Network Mail

From: Taft.PA

Date: 22 Feb 1980 3:40 pm PST (Friday)

Subject: Bldr

To: AltoProgrammers↑ (Alto Bcpl Programmers)

XEROX

A new version of Bldr (version 2.7) is released. The only change is the addition of a new global switch /D (i.e., "Bldr/D"), which, if present, causes the display to be turned off and attendant code and data structures (~3500 words) to be thrown away during loading, making more room for symbols and other things. This enables the loading of substantially larger programs than was possible previously.

Network Mail

From: Boggs.PA

Date: 29 Mar 1980 5:22 pm PST (Saturday)

Subject: New Swat

To: AltoProgrammers↑

XEROX

Swat version 30 of March 29 1980 is released. It contains accumulated bug fixes, some features to support IFS's extended memory emulator, and a few changes to commands. Most notable:

When a mulitple proceed breakpoint expires, it becomes a normal breakpoint rather than self destructing.

The search command has changed. It now searches from the currently open cell +1 to end of memory. Exp↑= searches for a cell containing Exp. Exp$↑= searches for a cell whose effective address is Exp. If Exp is omitted, it defaults to the last value searched for.

Network Mail

From: NetSupport

Date: 28 May 1980 9:02 am (Wednesday)

Subject: Growing pains

To: AllMonroe↑

XEROX

Enclosed are messages which explain many problems which people have been having this week. If you go into SWAT and get the message " Pup Routing Table Full" I suggest that you update the programs on your disks. I have noted that the new version of Empress.run described in the last memo needs version 18 of the operating system to work.

Bob Sperry

Date: 27 May 1980 11:38 am PDT (Tuesday)
From: Taft.PA
Subject: Growing pains
To: AltoUsers↑

If you discover today that various programs such as FTP and Bravo fall into Swat with the message "Pup routing table full", it is because you are running obsolete versions of these programs.

Apparently, today is the first day on which the number of simultaneously operating networks in the Xerox internet has exceeded 32. Versions of the Pup software more than about a year old were not capable of dealing with routing tables containing more than 32 networks. A version of the Pup software that can deal with an arbitrary number of networks has been available for slightly over a year, and most programs (FTP, Bravo, Chat, etc.) have been upgraded to incorporate this software.

(A few programs -- most notably Empress -- have not been upgraded. Maintainers of programs that use the BCPL Pup package should check carefully to make sure these programs are using current versions of the software, and release new versions of the programs where necessary.)

If you are in the unfortunate situation of having FTP not work (making it difficult to retrieve current versions of FTP.run and other programs), you should remember that it is also possible to invoke FTP via the NetExec; the network version of FTP is up-to-date, and you can use it to obtain FTP.run from your file server.

(If you have never updated your disk(s), you will probably be better off starting from scratch: make a copy of one of the standard "basic" disks, and copy your files onto it using FTP or Neptune.)

Date: 27 May 1980 2:53 pm PDT (Tuesday)
From: Hains.EOS
Subject: Printing from Bravo8.5
To: AltoUsers↑.PA
cc: AllEOS↑

If your Bravo 8.5 suddenly won't print due to the "Growing Pains" problem, the following set of macros will permit automatic printing through EMPRESS and a return to BRAVO with the same file. The printing is invoked with a "Q H" while in BRAVO8.5 with the desired filename in the file window.
Just copy the following into the [BRAVO] section of your user.cm and re-install bravo:

```
H.QUIT:"{6,1,0,0}b1'Thru Empress{6,2,0,0}hf'@4Press
{6,1,0,0}qJ
"
J.QUIT:"{6,2,0,0}q
EMPRESS.run @1 1/C ; Delete @1 ; Bravo/N @4
"
```

It is easiest to "Put" this message into a file and then merge from it into user.cm in BRAVO.
Both Macros are required, but they can be renamed to any other letters if necessary.
The "1/c" in empress is not necessary, but can be changed if more copies are desired.

Date: 27 May 1980 5:23 pm PDT (Tuesday)
From: Swinehart.PA
Subject: Updated Empress (sigh)
To: AltoUsers↑
cc: Swinehart

I have released a version of Empress which should thrive in an internetwork with a six (or more) bit address space. E.g., it should work correctly if the one you have now complains about full routing tables, etc. [Maxc]<alto>empress.run, [Maxc]<altodocs>empress.tty are current; sources will follow in a day or so.

If you are located outside of Palo Alto, please wait for a local announcement, unless your need is desparate.

In addition to its resistance to today's calamity, this version of Empress has the following modest enhancements:

Global switches:

/2 Requests two-sided printing, if the destination printer can handle it.

/S Sends the file to the printer, but does not print it. You must proceed to the printer workstation and issue a manual print request, supplying the password that was in effect when Empress sent the file. If you have not done this before, you will probably have to get some help from local printing experts.

/W After printing the file, waits until you type something before exiting. If you type <RETURN>, Empress queries the printer about the

progress of your file (printed, pending, etc.) If you type <DEL>, Empress will exit.

Local Switch:

password/S Sends the file, but does not print it until you supply the password. Identical to the global /S switch, above, but you get to pick a unique password for this transaction.

Network Mail

From: isa Date: 27 Feb 1979 10:01 am (Tuesday)

Subject: More OS16 stuff

To: ISADist↑

XEROX

Date: 24 Feb 1979 5:56 pm (Saturday)
From: Boggs
Subject: More OS16 stuff
To: AltoProgrammers↑

Swat.25, released with OS16 had a few small bugs which I have fixed (thanks to Doug Wyatt, Dan Swinehart and Ed Taft for reporting them). A new version, Swat.26 is available on [Maxc]<Alto>InstallSwat.run. I have added a new trap instruction: teleSwatTrap, 77412b which is equivalent to CallSwat(string1[], string2[]) followed by $$↑Y. There is no procedure in the OS to do this for you; here is how to make one for yourself:

and TeleSwatTrap(string1, string2) be (table [77412b; 1401b])()

I neglected to export two error procedures from the OS copy of BFS which are used by BFSInit and BFSFindHole. If you load these files in your programs, you must define DefaultBfsErrorRtn and BFSNonEx yourself. Look at BFSBase.bcpl to see what to do. They are both one-liners that call SysErr with appropriate arguments. OS17 will export them.

The problems with Find(Pkg), ScanFile, and Micro(D) were caused by an increase in the size of a BFS control block zone (CBzoneLength) and the size of a disk stream (1KS). Only people (Peter Deutsch in all cases reported so far) who deal with the BFS at the DoDiskCommand level are affected by the change in CBzoneLength. People who manipulate disk streams are affected by the change in 1KS. My apologies for not warning you; I didn't think anyone used them.

Ed [illegible] reports that not only did the [illegible] Stream [illegible] but the capitalization of some of its fields changed. The [illegible] I am trying to collect all Alto-specific stuff there. I added definitions for a BitBlt table (BBT) and the format of the UtilIn word in the I/O region of memory.

The system log was deimplemented in OS14, and the static fpSysLog in levFilePointers disappeared in this (OS16) release. fpSysTs also disappeared because no one ever used it. Files Sys.log and Sys.ts are not created by the OS whenever you boot anymore. I added storage for the FPs for SysDir, DiskDescriptor, WorkingDir and the name of the working directory in the BFS extension to the DSK structure (BFSDSK). They are initialized during booting when DP0 is opened. The FPs by the same name in levFilePointers are duplicates which I left for backward compatibility, except I deleted fpWorkingDir and NameWorkingDir, since I don't think any of them belong there and because

nobody uses multiple directories and so I hoped nobody depended on them. I haven't heard any screams of anguish.

The next substantial change to the Alto OS will likely be more support for multiple directories. This is caused by the imanent arrival of Dorados and D0s with large capacity disks. Larger disks mean more files and even slower directory lookups unless we break the directory into smaller chunks. Towards that goal, the BFS in this release of the OS pages the bit table making it reasonable to handle file systems up to the 65K page limit imposed by the 16-bit virtual disk address. While on the subject of directories, let me encourage you to NOT use the file versioning feature of the OS. Given the directory structure, it is difficult to implement, and will never work well. I am not inclined to support this mechanism and with just a little prodding, may even de-implement it.

/David

```
// <Alto>WRCNewOS.cm
// Last modified February 20, 1979  7:00 PM by Knox
// You must have at least 300 free pages to run this procedure!

// First, get the new OS
FTP WRC Directory/c Alto Retrieve/c NewOS.boot

// Install the new Operating System.
// Answer questions as appropriate.
Install NewOS.boot
Delete NewOS.boot

// Get new versions of subsystems, and
// update any Bcpl programmer's files that you already have
FTP WRC Directory/c Alto Retrieve/c InstallSwat.run ↑
Retrieve/u Sys.syms Sys.bk AltoDefs.d AltoFileSys.d Disks.d Streams.d SysDefs.d Bcp
**1Files.d Asm.run CleanDir.run EDP.run RamLoad.run Scavenger.run

InstallSwat
Delete InstallSwat.run


// Now initialize Bravo under the new OS.
// Type "Q return" when it is done.
Bravo/i
Delete NewOS.cm

// Delete Dumper.Boot if you wish.
// Swat users:  Consider obtaining the new file Swat.help.
// Mesa users:  Re-install the Debugger.
```

done on
Laurel disk
2/21

+ on Laurel NewUsers
3/12

Network Mail

From: Boggs Date: 20 Feb 1979 2:39 am (Tuesday)

Subject: OS version 16

To: AltoUsers↑

XEROX

Version 16 of the Alto Operating System is hereby released.

To update a disk, retrieve [Maxc]<Alto>NewOS.cm and say @NewOS to the Exec.

Unless you are a programmer, you need only skim the following material:

Henceforth new, uninstalled Operating Systems core images are called "NewOS.boot", not Sys.boot.

A new version of the Alto OS manual is available on [Maxc]<AltoDocs>OS.press. The changes from OS15 are described in more detail there; briefly they are:

1) The interface to Swat has changed, Swat version 25, with many new features is also hereby released.

2) Backward compatibilty with the 'old' operating system has been abandoned (the 'new' OS is 3 years old). If you have Asm, CleanDir, EDP, RamLoad or Scavenger on your disk, you should get new versions which are also newly released.

3) The BFS has been re-organized and released as a separate package. Documentation is in the OS manual.

4) The new Alto file date standard is implemented.

5) The format of type-S boot files (which the OS is) has changed slightly to help boot servers automatically distribute new versions of boot files.

New versions of the Packages and Subsystems manuals will be available in a few days.

A new version of BuildBoot which supports the boot file format change is released.

A new package, BootBase is released. BootBase allows you to make a boot file from most any run file without source level changes. Documentaion is part of BuildBoot.

New versions of the following boot files are hereby released: NetExec, CopyDisk, KeyTest, CRTTest, DiEx, MadTest, BFSTest, and PupTest. Most of these have been pre-released in Palo Alto for several months. The new official home for the standard boot files given out by boot servers is [Ivy]<BootFiles>. If you maintain a standard boot file, please rebuild it and re-release it.

New versions of the gateways implementing the new boot protocol will be released momentarily.

A new version of GateControl, the program for monitoring and controlling gateways is also released; it is available as a boot file and you are encouraged to explore the internet with it. Documentation is on [Ivy]<Portola>GateControl.press.

A new version of Peek, also implementing the new boot protocol, is released. People in Palo Alto have already seen the new DMT, which is also released. I am collecting cursor designs for DMT. If you have a favorite holiday, design a cursor and send it to me, and I will try to release a DMT commemorating it. The cursor is an array of 16x16 dots.

A new version of the Pup package is hereby released.

A new subsystem, ERP, for collecting Pup Event reports is released.

New versions of DDS and the Exec will be released shortly.

