem does not echo if it is actually connected to the host.

There may be a delay when the echo comes from the host. There-



fore many people will set their host default to no echo, and use a local, (HALF duplex mode), echo. This makes for less annoying typing and editing, but what you see on your screen may not be exactly what the host has recorded.

<H> <I>

Help/Info (XPROCs WizHelp, WizInfo)

The status bar has a message to remind you that ALT-H will display several help screens as a reminder of special key use while on-line. Info will show you some of the internal Wiz settings. You may not understand it all now, but it is interesting anyway. Hold down ALT and tap H for on-line help. A help screen will appear in the overlay normally used for the menu. There is more than one screen. Tap any key to continue to the end of the help screens.

Help and Info are expected to be two of the most often modified extensions to Pro.

<P>

Print (XPROC WizUtils, WizPro)

This option allows you to print messages as they are received from the host, on your printer (often called "print through"). It does not matter if your printer is fast enough to "keep up" with the baud rate, because Wiz Pro causes the Host to "wait" by sending it an X-OFF character before printing. When you first select this option, the message Print ON will appear at the top of your screen on the Status Bar. This message will remain as long as the printer is active or until you select another option. If the printer is already active option "P" will deactivate it and show the "Print OFF" message from the top of your screen.

Wiz Pro prints only when its buffer rotates. Since the buffer is 10k, and your printer may also have an 8k buffer, it may be a while before anything actually prints.

<Q>

Quit (XPROC WizQuit)

You will be asked if you want to save the session record. If you answer y or Y, you will be then asked if you want to use the same file that you used to autolog. The time you logged off, the elapsed time, and the host name will be recorded. If you have included billing information in the autolog file, the estimated billing for this session will be recorded also.

\$\$.\$\$

If you have included billing information in the autolog file, the estimated billing for this session will be recorded also.

By using the autolog file, the next time you use autolog to call that host, this info is re-displayed in the event window. If the host follows suit, (displays "last on" message), you can do a quick visual compare. You can also use this time record to reconcile your usage with any list you may get from the host. You can periodically delete session records, (using a text editor), to keep your autolog times short, or use the Start New Autolog file option during exit. Note that if you "save to another file", it must be an autolog file; and you can only "start new autolog file" if you autologged when entering Wiz.

Some of the Wiz parameters, such as screen color, font, baud rate etc., are stored in the autolog file. "Starting" a new autolog file will record the changes. This way, when you call that host again, the color etc. will be restored.

..... **<R>**
Reset (XPROC WizUtils)

The <R> option lets you reset your ontime, like for example when you call your local BBS and the phone is busy for 10 minutes. It also resets the host to NO FILE; you can use this when you want to call a new host without exiting WizPro.

Reset will also send a reset string to the modem. You can change the contents of this string using ProInstall. Reset also drops DTR.

..... **<S>**
Shell (XPROC WizShell)

The <S> option takes you back to the shell, in case you want to do a "dir" while on-line. You can use the break key here to "break" out of a long operation. The advantage of using Shell over just clear keying over to another window is that the shell will be run in the current data Wiz directory.

Shell runs in a plain 80x24 text screen. If you ever get an overlay screen stuck, (and I hope you never do), you can get rid of it by running a shell because WizPro closes all windows before calling shell.

..... **<U>**
Upload (text) (XPROC WizSend)

This option is used to send a text file from your computer to a host. When you tap ALT-U, you will be prompted for a handshake character, (if you are in FULL duplex). This character can be a carriage return (just tap ENTER), or any single character. You can also use a timed send. To send a message to cis, you could follow the following procedure:

```
Host sends  Function:
You type   L
Host sends  To:
You type   Bill Brady 70126,267 CIS
Host sends  Subject:
You type   On The Air with the Wiz
Host sends  1:
You type   <ALT-M> to get to menu .
You type   <U> and enter ":" (for handshake)
```

If you use a <CR> or single character handshake but the host does not actually send a <CR> or the handshake character, Wiz will send the next line anyway after about 2 seconds.

Upload uses a text filter, it will not send control characters if they are imbedded in the file.

Upload will insert a CR on lines exceeding a certain value. This can be changed via ProInstall or Configure. The default is 255.

If you are in the Full Duplex mode, the first prompt is: Handshake, Enter Character or <CR> to use CR. What you do here depends on how the host computer operates. In the example above the host sends you the short "1:" when it's ready for a line. The 1 is the line number. In this case you would want to enter a ":" as the handshake character. Other hosts use a ">" or nothing. If nothing, then just tap enter to use a CR as the handshake character. You are actually selecting the character that Wiz will wait to see echoed back from the host after each line it sends, before it sends the next. If you are in the half

duplex mode, the host may not send anything when it's ready for the next line. In this mode, Wiz simply waits for a pre-set amount of time, (see the configuration menu), before sending the next line.

Remember, most host programs were designed in the belief that the operator, (you), would be typing this text on a keyboard. Many cannot accept the data as fast as Wiz could send without either the delay or a handshake.

If you have any files in your current data directory named snap.n, the Upload extension will find out if you want to send these. It will display the first two lines of the file on the screen so you can make sure it's the file you want to send, then prompt you for a Y or N answer. See "Using Wiz". If you answer "N" or <ENTER>, Wiz will call the Wiz fmenu (see section 2).

Note that if you are in the HALF duplex mode the host does not echo back anything. Wiz has no way of telling if the host received the last line ok, so it simply delays for a while after each line is sent. You can set this delay from the Configuration Menu.

After you have composed your replies to the snap messages, you will want to send them to the host. You can do this by calling the host and using the <U> option from the menu for each snap file.

Working Directory (XPROC WizUtils)

<W>

You control which data directory that the Wiz uses. The default is whatever directory you are in when you start Wiz. This is indicated by "<@> Current Data Directory" displayed on the right hand side of the status bar. To change, tap ALT-W and enter the new directory name. Always use /dd or in the path to make sure you get back to the "root". If you start a new autolog file on exit, this data directory pathname will be saved. Note that this feature, along with the /DD/COM/PRO default for autolog files, means that you can run Wiz from any point in your directory structure. You never have to do CHD before running Wiz!

NOTE
fmenu also changes the working directory.

XModem (XPROC WPXmod)

<X>

Wiz has xmodem as a standard extension. If you type X from the menu, or ALT-X while on line, Xmodem will prompt you for: <U>pload, <D>ownload, <S>atellite Mode <T>ext upload, <Y>modem download or <Q>uit. The quit option is there in case you entered xmodem by accident. <S>atellite mode allows a longer xmodem delay to accommodate packet switched relays via satellite links. The Up and Download functions are to send and receive files from the host. You do not have to set any special port configuration before using xmodem, it is done automatically for you.

Xmodem can be used for either binary or text files.

If you select the U option, xmodem will ask you if you want to disable CoCoBin. The CoCoBin format is always best to use for xmodem file transfers since it includes a special block at the beginning that tells the download xmodem program the attributes to use for the file, its size, and the date last modified. This way the receiving xmodem will know how many blocks are coming. Another benefit is that the xmodem fill bytes will be stripped so that you do not have to process the file through any "filter utilities" before you use it. CoCoBin is especially important for sending Font files. Ymodem/CoCoBin II will automatically send any icon and AIF associated with that file.

The operation of the xmodem is fairly straight forward. If you upload with the text option, line feeds will be inserted after each carriage return found in the file. While xmodem is operating, the event window becomes a status indicator (graphics configuration only). It will "disappear" as the transaction becomes complete. On download this feature is only possible in the CoCoBin mode, and the file must have been uploaded in the CoCoBin format.

Note that when xmodem displays "block #" it is talking about the block # contained in the xmodem header. When it displays "blocks", it is giving you a number of blocks sent or received. They are not always the same. Xmodem block numbers are limited to 0-255, while Wiz can receive or send any size file. So these numbers will agree up to 255, then the block number will recycle to 0 while blocks keeps on counting. These numbers may also appear incorrect to you when using the T option, (due to the inserted linefeeds), and CoCoBin, (due to the extra block). The numbers indicate the count generated as a direct result of the data file only.

Xmodem automatically detects CoCoBin, and 1024 byte, (1k), blocks. It also automatically requests CRC checking, but will operate in the checksum mode.

When downloading, Wiz always creates a temporary file called 'xmodem.file1'. It will ask you to assign a new name when the transfer is successful. If you never receive the first block, no file is created. With YMODEM/CoCoBin II, the file will automatically be renamed... before it is created.

