

\$19.95 US
\$28.95 Canada

WEBER

OPERATING SYSTEM SERIES

WORKING WITH
PICK

PICK

Working with PICK®

by

Walter C. Stagner Jr.

**Weber Systems, Inc.
Cleveland, Ohio**

The authors have exercised due care in the preparation of this book and the programs contained in it. The authors and the publisher make no warranties either expressed or implied with regard to the information and programs contained in this book. In no event shall the authors or publisher be liable for incidental or consequential damages arising from the furnishing or performance of any information and/or programs.

Published by:
Weber Systems, Inc.
8437 Mayfield Road
Chesterland, OH 44026
(216) 729-2858

For information on translations and book distributors outside of the United States, please contact WSI at the above address.

Working with PICK®

Copyright© 1986 by Walter C. Stagner Jr. All rights reserved under International and Pan-American Copyright Conventions. Printed in United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise without the prior written permission of the publisher.

Typesetting and Layout: Edwards Bros., Laura Miralia, and Beth Danz.

Contents

1. Operating Systems and PICK	13
What is Software	13
Some Important Terminology	14
Tour of the PC/XT Keyboard	15
PICK IBM PC/XT Keyboard Differences	17
The Operating System	17
Monitor and Utility Programs	19
Operating System Selection	19
History and Development of PICK	21
PICK Design Philosophy	21
Multi-User vs. Multi-Tasking	22
How Multi-User Systems Work	22
File Contention	24
PICK Components	24
PICK PC/XT Hardware Requirements	25
Application Software	26
PICK Statistics & Capabilities	27
PICK Advantages	27
PICK vs. Other Multi-User Systems	29
Summary	29
2. PICK Installation	31
Hardware Requirements	31
PICK Systems PC/XT Package	32
Hard Disk Partitions	32
Disk Partitioning For MS-DOS	33
Disk Partitioning for PICK	35
Installing A New PICK System	36
Upgrading An Existing PICK System	39
On-line Documentation Supplement	39
PC/XT Configuration and Special Terminology	41
Summary	44

3. PICK Terminology and Conventions	45
PICK Conventions	48
Entering Command Lines	48
Special Control Sequences	49
Introduction To PICK File Structure	51
System Startup and Shutdown	51
Summary	54
4. Getting Started	55
Obtaining An ID	55
Logging On	57
Case Usage	58
The TCL Interface	58
Selected TCL Commands	61
Large Printing	61
Displaying File Contents on the Terminal Screen	62
Listing Attribute Definitions	63
Listing Dictionary Files	64
Listing Stored Procedures (PROC'S)	65
Listing Active Users	66
Listing Verbs	66
Logging Off	67
Setting Tab Stops	67
Setting Terminal Characteristics	68
Displaying The Time and Date	72
Displaying Who is Logged On	73
Logon PROCs	73
5. The File System	75
File Structure	76
Individual Files	76
File System	76
System Dictionary	77
Master Dictionaries	78
File Dictionaries	79
File Data	79
Overall View of the PICK File System	79
Items	80
Data Items	80
Dictionary Items	82
File Handling	83
File Creation	83
Modulo	84
Modulo Selection	86
Clearing Files	86
Removing Files	87
Copying to the Terminal Screen or Printer	89
Copying to Another Item in the Same File	90
Copying To Another File	92
File Ownership and Security	92
Ownership	93
Security	93

6. PICK Command Tutorials	97
Introduction	97
Leaving the System Unattended	98
Terminal Control Language	98
Stopping TCL Command Execution	98
TCL Statement	99
SYSPROG Account	100
PICK Command Summary	102
Session 1: Multi-User Operation	103
Session 2: ACCOUNT-RESTORE	105
Session 3: ACCOUNT-SAVE	106
Session 4: COLOR	107
Session 5: CREATE-ACCOUNT	109
Session 6: CT (Copy to Terminal)	111
Session 7: DELETE-ACCOUNT	114
Session 8: FDISK	115
Session 9: FILE-SAVE	117
Session 10: FFORMAT	120
Session 11: Listing Utilities	122
Session 12: The LOGON and OFF Commands	130
Session 13: MSG	133
Session 14: POVf	134
Session 15: POWER-OFF	136
Session 16: REBOOT	136
Session 17: Restoring the System	136
Session 18: RUNOFF	139
Session 19: SET-TIME and TIME Commands	141
Session 20: SET-DATE	142
Session 21: STARTPTR	143
Session 22: TERM, TERM-TYPE, SET-LPTR	147
Session 23: WHO	153
Summary	155
7. The Editor	157
Entering the Editor	158
Editing Data Items	158
Editing Dictionary Items	161
Tutorial on the Editor	165
Displaying Item Contents	165
Displaying the Current Line Number	166
Suppressing Line Numbers	167
Changing the Number of the Current Line	167
Moving Around in the Editor	167
Top	167
Bottom	168
Up	168
Down	168
To a Specific Line	169
Correcting Errors/Data Replacement	169
Finding a Specific Line	171
Inserting New Lines	171

Deleting Lines	174
Updating the Disk File	175
Editor Reference Guide	176
Invoking the Editor	177
Editor Command Reference	179

8. The PROC Language/Stored Procedures 195

Introduction	195
PROC'S Defined	196
Storing A PROC	197
Termination of PROC Control	198
Logon PROC'S	198
Components of the PROC Programming Language	200
Input/Output Buffers	201
Labels	201
Relational Operators	203
Moving Data Between Buffers: The A Command	204
Relational Testing: The IF Command	204
The C Command: Comments	205
Screen Control Commands	205
Constructing TCL Commands and Clearing Buffers	207
PROC Execution/Debugging and Termination	208
The GO Command	209
Clearing Buffers and Inputting Data	209
The O Command	210
The Link Command	210
Using the PROC Language	211
Getting Help	217
Summary	219

9. PICK System Maintenance 221

Introduction	221
Peripheral Attachment	222
Restoring Contiguous Disk Space	224
What to Do About Errors During FILE-SAVE/Restore	225
System and System-level Files	227
BLOCK-CONVERT	228
PROCLIB	228
ERRMSG	228
Accounting History File	229
SYSPROG-PL	230
Using PROC'S on Another Account	231
Sharing PROC'S Among Accounts	234
System Backup	239
Summary	241

10. Using PROC'S As Menus 243

Introduction	243
The Functions of PROC	244
Starting the System: The COLDSTART PROC	245
Logon PROC'S	245

The HELP PROC	246
Running Application Software From a Menu: The MASTER PROC	248
Performing System Maintenance: The UTILITY PROC	251
Some Examples of Utility Menu Choices	254
Summary	257

11. The PICK ACCESS Query Language 259

Introduction	259
Data Files	260
ACCESS Commands and Sentences	260
ACCESS Tutorial	261
Using ACCESS	262
ACCESS Syntax	263
ACCESS Error Messages	263
ACCESS Verbs	263
Selection Criteria, Modifiers, Options	264
Using ACCESS Verbs	265
Storing ACCESS Command Lines	273
Summary	273

12. A User's Guide To PICK Software 275

Introduction	275
PICK and Application Software	276
Reentrancy	276
Task Prioritization	276
Upward Compatibility	277
Why You Probably Haven't Heard of These Software Packages	277
Three Types of Software Everyone Can Use	278
Database Managers/Interfaces	279
The WIZARD Program	282
Word Processors	288
The JET Word Processing System	289
Cyclone III Word Processing	293
Spreadsheets	296
Cyclone III and CompuSheet	297
Office Automation Software: TotalWare	300
General Accounting Software	302
Vertical Market Software	303
Where To Find PICK Software	304
Summary	306

13. Further Considerations 307

Introduction	307
PICK Open Architecture Release 1.0	308
A Brief Discussion of Open Architecture Features	308
Open Architecture on the PC/XT	309
PICK on the IBM PC-AT	310
Unique Features of the AT Implementation	310
PICK Wish List	312
The Future of PICK	313
PICK Hardware Compatibility	313

Growing With PICK	314
Keeping in Touch with the PICK Community	315
PICK User's Groups	317
Where to Buy PICK Systems	317
Fujitsu Microsystems of America	318
Appendix A	319
Index	325

Acknowledgements

I would like to acknowledge the contributions of the many people and companies who helped me put this book together.

Frank Petyak, PICK Systems Inc.

Dianne McArthur, PICK Systems Inc.

Stephen C. Setzman, Automation Technology

Gus Giobbi, IDBMA

John Treankler and Sherry Elrod, Jet Software Corp.

John Thompson, Chicago PICK User's Group

Ken Simms, Automatic Programming Inc.

Mike Sutkowski and Steve Lazarov, Lazarov Software Systems, Inc.

Thomas Dunn, Kopic

Susan Eubanks and Carol Lovercio, Interactive Systems

Allen Stark, Innovative Business Systems, Inc.

Paul Barnaby Jr., BCS Computers

Barbara Stagner

John Kim, Kimtron Corp.

The author will be grateful for any suggestions from PICK users for future improvements to the book.

Introduction

This book is designed to be a non-technical tutorial and reference guide to using the PICK operating system, to illustrate a method of creating a “turn-key” PICK installation, and to provide a supplement to the documentation provided by PICK Systems Inc., the developers and marketers of the PICK operating system. This book can be used by managers or owners of businesses who may be considering the purchase of a PICK based computer system as well as those who are already PICK users.

The book is divided into two sections, the system manager section and the system user section. The system manager section is designed to be used by the more technically inclined person responsible for the day-to-day system operation. The system user section is designed for those who want to learn how to put the computer to work without learning “computerese,” or for those who want to learn about PICK application software.

For the purposes of this book we assume PICK to be installed on an IBM PC/XT.

1

Operating Systems and PICK

People who work with computers have a unique vocabulary. The chapters that follow assume that you are familiar with certain basic computer terms and concepts. In this chapter we will provide a working vocabulary for the later chapters, give you a tour of a terminal keyboard, which you will need to be familiar with in order to work a computer, and introduce you to the history and design philosophy of the PICK operating system. If you are already an experienced PICK computer user, familiar with the terminology, you may want to skip some of the material and start with the section on the history of PICK.

What Is Software?

Of all the areas of the computer business, the area of software is one of the most profitable, because all computer systems, regardless of size or cost, need software to operate. Software can be defined as the programs or lists of instructions that tell the computer what to do. Computer software is classified as either **applications software** or (operating) **systems software**. Applications software packages are designed to undertake a specific (generally business- orientated) tasks such as word processing, database management, financial modeling, and accounting. Systems software is generally invisible to the end-user but nonetheless performs important functions which will be described in this chapter.

Some Important Terminology

In this section, we will define some computer terms that will improve our understanding of what an operating system is. In order to begin familiarizing ourselves with PICK, we will indicate in parentheses the PICK equivalent term next to each section title if one exists.

Files

Files are similar to the file folders one stores in a file cabinet. One example of a file might be a name and address file. Another file might be all of your accounts receivable transactions for one month. To remember what is in your file folders, you always label them right? Similarly, your computer files have labels. You might file or store your names and addresses under the name of LIST. The word LIST is referred to as the **Filename**.

Records (Items)

Records are the individual items that make up each file. If the LIST file contains a person's Name, Address, City, State, and Zip Code (visualize an index card like the one below) each individual's information would constitute a record.

Name _____

Address _____

City _____

State _____

Zip Code _____ - _____

Figure 1.1. Example of a record

The number of records you can fit in each file is generally determined by the size of the file. The size of the file is determined by the storage capability of the hardware and the application software.

Fields (Attributes)

Each individual data item in a record is known as a **field**. In figure 1.1. Name, Address, City, State and Zip Code are fields. Each record or item can be composed of a number of fields.

Data

Data is information to be stored in files such as names and addresses, accounts receivable transactions etc. Another way of explaining data is to define it as information you put into the computer in a format that both you and the computer can understand. When you fill in the index card above you are filling in the data.

Programs

Programs are files (or a series of files) containing instructions that tell the computer what to do. Even though these programs are files, they are different for those files which contain your data.

A Tour of The IBM PC/XT Keyboard

In this section, we will describe the functions of the various keys found on a typical terminal keyboard as depicted in figure 1.2.

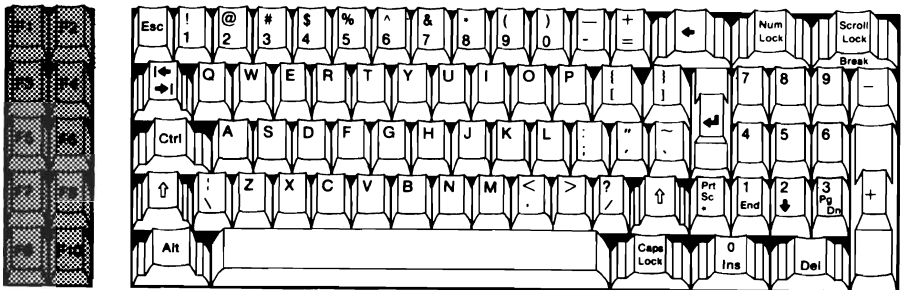


Figure 1.2. IBM PC/XT keyboard

The Return Key

As you begin to type, the characters entered will appear on the display screen. Although everything you type is *temporarily saved* in the computer's memory, the computer does not try to interpret what you type as an instruction or command until you press the return key. The return key tells the computer that you are finished typing. When you press return, the computer examines everything on the line just entered. If that entry constitutes an instruction that the computer can understand, it will act accordingly. If the computer does not understand the entry, an error message such as ? VERB may be displayed. The computer is telling you it doesn't understand your entry. The return key is labeled with a large left arrow (↵) .

The Shift Key

The shift key on the PC/XT is labelled with a large up arrow (⇧). This key allows letters on the keyboard to be generated in either upper or lower case. In addition, it allows some keys to produce two different characters. One character is generated by pressing a key with the shift key held down and another by pressing the same key without the shift key. The PC/XT keyboard has a caps lock feature which allows you to generate upper case letters without holding the shift down. Caps lock only affects the keys that type letters. The PICK system is always in upper case mode when it is first activated. All letters typed at the keyboard are interpreted to be upper case. The shift key can be used to input lower case letters. If the caps lock key is used, all keyboard input will be lower case. The user will usually want to leave the keyboard in the upper case mode because PICK does not "understand" operating system commands entered in lower case letters.

The CTRL Key

CTRL is a contraction of the word control. The CTRL key is always used in conjunction with another key just like the shift key. You hold the CTRL key down while you press and release another key. Examples of how to use the CTRL key will be provided later.

The Function Keys

These keys are labelled F1 through Fn where n is the number of function keys provided. The PC/XT keyboard has 10 function keys labelled F1 through F10. Function keys are generally used for specific commands in application software. There is a program provided with PICK which allows you to configure the function keys to work within the PICK system. See the PICK User Reference Manual for information on function keys.

The Other Keys

These keys include the letters of the alphabet, numbers zero through nine and various symbols. Many typists do not distinguish between the number zero and the letter O or the number 1 and the lower case letter i. When using computers, you must get into the habit of typing the exact characters you desire. Guard against substituting zero (0) and the letter O and 1's and the letter i.

PICK IBM PC/XT Keyboard Differences

Using the PICK system on the PC/XT causes the following changes on the XT keyboard:

- The numeric keypad only generates numeric characters
- Backtabs cannot be generated from the keyboard
- The print screen function is disabled
- The function keys labelled F1 through F10 are disabled but may be configured using the SET-FUNC program
- The ALT key is disabled except during a system reset sequence (CTRL-ALT-DEL) or if the ALT key is used in conjunction with a function key in the SET-FUNC program.

The Operating System

The operating system is a program, or a series of programs, which “connects” your hardware (the computer and its peripherals) and software (your programs) and interfaces them together with you, the user. When you want to type letters, fill-in cards or perform accounting functions, it is the job of the operating system to “call up” the applications program(s) from the disc or other storage device, put them into computer workspace (RAM) and allow you to put in your data. When you are finished with the program, control of the computer reverts to the operating system, and the computer waits for you to tell it what to do next.

Operating systems are generally classified as software. Since an operating system is software, a portion of the operating system must be stored in RAM in order to be used. Some of the operating systems are simple and occupy very little of the computer's memory, while others are large and require huge amounts of memory. The operating system manages the physical resources of a computer system. Physical resources include input/output devices (such as terminals and printers), storage devices (such as floppy disks and hard disks) and RAM.