/David

c: Jim Dorson
5/17/78

ADVANCED MULTIFUNCTION OFFICE STATION DOCUMENTATION

TABLE OF CONTENTS

PART I

1. INTRODUCTION
   Conventions Used Throughout Documentation

   STARTUP
   Advanced Multifunction Office Station Figure
   Loading Your Disk
   Loading The Disk Figure
   Booting The System
   The Keyboard
   Advanced Multifunction Office Station Keyboard Figure
   The Diablo HyType Printer
   Diablo HyType Printer Figure

2. THE EXECUTIVE

   The Executive Figure

   NAMING YOUR DISK
   Software System - INSTALL
   Install Figure

   CORRECTING TYPING ERRORS DURING KEYING

3. FILE AND DIRECTORY HANDLING

   File Naming Conventions
   Put
   Put Operating Instructions Figure
   FTP

PART II

BRAVO COURSE OUTLINE

PART III

DRAW

1. INTRODUCTION

The Advanced Multifunction Office Station

The Advanced Multifunction Office Station gives the user the capability of text and graphics creation, editing, formatting, and production of documents from the level of office memoranda up to the in-plant composition level. The Station's hardware is composed of a standard typewriter-layout keyboard, high resolution display, a mouse for specifying positions on the display screen, a processor, and on-line disk storage. Several software systems are available for performing text and graphic composition, and are discussed further in the documentation.

Conventions Used Throughout Documentation:

—	What the user types will appear in boldface and underlined-- e.g., **g_et** indicates the **g_et** Command, for which the user types only **g_**.
RETURN	The User types the RETURN key.
ESC	The User types the ESC Key.
CTRL	CTRL preceding a character indicates that the following character is a Control Character--e.g., **CTRL_b** means that the CTRL key is held down *while* **b** is typed.

STARTUP

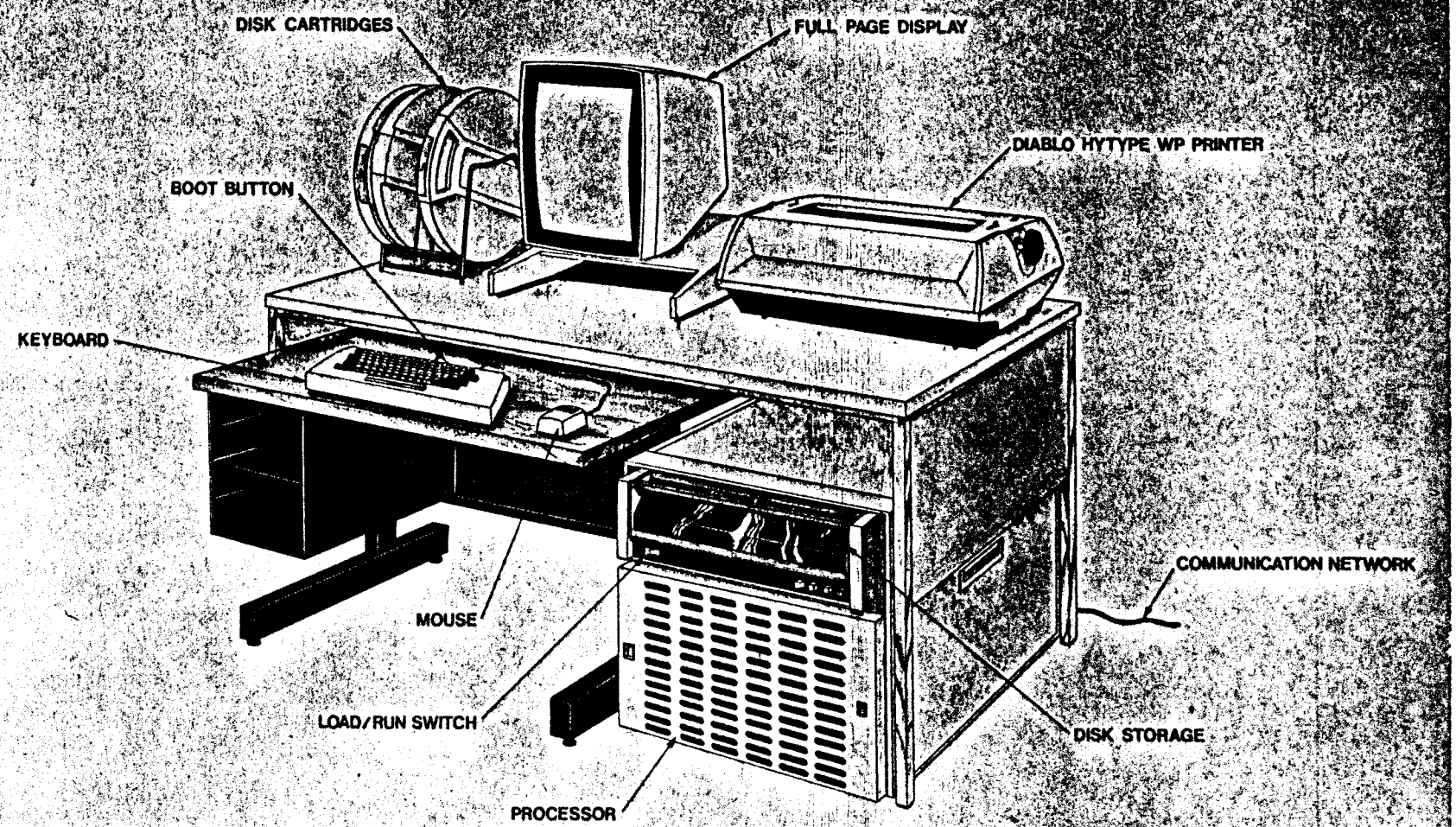
Several startup operations should be performed before you begin to use the software systems available on your Advanced Multifunction Office Station. Review the *ADVANCED MULTIFUNCTION OFFICE STATION figure* to locate the key components of your station's hardware.

LOADING YOUR DISK:

To start up the Office Station, check to see that the system power is on. Below the disk drive's glass door on the *left* is a white switch which is labeled LOAD in one position, and RUN in the other position. On the right is a white **Load light**, a yellow **Ready light**, a red **Check light**, and a red **Power light**. Place the switch in the load position insert your cartridge disk into the disk drive. See *Loading The Disk figure*. Close the disk drive's glass door and place the Load/Run switch in the RUN position. When the disk cartridge has speeded up for operation, the **Ready light** will come on.

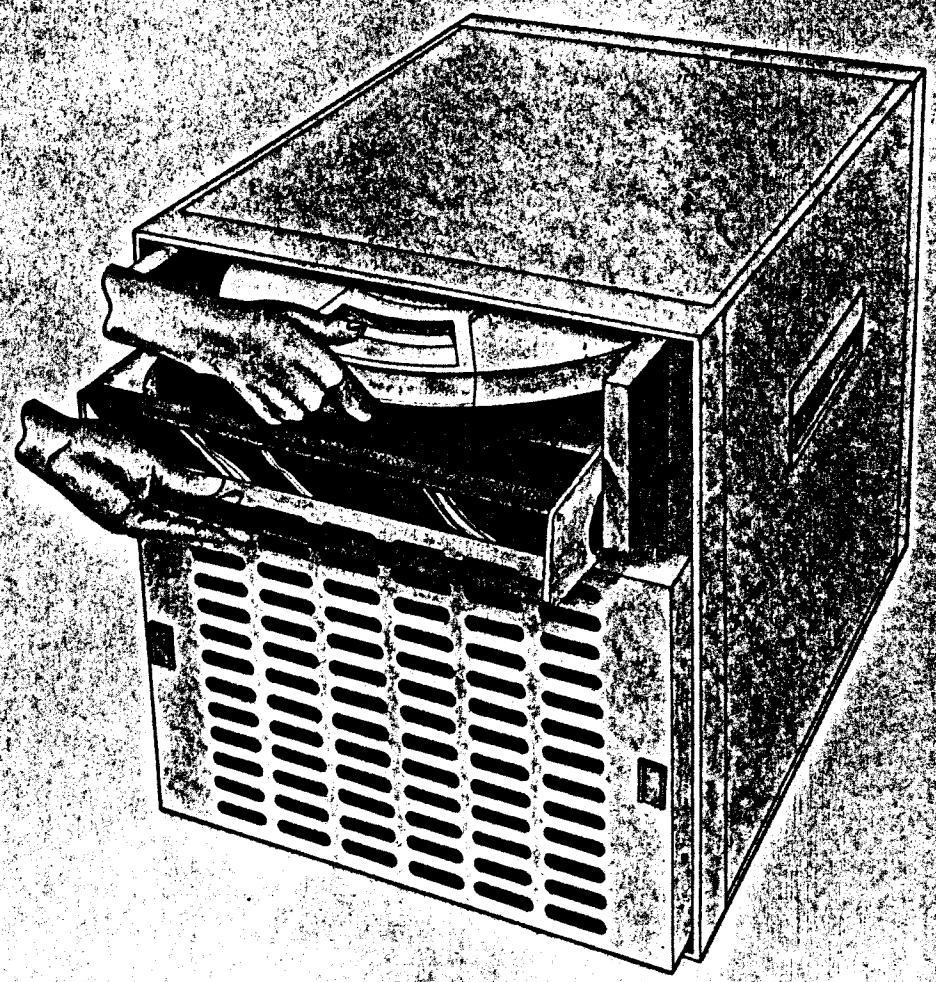
007798

ADVANCED MULTIFUNCTION OFFICE STATION



XEROX

LOADING THE DISK



XEROX

007799

BOOTING THE SYSTEM:

When the **Ready light** on the disk comes on, you can "boot" the system into operation by pressing the **Boot button** on the back of the keyboard. The **Executive** level of the system will appear on the display which allows you to proceed into any of the software systems.

THE KEYBOARD:

The keyboard, as shown in the *ADVANCED MULTIFUNCTION OFFICE STATION KEYBOARD figure*, is a standard typewriter-layout keyboard with several special function keys. It functions similarly to a standard typewriter with only one exception; the LOCK key causes only the alphabet keys to take on their uppercase representation, while the SHIFT key causes all keys on the keyboard to take on their uppercase representation. The SHIFT key will perform this function even with the LOCK key depressed. This is useful if you want to key in a document with all letters in caps. The LOCK key enables you to easily perform this function because the common punctuation keys remain unchanged. (Note that the period and comma do not appear also as shift characters on this keyboard.) The purpose of the other special function keys will be discussed further in the documentation.

THE DIABLO HYTYPE PRINTER:

The Diablo HyType Printer connected to your office station allows you to print documents and envelopes with a simple command. This printer uses varied "daisy" font wheels that can be easily changed depending upon the type of document you are printing. Your instructor will demonstrate changing font wheels and cartridge ribbons. When loading paper or envelopes into the Printer, care should be taken that the Platen Adjust Lever is set for the appropriate thickness of the paper and the Print Intensity Adjustment Lever is set to accommodate the font wheel you have chosen. The location of the levers and the meaning of the appropriate print intensity settings are shown on the *DIABLO HYTYPE PRINTER figure.*

2. THE EXECUTIVE

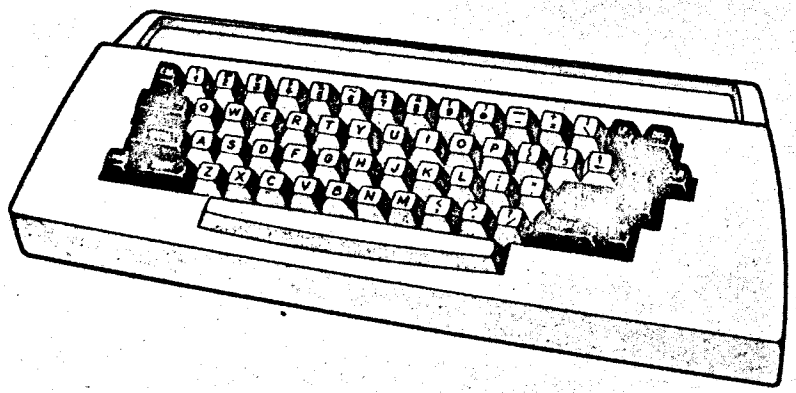
The **Executive** gives the user a base from which to call any of the numerous software systems for text and graphic creation and production. Upon completing a task in a software system, you will always be returned to the **Executive.**

On the top two lines of the display the **Executive** provides you with some operational information, along with identifying the Office Station's Executive and Operating System version on your disk. *THE EXECUTIVE figure* shows the display and the meaning of the operational information supplied.