You can de-archive on the fly. Once a download is started, go to another window and rename the file from 'xmodem.file1' to 'file.AR' then type 'ar -x file'.

To YMODEM download a group of files, first tell the host which files to send, the go to XMODEM and type 'Y'. Pro will then work with the host to download each file until all are on your disk.

If your host requires lfs, (line feeds), you can tell Wiz to send Xmodem text, it will insert a LF after each CR found in the file, otherwise you can use <U>. In <T>ext mode, CoCoBin is not an option.

XMODEM Block numbers always go from 0-255. The block number is not the same as 'total blocks'.

Wiz uses a fast parallel CRC calculation method as proposed by Greg Morse, the author of XCom9. Thanks Greg!

Presently GENie supports YMODEM. When you see the prompt 'Enter Download Request' just type all the file numbers seperated by commas. For example 3810,3004,2874,2700,2610. Then choose 'Y' from Pros xmodem menu.

Note : *True YMODEM is batch. Some hosts call XMODEM 1-K 'YMODEM', but if it is not batch, it is not YMODEM according to the YMODEM spec. (Batch means that you can transfer more than one file at a time).*

Operating Examples

Exactly how you use Wiz depends on the host computer that you are using it with and what you are trying to do. This manual cannot possibly tell you how to interface with the many BBSs and services that exist. But I can give you some examples of the things that I do with Wiz Pro.


Once or twice a day I check mail and messages on the various services. If Wiz Pro is not already running I just find any unused window (with the clear key, I usually run 4 immortal shells: [shell i=/w7&] in my start-up file.) Then I type: 'WizPro cis'. (I use shell+ so no ("") required).

CIS

Pro then calls 'cis', (Compuserve Information Service), and logs me on. If I have mail waiting, cis will take me to easypex; if not, directly to the OS-9 forum. If I have mail waiting, I may, or may not, read it right away. When I read it, I may respond to it while on-line and that's the end of that. Or I may either download the mail using the <D> option, or snap it to disk or the printer using Clipper. Lets say I am expecting a rather large text file from someone:

If I have not started reading the file already I will type ALT-D and Pro will ask me for a name. When I enter the name Pro opens a file and puts its name on the status bar: 'Downloading to:'. I then tell cis to 'read'. When the file is completely listed I simply type ALT-D again and the file is closed. Pro then puts a 'no file' message back up on the status bar.

If I have already started reading the file, I may hit the 'break' key to tell cis 'output off', then re-read and download as in the above paragraph. If the file is not too long, I may just let it list through to the end and then use the clipper:

Using ALT-Up Arrow, I call Clipper. Clipper then puts up a 12 by 80 window and displays the last few lines that cis sent. I use the up arrow  key and put the cursor on the last line of the message. Then I tap 'b' to mark the bottom of the message. Using up arrow again, perhaps shift-up arrow, I continue backing up until I have the cursor next to the first line of the message. Then I tap 't' to mark the top. If I make an error somewhere in this process, I tap 'u' and start over. (if Clipper detects certain errors it will do the <u>mark for me.)

At this point I have a block of lines 'marked' in the buffer, highlighted by Clipper. Depending on what the message is, I will save it to disk, the printer, or both. To save to disk I simply tap 's'. I usually override using the next snap number, and type a filename

when clipper asks. To save it to the printer, I tap 'p'. If I want the message to be on the paper by itself, I will also tap 'e' to eject the page. (form feed).

After reading and saving, I may want to reply to the message. I have a choice here: I can <q>uit Clipper and go back on line, or I can reply while still in the Clipper. If I choose the latter, (say I want to reply while viewing the message), the first step is to unmark the text I have highlighted. I do this by typing 'u'. I then use the arrow keys, or find, or goto to get the part of the text I want to view into the Clipper window, then I tap 'o' to go on-line. When I see the cursor in the bottom...'main screen', I know I am now talking to cis again. I then hit enter to get cis to give me a menu and select 'reply'. When cis says its ready, I begin typing the reply.

But I may want to 'quote' something out of the sender's message. (say he had a question, and I want to repeat the question, then type my answer). I will start my reply as in the paragraph above... I might say: 'In response to the following question"', then tap break to get back to the Clipper edit mode, mark his question, and tap 'm' for snap to modem. That clip is now inserted into my reply. Next I tap 'o' to go back on line and continue: 'my answer...'

At some point I will get to the OS-9 forum. Here I may have some messages waiting. I may choose to either read then, or read non-stop. If I do the latter, I keep the mouse handy to pause the screen at any point where I want to take a closer look. I usually un-pause with the spacebar.

I may next visit the data library. If I see a file I want to move to my computer, I tell cis to 'download' and choose the CIS B protocol. I then type ALT-T to call STERM to do the download. (I have STERM installed in my Wiz Pro menu.) CIS-B protocol is nice because it is fast and takes care of the filename. It doesn't handle ICONs and AIF's though... YMODEM/CoCoBin II will do that when it is available.



If I have a file to upload, I will use XMODEM. First I tell cis that I want to upload file 'example'. (upl example.bin). I then choose xmodem from the cis menu, and tap ALT-X. At the Pro Xmodem menu I will select 't' if I am sending a text file, (cis likes lf after cr, others don't), or 'u' if I am sending a binary file (or a unmodified text file). At this point Xmodem needs to know which file to send so it runs Fmenu. Fmenu opens a window and shows me 88 file names out of my current directory. If I am send

ing text, the filename that I want is usually in there. So I start typing the filename. By the time I get to the second letter Fmenu has usually found the file I want and placed it in the select window. I then tap 'enter' and the Xmodem starts sending the file.

If I am sending a binary file, however, it is not generally in my 'current data directory'. In this case I may simply type the full pathname: '/dd/cmds/programname', or I may use Fmenu to change directories. If I want to go 'up' a directory, I select '..' and tap enter. If I want to go 'down', I select the sub-directory name and tap enter. (by 'select', I mean I get the name into the small 'select' window at the bottom of the screen.) I tend to use the auto-search feature most, but I use the spacebar and arrow keys also.

If I decide that I really don't want to send anything after all, I tap 'break' to get out of Fmenu, and 'break' again to abort xmodem. Xmodem itself will tell cis that I aborted by sending CANcel. (some hosts don't respond to CAN, you may have to hit CTRL-C or CTRL-X to cancel the host xmodem). Next, I will tell cis 'bye'. When cis hangs up, I tap ALT-Q to call WizQuit. Here I will log the session <y> and return to Wiz, <y> again. To prepare Wiz to autolog to a new host I tap ALT-R. (I always use the 'x' to reset the modem here).

If I forget to do the reset, when I tap ALT-A, Wiz will attempt to call cis again. If this happens I quickly tap 'enter' then '@' to abort the autolog.

Calling WizAuto from 'already running' Wiz results in Fmenu again, this time showing the contents of /dd/com/pro. Here I tap 'u' for usersgroup, the file I use to log on to Delphi. Fmenu automatically puts usersgroup in the select window and I tap enter. On Delphi I do similar things to those I did on cis.

Delphi

On Delphi, however, impromptu conferences often occur. I may get called to a conference, or just join one, but once there the first thing I do is tap ALT-C to get me into the conference mode. Then I type /nam Bill which appears in the one-line 'event' window at the bottom of the screen next to a > which appeared there when I tapped ALT-C. Next I tap 'enter' to send /nam Bill to Delphi. /nam Bill will then disappear in the bottom window, and when Delphi echoes, it will re-appear in the main screen. (and my name will be Bill).

Often I will get into conversations with two different people. In which case I set up my two macros thus: First, I tap ALT-C to get to the configuration menu. There I tap 1 for macro 1 or 2 for macro 2. Config

will then put up a window, display the macro, and ask if I really want to change it. I tap 'y', then type in the new Macro (>>>Joe), followed by 'enter'. I then <q>uit Config and Config asks me if I want to save the changes. I hit <enter>... which means 'no', because I don't want to change my default values. Back on line I type a 'message to joe' into the bottom window. I then tap <F1>, <enter>. Delphi sees: >>>Joe 'message to joe'<CR>. Joe sees >>>Joe 'message to joe', as long as F1 is first.