Some examples of operating system functions include:

- Overseeing the transfer of data from RAM to a storage device such as a hard disk (the operating system makes sure that the data gets stored exactly as it was entered)
- Allowing you to store your data in files, each with a unique name for retrieval at a later time
- Keeping a directory of all files stored on your disks much like a table of contents in a book
- Allowing you to erase unnecessary files (and reuse the disk space)
- Allowing you to communicate with the computer via a terminal/keyboard
- Allowing you to output information stored in the computer to the printer

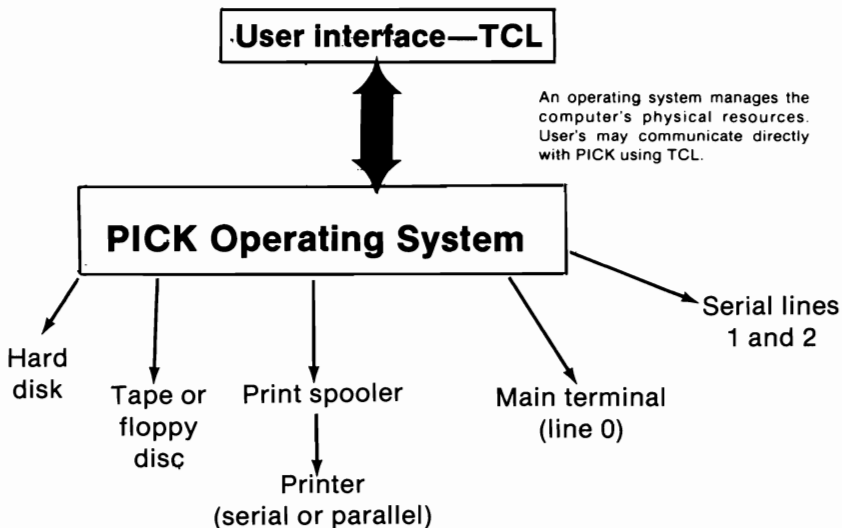


Figure 1.3. How an operating system works

Monitor and Utility Programs

The operating system is generally divided into two components:

- * **Monitor** (think of it as a supervisor or a policeperson directing traffic at a busy intersection)
- * **Utilities** a complete set of programs which manage computer “housekeeping” functions such as listing, copying and erasing files you no longer need

The monitor provides the interface (or a means of communication) between the hardware, the user and application software. When a computer is initially turned on, a special program called a *bootstrap loader* loads the required startup monitor programs into memory. The bootstrap loader is a permanent program stored in a special ROM chip called a **PROM** (short for Programmable Read Only Memory). These monitor programs are stored on the hard disk and transferred to RAM as needed.

Diagnostics for checking RAM are stored in the PC/XT’s ROM chip. If everything checks out, the PICK system will automatically boot from the hard disk after it has been installed.

Utility programs are operating system programs that allow you to erase files, make backup copies, print files etc. Since all of these programs would take up space in RAM and take space away from your data and application software, they are stored on the hard disk and only called into action when needed.

The monitor is the component of the operating system which actually accomplishes the task of “calling-up” the utility programs.

Operating System Selection

Choose carefully when selecting an operating system. The more popular the operating system, the more application software is available. What? You mean you have to take the operating system into account when looking for application software? The answer is YES. Selection of the wrong operating system can have a monumental negative effect on the functionality, ease-to-use and cost of your computer system.

In addition, if the operating system cannot accommodate growth and increase “computing power” dictated by your business and the application software, you may be in for some unhappy surprises. Consider the criteria depicted in figure 1.4.

- **Single or Multi-User**

In a multi-user arrangement, the computer is able to perform tasks for more than one user simultaneously. A time-sharing multi-user system can save you money by sharing a single microprocessor and requiring only add-on terminals to expand the number of users. The initial cost for the computer is somewhat higher than a single-user computer, but this cost may be justified in the long term.

- **Ease of Use**

This is important because most computer users are not computer “gurus”. The operating system should provide help menus, English (not computerese) based commands, and error handling routines that make it next to impossible to do harm to the system, programs and data files.

- **Security**

Security is important, particularly in a multi-user arrangement, to prevent unauthorized access to data files or parts of the computer system. Provisions for file and record locking should also be investigated to avoid damage caused to data files by two or more users accessing the same information simultaneously.

- **Portability**

Portability refers to the ability of the operating system to run on a variety of hardware. The more hardware manufacturers who use a particular operating system, the larger your selection of hardware. In addition, portability refers to the ability of applications software to be easily “moved” from the 8-bit version of the operating system to the 16-bit version.

- **Memory Requirements**

Memory requirements for the operating system will determine how much room is left to run applications software. A good 8-bit operating system will occupy no more than 20K of RAM. This means that in a system with 64K of RAM, there should be plenty of room left for most applications software packages.

- **Ability To Accept A Wide Range of Peripherals**

This is important because you do not want to be restricted to a limited selection of terminals, printers, modems, disk drives, etc.

Figure 1.4. Criteria and checklist for selecting an operating system

There is generally NO compatibility among operating systems. Once you decide on a particular operating system you cannot change to a different one (unless you completely change your application software) if you find one that better suits your needs later.

The History and Development of PICK

The Pick system has its roots in the mid-1960's. Dick Pick, a programmer for TRW, was chosen to develop an information management system. The system was to be used in the Cheyenne helicopter development project. No particular hardware was specified. To add flexibility to the system it was decided that variable length records be used to store files. This would permit a data file to expand or contract as a user's needs dictate.

In the late 1960's the resulting information management system, which Pick developed, was run as an application package on various mainframe computers. (At this point PICK was not a stand alone operating system. It used the machine's inherent operating system.) The system did not work particularly well as an application package. In the early 1970's, Dick Pick rewrote the system. He transformed it into a stand alone operating system with the information mangement system in its hub.

In 1973, the Pick Operating System was implemented for the Microdata Reality machine. Since that time the Pick system has been adapted to many other mainframe and microcomputers. With the advent of the more powerful 16-bit and lower priced microcomputers, the PICK system is now available for owners of small businesses via its recent implementation on the popular IBM PC/XT.

The Design Philosophy of PICK

The PICK system is specifically designed to manage information. It provides a means for entering, storing and retrieving information with a few keystrokes. It transforms an IBM PC/XT into a true multi-user system which can accommodate three users and a printer without the use of special software, which in many cases is untested. As can be seen in table 1.1 there are a number of implementations of PICK. The fact that it has been implemented on mainframe computers such as IBM, DEC and Hewlett-Packard should serve as a testimonial to its performance and reliability.

Table 1.1. Summary of PICK implementations

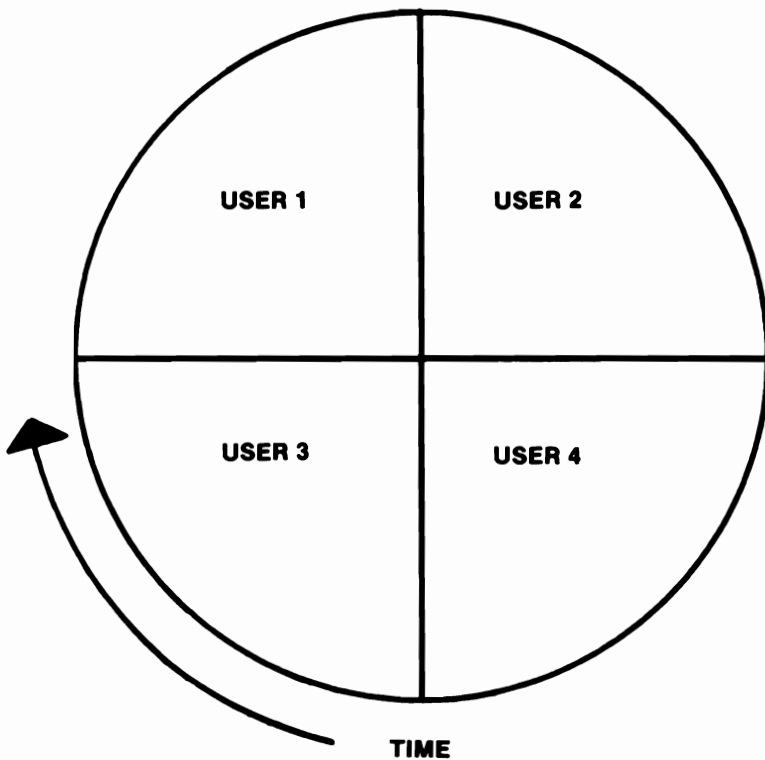
Microdata 1600
Intertechnique Multi-6
Evolution 280
Ultimate Honeywell Level-6
Ultimate DEC LSI-11
ADP Hewlett-Packard HP 3000/Series 30
ADDS Mentor Z8000
Datamedia Motorola M68000
C.D.I./IBM Series I
Altos I8086
General Automation Zebra M68000
S.M.I./IBM 4300
PICK Systems IBM PC/XT
S.M.I./IBM CS9000
Pertec Sabre
TAU M68000
Wicat M68000
Climax M68000
CIE 680 Series
Fujitsu I8086
Nixdorf 8090

Multi-User vs. Multi-Tasking Systems

Multi-user refers to a system that allows several individuals to share the use of a computer. Multi-tasking capability enables the computer to appear as if it is performing several “jobs” simultaneously. PICK is inherently a multi-user system. Its only multi-tasking capability is provided by the spooler which allows documents to be printed to the disk and printed when the attached printer is not “busy.”

How Multi-User Systems Work

A multi-user system allows more than one terminal to share access to the computer at one time. Although the computer itself has only “one brain” or microprocessor, the operating system allows users to get a “slice of time” allocated to them to perform their tasks. In this sort of time sharing arrangement it appears to each user that they have their own computer. Because of this sharing arrangement, only one copy of the operating system and application software is necessary for the computer.



Each user is allocated a “slice” of microprocessor time by PICK. A fast microprocessor is important to minimize system slowdown as more users are added.

Figure 1.5. Time slice arrangement

File Contention

When more than one user is working with the computer, a situation can arise when the users will want to work with the same disk file. It is possible that two users will want to make changes to the file without knowledge of the other user. Some method of file access must be provided to avoid *contention*. File contention is essentially a sort of competition between two users to access the same file. PICK BASIC eliminates this problem by providing a means of writing application software in a way that when a user is working with a file, all other users are temporarily “locked out.” This prevents data files from ending up in a state of chaos.

Note: this file locking mechanism is not an automatic feature of PICK. Programs must be written utilizing this feature of PICK BASIC. In addition, care must be taken when writing programs to avoid deadlocking. This occurs when any user is locked out if any file for an excessive amount of time.

The PICK Components

PICK can be divided into the following component parts:

- The ACCESS processor
- The PICK/BASIC processor
- Terminal Control Language (TCL)
- The PROC processor
- The EDITOR processor
- File management processors
- Utility processors

The ACCESS processor is an English-like data retrieval language used for accessing data files created within the PICK system. ACCESS provides the user with the ability to selectively sort and retrieve information and generate custom formatted reports which may be displayed on the terminal or printer.

PICK/BASIC is a programming language specifically designed for database management on the PICK system. It is an extended version of the BASIC programming language developed at Dartmouth college in 1963.

The Terminal Control Language (TCL) provides the primary means of communication between the user and the PICK system processors. The TCL prompt is a right arrow (>) displayed at the left edge of the terminal. Whenever the TCL prompt is displayed, the system is waiting for you to give a command. It is the key that allows you access to system and application programs and it serves as traffic director within the system.

PROC refers to stored procedures. PROC'S (pronounced "procks") are special instructions written in a language similar to the Job Control Language (JCL) found on large mainframe computers. The PROC processor allows you to store a complex series of TCL operations which can then be executed by typing a single command. In addition, customized menus can be created which effectively isolate novice users from the operating system.

The EDITOR processor permits you to modify any item in a PICK database. The EDITOR is the means used to create and/or modify PICK/BASIC programs, PROC'S etc.

The File Management Processors provide you with the ability to create and manipulate files within the PICK system.

The Utility Processors provide you with features such as print spooling, data backup, system security via creation of user accounts and system accounting, and starting and stopping the system.

To summarize, all of the above items together constitute the PICK operating system software. Each will be discussed in detail in later chapters.

PICK IBM PC/XT Hardware Requirements

The following components are required to run PICK on the PC/XT:

- IBM PC/XT or IBM PC with IBM expansion chassis
- A minimum of 256K RAM
- A monochrome or color graphics monitor (serves as main terminal)
- One or two additional terminals may be connected via IBM asynchronous communication port adapters
- Peripherals such as printers may be connected to the parallel port or to the optional communication port adapters

Note: If a printer is connected to a serial port this limits the number of users to two.

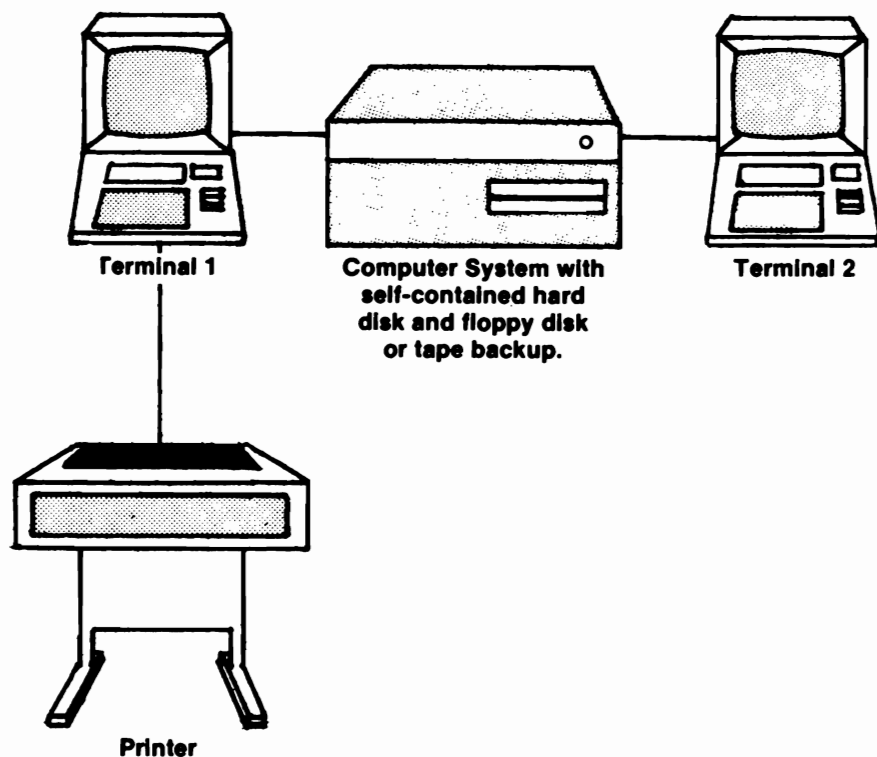


Figure 1.6. Typical PICK PC/XT system

Application Software For PICK

A wide variety of applications software is available for PICK systems from special purpose medical accounting packages to the more common spreadsheet and word processing programs. Since the PICK system has a built-in database manager there is no need to buy special software for this purpose.

PICK systems Inc., the marketers of the PICK system, publish an application software directory called PICK Hits. You can get it directly from the company. In part two of this book we will look in detail at some application software.

Some PICK Statistics and Capabilities

As of this writing, over 20,000 copies of PICK have been installed at end user locations world-wide. The popularity of PICK is due to the following attributes:

- PICK is a time tested working multi-user operating system.
- Each user shares a common microprocessor so adding more users to the system is accomplished by adding an inexpensive terminal.
- The user environment is "friendly."
- There is built-in security for important files, complete with the capability of assigning passwords.
- There is a built-in facility called PROC which can be used to create interactive "menus." This permits "housekeeping" functions and application software to be executed by making choices from menus. Novice users can easily learn to use the system.
- This facility ELIMINATES the need to purchase any extra "help" programs as with other operating systems. As new software applications are added, the menus can easily be modified to accommodate new programs.
- Print spooling allows all system users to share one printer.

Advantages of PICK

We have already discussed the advantage of PICK's database management facility. For the price of the operating system alone (\$495.00) you get the database manager too. Other strengths include:

- It is not new. It has been tested in the field for many years.
- It was designed from the ground up as a multi-user system.
- Relatively low cost.
- The Virtual Memory Manager design.

Virtual Memory Manager

Virtual memory systems have been used primarily in large main-frame computer systems. The PICK system has been designed to take full advantage of this concept. Virtual memory means that the hard disk storage device is treated as an extension of the basic 256K RAM

required to operate the system. Many similar operating systems require huge amounts of memory to operate efficiently. Not so with PICK.

Data in a PICK system is organized into 512 character segments called *frames*. These frames are stored on the disk. As a frame is needed for processing, if it is not already in RAM, it is transferred there automatically. The frames no longer needed are written back to the disk.