When returning to the **Executive** from any software system, review the number of computer pages remaining on your disk. If you drop below 150 computer pages, you should free some disk space by deleting files or transferring files to another disk. (A computer page is equivalent to approximately 500 characters of text.) See PUT documentation and FTP documentation for further instructions on how to free computer pages from your disk.

007797

ADVANCED MULTIFUNCTION OFFICE STATION KEYBOARD

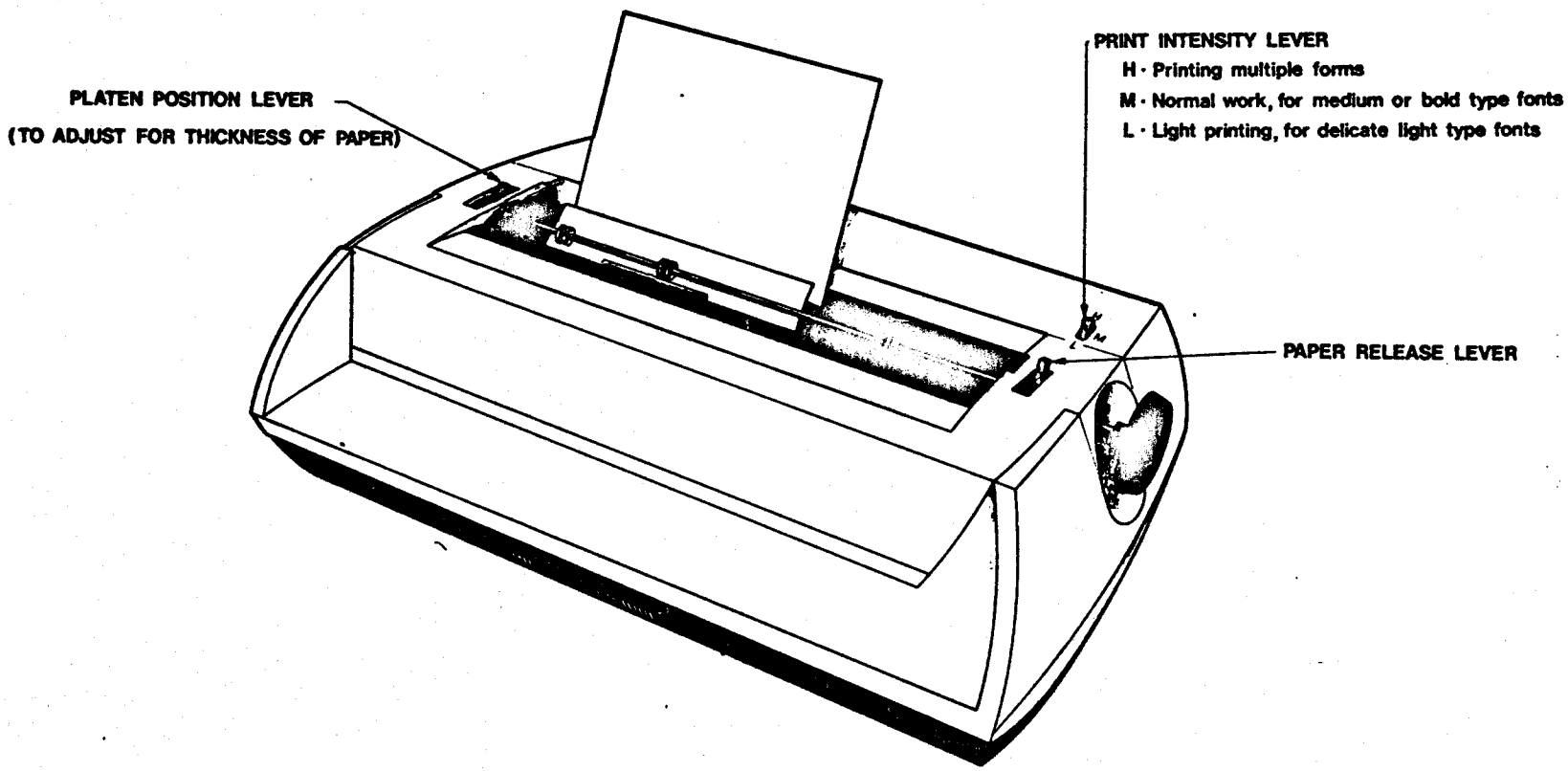


KEYS	EXPLANATION
DEL	Aborts Command.
LF	Shows page boundary.
ESC	Terminates & executes a command.
TAB	Executive - lists files. Bravo - tabs text.
CTRL	When held down makes a text key into a command key for looks during typing.
CLR	Clears all character looks.
LOCK	Turns only alphabet portion of keyboard to uppercase representation.
SHIFT	Turns all keys to uppercase representation.

XEROX

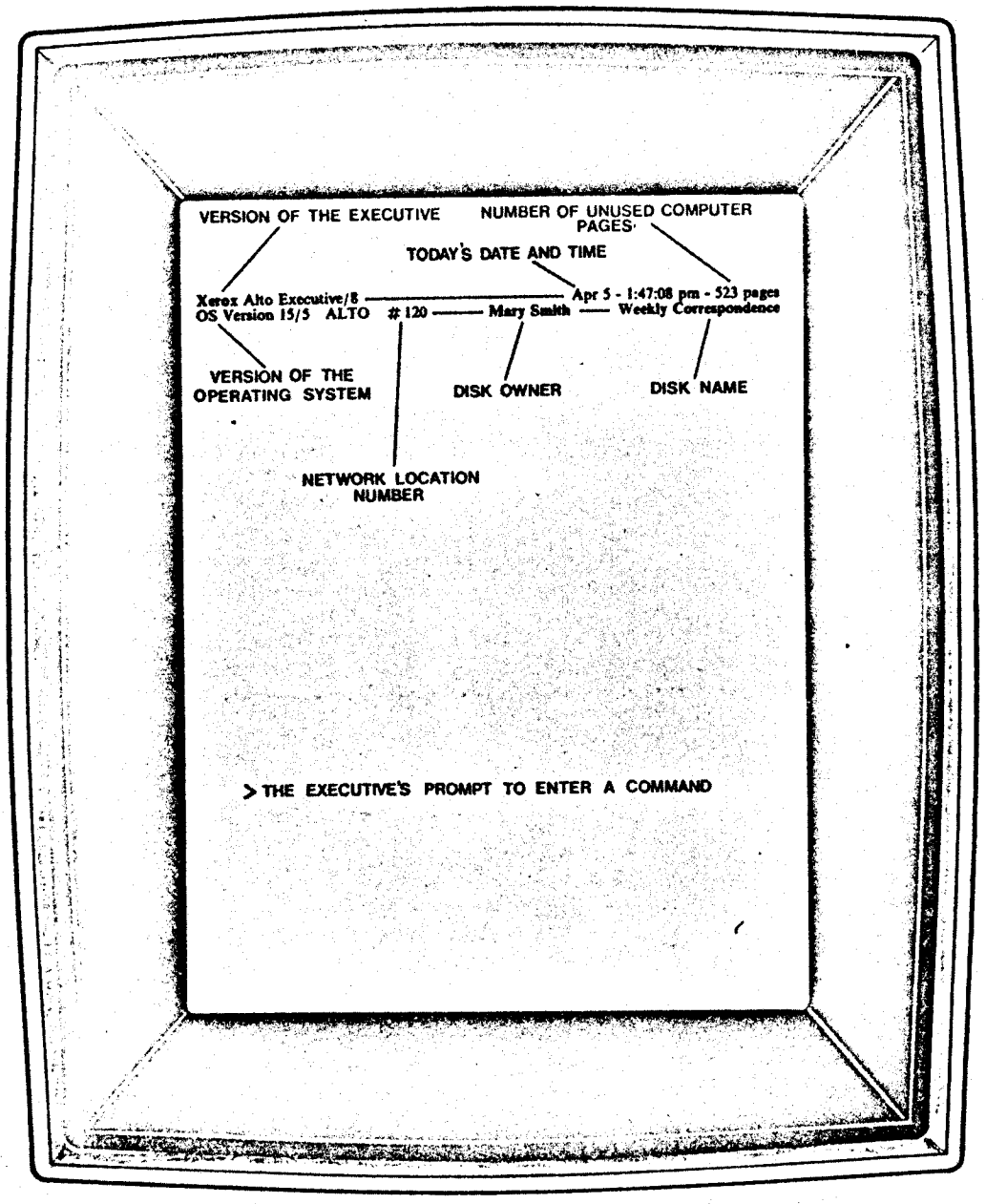
007804

DIABLO HYTYPE WP PRINTER



XEROX

THE EXECUTIVE



XEROX

Toward the middle of the screen is a marker " > " followed by a blinking bar. This indicates that the Office Station Executive is ready to accept a command.

NAMING YOUR DISK

When you are in the **Executive** a software system can be started simply by typing the software system's name. The software system INSTALL allows you to give a disk a name, identify the owner of the disk, and give the disk a password if desired.

To invoke the software system INSTALL type Install followed by the RETURN key to the executive.

> Install RETURN

See the *INSTALL figure* to review the interaction of your dialog with the computer. After calling the INSTALL software system, you will be asked whether you want the "long installation dialogue"; answer n for no. When it asks you for your name, type your name, followed by a RETURN. When it asks you for a disk name, choose a suitable one and type that in, again followed by a RETURN. Next it will ask you whether you want to give your disk a password; answer n for no. *If you answer* **y** *for yes, you will be asked for the password every time you "boot" the Station, and will not be allowed to do anything until it's provided correctly.*

CORRECTING TYPING ERRORS DURING KEYING

There are three keys that will be particularly useful to a new user when keying text. They are DEL, BS (backspace), and CTRL w (the CTRL key depressed while w is typed).

DEL is used to abort a command. Striking the DEL key will abort a command and allow you to start again with the command intended. The BS key backspaces one character at a time erasing the character from the display. The CTRL w keystroke backspaces one word at a time erasing the word from the display. These three keystrokes are useful whether errors occur during a command at the Office Station **Executive** level or while typing text into **Bravo** during document creation.