While in this conference, I may want to again 'quote' the message that I received while over in cis. (a few paragraphs back). First I use ALT-Up Arrow to call Clipper. I use the arrows, or find to see if the quote I want is still in the buffer. If not, I choose 'l' and via Fmenu, find the file that I snapped while I was on-line with cis, and load it into the buffer. I then mark my quote and tap 'm' to send it to the conference. Clipper knows we are in Conference and so uses half-duplex time delay mode automatically. (to keep us from getting 'gagged'). Once again I tell Delphi 'bye' and reset Wiz and the Modem. Here I have to reset my modem twice, because Delphi (telenet), doesn't hang up the phone, and I have to get rid of CD, because I use '<' in my autolog file. (see section 4).

GEnie

Again I tap ALT-A to autolog onto GEnie. On GEnie I download using Xodem 1K, but only for longer files. I use xmodem (128 byte) for the smaller ones. While on GEnie, rather than use Wiz Pro Half-Duplex, (GEnie will run full duplex but 'likes' half better), I use the Conference mode. This has the advantage of giving me full type-ahead at all times, which GEnie itself doesn't always allow. GEnie doesn't know the difference because nothing is actually sent until I hit CR. I use this trick on other services also. GEnie's structure encourages the use of mail rather than messages. If I have created a letter off-line, and now wish to send it:

First I get to GEnie mail: 'mov 200'. Once there I select 'type text letter', (little do they know, <grin>.) At the 1:> prompt I tap ALT-U to call WizSend. Send then asks what character to use for handshake and I tap '>' as in '1:>' above. Then, again via Fmenu, I select the file to send, and Send transmits it to GEnie, waiting for '>' after each line. I can see it go on the screen, and sometimes it looks like characters are missing, but they aren't, everything is getting sent but Send puts the emphasis on transmitting first, and the screen second, if the host accepts the data fast, Send skips the screen part. This way my local screen updates don't slow down the transfer. Send tells me on the status bar what it is sending and from what directory. Once Send is done, I may add a postscript, or just type *s. *s tells GEnie, I'm done.

Section 4. CUSTOMIZING WIZ PRO

In what ways you can add to or change Wiz Pro?

1. Make a new Fkey
2. Add an extension
3. Edit the WizEquates

~~~~~ Making new Fkeys.

When you tap one of the non-letter keys: 1-9 and !@#\$%^&*()_+{}:"<>?,.,) while ALT is held down, Pro will run WizFkeys with a ctrl.Mkey set to the decimal value of the key. (F1=177). WizFkeys gets the full parameter set, so a FKey can do almost anything. The primary difference between FKeys and ALT-A-Z extensions is that FKeys do not show up in the menu.

New FKeys are created by editing the proc "WizFkeys". The key is decoded by examining the variable cntrl.Mkey:

```
IF cntrl.Mkey=n THEN
```

Where n=decimal value of key..

~~~~~ Installing a XPROC or XPROG :

You can either run ProInstall or modify the ProStuff file directly; TYPE menuentry. There are 26 slots available. To install a new entry in one of these slots first decide which slot is to be used. Each corresponds to a keyboard alpha key a-z.

If you modify the WizEquates file, you must edit up to eight (8) fields:

mnu.sho - a 16 character string. This is the string that will be displayed in the Wiz Pro menu.

mnu.pro - a 16 character string that is the name of the procedure that Wiz Pro will RUN if the extension is invoked.

mnu.param - a 32 byte string that will be sent to the extension if called with PA=1 or PA=2 or PA=3.

mnu.PA (types) - A single byte that defines the parameter set type that will be passed to the procedure when RUN, and the window type.

mnu.funct - a single byte that can be used by extensions in any maneuver desired. It is primarily included for recursive calls.

mnu.trig1, mnu.trig2 - single byte triggers. See below.

mnu.trig 3 - a two byte entry that defines the "front end" trigger for invoking the extension. If these two bytes are detected by the port driver, the procedure automatically will be RUN. Only one trigger is active at any time, but the active trigger for a session may be selected in the autolog file.

Be careful about removing anything that uses WizUtils!

ProInstall has an 'other' category that lets you change other things like whether you run in text or graphics screens, the files loaded by Pro at startup, and the modem reset string.

Quick Reference Chart to Making Changes:

To Change:	Temporarily,(this session)	for Each Host	Default
Screen Type	--	--	ProInstall
RS-232 Port type	--	--	ProInstall
Serial Path	--	--	ProInstall
Modem Reset String	--	--	ProInstall
Installed XPROC	--	--	ProInstall
" XPROC	Use Shell ALT-S	--	ProInstall
Baud Rate / Parity, etc.	ALT - C	ALT - C then ALT- Q and <S>tart New File	ALT - C then Save ProStuff
Billing Estimate	ALT - C	"	"
Colors (ANY)	ALT - C	"	"
Font / Boldface	ALT - C	"	"
Printer Path	ALT - C	"	" - ProInstall
Half-Dux Delay	ALT - C	"	"
Macros	ALT - C	"	"
Line len before inserting CR (text upload)	ALT - C	"	--
Log on To A New Host	ALT - R - ALT - A	--	--
Working Directory	ALT - W	Fmenu or ALT-W then <S>tart New File	Current working directory
Autolog Directory	Type full pathname at OS-9 prompt	--	default fixed in code at /DD/COM/PRO
Autolog Shell Command		Use text editor, edit autolog file	- no default
Create New Host autolog file	--	ALT - N NewHost and <S>tart New Autolog File	- no default but you can create a new autolog file from scratch with any text editor
Break Key Value	ALT - B	ALT-Q log then <S>tart new file	
FKeys	--	--	Edit WizFkey Source

Section 5. Building WizPro

There are many many options here. You can leave procs on disk, load them in memory one at a time from shell, let Wiz load as needed, let Wiz pre-load them from WizLoads, merge & load, etc.

The simplest thing to do is to copy all the procs into the cmds dir and just run. This is clean, but slows you down when running. Most people will want to do a little Wiz Pro construction.

Construction guidelines:

GtKey, S9scall, WizCsum, & WizCRC should be merged either with runb or WizPro.

WizPro is supplied with GTkey, S9scall, Wizcsum, and WizCRC merged.

If two or more external procs are merged: Don't exceed 16k (16384), bytes of program size per merged file.

If you use WizLoads, the merged file **MUST** have the same name as the first proc in the set. (Like the OS9 Shell file.) So a file containing WizUtils and WizSend should be called WizUtils.

You can have WizPro pre-load up to 8 files, just put their names in WizLoads.

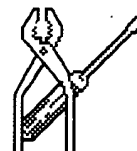
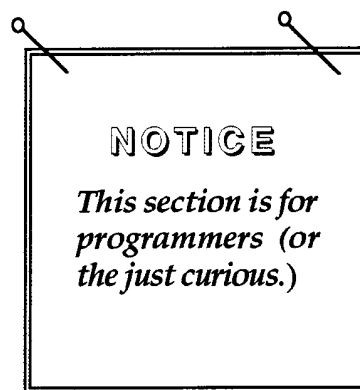
WizPro will also unlink these on exit. If a load is a merged file, make sure the first proc in the file has the same name as the file.

ProAcia and MWP should be placed in your bootfile. You should make a new boot. I put ProAcia just in front of Aciapak myself.

LOOK OUT!

Look out for patches. Some drastically alter the way OS-9 handles interrupts. Some of these are patches to 'clean up' glitches during animations, others are clock driver patches. **WIZ, AND THE COCO, HAVE BEEN TESTED REPEATEDLY WITH HIGH SPEED COMPUTERS UNDER ALL CONDITIONS UP TO 9600 BAUD... THEY DON'T LOSE CHARACTERS UNLESS SOMETHING IS WRONG WITH THE SYSTEM!** At 19.2k you can lose characters for any number of reasons, including cable length. 19.2k is the very edge of the rs-232 bandwidth.

BTW, Gtkey is the same as Inkey and S9Scall is the same as syscall. Pro uses these specially renamed versions to prevent memory mapping conflicts.



Technical Discussion

Wiz Professional was written using Basic09 and assembly language on a Tandy Color Computer 3 under OS-9.

Wiz Professional uses three OS-9 utilities, rename, s9scall and gtkey, and two machine code subroutines: WizCsum and WizCRC. Wiz Professional also makes extensive use of many OS-9 system calls.

Wiz Professional also has it's own device driver for the ACIA in the rs-232 pak called ProAcia. MWP uses the base address of FF68., MWP.Disto FF54. ProAcia is interrupt driven on receive and uses the FIRQ generated by the ACIA via the CART line, which is changed to an IRQ by the GIME. Once initialized, ProAcia runs all the time, collecting bytes coming in from the host.