The virtual memory design of the PICK system eliminates the burden of running out of memory for either data or programs.

Disk space allocated for PICK

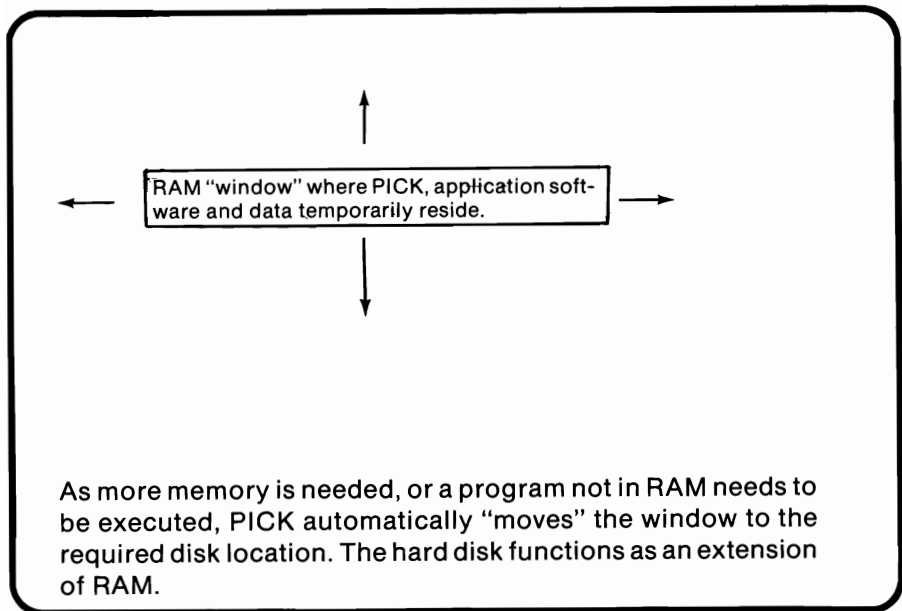


Figure 1.7. Virtual Memory Concept.

PICK Compared to Other Multi-User Microcomputer Operating Systems

Because the PICK system is a 16-bit operating system, comparisons can only be made to other 16-bit operating systems. In addition most personal computers have used only single-user operating systems such as MS-DOS or CP/M.

Other 16-bit microcomputer operating systems are:

- UNIX and XENIX
- THEOS 86/286

Both UNIX and XENIX were developed as multi-user systems. Neither one is particularly user-friendly nor do they have a built-in data base manager. However XENIX is a multi tasking operating system. It can run multiple tasks concurrently. XENIX also requires a minimum of 384K of RAM in order to install the complete version.

THEOS 86/286 were also developed as multi-user systems. They are user-friendly. It requires a minimum of 384K of RAM for installation.

PICK is unique in that as long as the minimum memory requirements are met it is not necessary to increase RAM. PICK's virtual memory design allows the hard disk to function as an extension of memory. This means that any programs and data files can be accommodated provided that the hard disk space has not been exceeded. Increasing the amount of RAM will improve performance.

Summary

PICK was designed to function as a multi-user system and to run ONLY business-oriented software such as accounting, word processing, order entry, etc. In addition, its powerful database management system makes it easy for you to store and retrieve your company information. If your needs dictate a multi-user business system, you may want to strongly consider PICK. Even if you need only one terminal at present, you can start with a basic IBM PC/XT system and add additional terminals (which is less expensive than buying individual computers) as the need arises.

2

PICK Installation

In this chapter we will discuss PICK installation. This chapter is designed to aid the individual responsible for getting a new PICK installation up and running. For our discussion we assume the computer hardware to be an IBM PC/XT with a minimum of 256K memory. The version of PICK we are using is the PICK PC System version 1.3. We also will discuss how it is possible for more than one operating system to reside on the same disk as PICK and be accessible for use. If you have purchased your computer system from a systems house or turnkey system supplier it is possible that PICK has been installed and is ready to go to work for you. Even if that is the case you may find a review of the PICK installation process helpful.

Hardware Requirements

The minimum computer hardware requirements to install PICK are:

- IBM PC/XT or IBM PC with expansion chassis
- A minimum of 256K RAM
- Monochrome or color graphics monitor
- 300 contiguous hard disk tracks
- Printers may be connected to the serial ports or parallel port

If you are uncertain of the PC/XT's memory, you can easily determine it. When the IBM PC/XT is switched on, it executes a memory checking routine. While performing this function, it displays the amount of memory available on the screen. The last number displayed before the operating system is booted is the amount of memory you have to work with.

The PICK Systems PC/XT Package

Included in every PC/XT package are the following items:

- 5 Diskettes, labeled:

- PICK PC SYSTEM #1
 - PICK PC SYSTEM #2
 - PICK PC SYSTEM #3
 - PICK PC DATA FILES #1
 - PICK PC DATA FILES #2

- PICK User Reference Manual
- Installation and upgrade guide
- End-user license agreement

If any of these items are missing contact your dealer immediately. If any of the diskettes are missing, installation cannot be completed.

Hard Disk Partitions

If PICK is to be the only operating system installed on the system, you can proceed to the following section entitled "Installing A New System." If more than one operating system is to be used, PICK must be installed last because during installation PICK assumes it will occupy all disk space not already partitioned. If you are currently using MS-DOS, you must be certain it does not occupy the entire 305 cylinder disk partition present on a standard IBM PC/XT hard disk because there would be no room for PICK.

The amount of disk space reserved for MS-DOS on a PC/XT system is expressed in cylinders. A cylinder is the unit of storage on the hard disk that is used to define partitions. The size of the hard disk determines the number of cylinders it contains as well as the size of each cylinder. A ten megabyte PC/XT disk contains 305 cylinders. Each cylinder contains 32,768 bytes of storage area.

You must create the appropriate size MS-DOS partition to match the requirements of your application software and data files. One megabyte of disk space equals approximately 32 cylinders. Consult your application software documentation for disk space requirements.

Note: Once partition sizes have been determined, they cannot be changed easily. When in doubt allocate a larger partition than you need.

Setting Up A Disk Partition For MS-DOS

To begin partitioning the hard disk:

- Insert the floppy disk containing the MS-DOS operating system into the floppy drive.
- Power the computer on.
- After the MS-DOS has booted from the floppy disk you will be prompted to enter the date and time. You can either enter the current time and date or press [RETURN]. When the system prompt appears on the screen, type:

>FDISK [RETURN]

The menu depicted in figure 2.1 will be displayed.

- Type 1 and press [RETURN]. This allows you to define a new DOS partition. The following prompt will be displayed:

Do you wish to use the entire fixed disk for DOS
(Y/N).....[Y]

Choose one of the following:

1. Create a DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Data

Enter Choice: []

Figure 2.1. The FDISK menu for MS-DOS

- The system automatically defaults to Y. We do not want to use the entire disk so type N[RETURN]. The screen will display:

No partitions defined

Total disk space is 305 cylinders.

Maximum available space is 305 cylinders starting at cylinder 0.

Enter partition Size.....[305]

- Type the value of the size of the DOS partition, in cylinders. Replace the default value of 305 with 35 and press [RETURN]. The screen will display:

Enter starting cylinder number...[0]

- The default value of 0 is ok so press [RETURN].
- Press the ESC key. The FDISK menu (see figure 2.1) will be displayed on the screen.

By allocating a 35 cylinder disk partition, we have allocated a disk partition of approximately one megabyte for MS-DOS. If you need a larger partition simply allocate more cylinders.

Setting Up A Disk Partition For PICK

Once all of the disk space has been allocated it cannot be reallocated (partitioned) without deleting one or more partitions. There is no option on the FDISK menu for changing partition size. Although we have allocated an MS-DOS partition, we assume that PICK is to be the primary operating system.

In order for PICK to be installed it must find a large portion of contiguous disk space. If other operating systems are to co-reside with PICK, PICK must be the last one installed.

PICK Storage Requirements

The hard disk supplied with the standard PC/XT is capable of storing up to 10 MB (ten million characters) of information. In this space the PICK operating system files plus data files must reside. Calculating disk storage requirements for PICK differs from the procedure for MS-DOS.

In order to calculate the amount of disk space required by the PICK operating system files, we must have the following information:

- Standard IBM 10 MB drives have 4 Heads and 305 cylinders
- When using drives with 4 heads PICK requires 75 cylinders (consult the chart in PICK documentation for other drive specifications)
- Hard disk space is allocated by cylinders. A standard IBM drive has 4 tracks on each cylinder.
- Each track can store 8704 bytes (characters)

Using the information we can see that each cylinder can store:

4 x 8704 bytes or 34,816 bytes

since PICK requires a minimum 75 cylinders for its files we can calculate that PICK occupies:

75 x 34,816 bytes or a total of 2,611,200 bytes

To convert this into Megabytes divide this number by 1 million.

On a PC/XT PICK occupies approximately 2.6 MB. Assuming that PICK were the only operating system on the hard disk, the remaining disk space would be available for data files.

Installing A New PICK System

The PICK PC system consists of five 5¼ inch floppy diskettes, one black and four grey ones. The black diskette is labelled #1 and is the diskette the system is booted from. The first two grey diskettes are labeled #2 and #3. The remaining grey diskettes are labeled Data Files #1 and #2 respectively.

To begin a new PICK installation:

- Insert the PICK PC SYSTEM #1 diskette and power on the system. The PICK installation sign-on and proprietary messages are displayed.

Note: if an error message such as "Protection Violation" is displayed, a hardware problem is possible. It is critical with PICK that the floppy disk drive be accurately calibrated and that the heads are not worn or damaged. If you cannot get past this point contact your PICK dealer.

- The screen will then display:

Enter number of hard disk(s) [1 or 2]=

Enter 1 and press [RETURN]

- PICK will display a menu as depicted in Figure 2.2

◇ THE PICK SYSTEM INSTALLATION MENU ◇

Copyright (c) 1985, Pick Systems

- | | |
|------------------------------|---------------------------------|
| 1. IBMPC XT 10 MB std. drive | 2. IBMPC+Sigma 33MB full height |
| 3. Compaq DiskPro/Plus 10MB | 4. ITT XTRA 10MB |
| 5. Option undefined | 6. NCR model PC4 10MB |
| 7. ERICSSON PC 10MB | 8. Option undefined |
| 9. Option undefined | 10. Option undefined |

99. *Custom Drive Definition option*

Enter drive selection for drive 0 =

Figure 2.2 PICK System Installation Menu

- In response to the prompt enter **1[RETURN]**
- The screen displays:

One moment, initializing Hard Disk drive 0

PICK begins to install itself by initializing its own disk partition. This process takes three to four minutes.

- When all of the information on the first disk has been read you will be prompted to:

load PICK PC system #2 then type 'C' to continue

Note: Enter 'C' only. It is not necessary to press [RETURN].

- Load each diskette when prompted and enter 'C' until all diskettes have been used.
- When the diskettes containing the data files are being read, the screen will display information messages such as:

```
SPOOLER STARTED
SYSTEM
BLOCK CONVERT
PROCLIB
DOC
etc.
```

These are some of the data files and messages that constitute a complete PICK system. These messages also indicate that installation is proceeding normally. Reading and copying of files takes three to four minutes.

- When the data from all of the diskettes have been read, you are asked to enter the current time and date.

Enter the time and date and press [RETURN] after each entry.

- Next, the PICK sign-on Messages are displayed along with the following:

```
Linking workspace for line 0.
The printer control block has been initialized.
verifying system modes, Please wait...
```

These messages indicate a normal system startup. During this time you will notice that the hard disk light flickers on and off while the PICK system boots. You are eventually placed at the logon prompt. When this occurs the PICK system has been successfully installed.

Upgrading An Existing Pick Installation

Upgrading an existing PICK system is accomplished by first making backup copies of your existing data files and then proceeding as if you were installing a new system. The purpose of making backup copies of the data files is so they can be restored to the system later. Assuming that you are currently running a version of PICK and you want to upgrade, the procedures to upgrade are:

- Execute ACCOUNT-SAVE's of all application accounts
- Delete the current PICK partition using FDISK from the SYSPROG account. At the Logon prompt, type SYSPROG [RETURN].
- After the PICK partition is deleted follow the directions for a new PICK installation.
- When the installation procedure is complete restore the previously backed up files using the ACCOUNT-RESTORE command. Do not re-boot the system and do a full file restore as this will overwrite the newly installed files.

Examples of how to use ACCOUNT-SAVE and ACCOUNT-RESTORE are provided in chapter 4.

On-Line Documentation Supplement

Improvements and enhancements are always being added to the PICK system. The printed documentation is printed weeks ahead of a new release of the operating system. It is because of this time lag that it is not possible to document all new operating system features. The purpose of the on-line documentation is to provide a means of documenting new features not explained in the PICK System Reference Manual. To use the on-line documentation log onto the SYSPROG account and type:

>ADDENDA[RETURN]

at the TCL prompt. The screen will resemble that depicted in figure 2.3.

```
Type 'ADDENDUM N' for detail on any addendum shown      current time/date
N....Brief description of addendum information.....
.
000 Philosophy of the documentation addendum utility.
001 New verb, POWER-OFF, included for orderly system shutdowns
002 Serial printer cabling information allows printer busy to control XT.
004 Information on how Pick serial ports are setup at boot time.
005 Okidata printer setup program included as a freeby.
006 ESC Code sequences given for those who do not want to use PRINT @ form.
007 WYSE-50 CRTs work with Pick PC XT software.
008 128K now legal with R83 - V1.3 software.
009 ABS frames which are available for user modes defined.
010 Trouble shooting tips for connecting the serial ports.
011 New SET-KBRD program allows line 0's keyboard to be redefined.
012 X-ON and X-OFF now supported on line 0.
.
.
.
.
.
```

Figure 2.3. PICK Addenda Screen

It is possible to review individual topics by typing:

>ADDENDA n[RETURN]

where n is the number of the addendum you wish to review. For example if you wish to review addendum 14 at TCL enter:

>ADDENDUM 14 [RETURN]

Note: To review any addenda prior to number ten you must enter leading zeros. For example to review addendum 1 enter:

>ADDENDUM 01[RETURN]

PC/XT Configuration and Special Terminology

The design of the PC/XT allows a maximum of four different operating systems to co-reside on the drive. Each operating system functions within the disk space boundaries specified by the FDISK command. This creates an interesting problem. How does the computer know which operating system to boot from? In addition to allocating disk partitions, the FDISK command allows you to specify which operating system will be “active” when the system is booted. Only one operating system may be active at any time.

The format of the PICK-supplied version of the FDISK is:

FDISK[RETURN]

```
IBM PERSONAL COMPUTER
FIXED DISK SETUP PROGRAM VERSION 1.00
(C)COPYRIGHT PICK SYSTEMS 1984
```

```
FDISK OPTIONS
```

```
CHOOSE ONE OF THE FOLLOWING:
```

1. CREATE PICK PARTITION
2. CHANGE ACTIVE PARTITION
3. DELETE PICK PARTITION
4. DISPLAY PARTITION DATA

```
ENTER CHOICE: [ ]
```

Figure 2.4. FDISK Screen for PICK

The PICK version of FDISK differs from the MS-DOS version in that the create partition option is non-functional. This is because the PICK partition is created automatically during the installation procedure. The delete partition option will not delete the PICK partition unless another operating system is first activated. This prevents accidental erasure of the PICK operating system files.

Line 0

Line 0 refers to the main terminal (usually an IBM monochrome or color graphics monitor) connected to port 0. The system is always booted or powered down from this terminal.

Terminals

Adding terminals to the system allows more than one person to use the computer at the same time. PICK can support many different types of terminals. A full listing of these terminals is given in figure 2.5.

A	Adds 580
B	Ampex 210
C	C. Itoh VT 52
D	Datamedia
I	IBM PC/XT (port 0, line 0)
L	Lear Siegler
M	Ampex 80
P	Pertec 701
R	Adds Regent
V	Adds Viewpoint
W	Wyse 50

Figure 2.5. PICK Supported Terminal Types

Other terminals may be used provided they can emulate any of those listed in figure 2.5. Line 0 plus 2 additional terminals are supported by the PC/XT version of PICK.

Make Physical Connection

To make the physical connection of the terminal to the computer, a serial connector is required. For the IBM PC/XT, the cable must have a female DB-25 connector on the computer end. This connector will be attached to the serial port on the back of the IBM PC/XT. The other end of the cable must have a connector that is compatible with the port on the terminal. Most terminals require male DB-25 connectors. Figure 2.6 diagrams the connection.