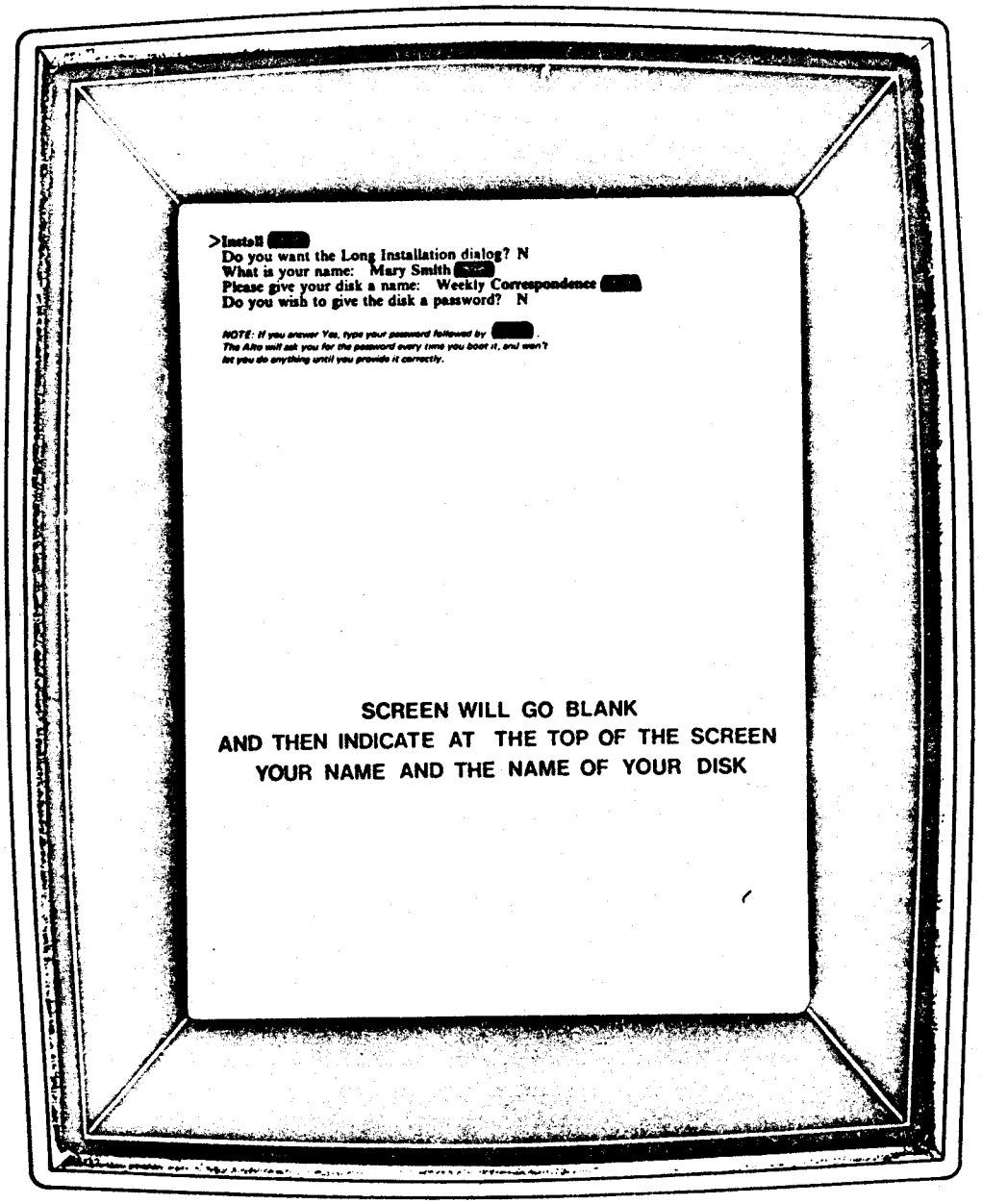
3. FILE AND DIRECTORY HANDLING

FILE NAMING CONVENTIONS:

It is good practice to name your **Bravo** files with the extension ".BRAVO". For example, the 4-24-78 Weekly Status Report, could be named 4-24-78-WSR.Bravo. Using this naming convention enables the system to organize automatically those files with the same extensions and thus facilitate accessing files created with **Bravo.**

XEROX

INSTALL



007803

XEROX

If further differentiation is needed, extensions can be used to indicate something about the kind of text document it is. For example, use ".Memo" for all memoranda created, ".Letter" for all letters, ".Report" for all reports, etc.

On your disk is a **Directory** of all of the files you have created. To display a *complete* listing of all files in your **Directory**, when you are in the **Executive** type the TAB key. The top third of the sceen will display the first portion of your **Directory**. They proceed in alphabetical order from left to right across the screen. At the bottom of this portion of the list, is the question "**More?**". Type y, or any other character for yes, to see more of the filenames; type n, for no, to discontinue the directory listing. Continue responding to the question "**More?**" until the complete directory has been listed.

In this form your directory may be long and somewhat difficult to decipher. You can specify a particular *set* of files by using a * in place of either the filename or the extension. In order to see only those files with the extension ".Bravo", type *.Bravo TAB. A list of only those files that end with the extension .Bravo will be displayed on the screen.

There is a limit of 39 characters for a filename. A filename cannot have a space character, but dashes are allowed. For example, "4-27-WSR.Report" is a valid name where "4 27 WSR.Report" is not valid. Abbreviated filenames are useful unless they are *so* clever and descriptive that they are easily forgotten. Checking your **Directory** for all files with a common extension will help you locate a forgotten file more easily.

DELETING FILES FROM THE DIRECTORY

Files that are no longer useful--that is, unlikely ever to be needed again--can be deleted from the disk. Realize, however, that once all versions are deleted the file cannot be retrieved.

There will be no immediate need to delete files from your disk. Until your disk begins to get full, do not worry about deleting. Once you decide to delete files, the command is given while in the Executive. It is:

>del filenameRETURN

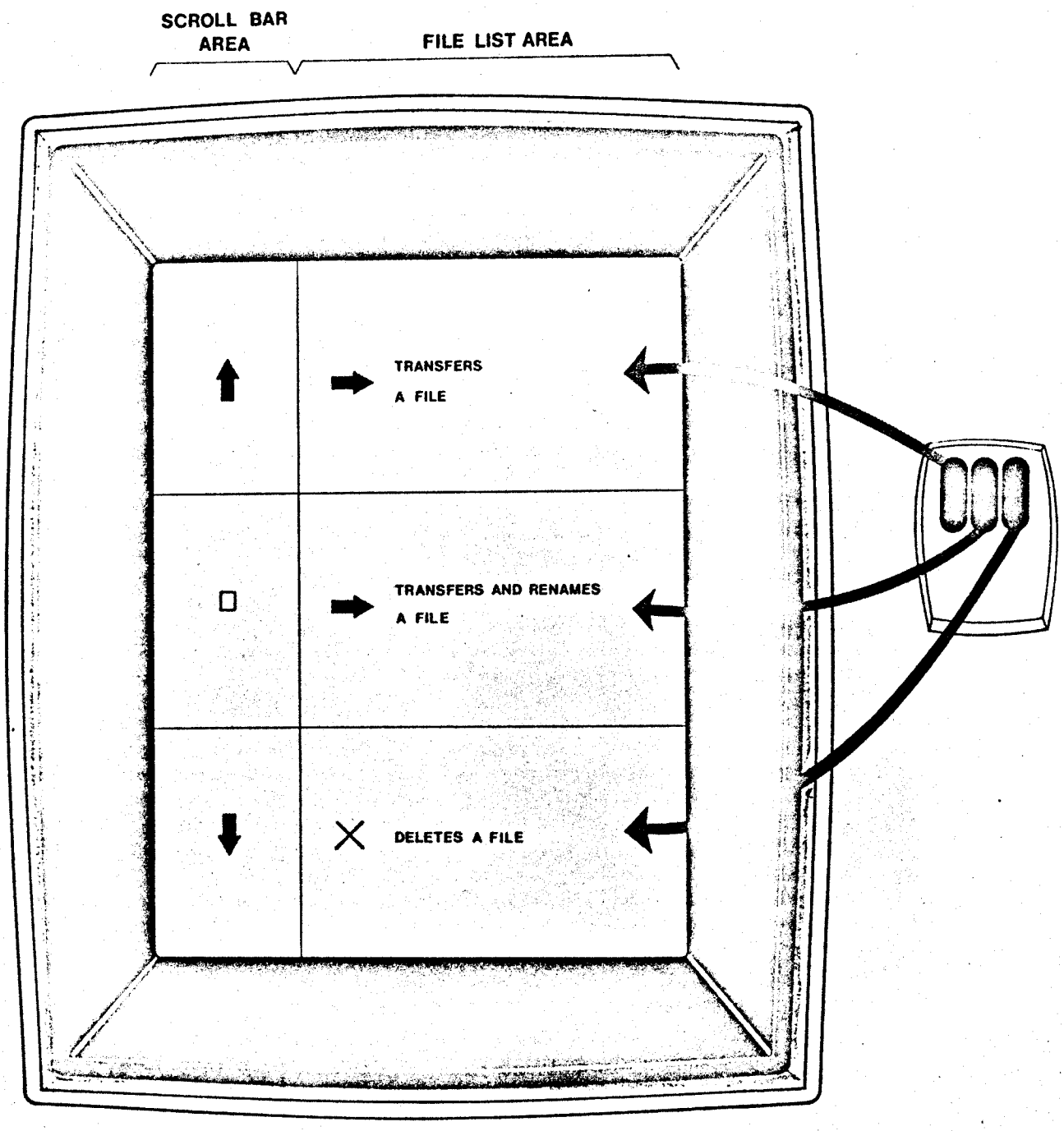
Type **del** (the characters d e l), *then a space*, then the **filename** to be deleted, then a RETURN to activate the command. The filename will be listed and deletion verified.

PUT:

The **Put** software system gives you an easy means of deleting and/or transferring files from one disk to another on *a dual disk* office station. (See *PUT OPERATING INSTRUCTIONS figure.)* To call **Put**, type Put RETURN to the **Executive**. **Put** displays a complete listing of all files on the disk(s) in side by side columns. The lists can be scrolled by moving the cursor to the left side of each column and pressing the mouse buttons as you would within **Bravo's** scroll bar area.

Selecting a file for deletion or for transfer to the other disk is done with the mouse. The *left button* indicates the file should be transferred to the other disk. The *middle button* indicates the file should be transferred to the other disk and given a new name. You will be

PUT OPERATION



XEROX

007800

asked to supply the new name after selecting a file with the *middle button*. Selection with the *right button* indicates the file is to be deleted. When all the selections have been made (they can be mixed) press the ESC key to perform the selected operations.

If during the selection process you would like to remove the deletion selection command or the transfer selection command from a file, reselect the file with any mouse button and the command to delete or transfer the file will be erased. If the middle button is used by mistake and the system asks you to input the new file name, type RETURN to abort the renaming transfer. Then reselect the file with another mouse button and the transfer with a new name is aborted. *Important Note*: After executing the deletion command with the ESC key, that file(s) is deleted from the disk and cannot be restored.

Subsets of files in the **Directory** can be listed by selecting the word "**List:**" with the left mouse button. The "*.*" will be replaced with a blinking vertical bar indicating you are to type in the descriptor of the subset of files that are to be displayed, followed by RETURN. For example, to have **Put** display all files with the extension ".bravo", type "*.bravo RETURN". The column listing will be changed to show only your Bravo files. To restore the column list to its default state displaying the entire directory, type R.

When you are finished with **Put**, type "q RETURN", and the system will return to the **Executive.**

Your instructor will demonstrate how Put is self-documenting. When you type a '?' to the system, the top window will begin to display the operating instructions.

FTP:

The FTP software system is used to transfer files from one Office Station to another. All Office stations in your facility are on the same communication network called **Ethernet.** In order to transfer files over the **Ethernet** from one station to another, both stations must be in a communication mode. In other words, both stations must call the software system FTP.

In order to get the stations in a communication mode, at both office stations simply type FTP followed by RETURN. Then at your station identify the station you want to communicate with by typing its Network Location Number, or name, followed by RETURN.

*120# RETURN (or if Location #120 is called Charlie...)

*Charlie RETURN

To retrieve a file from the *remote* office station and store it on your disk, type Ret followed by a space and the filename of the *remote file* followed by RETURN. After you type a space, the system will complete the instruction as shown below.

*Retrieve remote file 4-26-78-WSR.Report RETURN

After this instruction is given, FTP will prompt you to confirm that you want the file on your disk under the same name. If you want to store the file on your disk under a new name, type the new name followed by RETURN; otherwise just type RETURN. When the transfer is complete, the system will display "*Done*".