A word about speed. First, line speed is solely dependent on the time allocated to interrupt processing in your hardware and software environment. ProAcia's operation is independent of the Wiz. ProAcia handles the first level of protocol with the host, it talks to the host, and the Wiz plays "catch-up". OS9 itself takes some time away from the Wiz, and, of course, OS9 is multi-tasking, and will also share the available time with other tasks. Wiz actually sleeps a minimum of 4 ticks per port read.

ProAcia sends X-OFF to the host at critical points depending on its current x-off mode. Usually it sends an x-off when its buffer is near full. (ProAcia has a 1032 char buffer). Wiz can also request that ProAcia send an X-OFF via a setstat call. ProAcia never responds to in bound x-off's. No "S" type driver driver should both send AND respond to x-off, since it has no way of detecting its own x-off if it should be accidentally echoed back to it. (for T type drivers this is OK, since there is a human operator "out there").

ProAcia does not use interrupts or a buffer for writes (sending). This results in a smaller driver and reduced memory requirements at run time, but also has a side benefit of reducing interrupt handler overhead. The result is that, on a CoCo 3, ProAcia will run at 19.2 k baud, and is totally reliable with Wiz up to 9600. Above 9600, reliability will depend on factors like the interrupt environment in a particular machine, how the host handles X-OFF's etc. Note that, if you run another program, in another window for example, that disables interrupts, ProAcia will experience ACIA overruns.

More than one disk path may be open at the same time, one by the <D> option, another by autolog, xmodem or during a SNAP. If you are doing a <D> option the filename is displayed on the top on-line screen line, or in the menu.

In the FULL duplex mode, the <U>pload option sends a line then waits for a CR from the host, if the CR is not received within about ten seconds, the next line will be sent anyway. In HALF duplex it sends lines with a delay after each line. This delay is accomplished via an OS9 sleep call.

XMODEM cancels (aborts), transmission by sending a string of CAN's, whether or not the host responds to this depends on the host. It responds to the same if sent from the the host. There is no real standard for aborting xmodem transmissions.

The autolog feature of Wiz Professional has not been made "smarter" for two reasons: In my experience, no autolog program is smart "enough", they all fall down in some circumstance. The second reason is because the really good smart autologgers are very S-L-O-W, even on very powerful computers. The simple "watch and tap and restart" technique possible with Wiz Professional works with the minimum of hassle.

Wiz Pro is actually Basic09 with dozens of fast machine code sub-routines...OS-9 System Calls

Why Basic09? Well, originally, it was written in Basic09 just to show that it could be done; to demonstrate the speed and power of the language on a color computer. In fact, for large applications such as Wiz, the Basic09 version is actually smaller at run time, even when you count runb. Basic09 never manipulates the incoming data to the screen, or to disk. It sets up a buffer which OS-9 writes to and reads from. Basic09 also shines in string handling and parameter passing. Basic09 is powerful indeed.

How WizPro invokes an extension:

Each menu entry corresponds to one of the 26 alphabetical keys (A-Z). The procedure associated with a particular key will be run by Wiz Pro.

When the Menu is on the screen: When that key is pressed by itself, or at the same time as the shift key, (Capital), or at the same time as the ALT key. When the menu is not on the screen, the ALT-key must be pressed at the same time as the alphabetical key.

Wiz will then RUN that procedure, passing the parameter set and window type specified in the mnu(key).PA in the WizEquates file. If the procedure uses the Wiz parameter set, ie is a PA=4, it will have its own access to all of the Wiz parameters, buffers, etc.

Stand alone programs are called with a PA of 1,2 or 3. They are forked in the same manner as the shell does redirection. That is, a 'dummy' window is opened, its paths duped, and the program run in that window. When the process so created dies, control is returned to Wiz.

The only difference between PA 1, 2 and 3 is the window used. 1=80 col text (VT80), 2=40 col graphics (VT40... Multivue), and 80 col graphics, (gwin).

Procs run with a PA of 4 are called in the Wiz Windows, no particular window is set for them since they are passed all the variables and can make their own.

The currently active triggered procedure, as enabled by the user, will be RUN if the active trigger is received from the host. In this case the flag.mnu will be set false.

PA	Run Via	tmode	Window	params
1	Fork(params)	old	VT80	mnu.parm
2	Fork(params)	old	VT40	mnu.parm
3	Fork(params)	old	gwin	mnu.parm
4	RUN(Wizparams)	Current*	Current*	Type 4
		Wiz	Wiz	

*If mnu(n).funct=0 close all Wiz windows before running, even if PA=4.

Stand alone programs, such as STerm, or Multi-View accessories are run specifying a PA of 1,2 or 3, depending on what type of screen they require at startup. These programs are forked as a child process with re-direction, and with the path to the serial port closed.

Programs and procedures that can make use of Wiz-Pro resources are executed from the Basic09 RUN statement, and the entire WizPro parameter set passed. These are installed specifying a PA=4.. They are run in the Wiz standard window set unless they specify a funct=0, in which case all windows are closed prior to the call.

WizPro calls XPROCS with ProAcias buffer sized at 256 bytes. XPROCS which resize for a larger buffer must reset to 256 bytes.

WizPro calls XPROGS with ProAcia de-inized, (closed), and will reopen path when it's execution is resumed.

How to run an XPROC before WizAuto.

If WizPro is passed a parameter directly from the command line, it calls paths.nextproc, passing that parameter. The default for nextproc is WizAuto. Using ProInstall, nextproc can be changed. The xproc MUST exit with mkey set and flag.ex set if it wants to then call Autolog.

Remember that certain menu slots are reserved, see the end of the parameters definitions.

How to Run an extension directly from Autolog.

If Autolog encounters a #Kn within an autolog file, it will automatically set the proc associated with that menu slot to run upon exit. (flag.ex = true) During the autolog process, mnu(k).sho will be displayed in the event window to alert the operator.

For example: if the file 'cis' contains a #KT, since T is slot 20, and the "T" slot of the menu is configured for Sterm, Sterm will be run as soon as autolog exits after completing 'cis'.

Automatic Extension Operations

Extensions can be made to automatically initiate certain operations without operator intervention. Set flags.ex to tell Wiz to run, cntrl.MKey to identify which, and cntrl.Estep.

Xmodem	Estep=1	Download	
	Estep=2	Upload	mnu(x).parm=filename
	Estep=3	CoCoBin Upload	mnu(x).parm=filename
	Estep=4	Text Upload	mnu(x).parm=filename

If nextproc<>" and Estep>0 then set flags.ex upon exit to return control to caller.

Wiz Pro Parameters, their meaning and use.

This section discusses the meaning and use of Wiz Pro's Variables.

List the source file "Type4" to examine these types.

Each data type will be discussed.

The pseudo procedure TYPE4 is the complete Wiz Pro parameter set. It is what a proc gets if it is called and mnu(proc).PA=4. Use 'Type4' as a starting point for any Basic09 Extensions that you write.

Paths

```
TYPE wpaths=sp,spa,wpa,dpa,ppa:BYTE; siop,piop:STRING[3];
nextproc,ho,rxfile,host:STRING; spd(32),oldesc(32),newdesc(32):BYTE
DIM paths:wpaths
```

When passed from Wiz, these variables will contain the current information on the Wiz Paths. To see if a particular path is actually open, look at flags.

sp= path to ProAcia
spa= path to status, (top.. device window) screen or 'bar'
wpa= path to bottom screen or bar.
dpa= path to disk file, if open
ppa= path to printer, if open
nextproc= the extension run by WizPro if a filename parameter is given, usually WizAuto.
ho=host pathname, example: /dd/com/pro/GENie
host=host filename, example: GENie
rxfile=current/last download filename
spd= serial path descriptor, used to change/indicate port configuration.
oldesc= std input path descriptor when Wiz was called. Do not modify.
newdesc= current std path descriptor use to change pause echo etc, if you like.

Examples:

```
put #paths.wpa,cntrl.cls - clears the bottom window
put #paths.sp,bufs.mac(1) - sends macro 1 to the modem
siop= DD name "mwp" - piop= "p" or "p1"
ho="/dd/com/"+host - host= host file name - (ho is full pathname)
```

Calls

```
TYPE
calls=ststat,gtstat,cursr,screen,reed,wrt,wrtln,creat,sleep,pid,setpri,pag,bau,typ,pau,
alf,ech,eor,endof,quit,intrpt,opt,bsp,bso,bse:BYTE
DIM caw:calls
```

Various system calls and path descriptor offsets. Defined in Wiz Equates. Use if you like, don't change except in equates.