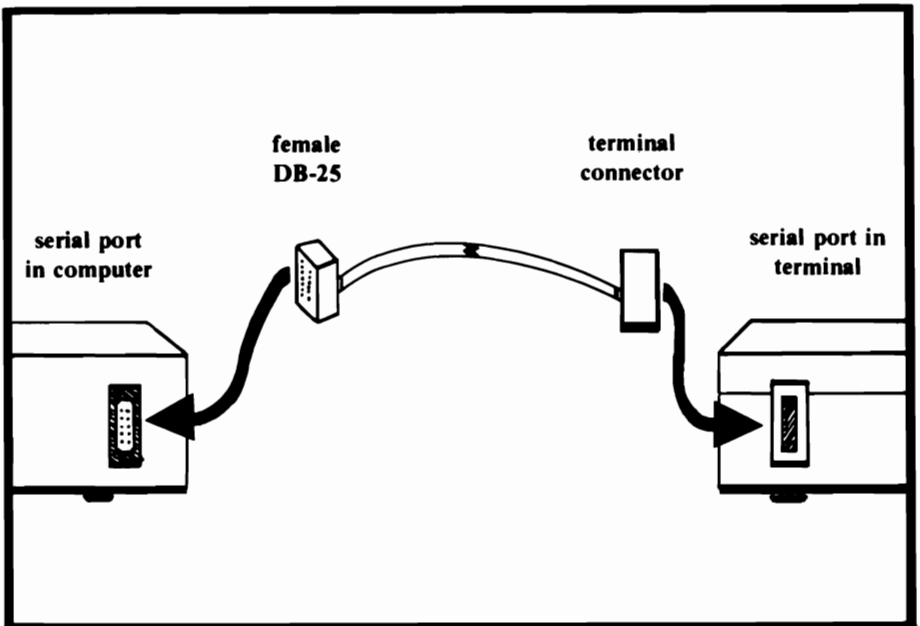


Figure 2.6. Diagram of a connection between an IBM PC/XT computer and a terminal.

Printers

A parallel printer is the default printer on a PICK system. Unless the system is configured otherwise, all hard copy output is directed to the parallel printer. A printer allows users to obtain hard copies of files and output. The PICK system provides a print spooler. This system places the desired print files in temporary disk files called queues

(pronounced “q’s”). Files are printed in a first in first out basis. The file is printed when the printer is not “busy.” After printing, each file is automatically erased. This routine allows all system users to share one printer.

A printer may be connected to one of the serial ports in place of a terminal. This would mean the loss of one user. We recommend using the default parallel printer.

Summary

In this chapter we discussed the hardware requirements for PICK and outlined the installation steps.

The topics of the FDISK verb, terminal connections and configurations were discussed.

3

PICK Terminology and Conventions

In this chapter we will discuss the knowledge required to begin using PICK. Topics of discussion include terminology, conventions, entering command lines, the PICK file system basics, and system startup and shutdown.

This chapter is intended to tutor the system manager on PICK startup and shutdown procedures. In the next chapter we will discuss the basics of using PICK. Some of the terminology from chapter two is repeated here for clarity. By the end of this chapter you should be able to start and stop a PICK system and be ready to begin using PICK.

As figure 3.1 illustrates there is a vocabulary associated with PICK which must be learned and understood before using the system. Study the words and their meaning before continuing with this chapter.

Account	Provides a means of security within the PICK system.
Attribute	Assume you have an index card with three fields of information written on it: name, address, and telephone number. The name, address, and telephone numbers are referred to as the attributes .
Command	An instruction given to PICK entered via the keyboard or program.
Command Line	A PICK command consisting of a verb and other parameters the verb may require to execute terminated by [RETURN].
Default Value	A value that PICK automatically assumes during execution of a command.
Delimiters	Special characters used to separate data while entering command lines. PICK recognizes three different kinds of delimiters: the backslash (\); parenthesis (); and quotation marks ("). The parentheses will be the delimiter of choice in this book.
Dictionary	A special type of PICK file which stores data.
Editor	Provides a means of entering data into a data base, writing PROCS, or entering source code for BASIC programs.
File	Analogous to a file folder. A file can contain the items (records) in a database.
Frame	Disk drive storage is divided into units called frames . PICK automatically allocates the file space it requires for files. Each frame is assigned a number called a frame-id or FID. The amount of data stored by a frame is machine dependent but is either 512 or 1024 characters per frame.

Figure 3.1. PICK Terms

Item	A file is composed of items. Each item in the file is composed of attributes. Assume a file to be a drawer containing ten index cards with names and addresses. Each index card is an item in the file. The names and addresses are the attributes .
Item-Id	The name of the sort key item in a data file.
Line	A communication link established with the computer via its serial ports, usually a terminal. The primary or booting terminal is designated line 0. The remaining serial ports are designated line 1 and line 2.
Master dictionary	A library of commands, programs, and data files accessible to individual user accounts. When a new user account is created a standard set of items is automatically copied into the master dictionary by PICK.
Modulo	This is a parameter specified when a new file is created. It is used to reserve a certain number of frames of disk space based on an estimate of the maximum number of characters to be contained in the file.
Monitor	The part of the operating system which handles "housekeeping" or maintenance functions such as the printing of files, storing of data on disks, display of data on a terminal screen, etc.
PROC	Pronounced "prock," this refers to stored procedures. It is similar to the job control language found on mainframe computers. It permits the automatic execution of complex command lines with a few keystrokes. Stored procedures may also be used to isolate novice users from the operating system via the use of menus.
Sysprog	The master controlling account. All system maintenance functions and orderly system shutdowns are performed from this account.

Figure 3.1. (Cont.)

TCL	This is the Terminal Control Language processor. It is the primary interface between the user and the PICK system. When the PICK prompt is displayed (>) the user is said to be at TCL. At this time the system is waiting for a command.
Value	The contents of an attribute. In the card file described earlier, if the value "Walter" was assigned to the attribute Name: "Walter" would be the value of that attribute.
Verb	An action word that specifies a PICK command. The PICK system has many verbs which describe an action for the operating system. Verbs are entered at TCL and only one verb can be entered and processed at a time.

Figure 3.1. (Cont.)

PICK CONVENTIONS

In order to enter PICK commands certain rules must be followed. These rules are referred to as **conventions**. We will now illustrate the syntax for entering PICK command lines.

Entering Command Lines

The proper syntax for entering command lines is shown below,

```
verb {...parameters...} { (options) } [RETURN]
```

where verb is the PICK command or PROC to be executed, parameters are file names, dictionary names, etc., and options are special parameters affecting the operation of verbs.

For example the following command line:

BLOCK PRINT WSI (P)[RETURN]

would cause the following output:

```

WW      WW      SSSSS  IIII
WW      WW      SS      II
WW      W      WW      SS      II
WW      WWW      WW      SSSSS  II
WW      WWW      WW      SS      II
WW      WWW      WW      SS      II
WW      WW      SSSSS  IIII

```

More examples of entering command lines will be provided in chapter four.

Special Control Sequences

When entering commands at TCL you may find it useful to use control sequences to assist you in data entry. A control sequence is entered by pressing the key labeled CTRL and another key simultaneously. These sequences are designed to aid you in editing command lines. Figure 3.2 depicts the most frequently used PICK control sequences.

Control Sequence	Function
CTRL-X	Cancels a command or operation. If TCL commands are entered the characters typed are ignored and the user is placed at the TCL prompt. It terminates a screen display and returns the user to TCL.
CTRL-H	Backspaces over the last character entered. The last character entered is deleted.
CTRL-W	Backspaces one word.
CTRL-R	Retypes the previously entered line. This sequence is especially useful after a command line has been edited. It saves the work of reentering the line.

Figure 3.2. PICK Control Sequences

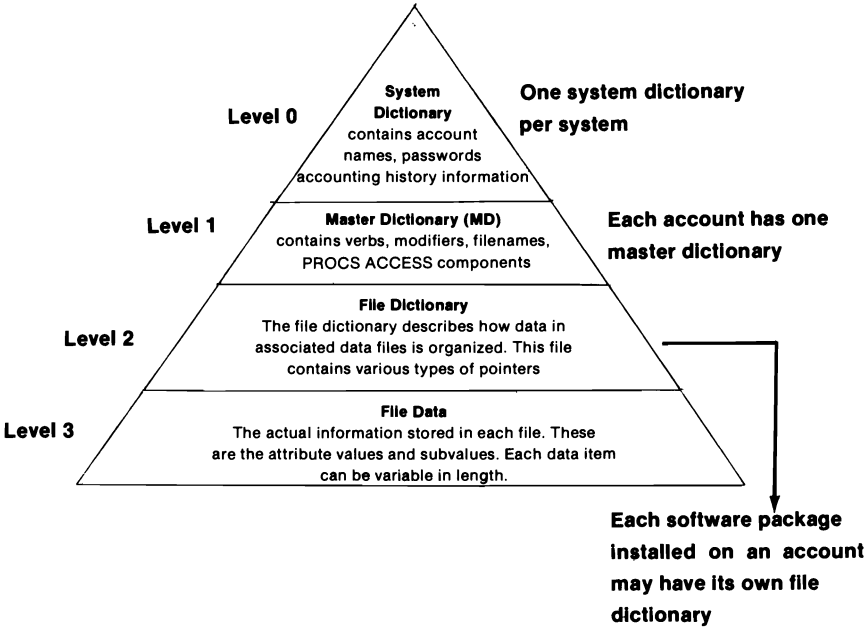


Figure 3.3. Hierarchical File Organization of PICK

Try entering some command lines or a few words to see how these editing sequences work.

Introduction to PICK File Structure

In order to use PICK it is necessary to understand how PICK files are organized. PICK files are organized in a *hierarchical* arrangement. This means that files at the highest level relate to files at the next lowest level. In this section the terms item, attribute, and item-id will be used.

Remember that item refers to individual records within a file, attribute is a data field within a file, and item-id is the name of the sort key in the file.

In a PICK system there are four levels of file organization. Each is assigned a level starting with zero. Starting with the highest level of organization they are:

- system dictionary (level 0)
- user master dictionaries (level 1)
- file level dictionaries (level 2)
- data files (level 3)

As figure 3.3 illustrates the system dictionary contains all of the security data necessary to determine access to other files within the system. The master dictionary contains the library of verbs (commands), parameters and options, and file names available for use by an individual account. The file dictionary contains the criteria that define the structure of data files. The data files contain the actual data.

The hierarchy means that if a user does not log onto the proper account, the user will not be able to access the files contained in that specific dictionary. Unless special circumstances have been provided for (via file/dictionary sharing) access is not permitted.

The PICK file structure is examined in detail in chapter five.

System Startup and Shutdown

Although the PICK system is much more complex than other popular personal computer operating systems such as MS-DOS, PICK provides an easy means for system startup and shutdown. This section details how to accomplish PICK system startup and shutdown.

The PICK system should never be halted by simply powering off the computer. Such an action may cause a system crash and damage data and operating system files.

Starting

We assume the PICK system has been properly installed on the hard disk according to the steps outlined in chapter two. The system can be started with the following steps:

1. Power on the computer. The machine will run through its hardware checks and display its check of RAM.

2. Be certain there isn't a floppy disk in the drive. The PICK system will boot itself from the hard disk and prompt you to enter the current time and date. The PICK system uses a military format of 24 hour time. Enter the time in the format HH:MM:SS. If the current time is 8:30 pm, enter 18:30 and press [RETURN]. You must use the colon as the delimiter between the hours, minutes, and seconds; otherwise the time will not be entered.

Enter the date in the format MM-DD-YY. For example if the current date is August 4th, 1986, enter 08-04-1986 and press [RETURN]. You must use the hyphen as the delimiter between the month, day, and the year; otherwise the correct month will not be entered.

PICK will automatically activate the system for two additional users, the print spooler and the parallel printer port. To activate a serial printer port see chapter 6.

Note: if the correct time and date are not entered during startup, you can always use the SET-TIME and SET-DATE verbs to set these parameters later.

3. After the date is entered, the system will display the message:

"The printer control block has been initialized."

Next, the message,

“Verifying system modes please wait...”

is displayed. At this time PICK is checking to see if all is well. The red light on the hard disk drive flashes on and off while it reads and verifies the system. Software verification takes three to five minutes.

4. When the system has been verified, it will display the following message,

RXX.X SYSTEM VERIFIED!

and the Logon prompt will be displayed.

Logon is the means of entering a user account. Before any data processing can be accomplished you must be logged on to an account. The display of the Logon prompt indicates that system startup was successful and is now complete.

Note: If the system does not verify, the message:

***** RXX.X SYSTEM DOES NOT VERIFY! *****
There are nn frames with mismatches.

is displayed.

RXX.X is the release number of the operating system. nn is the number of frames with mismatches. Do not attempt to use the system until the software has verified. To correct this problem see chapter five.

Stopping

To accomplish an immediate system shutdown, use the following steps:

1. Be certain users on all lines are logged off.

2. Log on as SYSPROG on line 0.
3. Type POWER-OFF[RETURN] at the TCL prompt. Execution of this command automatically closes all files and repositions the drive head in a safe area of the hard disk drive. The screen will display:

Flush done, power down OK

4. Power off the computer.

Summary

In this chapter we have provided you with the necessary concepts to continue your study of PICK. The best way to learn about PICK is to use it. In the next chapter we will show you how to get started.

4

Getting Started

In this chapter we discuss the basic procedures used to access the PICK system. We will include in our discussion an explanation of the terminology that is used within the PICK environment. We intend this chapter to be used by individuals who are unfamiliar with either the PICK system or computers in general.

Obtaining an ID

The method of obtaining a computer ID will vary from company to company. However, the final result will always be similar. For details concerning how to requisition an ID, consult with another user of the computer system. The ID will consist of two parts. The first part will be the user name that has been assigned to you. In the PICK system the user name is quite often related to the user's actual name

or initials. The second part of the ID will consist of the password assigned to your account. The account password must be given before anyone can gain access to the account. The password is a security measure. It helps keep your data safe from change by unauthorized users. The security of the files in your account depends on the security of your password. Do not tell other users your password.

Types of ID's

The PICK system has three types of ID's. These are detailed below:

- **SYS0** A SYS0 ID has the lowest security clearance. A SYS0 ID is often given to novice users. The commands that the user is allowed to execute with a SYS0 ID is limited. A SYS0 ID is adequate for individuals who will only be using existing data files to accomplish tasks such as creating reports based on the data in the data files.
- **SYS1** A SYS1 ID is reserved for more knowledgeable users. A SYS1 ID allows the individual to create and dispose of files as well as several other privileges not granted a SYS0 ID.
- **SYS2** A SYS2 ID has the highest security clearance given to a user. The SYS2 ID is given to veteran users. A SYS2 ID allows the user additional privileges such as access to the floppy drive.

Which security level is assigned to the account for an individual will be determined by the work that the individual is expected to accomplish. If a command referenced in this book will not execute and an error is displayed pertaining to the verb, then the command is probably not included in the individual's account. The LIST-VERBS command, which we discuss later, can be used to check for the occurrence of the verb in question in the Master Dictionary.

Logging On

The procedure used to access the computer is called logging on. In order to execute a logon, the terminal's screen must first display the following message on the screen:

```
Logon
```

Such a message is called a logon prompt.

To execute a logon, the user must type the user name that was assigned to him on his computer followed by pressing the return key. For example, an individual with the user name MEG would type the following:

```
MEG [RETURN]
```

If a correct user name was specified, the system will respond by displaying the following:

```
PASSWORD:
```

The user must now type in the password associated with the account, followed by pressing the return key. For example, if the password for the account named MEG was EZ15, the individual in the above example would type the following:

```
EZ15 [RETURN]
```

Note that the screen will not display the characters that compose the password as they are typed. This feature is an added security measure. If the password was input correctly, the system will display a message similar to the following on the screen:

```
<<< ----- THE PICK SYSTEM ----- >>>  
<<< --00:18:18 -- PC-XT Version 1.3 -- 14 MAR 1985 --- >>>
```

If an error is made in typing the user name after the logon prompt, the system will display the following message:

```
USER-ID?
```

The logon prompt will then be displayed once again. If a mistake is made when typing the password, the system will display the following message:

PASSWORD?

The logon prompt will then be displayed once again.

Case Usage

The PICK system is always in the upper case mode when it is first activated. All letters typed at the keyboard are interpreted as upper case. The shift key is used to input lower case letters. This arrangement can be reversed by pressing the CAPS LOCK key. The user will usually want to leave the keyboard in the upper case mode.