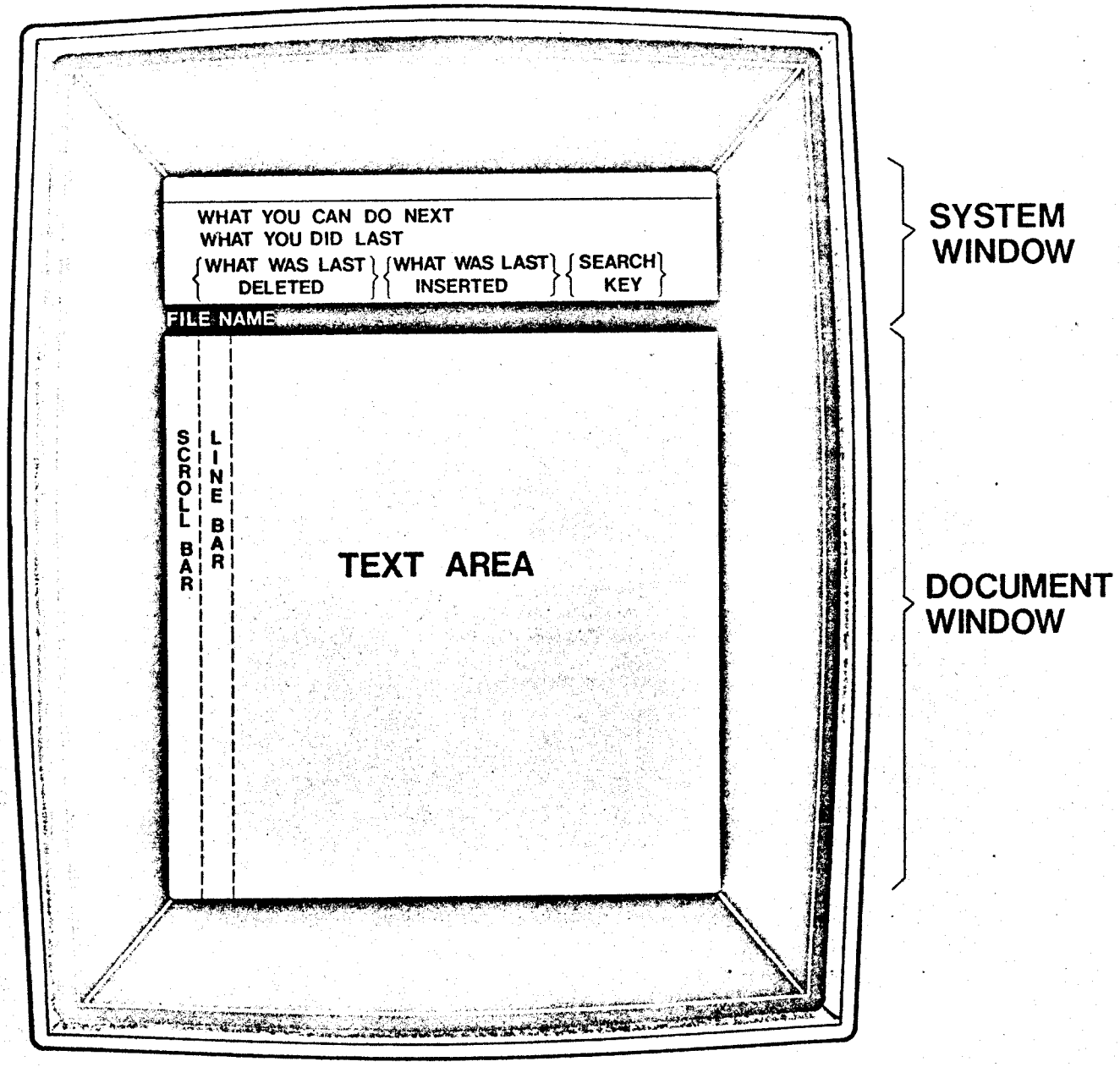
To store a file on the *remote* office station from your disk, type St followed by a space and the filename followed by RETURN. After you type a space, the system will complete the instruction as shown below.

*Store local file 4-19-78-WSR.Report RETURN

After this instruction is given, FTP again will prompt you to confirm that you want to store your file on the *remote* station's disk. If you want to store the file on the *remote* disk under a new name, type the new name followed by RETURN; otherwise just type RETURN. When the transfer is complete, the system will display "*Done*".

When all transferring is complete, type Quit RETURN. This command will close the connection and return you back to the **Executive.**

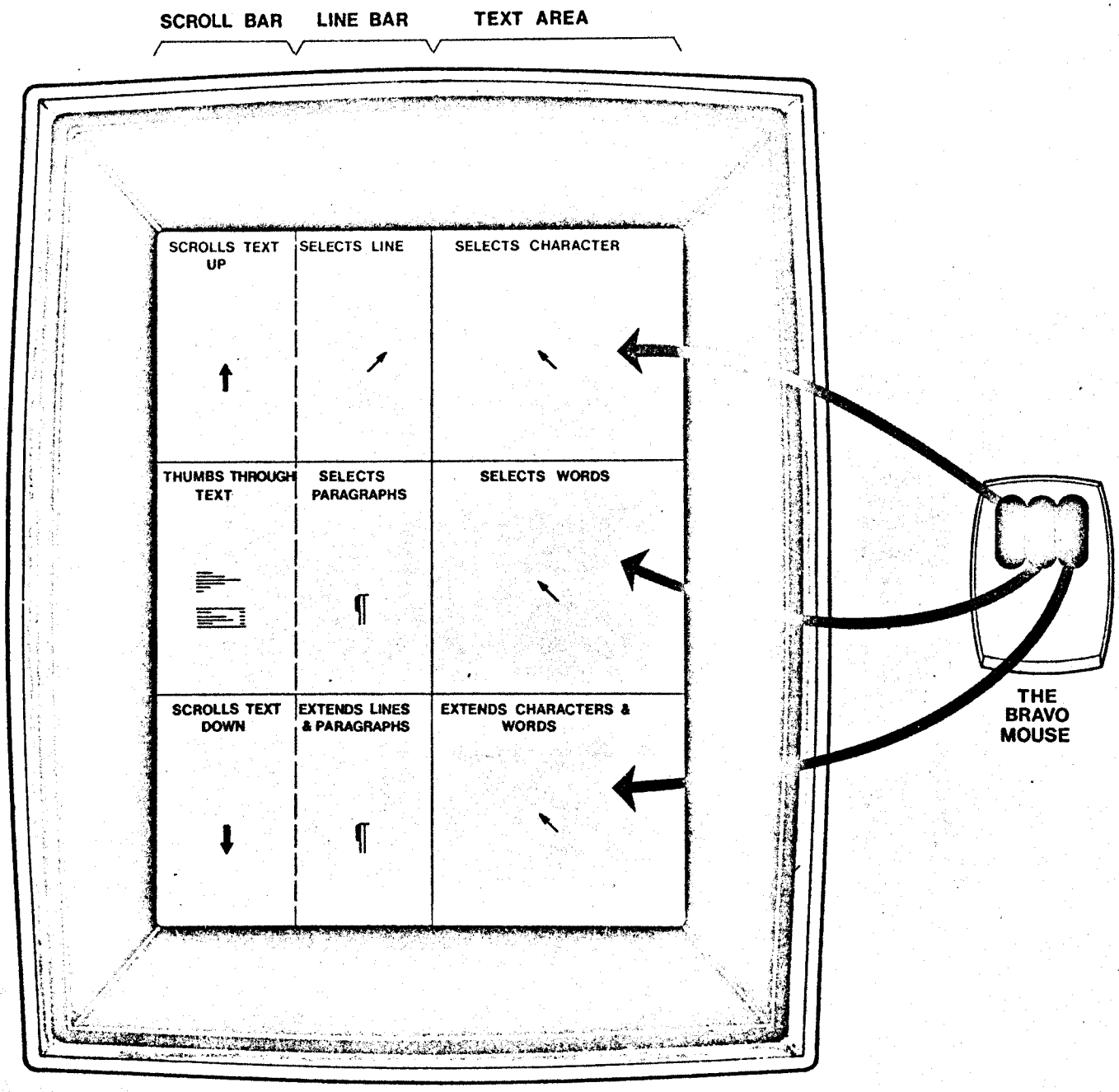
THE BRAVO DISPLAY



007796

XEROX

BRAVO MOUSE OPERATION

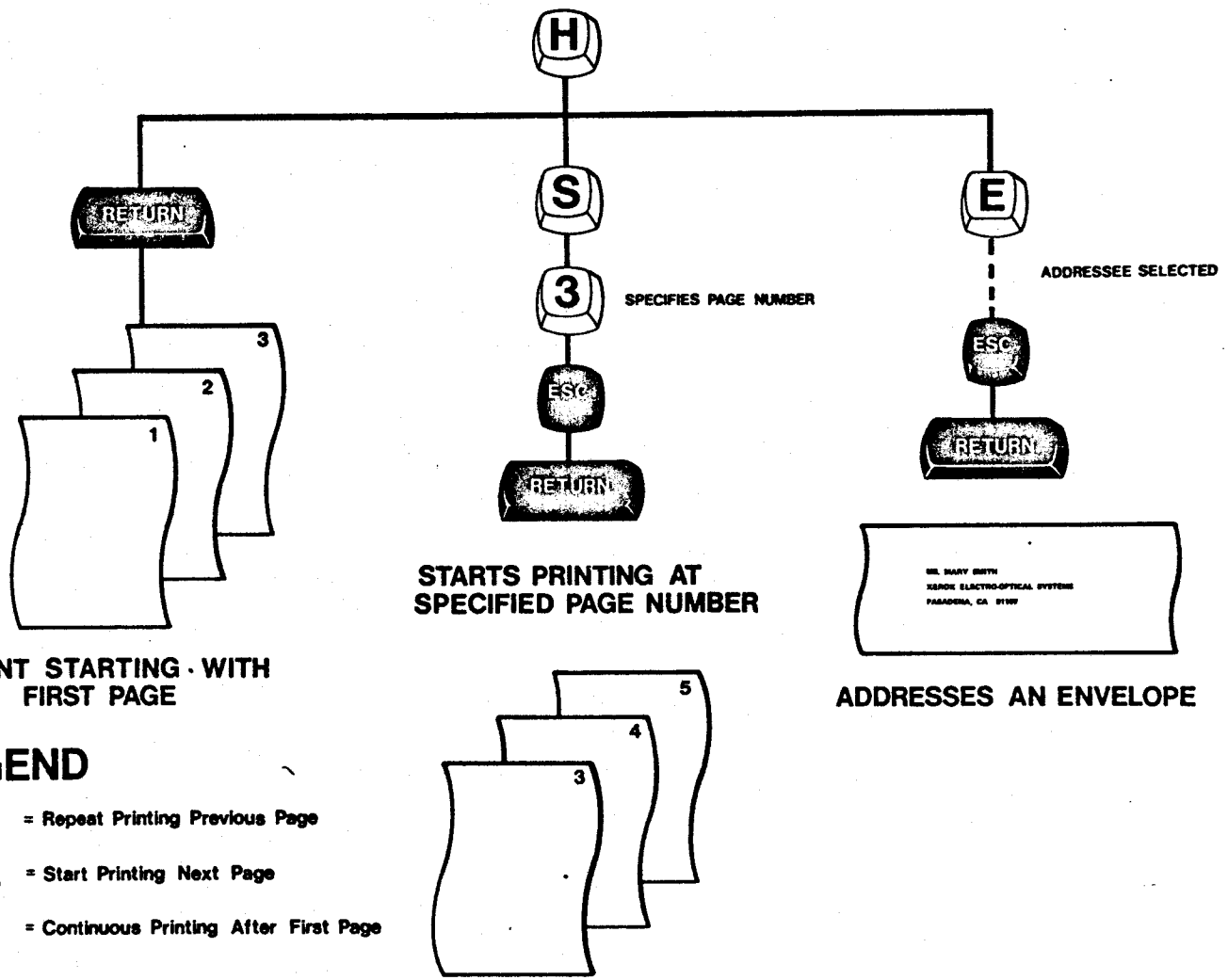


XEROX

007795

007802

BRAVO HARDCOPY COMMAND



LEGEND

R = Repeat Printing Previous Page

SPACE BAR = Start Printing Next Page

C = Continuous Printing After First Page

XEROX

XEROX XEROX

Alto II/Orbit/Sequoia Press file printer

Spruce version 5.1

File: telnet.help

Creation date: February 9, 1978 3:54 PM

Name: wrc

17 total sheets = 18 pages, 1 copy.