Paths

```
(* see if the serial port
has any characters
s.a=paths.sp\s.b=caw.opt
RUN s9scall(caw.gtstat,s)
(* s.y now contains the
count
(* read 'em
s.a=paths.sp
s.x=ADDR(bufs.buf+bufs.in)
RUN s9scall(reed,s)
inc=s.y
(* write 'em to screen
s.a=paths.sp
s.x=ADDR(bufs.buf+bufs.in)
s.y=inc\bufs.in=bufs.in
RUN s9scall(caw.wrtln,s)
```

CAW- OS-9 System calls.

```
(* sleep 10 ticks
s.x=10
RUN S9Scall(caw.sleep,s)
```


*Flags***Flags**

```

TYPE
flg=acia6551,TEXT,gmode,roll,bufroll,chdf1,vt52,vtansi,OLOPEN,spopen,WPAOPEN,SPA
OPEN,PSP,BLDF,conf,pfl,dfl,exit,ex,dup,mu:BOOLEAN
DIM flgs:flg

```

acia6551=TRUE most of the time - false for 6850.
 TEXT=TRUE Wiz is running in text (type2) screens.
 gmode=TRUE Wiz is in raw graphics mode
 roll=true buffer has just rolled over - buffroll=true it has rolled at least once
 chdf1=TRUE we have changed working directories. (may be re-defined)
 vt52=TRUE we are in that mode. Only one device window open (not used in Pro)
 vtansi same
 OLOPEN=TRUE indicates that an overlay is open please set true when you open an overlay,
 false when you close. I use (in all exits):

```

IF OLOPEN
put #1,cmds.owend\OLOPEN=FALSE
ENDIF

```

spopen=true the serial path (sp) is open

WPAOPEN=true (wpa) is open - SPAOPEN (spa) is open

BLDF=true we are using boldface - PSP=true we are using condensed fonts

conf=true we are in conference mode

pfl=true printer path (ppa) open dfl=true disk path (dpa) is open - use, do not modify.

exit=true normal exit from external proc. If you exit with this flag set to FALSE a shutdown
 will be executed. Without calling WizQuit.

ex=true request execution of another proc. Is used in conjunction with nextproc. Place the
 name of the proc you wish to run in paths.nextproc and set flags.ex before you exit. WizPro
 will then run that proc. It will be run with the same parameter set as the exiting proc. You
 may modify WizPro parameters that you know the new proc will get, and so control the ac-
 tions of your second proc. Ctrl.Mkey and mnu(n).funct are two that can be safely used by ex-
 tensions, but there are others as well. See CTRL.estep.

dup=true we are in full duplex

mu=true you were called from the menu =false you were called by a signal

*Ctrl***Control character variables.**

```

TYPE con=esc,CR,lf,BS,BL,home,cls,ptabl,lbrk,BK:STRING[1]; Mkey:BYTE
DIM cntrl:con

```

Most are self explanatory. lbrk=alt-break BK=break key value set for that host.
 estep = also used in conjunction with flags.mu, paths.nextproc, etc. Can be used by an ex-
 ternal proc to set a flag to itself or to a secondary proc.

Mkey = call modifier. Used by wiz to tell external proc which option to do for those that have options. Mostly used by Utils. You may set Mkey to some value and use with next-proc (to specify external proc name, and flags.ex to trigger.) ALT-A=Mkey of 1, ALT-Z=Mkey of 26.

Cmds

Commands short strings

Handy set of predefined, mostly window, commands. Just PUT #1,cmds.owend to close any overlay.

```
TYPE commands=dwend,dwsel,owend,defcolor,ulon,uloff,pal,hdds:STRING[2];
DLpos,PRpos,ATpos,br,bufnum,group,revoff,rev,prop,propoff,bold,boldoff:ST
RING[3]; four,palette:STRING[4]
DIM cmds:commands
```

owend, dwend, dwsel, defcolor, ulon, oloff, bufnum, group, rev, revoff, bold, boldoff, prop, propoff - are pretty much self explanatory. See WizEquates if in doubt.

pal= is the first two bytes of the palette command, the part that don't change.

hdds= string value of the current setting for half-duplex-delay-seconds. The delay to be used between sending lines to the host. In seconds 0-99. Good idea to use this in Conference mode also. (WizClipper uses the delay automatically if you enter Clipper while in the Co mode and snap to modem).

PRpos= cursor xy of printer message on status bar. Line 2 pos 0

DLpos= " download " " " 16

ATpos= " data directory " " " 30?

put #spa,PRpos\PRINT #spa,"Printer OFF"; (always use ';' in small device windows, or use PUT or use write, not writeln, else your message will 'disappear', or scroll right out of the window)

br= the string version of cntrl.BK (Break key sends value.)

four= four carriage returns, better than PRINT\PRINT\PRINT\PRINT

Buffers

```
TYPE buffers=buf(10240):BYTE; klin:STRING[256]; abuf(256):STRING[1]; mac
(2):STRING[80]; bsiz,absiz,in:INTEGER
DIM bufs:buffers
```

buf=main Wiz buffer.

Klin= buffer (used in Conference mode); use for anything you like, but not if we are in Conference mode. Please clear it when you are done.

abuf(128) last autolog sequence... may be increased to 256 bytes. (later)

mac(s) the two 80 char macros.

BuFs

bsiz= 8192 the size of the main buffer

absiz= 128 size of abuf

in= the offset to the last character in the main buffer. Or number of characters in the buffer, depending on how you like to think about it. An empty buffer= Bufs.in=0 and bufroll=false. bufs.in=0 is illegal offset, Wiz runs in BASE 1. - bufs.buf(1) is legal- bufs.buf(0) will get you an error 55.

Win

Windows long strings

Wanna open a window? Try put #1,win.flat, or PRINT win.flat;

```
TYPE windows=flat1,flat,stat,bottom:STRING[9]; vt80,vt40,gwin,main:STRING
[10]; menu:STRING[63]; mes:STRING[42]; tmenu,tmes:STRING[15];
ask,prmt:STRING[9]
DIM win:windows
```

Device windows:stat,bottom,vt's,gwin,main see equates for sizes

Overlays: flat,flat1,menu,tmenu,tmes,ask,prmt.

flat and flat1 are both 80-cols. One overlays the top half of main, the other the bottom. Use for entering macros, in clipper, by fmenu, etc.

menu=graphics type menu screen, complete with border. tmenu - text only version, no border

mes & tmes (message window) - smaller version of menu in 'other' colors. Use for things like: "Your Disk Just Died".

ask & prmt together make the porthole or select window for Fmenu. usage:

```
PRINT win.ask;" What do ya want";win.prmt;cntrl.cls; flags.OLOPEN=TRUE
INPUT a$
PRINT cmds.owend;cmds.owend; flags.OLOPEN=FALSE
```

PRNs - Colors Self explanatory. Use to build palette cmds, of course:

EXAMPLE:

```
colors.fore=63
cmds.palette=cmds.pal+chr$(PRNs.forePRN)+chr$(colors.fore)
PUT #1,cmds.palette
```

Palettes registers

PRNS

```
TYPE
PRNeq=forePRN,backPRN,menuforePRN,menubackPRN,otherforePRN,otherbackPRN,curPR
menucurPRN,othercurPRN:BYTE
DIM PRNs:PRNeq
```

Color Numbers

COLORS

```
TYPE cnos=fore,back,curs,menufore,menuback,menucur,otherfore,otherback,
othercur:BYTE
DIM colors:cnos
```

Font Group Buffers

FONT

```
TYPE fx=one,two,GRP,BFN:BYTE
DIM font:fx
```

Syscall variables (S)

S

```
TYPE regs=ccode,a,b,dp:BYTE; x,y,u:INTEGER
DIM s:regs
```

Strings Misc Strings

STRG

```
TYPE strings=ramf,nofont,nod,mdmclr,ddir,crtc,aktc,pnr:STRING; ontime:STRING
[8]; mdmdly,hh,mm:INTEGER
DIM strg:strings
```

mdmclr=modem clear string
ddir=current data or working directory
crtc="Carriage to continue... aktc="Any key.....etc. (see Equates)
pnr="Path not ready.....
ontime= the time we started this session
mdmdly=time delay for modem clear in ticks
hh,mm hours, minutes= version of ontime elapsed this session
te2= seconds of day at logon
ramf= Ram full or proc missing measge
nofont=Font error
nod=No carrier message

PORT

Port Parameters (PORT)

```

TYPE portset=bb,pp,ss,ll,sbaud,stype:BYTE; baud(7):STRING[4]; bcmd
(7):BYTE; PAR(5):STRING[4]; SB(2):STRING[3]; word(2):STRING[1]; pcmd
(5),scmd(2),wcmd(2),ppmax:BYTE; typcmd(2,3,2):BYTE
DIM port:portset

```

bb=1=300 baud etc.