The case of a letter is significant to the PICK system. The system uses the upper case exclusively for all the system defined files, verbs, and PROCs. The individual can use either upper or lower case. However, the names INVENTORY, Inventory, and inveNtory will represent three discrete entities to the system. Using all upper case letters simplifies using the system. The keyboard can be left in the upper case mode and the user need not worry about the case of letters in a name because the lower case letters are never used. It is important to note that the system will not recognize a verb or other system defined parameter unless it is input in the upper case.

The TCL Interface

TCL stands for Terminal Control Language. The TCL interface is used to gain access to all of the many facets of the PICK operating system. The system documentation makes a distinction between TCL verbs and ACCESS verbs; however, to the user that distinction is blurred. Both TCL commands and ACCESS queries are input at the TCL level. User defined PROCs are also accessed via the TCL interface.

The TCL Prompt

Immediately after displaying the message signaling that the logon has been accepted, the system will display the Terminal Control Language prompt. The prompt appears as a greater than symbol (>) on the screen. This prompt is often called the TCL prompt. This prompt signifies that the TCL interface of the PICK system is ready to process commands for the user.

TCL Command Format

A command to the TCL interface must start with a verb. The verb may be a TCL verb, an ACCESS verb or the name of a user defined PROC. The end of the command is signaled by pressing the return key. In the text, pressing the return key will always be indicated by a [RETURN]. The TCL interface recognizes many different verbs. A verb invokes one of the PICK systems processors to do a specified task. We discuss some of the more useful verbs in the following chapters. If the TCL interface cannot match the first word in a command to one of the verbs in its library, the system will display the following error message:

```
[3] VERB?
```

This error message can also be caused by a typing error. Check the original input for such a mistake.

Many TCL commands access files stored within the system. We discuss the various parts that compose a file in the next chapter. For now we will just explain the different formats that can be used to specify a file. The most general form of a file specification within a command is shown below:

```
{ DICT or DATA } dictname { , dataname }
```

DICT	The DICT parameter is used to specify the dictionary portion of the file to be accessed.
DATA	The DATA parameter is used to specify the data portion of the file to be accessed.

Note that we will specify which of these parameters are appropriate to use with a given command when we specify that command. The only parameter that always must be specified is the *dictname* parameter. Some commands access more than one file. In such a case, more than one file must be specified.

Options can be specified for many of the commands. We will detail the options available for a given command when we discuss that command in the text. Options must always be enclosed in parentheses. One option that is common to many commands is P. The P option is used to send the result of the command to the line printer instead of the default destination. The default destination will be used if another destination is not specified. Generally, the default destination is the terminal's screen.

Delimiters

If an item name or other parameter to a command has an internal blank, the system can become confused as to the intended meaning of the command. For example, the following command:

```
ED INVENTORY IBM PC XT [RETURN]
```

would be interpreted by the system to be a command to edit the three items IBM, PC, and XT in succession. The intention of the command was to edit the single item IBM PC XT. In order to avoid such confusion, delimiters can be used to set the item containing internal blanks off. Generally one of three standard delimiters is used. The three standard delimiters are parentheses (), the double quote (") and the backslash (\). The delimiter that is used must not appear in the name that is being delimited. Throughout this book we use parentheses as the delimiters. The delimiter used by the individual is left to personal discretion.

Output from the System

Generally, system output is routed to the screen of the terminal from which the request originated. Alternatively, an option to many commands allows the output to be routed to the line printer. The system formats the output so that it will fit the format of the specified device. This formatting includes matching the output line and page length to be compatible with the receiving device.

Another task that the system automatically performs is halting after each screenful of information. This feature allows the user to assimilate the provided information. The next screenful of information is accessed by pressing the return key. The TCL prompt will appear after the last screenful of information has been displayed. All of the output need not be displayed in this manner. The user can escape from viewing the rest of the output by typing control-x.

Selected TCL Commands

In this section we will present tutorials on the usage of selected TCL commands. Most of the commands that are not included in this discussion are covered in other chapters. File handling verbs are covered in the chapter on the file system. User defined PROCs and ACCESS verbs are discussed in chapters devoted to those topics. The EDIT and RUNOFF verbs are detailed in the chapter on the Editor. The PICK User Reference Manual contains a complete list of the commands that can be processed by the TCL interface. The commands that we discuss are listed in alphabetical order.

Large Printing

A character string can be printed in large block letters using the BLOCK-PRINT command. The resulting display can be routed to either the terminal's screen or the printer. The BLOCK-PRINT command has the format shown below:

```
BLOCK-PRINT string (P) [RETURN]
```

where *string* represents the character string to be printed using the large characters. The P option is used to route the output to the printer. If the P option is not used, the output from the command will appear on the screen.

For example, the following command:

```
BLOCK-PRINT WSI (P) [RETURN]
```

would cause the following printer output:

```
WW      WW      SSSSS  I I I I
WW      WW      SS      SS  II
WW  W  WW      SS      II
WW  WWW  WW      SSSSS  II
WW  WWW  WW      SS      II
WWW      WWW      SS      SS  II
WW      WW      SSSSS  I I I I
```

Also, depending on how the line length of the printer is defined, the output resulting from a BLOCK-DISPLAY command is not always done correctly. To set the line length correctly, use the TERM command. We discuss how to use this command later in this chapter.

Displaying File Contents on the Terminal Screen

The data stored within a file can be displayed at the terminal using the CT command. The format of the CT command is shown below:

```
CT {DICT} name items {(options)} [RETURN]
```

where DICT and *name* are used to specify the file name as we discussed earlier. The *items* parameter is used to specify which record in the file should be displayed. We discuss file items in detail in the next chapter. The *options* available to the CT command are the same as those available to the COPY command. We detail these options in the next chapter. For most file display applications, the options are not needed.

For example, the following command:

```
CT SYSTEM logon name [RETURN]
```

where *logon name* represents the user's own logon name, would cause the system to display the definition of the user's Master Dictionary in the System Dictionary on the terminal's screen. Such a display would be similar to the following:

```
          SULLY1
001  D
002  2858
003  29
004  1
005
006
007
008  SYS1
009  L
010  10
```

Listing Attribute Definitions Contained in a Dictionary

A complete sorted display of all attribute definitions contained in a dictionary file can be obtained using the LISTDICT¹ command. We discuss attribute definitions in the next chapter and in the section about the ACCESS processor. This command is useful when checking attribute definitions in files used for data base applications. The LISTDICT command has the following format:

```
LISTDICT dictname {LPTR} [RETURN]
```

where *dictname* represents the dictionary from which the attribute definitions should be taken. The optional LPTR parameter allows the resulting list to be sent to the line printer instead of the terminal screen.

For example, the following command:

```
LISTDICT MD [RETURN]
```


would cause all of the attribute definitions in the user's Main Dictionary to be displayed on the screen. In a newly created account, there are approximately fifty attribute definitions. The listing of these attributes typically takes 16 PICK terminal display pages.

Listing Files Contained in a Dictionary

Files and file synonyms contained in a dictionary can be listed using the LISTFILES command. The form of the LISTFILES command is shown below:

```
LISTFILES dictname {LPTR} [RETURN]
```

where *dictname* is used to specify the dictionary from which the files should be listed. The optional LPTR parameter is used to specify that the listing should be made using the line printer. If LPTR is not included, the listing will appear on the terminal screen.

For example, the following command:

```
LISTFILES MD [RETURN]
```

would cause the system to display the following information:

```
PAGE    1    *** MD *** FILE DEFINING ITEMS      31 OCT 1984
MD..... CODE F/BASE..... F/MOD..... F.SEP...

ACC      Q                      ACC                      ACC
BLOCK-CONVQRT      BLOCK-CONVERT
ERRMSG    Q                      SYSPROG                ERRMSG
M/DICT    Q
MD        Q
POINTER-FIQE      SYSPROG      POINTER-FILE
PROCLIB  Q          PROCLIB
SULLYO    Q
SYSPROG-BPQ      SYSPROG                BP
SYSTEM    Q          SYSTEM

10 ITEMS LISTED.
```

¹ The LISTDICT command is improperly referenced as LISTDICTS in the system documentation.

Listing Procedures Contained in a Dictionary

The procedures contained in a dictionary can be listed using the LISTPROCS command. The listing will contain both user defined PROCs and system provided PROCs. The listing will contain two columns. The first column will contain the names of all the PROCs in the specified dictionary. The second column will contain a short description of each of the PROCs. The LISTPROCS command has the following format:

```
LISTPROCS dictname {LPTR} [RETURN]
```

where *dictname* is used to specify from which dictionary file the PROC listing should be taken. If the optional LPTR parameter is included, the listing will be output on the line printer instead of on the terminal screen.

For example, the following command:

```
LISTPROCS MD [RETURN]
```

would cause all of the procs in the user's Master Dictionary to be listed. Such a listing would include information similar to the following:

```
PAGE      1
11:57:26  01 NOV 1984

MD.....  Description.....
CAT        (PROCLIB  CAL)
CHOO-CHOO  (PROCLIB  CHOO-CHOO)
CT         (PROCLIB  CT)
DELETE     (PROCLIB  DELETE)
EXCHANGE   (PROCLIB  EXCHANGE)
LISTACC    (PROCLIB  LISTACC)
LISTCONN   (PROCLIB  LISTCONN)
LISTDICT   (PROCLIB  LISTDICT)
LISTFILES  (PROCLIB  LISTFILES)
LISTPROCS  (PROCLIB  LISTPROCS)
LISTU      (PROCLIB  LISTU)
LISTUSERS  (PROCLIB  LISTUSERS)
LISTVERBS  (PROCLIB  LISTVERBS)
LOOP-ON    (PROCLIB  LOOP-ON)
SET-FILE   (PROCLIB  SET-FILE)
SET-LPTR   (PROCLIB  SET-LPTR)
T-SPACE    (PROCLIB  T-SPACE)

17 ITEMS LISTED.
```

Listing Active Users

The LISTU command is used to obtain information about who is currently using the system. The LISTU command has the following form:

```
LISTU [RETURN]
```

The output from the LISTU command cannot be sent to any device other than the terminal screen.

For example, the following command:

```
LISTU [RETURN]
```

would cause information about each user who is logged onto the system to be displayed. The provided information includes the name of the user, the location at which that user is logged, and the time at which the logon was accepted by the system. The user who requested the list will be marked with an asterisk.

Listing Verbs Contained in a Dictionary

The LISTVERBS command is used to obtain a list of all verbs that are contained in a specified dictionary. The form of the LISTVERBS command is shown below:

```
LISTVERBS dictname {LPTR} [RETURN]
```

where *dictname* specifies the dictionary from which the list of verbs is to be taken. The optional LPTR parameter allows the output from the command to be sent to the line printer instead of being displayed on the terminal screen.

For example, the following command:

```
LISTVERBS MD [RETURN]
```

would cause all of the verbs in the user's Master Dictionary to be displayed. A newly created account contains over 100 verbs in its Master Dictionary.

Logging Off

The OFF command is used to end a session with the PICK operating system. The system can no longer be accessed by a user once he has logged off. The OFF command has the following format:

```
OFF [RETURN]
```

When an OFF command is executed, the system will respond by displaying information similar to the following:

```
[ CONNECT TIME= 0 MINS. ; CPU = 7 UNITS ; LPTR PAGES=0 ]  
[ LOGGED OFF AT 11:42:47                ON 31 OCT 1984    ]
```

The system will then display the logon prompt. When the logon prompt is displayed, another user can access the system by logging on.

Setting Tab Stops

Tab stops may be set for both input and output. Output tabs are used with printing terminals. Most microcomputer systems do not include this type of terminal. Tabs for output should not be set for use with a CRT terminal. Setting tab stops for input allows the user to move the cursor to the next tab stop by using either the control-I sequence or the tab key. On the IBM PC XT the tab stop key is labelled $\Rightarrow$. Up to 15 tab stops may be set using the TABS command. The form of the command to define input tab stops follows:

```
TABS I number { , number , , } [RETURN]
```

where each occurrence of *number* is used to define a tab stop. The tab stops must be listed in ascending order, that is from lowest to highest.

For example, the following command:

```
TABS I 5,20,40,60 [RETURN]
```

would cause tab stops to be set at columns 5, 20, 40, and 60. Note that the positions of the current tab stop settings can be displayed using the following command:

TABS [RETURN]

the display from such a command would appear as follows:

```

      1      2      4      6      7
12345678901234567890...0...0...1234567890123456789
      I      I      I      I

```

Each I represents an input tab stop. To clear the existing input tabs from the system, the following command is used:

TABS I [RETURN]

Setting Terminal Characteristics

Certain characteristics of how the terminal functions can be set using the TERM command. Alternatively the TERM command can be used to check the current value of the terminal settings. The TERM command is also used to set characteristics of the line printer. The form of the command to check the settings of terminal characteristics is shown below:

TERM [RETURN]

A typical output generated by such a term command is shown below:

```

                TERMINAL PRINTER
PAGE WIDTH:      79          80
PAGE DEPTH:      23          60
LINE SKIP  :      0
LF DELAY   :      0
FF DELAY   :      2
BACKSPACE  :      8
TERM TYPE   :      T

```

>

The parameters listed in the display include all of the terminal characteristics that can be modified using the TERM command. The characteristics shown above are the default settings given by the system when the user logs on. We discuss how to have the system automatically set the terminal characteristics to a different set of values in the section on logon PROCs which we present later in this chapter.

The form of the term command to alter a terminal settings follows:

```
TERM { tll }, { tpl }, { tbi }, { lfd }, { ffd },
{ bks }, { pll }, { ppl }, { ttc } [RETURN]
```

All of the parameters are optional. However, if a parameter is to be skipped, the comma that would normally separate the parameters must still be typed. This action is necessary because the system determines which parameters are being defined by their position among the commas. Therefore, it is important to place the value of the parameter being defined in the correct position. Parameters that take their meaning from the position they inhabit are called positional parameters. The following list details the meaning of each of the parameters to the TERM command:

- *tll* The *tll* parameter specifies the terminal line length. This line length must be between 16 and 140 characters. For most terminals, the *tll* parameter should be 79. The *tll* parameter sets the PAGE WIDTH characteristic for the terminal.
- *tpl* The *tpl* parameter specifies the number of terminal print lines on each page of output from the system to the terminal. This parameter specifies how many lines will actually appear on the screen. The maximum number of lines that a terminal screen can display is typically in the range of 20-25.

- *tbl* The *tbl* parameter specifies the number of terminal blank lines on each page of output from the system to the terminal. This parameter along with the *tp/* parameter specifies the PAGE DEPTH characteristic for the terminal. Generally, the *tbl* parameter is set to 0.
- *lfd* The *lfd* parameter specifies the line feed delay. The line feed delay is defined as a number of characters. The line feed delay is used with terminals that cannot receive data as rapidly as the system generates them. The system will wait for the amount of time that it would take to send the number of characters specified by the *lfd* parameter after each line feed. For most newer equipment, the line feed delay can be set to zero. The *lfd* parameter sets the LF DELAY characteristic.
- *ffd* The *ffd* parameter specifies the form feed delay. The form feed delay performs a function similar to the line feed delay. However, the form feed delay is used only after a top-of-form is sent. The value of the *ffd* parameter determines how it is to be used. A value of 0 will cause the system to generate no form feed after a page of output. A value of 1 will cause the system to generate a form feed after each page of output to a printer. A positive value greater than 1 will cause the system to generate a form feed for the terminal as well as the printer. Also, the system will then wait the specified amount of time before sending more data to the terminal. The *ffd* parameter sets the FF DELAY characteristic.

- *bks* The *bks* parameter specifies the backspace character. The system always outputs a control-H to represent a backspace. The *bks* parameter allows the control-H to be changed to a character suitable to the terminal. Generally, an 8 is used. An 8 is the ASCII code for a backspace. If the terminal in use does not backspace properly using a backspace character of 8, consult the specifications of the terminal to find which code it does use to backspace. If the terminal cannot backspace, a character string such as [bksp] can be used to visually represent the backspace. The display on the screen will be messier using this technique. However, the data maintained within the system will be correct. The *bks* parameter sets the BACKSPACE characteristic.
- *pl* The *pl* parameter specifies the printer line length. Generally, printers used with microcomputers have a line length of 80 characters. The *pl* parameter sets the PAGE WIDTH characteristic for the printer.
- *ppl* The *ppl* parameter specifies the printer page length. Generally, paper used in printers have 66 lines on each page. However, the page length is typically specified to be a few lines less than 66 to allow for a top and bottom margin. The *ppl* parameter sets the PAGE DEPTH for the printer.
- *ttc* The *ttc* parameter specifies terminal type code. The system uses the terminal type code to set the form feed character to its correct value and to define the BASIC cursor addressing correctly. Chapter 2 contains a list of the terminals supported by the PICK system along with their related type codes. The *ttc* parameter sets the TERM TYPE characteristic.