XEROX XEROX

?Telnet User Guide

User Telnet (hereafter called Telnet) provides facilities for communicating with host computers via the ARPA network utilizing the TELNET protocol. The purpose of the Telnet program is threefold. It converts various terminals connected to TENEX into a standard type of terminal called a network virtual terminal (NVT) by interposing programs in the character streams between the terminal keyboard and printer and the terminal port on the host computer. Secondly, it provides information about the network to assist a user in establishing connections. Thirdly, it multiplexes the terminal among several remote jobs.
?

Telnet ?Command-Interpreter

Instructions to the Telnet program are given via the Telnet Command Interpreter. When in command mode (see below), characters typed on the user's terminal are read by the Telnet command interpreter and decoded as commands to perform various actions by Telnet.

The Telnet command interpreter has two unique features. The command interpreter will refuse to hear anything it does not understand. With full-duplex terminals, this means that no echo will appear for characters which are not valid successors of the previous input. In any case, the character is ignored and a bell is typed out. The input stream that has already been typed is not forgotten however. Therefore, it is only necessary to type the correct character and not the complete command. This feature may be turned off with the "no fancy.command.interpret" command.

The other unique feature of the Telnet command interpreter is the use of question mark to discover what the command interpreter expects next. Typing a "?" at any time in command mode will elicit a list of words the command interpreter is expecting. Thus, typing a "?" when nothing has been typed will yield a list of all possible top-level commands. Typing "co?" will yield a list of all commands starting with "co". Typing "connection.to ?" will yield a list of possible arguments to the "connection.to" command.

The command interpreter provides command completion whenever a terminator is typed (full-duplex terminals only) and an exact match is achieved with some command or a unique initial substring is typed. Command completion may be suppressed with the "concise" command. Terminators are space, comma, alt-mode, and carriage return. Terminators are often not distinguished and are thus equivalent. Where necessary, comma is used to separate list items, space terminates a command or option and signals the desire to specify more options, carriage return ends a command unless more information is necessary. Altmode is the same as space except that it will cause command completion in those modes where it is normally suppressed.
?

?Command/Remote-Mode

As mentioned above, characters typed on the terminal keyboard may be used in two ways: either as commands to Telnet, or as input to the remote host. The choice is made on the basis of whether Telnet is in remote mode or command mode. In command mode, characters typed on the terminal keyboard are read by the Telnet command interpreter and decoded as commands to perform various actions. TELNET is initially in command mode and will revert to command mode whenever the Telnet escape character (see below) is typed.

The opposite of command mode is remote mode. In remote mode, characters typed on the keyboard (with certain exceptions) are not examined by Telnet at all, but are merely passed on to the remote host computer. Remote mode is normally entered after any command is executed when the current connection exists. The "local.mode" command may be used to defeat this. The effect of the "local.mode" command is cancelled by the "remote.mode" command or by the "connection.to" or "retrieve.connection" commands.
?

?Escaping Back to Command Mode

At any time, typing the Telnet escape character (initially control-Z (SUB)) will cause Telnet to stop whatever it is doing and return to command mode. Occasionally, a slight delay may be experienced due to the need to clean up whatever was happening at the time. Telnet announces the switch to command mode by the appearance of a sharp sign "#" at the left margin. Telnet also indicates the transition out of command mode by the appearance of another sharp sign followed by a new line.

WARNING: If you have control-Z anywhere in your programming, you should change your escape character for Telnet to other than control-Z to avoid mishaps.
?

?Making-a-Connection

There are two ways to make a connection. Typing "connection.to <Host> [<qualifiers>]" or simply typing "<Host> [<qualifiers>]" will cause a connection attempt to be made. If successful, the connection will be said to be complete and the terminal will be placed in remote mode. If unsuccessful, the connection will be said to be "incomplete because ---" with a reason given; also if the remote host is down, a line is typed telling why and for how long. By terminating the host name with a space, one or more qualifiers may be specified. Ordinarily socket 1 is assumed. Thus without a qualifier, the connection will be made to the "logger" on the remote system. By using an octal number as a qualifier, the connection will be made to the socket so specified. A set of names is available for specifying the socket desired. This set consists of names for all the standard sockets.

The "wait" qualifier may be used to camp-on the connection. This qualifier causes Telnet to repeat the attempt to connect in the

event of a failure until it finally succeeds. An initial failure causes a message to that effect to be printed. When the attempt finally succeeds, bells are typed out to wake the user up. The attempt to connect may be aborted by typing the Telnet escape character.

The "load.settings.from..." qualifier (possibly qualified with "no") may be used to cause (inhibit) the mode flags to be initialized from the mode file. When inhibited, the current modes are used.

The "name.for.connection" qualifier may be used to specify a name for this connection other than the one assigned by Telnet. A name for the connection may also be given later by the "name.for.current.connection" command.
?

?Disconnecting

The "disconnect" command is used to close the current connection. This will not necessarily log you out from the remote host so you should perform the logout procedure for that host before disconnecting. The disconnect command takes an optional argument specifying the name of a particular connection to be disconnected. See multiple connections and connection names below.

In the event that the network connections are severed by a network failure, the message "IO error for connection <name>" is printed, the connection is disconnected, and Telnet reverts to command mode. This may happen even if the error occurs on a connection which is not current. If the remote host initiates a disconnect, a message to that effect is printed and the same action is taken.

If the remote host on the current connection stops responding when input is being sent, a line is typed, "Host not responding on connection xxx." (In this case the connection is NOT lost.) When the remote host resumes operating, the user is informed: "Service restored on connection xxx."
?

?Echo-Control

Telnet allows several options concerned with echoing. Echos may be generated by the terminal, by Telnet, or by the remote host. Telnet determines if the terminal is generating echoes when started by examining the mode word for the terminal. The "terminal.type.is" command may be used to change this.

If the terminal is echoing, then Telnet will do everything possible to cause the remote host to not generate echoes, and Telnet will not generate echoes itself. If the terminal is not generating echoes, then Telnet determines whether it should echo or not by information in the mode file (if any) or by the "echo remote"/"echo local" commands, or by information sent from the remote host.

Telnet keeps the remote host informed about how echoing is being done and if the remote host is suitably equipped, it will follow along. If not, then the user will have to give commands to the remote host to achieve the proper echoing. Telnet also will respond to commands from the remote host concerning who should be echoing. If Telnet believes the terminal is doing its own echoing, it will respond to any request from the remote host to not echo by an "I'll echo" command.
?

?Line-Buffering and End of Line Conventions

Telnet provides an optional line buffer for use with line-oriented operating systems. In this mode, characters typed in remote mode are stored in a local buffer up through an end of line. Prior to the end of line, the currently buffered line may be edited using control-A (SOH) or control-H (BS) to delete characters, control-X (CAN) to delete everything, and control-R (DC2) to retype the current contents. Telnet always converts the TENEX EOL into the NVT EOL. TENEX in turn converts a carriage return into the TENEX EOL. Thus typing a carriage return will cause the buffered line to be transmitted. Linefeed may also be used to terminate a line. In this case, the transmitted line will end with only linefeed, not the NVT EOL.

Telnet provides an optional linefeed echo for carriage return. If the remote host provides a linefeed also, then the echo generated by Telnet should be suppressed with the "echo no linefeed.for.carriage.return" command. In remote echo mode, Telnet generates no echos whatsoever. In this mode, all echos must be provided by the remote host.
?

?Status-Commands

Several status commands are available for discovering facts about the network. None of these commands will affect the state of the current connection. The status commands include where.am.I, status.of, netstatus, and socket.map. These commands are summarized below.

?

?Special-Characters

Several commands are available to send characters which do not appear on the terminal. "Code" takes an octal (decimal if preceda by "D", hexadecimal if preceded by "H") argument and sends the character with that code. The word "code" may be omitted and just the argument typed. "Control" takes a character argument and sends the corresponding control character (the low order five bits of the character) is sent. The "!break!" command sends the NVT break character which is mapped by some systems into the equivalent of the attention, quit or break key which appears on some terminals.

To facilitate operation with systems requiring frequent use of

special characters or lower/upper case graphics which a particular terminal may lack (e.g. 33 Teletypes have no lower case), case shift characters may be defined for upper/lower character/lock shifts and characters may defined which will translate into attention or break (NVT 201), and the synch sequence. The "case.shift.prefix.for", "attention.character=", and "synch.character=" commands are available to independently set each of these characters. In addition, a character may be defined ("quote.prefix" command) to be a single character quote. The character following this character is always sent regardless of any special action it may otherwise have.

If possible, case shift characters will be used to indicate the case of both input and output. Thus the case shift characters may not be echoed when typed but rather before the output.

All special characters are listed by the "current.modes.are" command. This includes the escape character and the clear output buffer character.
?

?Leaving-Telnet

To leave Telnet, it is first necessary to return to command mode by typing the escape character. This is because while in remote mode all characters except the escape character are passed on to the remote host or modify characters passed to the remote host. Once in command mode, you may return to the EXEC by typing control-C (ETX) or by using the "quit" command. Continuing from the EXEC will resume with no loss. The "logout" command will disconnect from any remote job and logout your local job. The "exec" command will start up an inferior EXEC under Telnet. From this inferior EXEC, it is possible to perform assemblies or any other task involving the running of subsystems. The "run" command allows an arbitrary program to be run in an inferior fork of Telnet. The "run" may be interrupted by the Telnet escape character.
?

?Multiple-Connections

Telnet provides a facility for multiplexing a user's terminal among several remote jobs thus allowing several simultaneous activities. This is done by giving a name for each connection as it is created. The user may specify the name, or Telnet will default the name to a number. The "retrieve.connection..." command causes the named connection to be made current and remote mode to be entered. Non-current connections remain active, but any output received is buffered until that connection again becomes active. Terminal input goes only to the currently active connection.

Telnet may be made to announce the receipt of output on a non-current connection with the "signal.waiting.output" command; it may also be caused to hunt for and switch to any active connection-- see "wait.for.any.active.connection" and "auto.switch.to.active.connection" commands.

The name of the current connection may be changed after it is established by means of the "name.for.current.connection" command. The name so specified may be up to 6 characters in length and must be unique.
?

?Typescript-File

Telnet provides a means of saving on a file a copy of the typescript for a session. This is useful for producing hard copy of the session when using a scope terminal or for producing documentation of procedures or demonstrations. Telnet is started with no typescript file assigned. The "typescript.to.file" command may be used to assign one, either the default temporary file TELNET.TYPESCRIPT;S in the LOGIN directory, or one named by the user. The typescript consists of a nearly exact copy of what appears on the terminal with the exception of that which occurs during the execution of the "exec" or "run" or "ddt" commands. "Nearly" refers to slight differences in the spelling of file names in certain Telnet commands. For privacy, the typescript file is given a protection that allows no access to anyone but "self".
?

?Diverting-Output

The output stream may be diverted to some other file with the "divert.output.to.file" command. While diverting output, Telnet sends all output to the indicated file and sends a line to the terminal only when the terminal's output buffer is empty. Thus the terminal monitors the transmission of the stream to the file. The diverted output consists only of characters from the remote host. Telnet commands and responses do not appear in the diverted information. This mode is useful as a primitive file transfer mechanism or to allow printing of large amounts of terminal output to be done with the lineprinter. It is cancelled by "no divert.output ...".