MNU

Menu Entry

```

TYPE menuentry=parm:STRING;sho:STRING[16]:PRO:STRING
[12];pa,funct,trig1,trig2:BYTE;trig3(2);PARAM mnu(26):menuentry

```

Example:

mnu(24).sho="<X>modem" "\mnu(24).PRO="WPXmod" \mnu(24).pa=4

An extension with PA=1 will set shell line params in mnu.parm, see page 36.

mnu(n).funct 0 = run after closing Wiz windows

Reserved Menu Slots

The following slot numbers are reserved, ie, cannot be used by any except replacement procedures:

- mnu(1)=WizAuto... autolog xproc
- mnu(21)=WizSend... text upload xproc
- mnu(24)=WPXmod... xmodem xproc

ProAcia Reference

Summary of System Calls and Signals for WizAcia and ProAcia

SETSTATS

```
$28 ComStat Call set new port configuration
128 Set work buffer size; Y=size-1
130 Set EOB signal count; Y=count-1
132 Enables Driver to send x-off on buffer near full
133 Disenables Driver " " "
134 Request from application to send true line break
135 Enable text mode filter
136 Enable binary mode (8-bits no parity, no filter)
137* Toggle DTR off for 1 second
140 Application requests x-off condition
141 Application requests x-on condition
142 Enable VT52 filter
143* Enable trigger 1
144* Enable trigger 2
145* Enable trigger 3
146* Enable eob trigger
147* Disable all triggers
```

GETSTATS

```
ss.rdy Standard system call #chars returned in B
128 Reserved Diagnostics
129 s.rdy # chars returned in Y and B
130 return acia status register in B
131* deleted use 130
```

DEVICE DESCRIPTOR (MWP)

```
PD.NUL (7)* Filter bypass character (this character bypasses
filter)
PD.PAG (9)* Filter 0=No filter FF=Text filter
PD.BSP (10)* Single byte signal trigger 1 (signal 128)
PD.DEL (11)* " " " " 2 (signal 129)
PD.RPR (14)* Flow-control (x-on/x-off) 0=off $FF=on
PD.XON (25)* Byte 1 two byte signal trigger
PD.XOF (26)* Byte 2 two byte signal trigger (signal 130)
PD.BAU Same as AciaPak
PD.TYP Same as AciaPak
```

Notes:

- 1) * New items for ProAcia - ProAcia was previously referred to as "version 8"
- 2) ProAcia defaults to binary mode with x-off enabled
- 3) Setstat 130 sets a count, which when reached, causes an OS-9 Signal 211 to be sent to the application. (intercept with Basic09 ON ERROR GOTO.)
- 4) SetStat 132 ProAcia sends x-off at buffer 1/2 full and 3/4 full if host ignores first. WizAcia has 256 byte buffer. WizPro sends xoff at bsize-160 and bsize-80. bsize=1032

- 5) SetStat 135 Text Mode filters all chars below 31 except CR BS Bell & CLS.
- 6) SetStat 136 Immediately puts port in binary mode (disables text filter and xoff, Com-Stat required if not already in 8 bit mode.
- 7) SetStat 140/141 Allows application (Wiz) to send x-off/x-on to the host, via the driver, without confounding driver and host logic.
- 8) The getstat ready call returns number of characters available in . B is only accurate if you have set a working buffer size of 256 bytes or less. (Working buffer size and B reg use are included for older applications.) Use #129 call for larger buffers. ("Y") reg.
- 9) ProAcia does not have a VT-52 filter.
- 10) Numbers in () are paths.SPD() values.

TRIGGERS

ProAcia can be configured to detect three different triggers of two types:

- a) One byte Trigger(s): If that single byte is received from host a signal is sent to the application. (signals 128 & 129)
- b) Two Byte Trigger: If byte 1 is received a flag is set, if byte 2 is received with the flag set then signal 130 is sent to application, if flag is set and any other than byte 2 is received then flag is reset.

Triggers may be used to detect protocol switches, xmodem starts, identify requests, and VT/ANSI control sequences. They eliminate the full-time need for the application to examine incoming data on a byte for byte basis.

Triggers are active in text mode or binary mode but must be enabled via setstat 144-146, and may be disabled via setstat 147. They are one-shot, and must be re-enabled.

The caller may receive signal 211 (eob) at a time when there are several characters backlogged in the buffer that precede the trigger character in time.

Signals for triggers 1,2 & 3 are sent only when a read of the port is in progress, not when the character is received. The signal will occur at the precise point when the character is read. The next read will return the trigger character.

SIGNALS

- 211 End of block.. WizAcia has accumulated the number of characters specified in PD.DUP & PD.PSC
- 128 Single Byte specified in PD.INT detected
- 129 " " " PD.QUT detected
- 130 Two Bytes, specified in PD.XON and PD.XOFF detected
- 141 (Wiz, not ProAcia) Mouse Pause Signal (toggle).
- 244 A Buffer overflow has occurred

Signals are "caught" in BasicO9 by the ON ERROR GOTO.

Section 8 If you have problems

Here are some hints and tips gained over the past two years. Each item is followed by causes, a discussion if appropriate, followed by fixes/diagnostic work-arounds.

RAM FULL? Proc missing? - in the upper left hand corner on starting up Wiz.

Too many modules packed into one file. (more than 16k worth).

WizPro, s9scall, Gtkey, WizCsum, and WizCRC not packed together.

One of the Wiz modules got "stuck" in memory because you 'loaded' it.

The 'e' attribute bit not set on a file. (xproc)

Error 216 when autologging.

No COM or COM/PRO directory.
You must have a COM directory even if there is no file in it.

No autolog file by that name.

Dropped characters. (some characters appear on the screen)

RS-232 pak/ACIA not performing at the CoCo-3 2MHz rate.

Unmodified multi-pak.

Incorrect port configuration. (Try 7 bits space parity)

Software is disabling interrupts

Some rs-232 paks contain s6551 acia chips. I have been told that this chip is spec'd for 1 MHz. The s6551A, however, is spec'd for 2MHz.

If the multi-pak is not modified for the CoCo-3, it appears that some interrupts "get lost", possibly due to address problems. This problem seems worse with the "small" MPI.. Some of the small MPI's appear to have more problems AFTER the mod, suggesting an improper hardware mod.

Try borrowing an rs-232 pak from someone. If the borrowed pak works try the following: If your Pak contains a plug-in ROM.. remove it, if it is soldered in, try replacing the s6551, (use a s6551A). If the 6551A don't fix the problem, and the ROM is soldered in, you may, if you possess the skill, remove the ROM anyway. (I have seen this fix the problem). Note that the ROM is the smaller of the two "big" chips. If you still have problems.. look at your "rock". (the rock is the crystal, the squarish "can" laying in its side with two leads coming out of it.. & sometimes a third wire from the top of the "can" to ground.) If the rock is of the smaller type, (3/8" or so square), and you have a large one, (1" or so square), then try the larger rock. (this has been known to work also in cases where smart modems "ignore" the CoCo commands, (but properly echo characters), at higher baud rates.. (1200+). If you have an unmodified MPI, or none of the above has worked, borrow a MPI if you can and try it. If you have a "Y" cable you might try that also, although I have not seen a "Y" cable work when directly plugged into a CoCo 3. (Some users have reported success in using "Y" cables.)

If the host port configuration is not the same as the CoCo, characters can be lost due to parity errors. Try different port configurations via the configuration menu.. 7

Kevin Darling's PMAP utility is provided on the Pro Disk. See the PMAPS below.