The TERM command is most effectively used in a logon PROC. We discuss logon PROCs later in this chapter. We now present some examples using the TERM command. The following command:

```
TERM , , , , , , 60 [RETURN]
```

would cause the terminal characteristics to be changed from their default setting to the following:

TERMINAL PRINTER		
PAGE WIDTH:	79	80
PAGE DEPTH:	23	60
LINE SKIP :	0	
LF DELAY :	0	
FF DELAY :	2	
BACKSPACE :	8	
TERM TYPE :	I	

Note that the PAGE LENGTH characteristic was changed from the default setting of 64 to the 60 specified by the ppl parameter in the TERM command. Also, the seven commas that appear before the page length were required by the system in order to correctly identify the parameter.

As a second example, consider the following command:

```
TERM ,15,9 [RETURN]
```

This command would not change any of the characteristics displayed by the system with the TERM command. However, the number of lines actually printed on each full screen would be reduced to 15. There would be 9 blank lines on the screen.

Displaying the Time and Date

The time and date kept by the system can be displayed using the TIME command. The format of the TIME command follows:

```
TIME [RETURN]
```

The output from the TIME command can only be displayed on the screen. The time of day will be shown first. The time is displayed as military time. Military time operates on a 24 hour cycle so that 1 pm is shown as 13.

For example, the following command:

```
TIME [RETURN]
```

would cause the time and the date to be displayed as follows:

```
13:03:30    02 NOV 1984
```

Displaying Who Is Logged On

A list of the users logged onto the system with no other information can be displayed using the WHO command. The output from this command will list each user who is logged on, one to a line. The form of the WHO command is shown below:

```
WHO [RETURN]
```

Logon PROC'S

A logon PROC is a procedure that is executed each time the system accepts a logon from a user. Each user can define his own logon PROC. The most important task an individual's logon PROC accomplishes will generally be the task of setting terminal characteristics to coincide to the terminal from which the user most often accesses the system.

We will present a procedure to create a logon PROC that will accomplish the setting of terminal characteristics. We will not explain the details of how and why we are issuing particular commands. We explain more on how PROCs work in the chapter detailing PROC usage. If the individual uses the main terminal to access the system, the logon PROC is not needed because the default terminal settings are for that machine. The main terminal is the terminal from which the system is booted.

The logon PROC must have the user's logon name as its name. The system will only automatically execute the PROC whose name matches the user name given at the logon prompt. The sequence of commands listed below will accomplish the task of creating a logon PROC. The appearance of *user name* should be replaced with the individual's own user name and TERM ---- should be replaced with the TERM command to correctly define the terminal characteristics. We discussed the TERM command earlier in this chapter. The command sequence follows:

```
ED MD user name [RETURN]
I [RETURN]
PQ [RETURN]
HTERM --- [RETURN]
P [RETURN]
[RETURN]
FI [RETURN]
```

The system will then display:

```
" "USER NAME" "
FILED.

>
```

5

The File System

In this chapter we discuss the PICK file system. It is important to understand the structure and usage of both the entire system and the individual files. The first section in this chapter offers a discussion of both of these topics.

The section on file definitions is somewhat technical and is provided for reference. In the majority of business installations, even the system manager need not be concerned about the file definitions. When data files for application software are created, the file definition parameters are automatically determined during software installation.

The following topics are also discussed in this chapter:

- File handling, including file creation and deletion
- File ownership
- File security

Structure

The structure of the PICK file system is dependent on the structure of the individual files within the system. In this section we discuss the structure of individual files. We will expand that explanation to describe the overall structure of the file system.

Individual Files

Individual files within the PICK operating system include two parts, the dictionary portion and the data portion. The data portion, as its name implies, is used to store the actual data. The dictionary portion of the file is used to define how the data is to be used.

Both the data and dictionary portions of a file are composed of items. However, the function of items in the dictionary portion is quite different from that of items in the data portion.

Items in the data portion of a file are analogous to records in the more conventional vocabulary. The only function of data items is to contain data.

Items in the dictionary portion of a file are used to accomplish several system-oriented tasks. These are listed below:

- File definitions
- Synonym definitions
- Definitions detailing the layout of the data within data items

We discuss the details of each of these types of dictionary items later. For now, note that dictionary items are used to define system parameters. Dictionary items never contain data.

File System

The overall structure of the PICK file system is hierarchical in nature. A hierarchical file system is one in which the data is stored in levels. For example, the MS-DOS file structure is hierarchical in that any directory can contain subdirectories. The subdirectories are located one level below the directory that contains them. Figure 5.1 illustrates this concept.

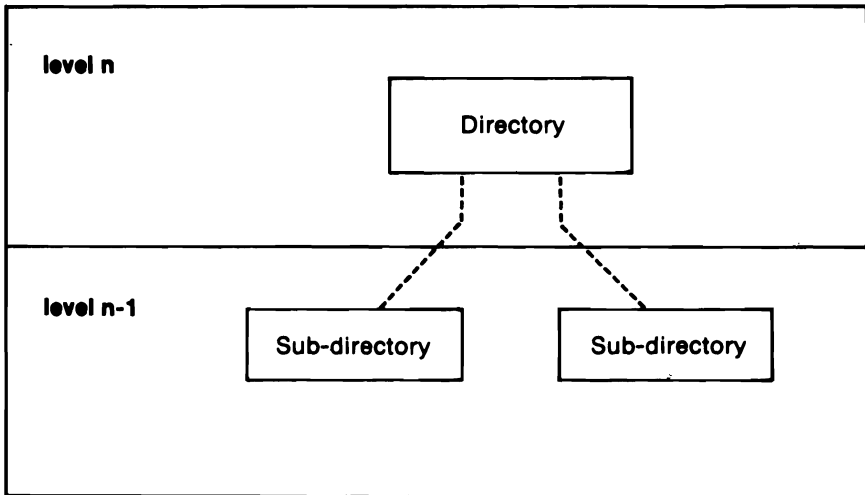


Figure 5.1. Example of a hierarchical file system.

The PICK file system has only four levels in its structure. These are numbered levels zero through three. Level zero is called the System Dictionary. Level one is named the Master Dictionary. Level two is called the File Dictionary, and level three is named File Data. Each of these levels will be discussed in detail in the following sections.

System Dictionary

The System Dictionary is the highest level in the PICK file system. There is only one System Dictionary in a PICK system.

The System Dictionary is used to store user account logon information, system security codes, and special features reserved for use by the system programmer account. The system programmer account is used by the system manager for special tasks such as adding new users to the system. Generally, most users do not have the security clearance to alter the System Dictionary. However, the average user can examine The System Dictionary to view its contents. Type the following command from TCL:

```
LISTFILES SYSTEM [RETURN]
```

The system will respond with a display similar to the following:

PAGE 1

12:28:37 22 OCT 1984

SYSTEM..... 1..... AMC

BLOCK-CONVERT	D	2175
SYSTEM-ERRORS	D	2197
LOGON	L	H logon +
COLDSTART	Q	SYSPROG
PROCLIB	D	2230
SYSTEM	D	2164
SYSPROG	D	2261
SULLYO	D	2829
SULLY1	D	2858
SULLY2	D	2887
ACC	D	2812

11 ITEMS LISTED.

The name SYSTEM represents the system dictionary. Note that each entry in the system dictionary is an item.

Master Dictionaries

The Master Dictionaries compose the second highest level in the PICK system. A PICK installation can include a number of Master Dictionaries. In fact, each user will usually have his own dictionary.

The Master Dictionary is used to define the commands which the user can execute and the files which the user can access. The pointers to any files the user creates will reside in his Master Dictionary. The individual can customize his PICK environment by defining synonyms for command names and files.

The contents of the Master Dictionary can be listed using the following command:

LISTFILES MD [RETURN]

where MD represents the Master Dictionary. The output from this command is too voluminous to insert into this text. Many screenfuls of information will be displayed as a result of this command. Note that one screenful is called a page in this situation. The system will stop after each page has been displayed. To continue to the next page, press the return key. Note that any new files or synonyms that the user defines will also appear somewhere in the Master Dictionary.

File Dictionaries

The third level in the PICK operating system is made up of file Dictionaries. Each file that a user creates is composed of a dictionary portion and a data portion. The File Dictionary is used to store the dictionary portion of a file. Each Master Dictionary in The System Dictionary can contain a number of File Dictionaries.

File Data

The File Data is the lowest level of the PICK system. The File Data level is composed of the data portion of the files stored in the Master Dictionary. The data portion of each file is composed of items. Several items compose a File Data set. There can be several File Data Sets for each File Dictionary in the Master Dictionary. Each data file that is associated with a File Dictionary will be defined by an item in that File Dictionary.

Overall View of the PICK File System

From the preceding discussion, we can draw a picture representing the PICK file system shown in figure 5.2. Note that only part of the total system is represented in figure 5.2.

In figure 5.2 the system Dictionary is depicted on level 0. Level 0 is the highest level in the PICK file structure. Note that there is only one System Dictionary for the entire system. The bulk of the items in the System Dictionary define user accounts. Each user account points to a Master Dictionary on level 1.

There are two Master Dictionaries depicted in figure 5.2. They represent the accounts, SULLY and MEG. A working PICK system

will have many Master Dictionaries on Level 1. The Master Dictionary contains items that define which commands are available to the user logged on to the account. Also, the Master Dictionary contains items that define file and command synonyms. Finally, the Master Dictionary contains items that point to file Dictionaries on level 2.

There are two File Dictionaries depicted in figure 5.2. These File Dictionaries are defined in the Master Dictionary SULLY. They represent the dictionary portion of the files INVENTORY and SALES. Each Master Dictionary can define many File Dictionaries. The depicted File Dictionaries contain items that point to File Datas on Level 3. Other items in the File Dictionary contain attribute definitions for the File Datas. (We discuss attribute definitions later in this chapter.)

There is only one File Data depicted in figure 5.2. There can be several File Datas defined for each File Dictionary. The depicted File Data represents the data file SALES. Note that the first attribute is defined as NAME while the second attribute is defined as TERRITORY. The items in the data file SALES contain the actual values that are stored in the file SALES.

Items

As we discussed earlier, items may appear in both the dictionary and data portions of a file. Items in the data portion of a file are equivalent to records in the standard data processing terminology. The sole function of data items is to contain data. Items in the dictionary portion of the file are used to store system parameters. Dictionary items have special rigid formats. We discuss the different types of dictionary items after we detail how data is stored within a data item.

Data Items

A data item is the equivalent of a record in common data processing terms. Each data item is divided into attributes. An attribute is similar to a field in the more usual terminology. Attributes can be further subdivided into multiple values. A value can be divided once more into multiple subvalues. The PICK system's values and subvalues have no corresponding equivalent in the common file's structure.

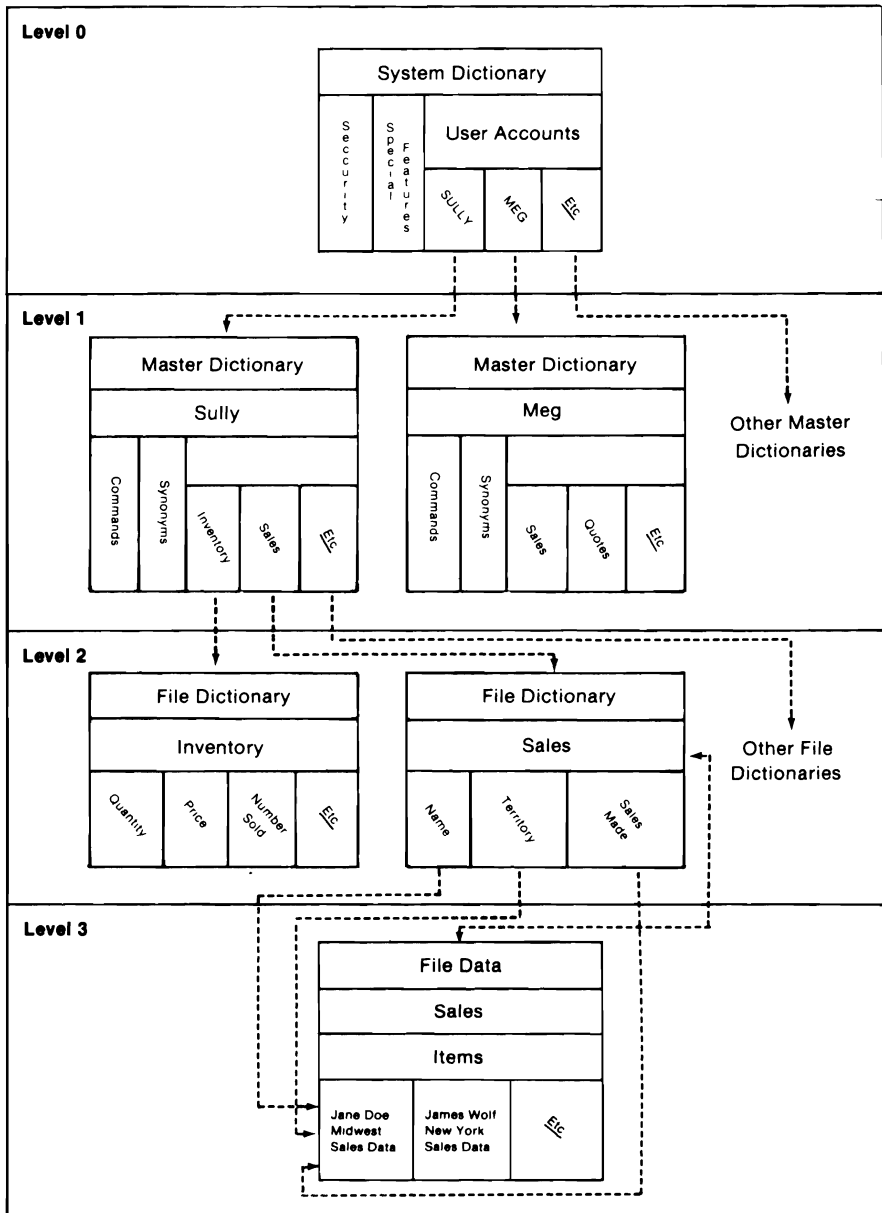


Figure 5.2. Representation of the PICK file system.

Figure 5.3 illustrates the structure and possible use of the PICK data item. The example data item would relate to one of the items in the file data SALES on Level 3 of figure 5.2. The attribute names in figure 5.3 would relate to the items in the File Dictionary SALES on Level 2 of figure 5.2. The values and subvalues would relate to the contents of the data item in the File Data.

In this particular example, the NAME attribute has a single value of JANE DOE with no subvalues. The TERRITORY attribute is also single valued with no subvalues. The SALES MADE attribute contains two values, CROSSLY’S SPORTING GOODS and BIGBILL’S, which represent customer names. Each value includes subvalues that contain the amount of each individual sale made to that customer. Finally, the item name, MW 9501, contains the identification number for the salesperson Jane Doe.

So, the data item MW9501 contains the information that Jane Doe in the midwest territory made sales to Crossly’s Sporting Goods of \$281.55 and \$5781.33. She also made sales of \$88.09, \$55.67, and \$987.10 to Big Bill’s.

Defining, inputting, and handling multiple values and subvalues is accomplished using PICK/BASIC Programs and the ACCESS processor to generate customized reports.

Item Name : MW9501

<u>Attribute Name</u>	<u>Value(s)</u>	<u>subvalues</u>
NAME	JANE DOE	- -
TERRITORY	MIDWEST	- -
SALES MADE	CROSSLY’S SPORTING GOODS	28155 578133
	BIG BILL’S	8809 5567 98710

Figure 5.3. PICK data items

Dictionary Items

PICK includes several different types of dictionary items. These are:

- File definitions
- File Synonyms
- Attribute definitions

File definition dictionary items are used to define a file to the systems. The file definition tells the PICK system where the file resides on the disk and how to access the file from the disk.

File synonym dictionary items tell the system which file is related with the synonym name. when the synonym name is used, the system will access the related file.

Attribute definition dictionary items are used to describe the information contained in a single attribute in a data item. individual definition dictionary items help facilitate the accessing of data contained within data items.

File Handling

The PICK system includes four processors designed to manage files:

```
CREATE-FILE  
CLEAR-FILE  
DELETE-FILE  
COPY
```

The function of each of these processors is evident from the name. We will discuss the details of using each of the four processors in this section.