?

?Input-from-a-File

The input stream to a remote job may be taken from a file instead of the local terminal by means of the command "take.input.stream.from.file". Telnet blocks terminal input to the connection current when the file is specified, and transmits characters from the named file (echoing as usual according to current modes). However, input to other connections and in command mode is from the user's terminal. When the given file reaches EOF the file is closed and released, and input reverts to the terminal. The user may also manually cancel file input by escaping to command mode and giving "no take.input ...". This mode is useful for routine sequences performed in the remote job. Note that a connection must be established and current when input to it is diverted to a file. Note also that file input is suspended when TELNET is returned to local mode or when another connection is made active; it is not possible to let an input file run while attending another connection.

?

Telnet Command Summary

?Connection.to <host> or host name

Performs ICP to connect to the indicated host. Options are available for specifying initial connection socket name or number, and initializing modes from the mode file via the following subcommands. Note that if <Host name> is used as a command, only the NAME of a SERVER host may be given (e.g., BBN-TENEX). The argument for "Connection.to" may be any host name or an octal host number.

<octal number>

An ICP is performed to connect to the indicated service socket. Normally socket 1 is assumed.

Logger

Sets socket to 1.

Wait

The connection attempt is repeated until successful.

Name.for.connection.is <name>

Sets the name for this connection as specified.

[no] load.settings

Determines whether to use current mode settings or to load new ones from the mode file.

?

?Disconnect <cr>

Disconnects the current connection. This will not necessarily log you out from the remote Host. Perform the necessary operations before disconnecting.

Disconnect <name>

Disconnects the connection with the specified name.
?

?Net.exec

Connects to BBN socket 15600031 where-in the RSEXEC (Resource-Sharing Executive) is found.
?

?Status.of <host>

Performs ICP with the indicated host and prints its status.
?

?Echo.mode.is

Sets echo mode according to the following subcommand.

[no] remote

Turns off echoes generated by Telnet and signals the remote computer to generate echoes. Some Hosts are not yet equipped to handle this signal and may require additional action to cause the remote computer to generate echoes. If Telnet believes it is connected to a local half-duplex terminal, it will complain about remote echoes but do it anyway.

[no] local

Turns on Telnet generated echoes and signal the remote computer to not generate echoes. Note that Telnet never generates echoes for terminals it believes have local echo of their own.

[no] linefeed.for.carriage.return

TENEX translates carriage return to EOL, Telnet sends the EOL as the TELNET EOL (i.e. carriage return-linefeed). For some systems, the TELNET EOL is translated into carriage return. For these systems, the appropriate echo is carriage return. Other systems translate the TELNET EOL into carriage return-linefeed. For these systems the appropriate echo is carriage return-linefeed. This subcommand causes

the latter echo to be generated.

[no] control.character.echo.for <list of characters>

Turns on local echoes for the indicated control characters. Normally only control-G,J, and M (bell, linefeed, and carriage return) are enabled.

?

?Terminal.type.is

Allows the user to change Telnet's opinion of his terminal according to the following subcommands. Each command may be preceded by the word "no" to negate its meaning.

Half-duplex

Terminal generates its own echoes.

Full-duplex

Terminal does not generate its own echoes.

[no] lower.case

The terminal has lower case characters.

?

?Local.mode

If connected, this command prevents Telnet from returning to remote mode after each command.

?

?Remote.mode

If connected, this command causes Telnet to return to remote mode after each command. If not connected, it does nothing.

?

?No

May appear before some commands to reverse their action.

?

?Current.modes.are

Prints the state of connection terminal mode

flags, and all special characters.
?

?[no] character.mode

Causes each character typed to be transmitted as it is typed.
?

?[no] line.buffer

Causes Telnet to accumulate a line of text before transmitting. A line ends on linefeed or EOL or altmode (esc). The line may be edited with control-A, X, and R.
?

?[no] raise

Causes lower case letters to be transmitted as their upper case equivalents.
?

?[no] lower

Causes upper case letters to be transmitted as their lower case equivalents.
?

?[no] transparent.mode

Causes all characters to pass through Telnet and TENEX untouched. This is needed for special terminals such as the IMLAC using special character stream protocols.
?

?[no] case.shift.prefix.for

Allows the specification of the four case shift characters according the following four subcommands.

Lock.lower.case

Same as the "Lower" command. Subsequent upper case input will be converted to lower case.

Char.lower.case

Converts the following letter to lower case.

Lock.upper.case

Same as "Raise" command. Subsequent lower case input will be converted to upper case.

Char.upper.case

Converts the following character to upper case.

?

?[no] unshift.prefix

Causes all following characters to be unshifted. I.e. undoes both an upper case lock and a lower case lock.

?

?[no] quote.prefix

Causes the following character to be transmitted without regard to any special significance it may have.

?

?[no] synch.character

The specified character will be converted to the TELNET synch sequence. The TELNET synch sequence is used to cause the remote host examine its input stream to the current point for any special characters (interrupts, attentions etc.). All non-special may be thrown away.

?

?[no] attention.character

The specified character will be converted to the TELNET break or attention character. This character is equivalent to the attention, quit, or break key on certain terminals and may be necessary for using some systems. The !Break! command generates the same character.

?

?Concise

Turns off automatic command completion. Saves typeout at the expense of readability.

?

?Verbose

The opposite of concise.

?

?[no] fancy.command.interpret

Commands are checked character by character. If a character does not fit, it is ignored and not echoed (full duplex terminals only).
?

?[no] divert.output.stream.to.file

Causes all subsequent output from the remote computer to be written on the specified file. Use "No divert..." to stop this.
?

?[no] take.input.stream.from.file

Causes subsequent input to the remote host on the current connection to be read from the specified file; input to other connections and in command mode is still from the user's terminal. File is automatically closed and released at EOF; user may force this by "No take.input...", after escaping to command mode. File input, like terminal input to a connection, is active only in remote mode and when connection is current.
?

?[no] typescript.to.file

A record of the session is kept on a file including both input and output. This is useful for providing hard copy with scope terminals.

typescript.to.file <cr>

The file kept is TELNET.TYPESCRIPT;S in the LOGIN (not connected) directory.

typescript.to.file <filename> <cr>

The named file receives the typescript.

no typescript.to.file

The typescript file (if any) is closed and released; subsequent terminal activity is not saved.
?

?Escape.character=

The specified character becomes the Telnet escape character. This character must be a TENEX interrupt character. "?" will type what these are.

WARNING: If you have anywhere in your programming a control-Z you should change your escape character in TELNET to other than control-Z to avoid mishaps.
?

?Clear.output.character=

The specified character becomes the clear output buffer character. Typing this character generates an interrupt which causes the terminal output buffer and any accumulated output to be cleared.
?

?Help

Prints the file <SYSTEM>TELNET.HELP on the user's terminal.
?

?Describe <identifier>

Looks up the given identifier in the Help file and prints the accompanying description; an efficient way to read the Help text. Type "describe ?" to get a list of identifiers; command recognition operates on input of identifier.
?

?Netstatus

Runs <SUBSYS>NETSTAT.SAV.
?

?Socket.map

Prints a list of all current connection on the system. Optional arguments may be used to select a particular host and a particular connection state.
?

?Run

Runs the specified file. Like the EXEC's run command.
?

?Quit

Returns from Telnet to the superior fork (usually the EXEC). May be continued with no loss.
?

?Logout

Logs out the local job (not the remote one). Requires confirmation with a carriage return.

?

?Reset

Re-initializes Telnet producing an essentially virgin copy.
?

?Ddt

Enters ddt. If ddt is not loaded, this will result in an unexpected interrupt. No harm is done if this happens.
?

?Exec

Starts up an inferior EXEC under Telnet. This EXEC may be used like an ordinary EXEC to run subsystems etc without disturbing any existing connections. The Telnet escape character will return to Telnet however.
?

?Code

Transmits the character specified by the argument. The argument is a taken as an octal number unless preceded by "d" for decimal or "h" for hexadecimal. The argument may be preceded by "o" for octal.

The "code" command argument may be used as a command by itself and will cause the indicated code to be transmitted.
?

?!break!

Transmits the TELNET break character.
?

?!synch!

Transmits the TELNET synch sequence. Occasionally the "!synch!" command will work where the synch character will not since the command bypasses the buffering which may interfere with the use of the synch character.
?

?Write.modes.for.host

Causes the current mode flags to be saved on the <SYSTEM>TELNET.MODES file under the specified host. Requires write access to the file and is thus not available to ordinary users.

?

?Retrieve.connection.under.name

Retrieves the connection previously saved under the specified name.
?

?Wait.for.any.active.connection

Used with multiple connections to wait for and switch attention to the next connection that has any output waiting. Useful when several independent tasks are being run and you wish to know when one completes and switch to that task.
?

?[no] auto.switch.to.active.connection

Used to switch between tasks on several connections which may each be inactive for long periods. If the current connection is inactive on both input and output for a given number of minutes, Telnet will begin to hunt for any other active connection. If and only if one is found, that connection is made current. The "inactivity time constant" may be specified as any positive integral number of minutes if the "auto.switch..." command is terminated by a space. A <cr> terminator invokes the default value of 2 minutes. "No auto.switch..." disables this feature (current connection remains current until manually changed.)
?

?Where.am.I

Prints a summary of the local job, system, user, terminal and the remote host and socket.
?

?[no] Signal.waiting.output

Causes all non-current connections to print a message when output becomes available.
?

?Host.names

Lists all current host names with corresponding octal host numbers.
?

?List.connections

Lists the name, local socket, foreign host, and foreign socket of all connections.
?

?Flush.host

Marks all connections to the specified host as dead and sends a reset to that host. Requires wheel or operator special capability.
?

?Comments

An initial semi-colon causes the remainder of the line to be ignored. Useful for comments or typing to links.

XEROX

Internal Memo

R. Sauvain

To

Distribution

From

L. Bayles Holt

WRC/ISyL W114
Systems Research Area
Extension 25169

Subject

ECP and Alto User Notes

Date

18 January 1978

Here are a few notes that wont be published in the Whole Alto World Newsletter nor have they been otherwise documented, but which you may find helpful in one way or another.

1. Users who desire to install or re-install new versions of Bravo on some particular disk will find it helpful to remember that the following files are a minimum set of those required for a successful installation:

Bravo.Run, Bravo.Error, Bravo.Messages, Fonts.Widths, User.cm and a suitable font for FONT:0 as declared in the User.cm file. This also implies that User.cm cannot be empty. Anything else (except other fonts) is superfluous. In order to perform the install, one types:

Bravo/i cr

In order to perform a repeat of a previous Bravo session, the Bravo documentation suggests running BravoBug which repeats the session as desired and reports any errors or bugs encountered to an error accumulation file on Maxc. Usually this error reporting is not necessary and is indeed undesirable if the Gateway is down and there is no pathway to Maxc. A repeat session alone may be obtained by typing:

Bravo/r cr

Unless you are a frequent user of BravoBug.Run, I would suggest deleting it from your disk.

2. There is now a new version of Alto memory diagnostic, DMT.Boot which is more accurate, more reliable and allows Altos with disks left in Run mode to be awakened without jumping off the deep end. Previous versions were known to have problems with this feature. If you wish to be compatible with the rest of the Alto world, delete DMT.boot from your disk altogether. Your alto will retrieve it automatically from the ethernet at the appropriate times.

3. For those who occaisionally need hardcopy output which doesn't require the beauty of a Diablo or Slot printer there is a new version of ECP.run on <holt>ECP.Run which can be used to print files on the RTCC Eclipse driven

Versatec printer. A brief description of its use is described here.