PMAP WizPro in 'WAIT'

ID	01	23	45	67	89	AB	CD	EF	Program
1	00	15	05	01	02	03	04	3F	SYSTEM
2	07	08	Shell
3	10	12	13	14	09	0A	RunB
4	16	08	Shell
5	0B	08	Shell
6	0C	08	Shell
7	0E	08	Shell
8	0F	08	Shell
9	20	0D	..	PMAP
10	23	24	25	21	Arc
11	26	0D	..	PMAP

Blocks reserved for XPROCs to run in WizPro

PMAP WizPro running small XPROC (help)

ID	01	23	45	67	89	AB	CD	EF	Program
1	00	15	05	01	02	03	04	3F	SYSTEM
2	07	08	Shell
3	10	12	13	..	16	14	09	0A	RunB
4	20	0D	..	PMAP
5	0B	08	Shell
6	0C	08	Shell
7	0E	08	Shell
8	0F	08	Shell

PMAP WizPro running large XPROC (Wpxmod)

ID	01	23	45	67	89	AB	CD	EF	Program
1	00	15	05	01	02	03	04	3F	SYSTEM
2	07	08	Shell
3	10	12	13	1D	1E	14	09	0A	RunB
4	16	08	Shell
5	0B	08	Shell
6	0C	08	Shell
7	0E	08	Shell
8	0F	08	Shell
10	24	0D	..	PMAP

See the Memory directory on the next page. You can use PMAP and DIRM to identify 'RAM FULL-invaders' in Pros space if they should occur.

bits space parity works for me with all hosts. Remember, Wiz, like most host programs, automatically switches to 8 bits for xmodem. YOU DO NOT HAVE TO RUN WIZ IN 8 BIT MODE JUST FOR XMODEM TO WORK.

Characters go out, but don't come in.

You know this when you are in full duplex, typing "against" the modem only, (not connected to a host), and you can see the TD and RD lights BOTH blink when you hit a key, but nothing appears on the screen.

Caused by interrupts not making it back to the GIME chip.
No CD, (carrier detect).

There should be two jumpers in your rs-232 pak. One is the interrupt line to the "CART" of the CoCo. If these jumpers are missing or cut, try reconnecting if you can. Some paks were apparently made without the jumpers, or they may have been cut earlier. Note that many terminal programs before Wiz, many CoCo DOS programs, did not use interrupts and so work fine without the jumper. If this is not the problem, try the MPI... I know some people that have had to run the interrupt to the CoCo itself with a separate wire... they just could not get the MPI to work. Under most circumstances, Wiz Pro will warn you when there is no CD. You will see a message at the upper right hand corner of the screen. Try an xmodem download... if there are no errors, its a bad configuration, if there are SOME errors, its a bad data line. (note that "data line" here includes the modems on both ends...you may have a flaky modem).

GShell disables interrupts for excessive periods of time. If you switch in-and out of the Multi-View window, characters will be lost if they are coming in. Perhaps this problem will be fixed, it has been identified and it is unnecessary for GShell to do this.

"No modem CD you cannot receive" appears in the bottom window, but you receive characters anyway.

CD line not properly terminated, ie is open. (note this can cause crashes when operating a remote terminal).

Modem defective, or improperly configured.

Typing alt-c alt-c will get rid of this message.

ERROR 55 OR 67 during autolog.

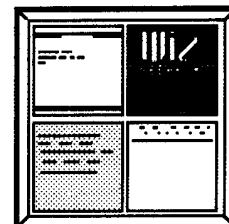
Bad variable in hand prepared autolog file. Use Wiz Pro to re-make file by WizQuit's <S>art new autolog file option.

A Wiz Pro Memory Directory						
Bk	Ofst	Size	Ty	At	Link	Module Name
-3F	D06	12A	C1	81	0	REL
3F	E30	1D0	C1	81	1	Boot
3F	1000	ED9	C0	88	0	OS9p1
-1	700	CAE	C0	82	1	OS9p2
1	13AE	2E	C0	81	1	Init
1	13DC	1EE	C1	81	2	Clock
1	15CA	9F3	C1	81	1	IOMan
1	1FBD	122E	D1	81	135	RBF
2	11EB	6FF	E1	85	135	WDDisk
2	18EA	36	F1	81	5C	H0
2	1920	37	F1	81	0	H00
2	1957	36	F1	81	79	DD
2	198D	36	F1	81	0	H1
2	19C3	36	F1	81	60	D0
2	19F9	36	F1	81	0	D1
2	1A2F	36	F1	81	0	D2
2	1A65	36	F1	81	0	D3
2	1A9B	5E3	D1	81	1E	SCF
3	7E	C36	E1	81	16	CC3IO
3	CB4	1D4B	C1	81	1	WindInt
4	9FF	CE5	C1	81	0	VDDInt
4	16E4	45	F1	81	6	Term
4	1729	45	F1	81	4	W
4	176E	45	F1	81	1	W1
4	17B3	45	F1	81	1	W2
4	17F8	45	F1	81	0	W3
4	183D	45	F1	81	0	W4
4	1882	45	F1	81	0	W5
4	18C7	45	F1	81	0	W6
4	190C	45	F1	81	0	W7
4	1951	45	F1	81	3	W8
4	1996	45	F1	81	3	W9
4	19DB	45	F1	81	3	W10
4	1A20	45	F1	81	3	W11
4	1A65	45	F1	81	0	W12
4	1AAA	45	F1	81	0	W13
4	1AEF	45	F1	81	0	W14
4	1B34	45	F1	81	0	W15
4	1B79	45	F1	81	0	W16
4	1BBE	439	E1	82	4	ProAcia
4	1FF7	3C	F1	81	4	MWP
3F	33	471	E1	82	3	ACIAPAK
3F	4A4	3F	F1	81	4	T2
3F	4E3	17A	E1	81	1	PRINTER
3F	65D	3C	F1	81	2	P
3F	699	14B	E1	80	0	FT
3F	7E4	1F	F1	80	0	FTDD
3F	803	219	D1	81	0	PipeMan
3F	A1C	28	E1	81	0	Piper
3F	A44	26	F1	81	0	Pipe
3F	A6A	1AE	11	1	1	CC3Go
-6	0	1FFC	C1	81	1	GrfDrv
-8	0	1887	11	82	7	Shell
8	1887	A5	11	81	0	Del
8	192C	36B	11	81	0	Dir
8	1C97	84	11	81	0	Display
8	1D1B	4F	11	81	0	List
-9	0	2F99	11	81	1	RunB
-D	0	68	11	81	1	Merge
D	68	42	11	80	0	FF
D	AA	11D	11	81	0	Rename
D	1C7	1F8	11	81	0	PMap
D	3BF	2E7	11	81	0	Copy
D	6A6	1516	11	81	0	Edit
D	1BBC	7A	11	81	0	Iniz
D	1C36	301	11	81	0	Tmode
-14	0	1E10	22	81	1	WizPro
14	1E10	5E	21	81	1	GtKey
14	1E6E	63	21	81	1	S9scall
14	1ED1	41	21	81	0	WizCsum
14	1F12	54	21	81	0	WizCRC
-16	0	7C45	11	81	1	ds
-17	0	1A1E	22	81	1	WizUtils
17	1A1E	513	22	81	0	WizKeys
17	1F31	14B	22	81	0	WizDmo
18	7C	508	22	81	0	WizShell
18	584	14AF	22	81	0	WizSend
-19	0	3085	22	81	1	WizConfig
-1B	0	2F1D	22	81	1	WizClipper
-1D	0	3B55	22	81	1	WPKmod
-1F	0	1C5C	22	81	1	WizAuto
-22	0	15FF	11	81	1	cp
-28	0	1A7	11	81	1	DirM