Creating Files

In this section we will discuss file creation in the PICK system. Creating a file is more difficult in the PICK system than in other operating systems. Information about the length of items in a file as well as the number of those items must be used to calculate a number that is used by the system to determine the manner in which the data in the file is stored on the disk. This number is called the modulo or F/MOD.

The modulo will affect both the efficiency of the disk storage and disk access speed. Disk access speed is critical to the performance of the system. It is important to choose an appropriate modulo so that the system does not waste disk space and does not suffer a degradation of performance. A discussion of the functions of the modulo will follow in the next section.

The Modulo

A modulo must be selected for both the dictionary and data portions of the file. The PICK system uses the modulo to store and to find an item within the file. A modulo is required to locate both dictionary and data items. The modulos for the dictionary and data portions of the file need not match.

Figure 5.4 conceptualizes the techniques used in storing and accessing a particular item in a file. The figure uses a data item in a File Data. However, the technique is identical for dictionary items in Dictionary files.

The system uses the F/BASE of the specified file to find the area on the disk in which the file is stored.

Items within the file are stored in groups. A group is a number of items that are stored as a single unit on the disk. Each group has its own ID. The system uses a hashing routine to obtain the Group ID of the item of interest. For our purposes, we can treat the hashing routine as a “black box” that takes the modulo and the item name as input and returns the Group ID of that item.

Once the proper group has been identified, the system can easily locate the item of interest by sequentially searching the items that compose the group.

In Figure 5.4 we depicted only a few groups. However, in a real application, a file will contain many groups. Dividing the items into groups allows the system to locate an item quickly.

The modulo provides the system with information regarding the expected size of the file. The modulo specifies the approximate number of groups that are reserved for the file. Selecting a proper modulo is critical in maintaining system performance. Selecting too large a modulo wastes disk space. The wasted space is taken up by groups that are underused and the overhead involved with setting up those groups. Selecting too small of a modulo will cause the time needed to access a particular item to greatly increase. The access time is increased by the searching through groups that contain too many items in order to find the item of interest.

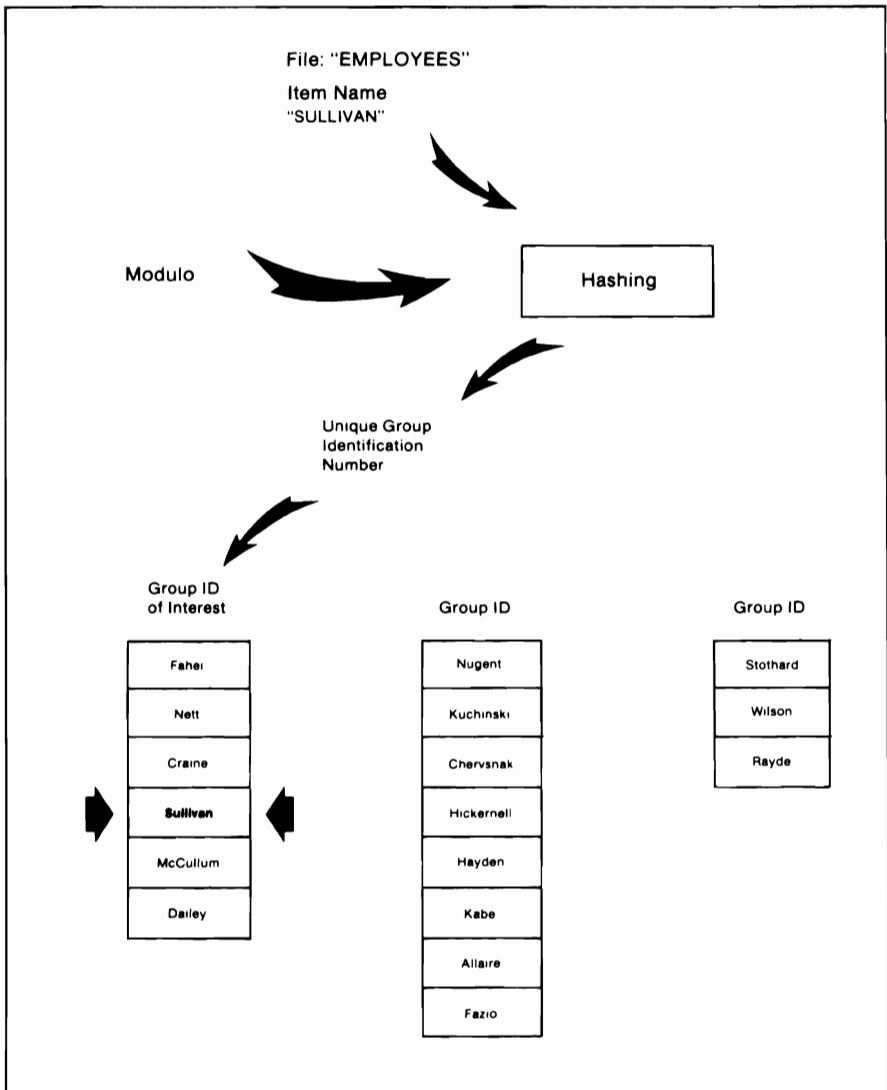


Figure 5.4. Storage and access of PICK files.

Modulo Selection

To select an appropriate modulo for a file, the approximate number of items the file will contain and the size of each item in bytes must be known. Fortunately, the current trend in PICK application software is to automate the selection of the appropriate modulo when a new data file is created. Instead of going through a complex series of calculations you might be prompted to:

- Enter the name of the file to create
- Enter the approximate number of items to be stored in the file
- Enter the approximate number of characters in each item

When each question has been answered, a new file will be created and the appropriate amount of disk space allocated.

Clearing Files

The contents of a file can be cleared without that file itself being destroyed. This procedure is useful for cleaning out a file where data regularly changes, such as a file used to accumulate weekly sales data. Either the dictionary portion or the data portion of the file may be cleared. The form of the command to clear the dictionary portion of a file follows:

```
CLEAR -FILE DICT dictname [RETURN]
```

where dictname indicates which dictionary should be cleared. Note that clearing a dictionary file does not cause the items that define data files related to that dictionary to be cleared. Only the items such as attribute definitions are removed from the dictionary. The command to clear a data portion of a file has the following format:

```
CLEAR -FILE DATA dictname(, dataname) [RETURN]
```

where dictname indicates which dictionary file the data file with the name dataname is related to. The data file dataname will be cleared

of data items by this command. Note that if `dataname` is not specified, only the data file with the same name as `dictname` will be cleared.

For example, suppose we have the dictionary file `INVENTORY` with related data files `INVENTORY` and `CLEVELAND`. The following command:

```
CLEAR-FILE DICT INVENTORY [RETURN]
```

would cause all of the dictionary items in the `INVENTORY` dictionary file except the data file definitions, `INVENTORY` and `CLEVELAND`, to be removed. Also, the following command:

```
CLEAR-FILE DATA INVENTORY [RETURN]
```

would cause the data file `INVENTORY` in the dictionary file `INVENTORY` to be cleared of all items. Finally, the command shown below:

```
CLEAR-FILE DATA INVENTORY ,CLEVELAND [RETURN]
```

would cause all of the data items to be removed from the data file `CLEVELAND` in the dictionary file `INVENTORY`.

Removing Files

Files that are no longer needed should be removed from the file system. The `DELETE-FILE` processor is used to remove files. Either the dictionary or the data portion of the file can be removed. Alternatively, the dictionary file and all related data files may be removed.

The command to remove only the dictionary portion of a file has the following format:

```
DELETE-FILE DICT dictname [RETURN]
```

where `dictname` specifies which dictionary file is to be removed from the Main Dictionary. Note that a dictionary file that has one or more data files related to it cannot be removed in this fashion.

The command to remove only the data portion of a file has the following form:

```
DELETE-FILE DATA dictname{,dataname} [RETURN]
```

where dictname specifies the data file to be removed and dictname specifies the dictionary file that the data file is related to. As with the CLEAR-FILE processor, if no dataname is specified, the data file with the same name as dictname will be removed.

The form of the command to remove a dictionary file with all data files related to it is shown below:

```
DELETE-FILE dictname [RETURN]
```

where dictname specifies the dictionary file to be removed.

For example, suppose we have a dictionary file INVENTORY with related data files INVENTORY and CLEVELAND. The following command:

```
DELETE-FILE DICT INVENTORY [RETURN]
```

would cause the following message to be displayed:

```
[420] DICTIONARY FILE DELETION CANNOT BE  
DONE WITHOUT DELETION OF DATA SECTION(S)  
FIRST,
```

The following command:

```
DELETE-FILE DATA INVENTORY [RETURN]
```

would cause the data file INVENTORY in the dictionary file INVENTORY to be removed. In addition, the following command:

```
DELETE-FILE DATA INVENTORY ,CLEVELAND [RETURN]
```

would cause the data file CLEVELAND to be removed from the dictionary file INVENTORY. Now, the command:

```
DELETE-FILE DICT INVENTORY [RETURN]
```

would cause the file to be removed if the command was reissued.

The command shown below represents an easier method of accomplishing the removal of the entire file:

```
DELETE-FILE INVENTORY [RETURN]
```

This command will cause the dictionary file INVENTORY along with all related data files to be removed.

Items may be copied from any dictionary or data file to several different destinations. These destinations include other dictionary or data files, the terminal screen, or the line printer.

Copying to the Terminal Screen or the Printer

The contents of either dictionary or data items can be viewed by copying the item of interest to the terminal. If a permanent copy is required, the items of interest can be copied to the printer. The command to accomplish such a task has the following format:

```
COPY {DICT} (name) (sending items) (options) [RETURN]
```

where DICT is used to specify that the dictionary portion of the specified file should be the source of the items to be copied. If DICT is not specified, the items to be copied will be taken from the data portion of the file.

The parameter (name) is used to specify the file from which the items to be copied should be taken. Note that (name) can be of the form dictname or dictname,dataname as discussed in the previous sections.

The parameter (sending items) is used to specify which items should be copied. One or more items can be specified. If multiple items are specified, the item names must be separated by blanks. Note that an asterisk (*) can be used to specify all of the items in the file.

The parameter (options) is used to specify the destination of the copy and the manner in which the copy will be performed. The following options may be used to specify either the terminal screen or the printer as the destination:

- P** The P option is used to specify the printer as the destination of the copy.
- T** The T option is used to specify the terminal as the destination of the copy.

The options listed below can be used to modify the manner in which data is displayed on the terminal, screen, or the printer:

- F** The F option is used to specify that each item is to appear on a new page.
- N** The N option is used to specify that no pause should occur at the end of a page. Usually, the system will wait until the return key is pressed before proceeding to send the next page to the terminal.
- S** The S option is used to suppress the line number display that usually appears to the left of each line in the item.
- X** The X option is used to specify that the items should be displayed by their hexadecimal representation. The hex code listing is useful in finding characters that are not usually printed.

For example, the following command:

```
COPY INVENTORY / ( PFN) [RETURN]
```

would cause the contents of all of the items in the data file INVENTORY related to the dictionary INVENTORY to be listed on the printer. Each item would be listed on a separate page, and there would be no pause between pages as the output was being generated.

Copying to Another Item in the Same File

Items may be copied to another item in the same file. The command to accomplish such a task has the following format:

```
COPY{DICT} (name) (sending items) {(options)} [RETURN]
```

When the return key is pressed, the system will respond with the following prompt:

TO :

The names of the items that are to receive the copy of the (sending items) should now be input. Multiple items must be separated by a space. The last item must be followed by pressing the return key. The meaning of the parameters DICT, name, and sending items, remain as discussed in the preceding section. When copying to other items, the options parameter can have the following values:

- D** The D option causes the (sending items) to be deleted after the copy is made.
- I** The I option will cause the display of the item name to be suppressed in the diagnostics generated as the command is executed.
- N** The N option is used to specify that no new items should be created. The copy will only take place if the destination item already exists.
- O** The O option is used to specify that existing items should be overwritten. Usually, only new items are copied to.
- S** The S option is used to suppress error messages. Error messages are generated when items are not copied.

Note that some of these options have different meanings when used with a copy to a file rather than to a device.

For example, the following command sequence:

```
COPY NUMBERS ONE TWO THREE [RETURN]
```

```
TO: UNO DOS TRES [RETURN]
```

would cause the contents of the items ONE, TWO, and THREE to be copied to the items UNO, DOS, and TRES, respectively.

Note that the option N will always cause no copies to be made unless the O option is also specified. The copy command will not

overwrite existing items unless it is specifically told to do so with the O option.

Copying to Another File

Items in one file can be copied to items in a different file. The form of the command to copy to a different file is identical to the form to copy to the same file, except the TO: prompt is answered in a different manner. Instead of responding with just a list of destination items, a destination file is also specified. The response to the TO: prompt to copy to a different file has the following form:

```
{(DICT)(name)} {receiving items} [RETURN]
```

The DICT parameter is used to specify a dictionary portion of a file. The (name) parameter is used to specify the name of the receiving file. The parameter (receiving items) is used to specify the item names that are to receive copies of the data. Multiple items must be separated by a blank. If no (receiving items) are specified, the names of the sending items will be used.

For example, the following command sequence:

```
COPY ACCUMULATOR / (D) [RETURN]
```

```
TO: (1984,OCTOBER) [RETURN]
```

would cause all of the data in the items of the file ACCUMULATOR to be transferred to the data file OCTOBER in the dictionary file 1984. After completion of this move, no data items would remain in the data portion of the file ACCUMULATOR.

File Ownership and Security

In this section we discuss the important topics of file ownership and file security, which control the privacy and integrity of any data stored within the PICK system.

Ownership

Each file is owned by the individual user who created the file. When a dictionary file is created using the CREATE-FILE processor, that file appears as an item in the user's Master Dictionary. Data files owned by the individual appear in the appropriate dictionary file.

Only the system user who owns a file can remove that file from his Master Dictionary. Also only the owner can use the CLEAR-FILE processor on the file. Other system users can retrieve and update that file depending on the lock values, as discussed next. A file owned by another user can be accessed only by using a file synonym. File synonyms were discussed earlier in this chapter.

Security

Both the retrieval and the update of a file is monitored by the system. Only those individuals having L/RET (retrieval lock) and L/UPD (update lock) codes that match those of a file are permitted access to that file. The L/RET and L/UPD codes for each individual user are specified in the user's identification item in the System Dictionary. Only the system administrator can change these lock codes.

The file retrieval lock codes are stored in attribute 5 of the file definition item in the owner's Master Dictionary. A retrieval lock code should not be instituted or changed for a file without first checking with the system administrator. If a file's retrieval lock code does not match one of those stored in the owner's user identification item, the owner will be denied access to his own file. A match with the file retrieval lock code is required in order to invoke all file processors except those that can change the file contents. Usage of these file processors is controlled by the update lock codes.