ECP has three windows through which the user interfaces to the ECP operations. The first displays messages for the user to read when specific directions are required by ECP. It also echos the responses of the user to these requests. The second window shows the title and display of ethernet activity. After the user has selected a transfer command (described below) and no subsequent activity occurs in this window then it might be reasonably assumed that the Eclipse is incommunicado for the time being. There is no response explicitly from ECP for this occurrence and this is not likely to change due to the non-standard way in which the ethernet communications software was implemented.

The third window serves two functions, first as a menu display for the user selection of specific operations, and second, as a file content display window for files that reside on the Eclipse.

Operation of ECP takes place in two modes, one directly from user keyed commands and the other from a command file. Let it be noted that any command may be aborted by hitting the DEL key before a command has completely executed. In the key mode, all operations are displayed in the menu format and are reasonably self explanatory. Selections are made from the menu by bugging any mouse key with the cursor over the appropriate selection. These selections are:

Send a file to the Eclipse... which then requests a file name to be typed into the first display window. If the file does not exist, it will say so. A carriage return or space terminates the file name, whereupon a second carriage return will select the same name for the Eclipse file name or a new file name may be typed.

Retrieve a file from the Eclipse... reverses the above operation with the same protocols.

Delete an Eclipse file... requests the name of a file which is to be deleted. If the file doesn't exist, it will say so.

Display the Contents of an Eclipse file... again requests a file name, but this time the menu disappears and the contents of the requested file are regurgitated until the user hits any keyboard key. The display remains frozen until another key is hit. Then the display will continue until another key is pressed or else to completion, when it will request another key to return to the menu.

Other commands are used to select parameters that control the printing of an Alto file such as: the tab size in number of characters, the width of the left margin in number of characters, the length of the line, whether or not to automatically insert carriage returns and right-justify the text at the specified line length, and whether or not to print the headings at the top of the print out. The heading contains the file name, the date and the time.

Quit does just that.

Use Command File... causes the second mode of operation to commence. Commands for this mode, instead of being selected by the mouse, are taken from a command

[illegible] directly from a file [illegible] be typed [illegible] commands and [illegible] recommendations.
[illegible] form [illegible]

[illegible] or

[illegible] and parameters [illegible] and they are [illegible] of parameters, [illegible] are:

[illegible] and [illegible] file to [illegible] or @ [illegible] file...' or @ [illegible] the [illegible] current [illegible].

[illegible] editorial format -- [illegible] [illegible].

@ command -- [illegible] command for [illegible] the file [illegible] from the [illegible] executing command file [illegible] that this [illegible] also permits an infinite [illegible] command [illegible] current [illegible] warm [illegible] [illegible].

[illegible]

[illegible] [illegible] [illegible] [

file named ECP.cm or, if it doesn't exist, from a file name which must be typed in by the user. Commands for this mode are a superset of the menu commands and of course have character representations rather than menu representations. The command format is:

x par1 par2 cr

where x is the command character, and par1 and par2 are the command parameters desired. In some cases only one or no parameters are required and they are simply left blank. Failure to supply the required number of parameters, though, will produce bizarre results. The available commands are:

f newcommandfile — changes the currently executing command file to newcommandfile. The new command file is not automatically executed but is merely the next one in line to be executed when the 'Use Command File...' or @ command is specified by the user or current command file.

e eclipsefilename — deletes or expunges an Eclipse file.

@ commandfile -- will cause cammands for execution to be taken from the file commandfile. If no commandfile is specified, commands will be taken from the currently executing command file beginning at the beginning. Note that this will also continue ad infinitum and is not recommended. When a new command file is specified, execution of whatever remainder there is is the current commandfile will be lost forever. (This could change with urgent requests.)

d eclipsefilename -- displays a file.

r eclipsefilename altofilename -- retrieves an Eclipse file. If altofilename is omitted, eclipsefilename is used for both names.

s altofilename eclipsefilename -- the transpose of r.

p altofilename -- prints the named file on the Eclipse Versatec printer.

t nnn -- sets the tab size for printing to nnn.

m nnn -- sets the left margin to nnn.

l nnn -- sets the line length to nnn.

j -- compliments the switch which selects automatic carriage return insertion and right margin justification.

h -- compliments the switch which specifies whether or not to print headings.

The use of ECP at present is somewhat clumsy and non-standard due mainly to the large number of authors (at least 3) and is not likely to change significantly until a completely new package is written. However, suggestions for improvements to this version will be entertained.

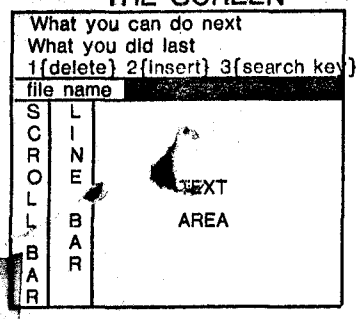
Distribution:
A.R.Axelrod 128
A.S.Ben David 128

Rayles Holt

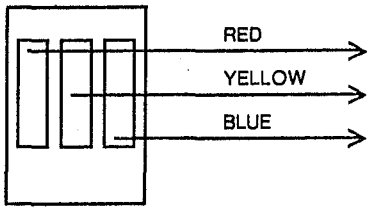
J.R.Blanchard 128
D.F.Blossey 128
J.E.Bollman 128
D.E.Damouth 114
O.M.Densmore 128
E.C.Faucz 114
T.M.Frey 114
R.N.Goren 114
L.B.Holt 114
J.A.Iverson 128
E.D.Jonas 128
K.T.Knox 128
S.J.Lanzatella 128
S.W.Marshall 128
D.M.Murray 128
P.G.Roetling 128
[redacted]

BRAVO VERSION 6.0 SUMMARY

THE SCREEN



THE MOUSE



WHAT MOUSE BUTTONS DO

Scroll bar	Line bar	Text area
Scroll up	Select line	Select character
Thumb	Select paragraph (sets cursor top)	Select word
Scroll down	Extend	Extend

The blank key to the right of BS is called CLR

COMMANDS

CHANGING TEXT

Delete selection

The following make a new paragraph if a paragraph is selected
- Insert <text> before selection
- Append <text> after selection
- Replace selection by <text>

<text> can be typed or selected. ESC gets the current buffer. Buffer n ESC makes buffer n current.

During type in:
- Backspace character with BS or ctrl-A
- Backspace word with ctrl-W
- Separate paragraphs with ctrl-CR
- Enter ctrl-<char> as <char> ctrl-S
- Enter TAB n as n with TAB held down

FILING

Get to read in a file
ctrl-Z for unformatted text
Put to write out a file

Hardcopy to print. Options
- Copies n
- Ears file
- CR to do it

Diablo output, with sub-options
- Start on page n
- Space to start or abort
- Continuous printing
- Reprint last page

File name goes into bar above window. ESC gets you one already there. Can edit file name like any other text.

MISCELLANEOUS

Undo works on most recent command.
ESC repeats most recent command, using current selection.
Everything selects whole document.
Normalize moves current selection to top of window.
([{ < ' " put indicated brackets around current selection.
)] } > extend current selection to closest matching brackets.
LF takes current selection as top of page, and moves selection to top of next page.

DEL cancels what you are doing.
Quit CR exits from Bravo
Time inserts current date and time.
Buffer 123 <text> sets buffer and makes it current. ESC instead of <text> just makes it current.

LOOKS

Basic looks. Mostly, SHIFT means NOT. Thus, Look b turns on bold, Look B turns it off.
During type-in, use ctrl-b for bold, etc. In command mode, give the Look command, then the letter alone (no ESC).

Text:
- bold
- italic
- visible
- graphic
- 0-9 font
- underline
- BS overstrike
- ←/↑↑ sub/superscript
- >/< upper/lower case
- CLR to reset

Paragraphs:
- centered
- justified
- nested (indent)
- open - add 12 pts space before paragraph
- q - add 6 pts space before paragraph
- ; document profile

Standard Fonts
- 0 Times Roman 10
- 1 Times Roman 8
- 2 Logo
- 3 Math 10
- 4 Hippo 10
- 5 Times Roman 12
- 6 Helvetica 10
- 7 Helvetica 8
- 8 Gacha 10
- 9 Helvetica 18

There are samples in the manual

Look has more cases, which take a distance as parameter. These cannot be used during type-in.

Text:
- Up raises text; default 4 pts.
- Down lowers text; default 4 pts.

Distances (measured from left or bottom page edge):
- As a number: in points (72/inch) - 123 or 123 pt.
  - in inches - 4.5 or 4.5" or 4.5 in
  - in cm - 4.5 cm
  - in blank widths - type that many blanks
- As an increment to current value: + or - followed by a number as above
- As the left edge of a character: select with RED
- Copied from an existing paragraph: select with YELLOW
- As a position on the screen: point with BLUE
- Defaulted to a standard value: type \.

Paragraphs:
- Left margin; default 85 pt
- F - left margin of first line
- P - left margin of other lines
- Right margin: default 527 pt
- X - space between lines; default 6 pt
- Y - space between paragraphs; default 12 pt
- Z - vertical tab; default none
- Keep on current page; default 0; 11" forces new page
- T d d n - sets n stops for table
- TAB n d sets tab n; default not set
- TAB = d sets even tab stops, default 36 pt

Look Same <selection> sets all text looks of current selection to be the same as those of <selection>
Look All <selection> sets all paragraph looks of current selection to be the same as those of <selection>
Look ? displays looks of selected text or paragraph in buffer 3.
Look hardcopy makes screen match hard-copy with 10% magnificatiobook Hardcopy clears it.
Look Magnify n does Look hardcopy with n0% magnification. Look Magnify Diablo n matches Diablo hardcopy.
Look \ adjusts margins according to Column line in document profile.Look | undoes it (but not after a Put)

PAGE FORMATTING

The following can be in the document profile; it must be the first paragraph, and have the profile property (Look;):
- Private Data Stamp: Yes/No X:d Y:d
- Columns:n Edge Margin:d Between Columns:d
- Margins: Top:d Bottom:d Binding:d
- Line Numbers: Yes/No Modulus:n Page-relative First Line:n
- Page Numbers: Yes/No X:d Y:d First Page:n Roman Uppercase Not-on-first-page
- Heading or Odd/Even Heading: Not-on-first-page followed by a heading paragraph, also with profile property

d is a distance: use - to measure from top or right
ctrl-L causes page (or col) break after current line
Use two ctrl-Ls for page break with multiple cols
Also note Look Keep and Look Z (vertical tab)

WINDOWS

Window
- RED to move boundary
- YELLOW to split (make new subwindow)
- BLUE to make new window

Kill
- RED to merge with window above
- YELLOW to erase contents, leave the window
- BLUE to merge with window below

Window Edge <distance> sets the left edge of the window at the specified point on the page; the default is 85 pt.

SEARCH

[illegible]n <text> starts search at second line, doesn't move selection
[illegible] <text> starts search after current selection, moves selection to the string found
Substitute <text> for <text> - works on the current selection. Usually you want to do Everything first.
Yes is equivalent to Replace ESC Find ESC; use Yes when you want to confirm substitutions.

REPLAY

>Bravobug will start a replay. >Bravobug/R will report a bug to the Bravo maintainers first, then replay.
Space will stop the replay and show you the number of the next command. You can then type:
- Slow to slow down typein
- Quick to speed it up
- Space to replay one more command
- Proceed to continue replay full speed
- Break n to stop before command n
- Terminate CR to stop replaying

ARITHMETIC

Operators are +* - *+ % / \(enter). An operator as a command combines the selection with the top of stack.
You can also type a number, followed by an operator. The stack is just like the one in a Hewlett-Packard calculator.

Also:
- ~ exchanges x and y
- ↑↑ rotates the stack

Calculator, with options:
- Fixed
- Scientific
- Engineering

operator to operate on x and y
0-9 to set digits after decimal pt
Radix n to set the radix (n=H for hex)