Index

Item.....	page	Conference.....	16
/dd/cmds/programname.....	31	conferencing.....	16
/dd/com/pro.....	8	Configuration.....	23
26 alpha keys.....	36	constat.....	44
2MHz.....	46	CRC.....	29
8 bit binary data.....	23	ctrl-o.....	12
8 bit mode.....	46	CTRL-X.....	31
@.....	21	delay.....	23
@.....	31	Delphi.....	31
aciapak and t2.....	8	device descriptor.....	44
alt-break.....	17	Edit mode.....	14
alt-down-arrow.....	16	elapsed time.....	13
alt-spacebar.....	22	Enables.....	25
Autolog.....	20	error 216.....	46
autolog directory.....	33	error 55.....	47
backslash.....	21	error 67.....	47
Basic09.....	35	F1.....	18
boldface.....	6	file menu.....	19
bootlist.....	8	Fkeys.....	18
Break.....	22	flags.....	39
break key value.....	33	fmenu.....	28
buffer re-size.....	36	fmenu windows.....	13
bufs.....	40	font.....	42
bufs.buf.....	40	Fonts.....	24
calls.....	38	Full/Half duplex.....	25
CAN.....	31	GEnie.....	31
CART.....	47	go.....	11
caw.....	38	graphics screen.....	24
cc3go.....	7	groups.....	42
CD.....	20	GShell.....	47
CD.....	47	gtkey.....	34
change data directory.....	6	gwin.....	36
change macro.....	18	handshake character.....	27
CIS.....	30	hardcopy path.....	23
CIS B protocol.....	30	help.....	26
Clipper.....	14	info.....	26
cmds.....	40	installing a xproc or xprog.....	32
cntrl.....	39	invoking.....	36
CoCoBin.....	29	Kevin Darlings save command.....	7
Colors.....	24	LS.....	7

macro.....	18	Shell.....	27
main wiz buffer.....	40	shellplus.....	12
mcmclr.....	42	signals.....	45
mdir.....	10	Snap mode.....	15
Menu.....	17	snap.n.....	28
menu entry.....	43	spacebar.....	19
mnu.....	43	Start New Autolog file option.....	26
mnu.sho.....	32	STerm.....	30
modem.....	46	Stop.....	17
MODS directory.....	7	strg.....	42
mouse paws.....	3	syscall.....	42
MPI.....	46	system calls.....	38
multi-pak.....	7	system calls.....	44
mwp.....	34	text recorder.....	24
mwp.....	44	toggle options.....	17
new fkeys.....	32	triggers.....	45
ON ERROR GOTO.....	45	tsmon.....	20
on-line mode.....	12	type 4.....	38
OS-9 Forum.....	30	Upload.....	27
other terminal programs.....	3	usage log.....	6
PAL upgrade.....	7	viewer.....	46
palettes.....	42	VT80.....	36
password.....	20	Wait.....	16
paths.....	38	win.....	41
paths.nextproc.....	37	windows.....	41
pause.....	3	Wiz.....	16
Pause.....	18	wiz process space.....	11
PD.XON.....	45	wizacia.....	44
port.....	43	wizcrc.....	34
Print.....	26	Working directory.....	28
proacia.....	44	X-OFF.....	35
proacia and mwp.....	34	XCom9.....	29
proc missing.....	46	XModem.....	28
proportional.....	6	xprocs.....	10
quick reference chart.....	33	xprogs.....	10
Quit.....	26		
ram full.....	46		
reserved menu slots.....	43		
Reset.....	27		
ROM.....	46		
rs-232 pak.....	7		
s6551.....	46		
setstat.....	44		



Wiz Professional Registration

If you have not already supplied this information, Please complete this page and return to :

WBW
1503-I Flanders Lane
Harwood, MD 20776

Registration entitles you to several benefits including upgrades and a subscription to the Wiz newsletter.

Serial number from the disk: _____

Your Name: _____

Company: _____

Street Address: _____

City: _____

State: _____

Zip Code: _____

Country: _____

Electronic Address: CIS _____ Delphi _____ GENie _____

Do You wish to become a Wiz Pro developer? _____

Programming Example

The fast way to create a new proc is to start with the file 'Type4'. This file has all the PA=4 or type 4 equates already paramed. Just load this file into Basic09, rename, and start editing.

It took me about 5 minutes to create the example extension called 'viewer', and another 5 minutes to install it in Wiz. Viewer is a simple proc that lets you see the hex values coming in the port. It is also a simple terminal program. You can find it on the Wiz Delivery disk, but I'll present it here as well:

```
PROCEDURE viewer
ON ERROR GOTO 100
TYPE wpaths=sp,spa,wpa,dpa,ppa:BYTE;
siop,piop:STRING[3];
nextproc,ho,rxfile,host:STRING; spd(32),oldesc
(32),newdesc(32):BYTE
PARAM paths:wpaths
(* **
TYPE
calls=ststat,gtstat,cursr,screen,reed,wrt,wrtln,
creat,sleep,pid,setpri,pag,bau,typ,pau,alf,ech,e
or,endof,quit,intrpt,opt,bsp,bso,bse:BYTE
PARAM caw:calls
(* **
TYPE
flg=acia6551,TEXT,gmode,roll,bufroll,chdf1,vt52,
vtansi,OLOPEN,spopen,WPAOPEN,SPAOPEN,PSP,BLDF,co
nf,pfl,dfl,exit,ex,dup,mu:BOOLEAN
PARAM flags:flg
(* **
TYPE
con=esc,CR,lf,BS,BL,home,cls,tnum,lbrk,BK:STRING
[1]; Mkey,Estep:BYTE
PARAM cntrl:con
(* **
TYPE
commands=dwend,dwsel,owend,defcolor,ulon,uloff,p
al,hdds:STRING[2];
DLpos,PRpos,ATpos,br,bufnum,group,revoff,rev,pro
p,propoff,bold,boldoff:STRING[3];
four,palette:STRING[4]
PARAM cmds:commands
(* **
TYPE buffers=buf(8192):BYTE; klin:STRING[256];
abuf(128):STRING[1]; mac(2):STRING[80];
bsiz,absiz,in:INTEGER
PARAM bufs:buffers
(* **
TYPE windows=flat1,flat,stat,bottom:STRING[9];
vt80,vt40,gwin,main:STRING[10]; menu:STRING[63];
mesw:STRING[42]; tmenu,tmes:STRING[15];
ask,prmt:STRING[9]
PARAM win:windows
(* **
TYPE
PRNeq=forePRN,backPRN,menuforePRN,menubackPRN,ot
herforePRN,otherbackPRN,curPRN,menucurPRN,otherc
```

```
urPRN:STRING[1]
PARAM PRNs:PRNeq
(* **
TYPE
cnos=fore,back,curs,menufore,menuback,menucur,ot
herfore,otherback,othercur:BYTE
PARAM colors:cnos
(* **
TYPE fx=one,two,GRP,BFN:BYTE
PARAM font:fx
(* **
TYPE regs=ccode,a,b,dp:BYTE; x,y,u:INTEGER
PARAM s:regs
(* **
TYPE strings=mdmclr,ddir,crtc,aktc,pnr:STRING;
ontime:STRING[8]; mdmdly,hh,mm:INTEGER; te2:REAL
PARAM strg:strings
(* **
TYPE portset=bb,pp,ss,ll,sbaud,stype:BYTE; baud
(7):STRING[4]; bcmd(7):BYTE; PAR(5):STRING[4];
SB(2):STRING[3]; word(2):STRING[1]; pcmd
(5),scmd(2),wcmd(2),ppmax:BYTE; typcmd
(2,3,2):BYTE
PARAM port:portset
TYPE menuentry=parm:STRING; sho:STRING[16];
PRO:STRING[12]; pa,funct,trig1,trig2:BYTE;
trig3(2):BYTE
PARAM mnu(26):menuentry
1 (* DIM Local Vars here
DIM char:STRING[1]
DIM ch:BYTE
10 (* proc start
REPEAT
  RUN gtkey(paths.sp,char)
  IF char>" THEN
    PRINT USING "s1,x1,'$',h2,",char; char;
    PRINT " ";
  ENDIF
  RUN gtkey(ch)
  IF ch<>255 THEN
    PUT #paths.sp,ch
    IF ch=5 THEN 20
  ENDIF
UNTIL char=cntrl.CR
IF char=cntrl.CR THEN
  PRINT
ENDIF
GOTO 10
(*
20 END
100 en=ERR
END
```

As you can see, its starting point was Type4 and it uses paths.sp, (the serial port), and cntrl.CR, (a carriage return) both of which are already set by WizPro when viewer is called.

I installed viewer in slot 22 of the menu using Proinstall. sho=<V>iewer, pro=viewer and PA=4.

Appendix B Other Wiz Pro Extensions:

Zero Buffer (XPROC WizZero)

<Z>

The ability to zero, or reset, the Wiz Pro buffer was inadvertently left out of version 1.0 of Pro. It is useful to zero the buffer when starting a <D>ownload if you do not wish to save characters that have already come in. (Download normally saves the complete buffer, back to the beginning).

You will have to install WizZero using ProInstall. Install as an XPROC, and show in the menu <Z>ero Buffer. Install in slot 26 or the 'Z' key.