The update lock codes are stored in attribute 6 of the file definition item in the owner's Master Dictionary. An update lock code should not be put in use or changed for a file without first checking with the system manager. If a file's update lock code does not match one of those stored in the owner's user identification items the owner will be unable to update his own file. A match with the file retrieval lock code is required in order to invoke a file processor that can change

the content of the file. Such file processors are considered to be the following:

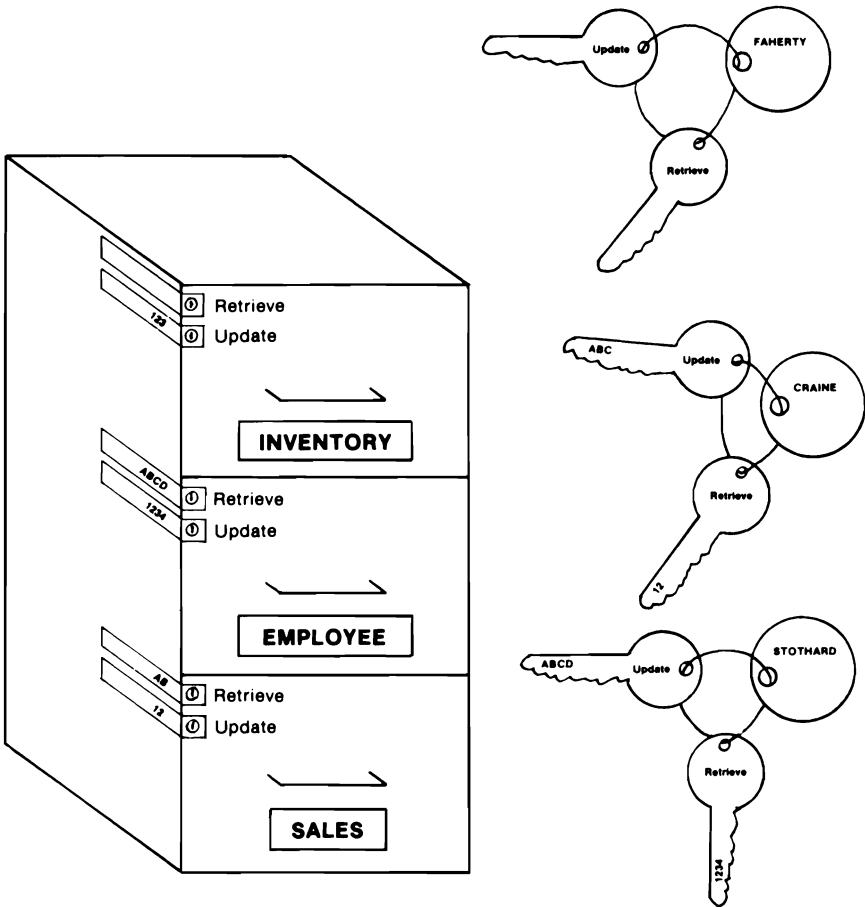
- COPY
- ED (The editor)
- PICK/BASIC (When writing to a file)
- RUN (Execute a PICK/BASIC program)
- The Assembler

The following message indicates that the individual does not have the proper L/RET and L/UPD code:

```
[210] FILE (name) IS ACCESS PROTECTED
```

Figure 5.5 contains an illustration that conceptualizes the lock codes used by the file system. The user's L/RET and L/UPD codes are represented as keys. Note that each user has a retrieve key and an update key. The file's L/RET and L/UPD codes are represented as locks on the file cabinet. Note that each file has a retrieve and an update lock.

The table below the filing cabinet shows the access permissions resulting from the locks and keys depicted. For example, the user FAHERTY is allowed to retrieve the inventory file, but he cannot change that file. Note that the INVENTORY file has a blank lock code. Any key will match a blank lock key.



User Name	INVENTORY		EMPLOYEE		SALES	
	Retrieve	Update	Retrieve	Update	Retrieve	Update
CRAINE	YES	NO	NO	NO	YES	YES
FAHERTY	YES	NO	NO	NO	NO	NO
STOTHARD	YES	YES	YES	YES	YES	YES

Figure 5.5. Conceptualization of file system lock codes.

6

PICK Command Tutorials

Introduction

The purpose of this chapter is to provide a series of tutorial sessions that will allow you to begin using PICK. The first session will discuss multi user operation. The remainder of the sessions will focus on the other PICK commands. In order to maximize comprehension and hold your attention, we have divided each PICK command into a working session. You can work with each of the commands in any order you prefer but it would be best if at first you worked with them in the order presented. Since many readers already have some experience with MS-DOS we have indicated, where possible, an MS-DOS command which corresponds to a PICK command in order to aid your comprehension.

Leaving the System Unattended

It is important to remember that since the PICK system utilizes the virtual memory concept, RAM contents should be emptied and written to disk when you leave the system unattended for an extended period of time. This can be accomplished by using the OFF command as described in tutorial session 12.

Terminal Control Language

The Terminal Control Language (TCL) provides the means of communication between the user and the computer system. After a successful system start is accomplished and a user logs on, control of the system is turned over to TCL, and the operating system prompt (>) appears on the left-hand side of your terminal screen. It is the TCL that allows entering of PICK commands and allows immediate execution of those commands when the [RETURN] key is pressed.

The [RETURN] key must be pressed after entering any PICK command. If you forget to press the [RETURN] key, the system will not respond.

Stopping TCL Command Execution

Cancelling TCL command execution can be accomplished by pressing the key labelled CTRL while simultaneously pressing the BREAK key. When this sequence is entered, command execution is terminated and you are placed in the DEBUG state. The screen will display the following:

```
I x , d
!
```

where x and d describe the location within the PICK software where the interruption occurred. The exclamation point (!) is the prompting character for the DEBUG mode. To exit the DEBUG mode type:

```
END[RETURN]
```

after the ! and return to TCL

or type:

```
OFF[RETURN]
```

after the ! and logoff the system

Note: Do not use the CTRL BREAK key sequence indiscriminately. Although you are prevented from exiting commands during critical processing, it is possible to cause disk errors. The sequence CTRL-X can be used to terminate lists of data displayed on the screen.

TCL Statement

A TCL statement is simply a PICK command line. The proper syntax for entering a TCL statement or command line is depicted below:

```
verb parameters ~options,,° [RETURN]
```

Parameters are usually file names. Options affect the operation of a verb and may specify routing to either the terminal or printer. Some verbs may permit multiple options. As each command is discussed the options of importance will be indicated.

The following editing functions are useful when entering TCL statements:

CTRL-X	Deletes entire line and reprompts the user for input.
CTRL-W	Backspaces over the last word entered.
CTRL-H	Backspaces over the last character entered.
CTRL-R	Retypes line after editing.
←	(backspace) deletes previous character.

Note: At the Logon prompt, when the back arrow is used to delete the previous character, characters such as \$\$\$ may be displayed...one for each time the key is pressed. This is caused by a "bug" within the PICK software, and is not a serious problem. If a mistake is made while entering an account name to logon to, use the sequence CTRL-X[RETURN] to redisplay the Logon prompt, or Enter TERM-TYPE-[RETURN] at TCL.

The SYSPROG Account

The SYSPROG (System Programmer) account is the system maintenance account and is automatically assigned the highest privilege level (SYS2) in the system. In addition, the ERRMSG (error message) file and the NEWAC file are defined in this account. Although these files “live” in the SYSPROG account, the ERRMSG file is accessible by **all** logged on users and is used to display various system error and information messages. The NEWAC file contains the verbs used by all accounts **other than** the SYSPROG account. When a new account is created, the verbs in this file become the master dictionary for the new account.

Table 6.1. Partial listing of ERRMSG and NEWAC files

Error Messages

YOUR OPEN FILES WERE CLOSED
FILE HAS NOT BEEN OPENED
VERB?
THE SPOOLER IS PRINTING A FILE ON THE PRINTER
Invalid floppy drive parameter
LOGGED OFF AT
Requested floppy drive is not installed
THE CURRENT JOB HAS BEEN ABORTED
PC-AT System verified.
THE SPOOLER IS INACTIVE
*** PC-AT RELEASE 1.0- SYSTEM DOES NOT VERIFY! ***
NO ITEMS PRESENT

NEWAC Verbs

BLOCK-PRINT
QSELECT
SET-FLOPPY
LISTPROCS
STARTPTR
SP-KILL
TERM
LISTVERBS
CHARGE-TO
LISTPTR

In a newly installed PICK PC/XT system, the only user accounts

on the system are the SYSPROG account and a tutorial account called TUTOR.

Note: Some PICK PC/XT systems may also include the WIZARD account which contains the WIZARD applications development system.

When a PICK PC/XT system is powered-on for the first time, usually after installation of the PICK operating system, it asks the user to enter the time, date and will display the message:

Verifying system modes...

If all is well the Logon message is displayed prompting the user to enter an account name. If SYSPROG[RETURN] is entered the user is placed at TCL and PICK waits for the next command to be entered.

When executing PICK commands it is important to note which account you are currently logged onto. Some commands require being logged onto the SYSPROG account in order to be executed. The SYSPROG account is also the account whose master dictionary contains the verbs required for system maintenance. For example, the execution of such commands as

- CREATE-ACCOUNT
- DELETE-ACCOUNT
- ACCOUNT-RESTORE
- SEL-RESTORE (selective file restore)
- FILE-SAVE
- POWER-OFF
- REBOOT

require that you logon to SYSPROG.

Note: If a command cannot be executed from the currently logged account the message:

[3] VERB?

is displayed after the command line has been entered and the [RETURN] key pressed. The system is informing you that the verb required to execute the command is not present in the currently logged account's master dictionary. A further qualification with certain

commands is that you be logged onto a terminal attached to line zero. When this condition is encountered the message:

VERB is a reserved verb for line 0

is displayed. VERB is the verb which cannot be executed on the currently logged line. One example of such a verb is the COLOR verb. The COLOR verb can only be executed from line 0. Whenever either qualification must be met we will say so during the tutorial on the command.

PICK Command Summary

There are a number of commands associated with PICK, but it is beyond the scope of this book to cover all of them. All commands can be found in the documentation provided by PICK Systems, Inc. All of the commands we will cover in this chapter are depicted in figure 6.1.

Command Name	Command Function
ACCOUNT-SAVE	Backup individual accounts
ACCOUNT-RESTORE	Restore backup data to an account
COLOR	Set primary color of main terminal
CREATE-ACCOUNT	Create new accounts
CT	Copy file contents to terminal
DELETE-ACCOUNT	Deletes an account and its files
FDISK	Creates/changes active disk partition
FILE-SAVE	Complete system backup
FFORMAT	Formats floppy disks
LISTACC	Lists logon data for each account
LISTCONN	Lists ACCESS connectives
LISTFILES	Lists the files on an account
LISTPROCS	Lists the PROC's for an account

LISTU	Lists all currently logged on users
LISTVERBS	Lists the verbs for an account
LOGTO	Logon to another account
MSG	Send message to another user
OFF	Logoff system
POVF	Display the available disk space
POWER-OFF	Flushes RAM to disk for shutdown
REBOOT	Reboots system without power down
RESTORE	Restores operating system/data files from floppy disk
SYSTEM	
RUNOFF	Formats edited text for printing
SET-TIME	Set current system time
SET-DATE	Set current system date
STARTPTR	Activate port and spooler for serial or parallel printer
TERM	Set up terminal/printer parameters
TERM-TYPE	Automates TERM command
TIME	Outputs system time and date
WHO	Outputs current account status

Figure 6.1. PICK command summary

Some of the PICK commands are actually a series of command lines stored in a special file called a PROC. We will learn more about PROCS in chapter 8.

In each of the following sessions the use of each command will be illustrated in detail. We recommend that you sit at a terminal and try each example as you read. Some commands have options associated with them and allow you to use more than one option at a time. User entries will be indicated in underlined type.

Session 1: Multi-User Operation

We assume your PC/XT has been configured with two serial ports properly installed. If the hardware is not configured properly the PICK software will not recognize any devices attached to these ports. When a PICK PC/XT system is booted, multi-user operation is automatically invoked *provided that* the the steps described below are followed.

During start-up the PICK software is automatically verified. Also, at this time, the spooler with the default parallel printer is activated. The parallel printer port is utilized to permit the use of two terminals attached to the serial ports. If the software does not verify multi-user operation is permitted, but not recommended.

To invoke multi-user operation follow these steps:

1. Boot the system. A screen similar to figure 6.2 is displayed:

Hard Disk Boot , PICK Systems IBM PC XT , 1985

Copyright (C) 1985 Pick Systems, as an unpublished work. All rights reserved. This work is the property of, and embodies trade secrets and confidential information proprietary to Pick Systems, and may not be reproduced, copied, used, disclosed, transferred, adapted or modified without express written approval of Pick Systems.

SPOOLER STARTED

Linking WORKSPACE for line 0

Logon

```
<<<    ---THE PICK SYSTEM---    >>>
<<<--00:00:24 PC-XT Version 1.3-14 MAR 1985 - >>>
```

Please enter the time >

Figure 6.2. PICK boot screen.

2. Enter the time and date when prompted. After the date has been entered, the messages:

```
The printer control block has been initialized.  
Verifying system modes, Please wait...
```

are displayed.

3. Allow the system to verify the software.
4. When the "System verified" message is displayed, the system is ready for use with the main terminal, two additional terminals and a spooled parallel printer. The Logon message is then displayed.
5. If the software does not verify see the procedures described in the session on the RESTORE command.
6. If a terminal is attached to one of the serial ports, press [RETURN] to display the Logon message.

Session 2: ACCOUNT-RESTORE

The ACCOUNT-RESTORE command permits PICK users to restore files selectively by account. It is possible to restore files from a complete system backup (FILE-SAVE) or from an ACCOUNT-SAVE.

The format of the ACCOUNT-RESTORE command is:

ACCOUNT-RESTORE

To use the command, follow these steps:

1. Log onto the SYSPROG account.
2. Insert a PICK formatted floppy disk in the drive.
3. Type:

```
ACCOUNT-RESTORE new accountname[RETURN]
```

at TCL. PICK will prompt you to enter the name of the account you wish to restore.

Note: It is not permitted to restore an account which already exists. If an existing accountname is entered the message:

ACCOUNTNAME EXISTS ON FILE

is displayed.

The primary purpose of the ACCOUNT-RESTORE verb is to add a new account or an application software package and its files to a pre-existing PICK system.

Session 3: ACCOUNT-SAVE

The ACCOUNT-SAVE command provides the means of making backup copies of files which belong to individual accounts. In order to use the command you must be logged onto the SYSPROG account. ACCOUNT-SAVE is similar to the BACKUP command in MS-DOS but backs up an individual account (sub-directory) rather than an entire disk.

The format of the ACCOUNT-SAVE command is:

ACCOUNT-SAVE

To use the command, follow these steps:

1. Log onto the SYSPROG account.
2. Insert a PICK formatted floppy disk in the drive.
3. Type:

ACCOUNT-SAVE[RETURN]

at TCL. The screen will display:

TAPE ATTACHED BLOCK SIZE: XXX
FILE-SAVE TAPE LABEL =

where XXX indicates that the disk block size in bytes and tape label is an optional entry. If you choose not to give the disk a label just press [RETURN]. The system will then display:

ACCOUNT NAME =

Enter the name of the account to back up and press [RETURN]. As the system proceeds with the backup each file will be displayed on the screen.

Session 4: COLOR

The purpose of the COLOR command is to utilize the features of the IBM color monitor. With this command it is possible to determine the color of the text displayed as well as set the various screen attributes such as reverse video, blinking or non-blinking cursor, etc. The colors available for text display are:

- BLACK
- BLUE
- GREEN
- CYAN
- RED
- MAGENTA
- BROWN
- WHITE

Use of black should be avoided. Yellow and brown are recommended because they are easy on the eyes when using a terminal for extended periods.

The screen attributes/switches available are:

- /B or /BLINK
- /NB or /NOBLINK
- /F or /FULL
- /H or /HALF
- / or /REVERSE
- /NR or /NOREVERSE

Use of this command is restricted to the SYSPROG account and can only be executed from line 0.

The format of the COLOR command is:

COLOR foreground color background color/attribute

To change the foreground color type:

COLOR GREEN[RETURN]

To change the background color type:

COLOR ,WHITE[RETURN]

The command line:

COLOR MAGENTA,WHITE FULL[RETURN]

will set the foreground color to MAGENTA, the background color to WHITE, and the FULL intensity attribute.

Syntax is important with this command. For example, if you were to type:

COLOR BROWN,NOB[RETURN]

The message:

```
***ERROR*** "NOB" is unexpected. We are ex-
pecting a background color.
```

is displayed. There are two errors in this command line. After the comma, there must be a color and NOB is not the switch for NOB-LINK. You must either spell out the attribute or use its abbreviation of NB.

If you were to type:

COLOR BROWN/NOB[RETURN]

the message:

```
***ERROR*** "NOB" is unexpected. We are ex-
pecting a switch argument.
```

is displayed.

The error in this command line is that NOB is not an acceptable switch for the NOBLINK attribute. This procedure can be automated at the time of logon through the use of a PROC. We will see how to do that in chapter eight. You may experiment with this command. However, it is best to limit your experimentation to foreground colors or attribute switches.

Session 5: CREATE-ACCOUNT

This command is used to create new user accounts. CREATE-ACCOUNT is analogous to the MD (make directory) command in MS-DOS which creates sub-directories on the hard disk.

CREATE-ACCOUNT can only be executed from the SYSPROG account. The format of this command is:

CREATE-ACCOUNT

To create a new account enter:

```
CREATE-ACCOUNT[RETURN]
```

The system will display:

```
ACCOUNT NAME?WALT[RETURN]
L/RET-CODE(S)?[RETURN]
L/UPD-CODE(S)?[RETURN]
PRIVILEGES?SYS1, SYS2 or [RETURN]
MOD, SEP?[RETURN]
CREATE-FILE (DICT WALT 29,1)

[417]FILE 'WALT' CREATED; BASE = 4555,
MODULO = 29, SEPAR = 1,

244 ITEMS COPIED
'WALT' ADDED
'WALT' UPDATED
PASSWORD? enter a password or press[RETURN]
FINISHED
```

>

Usually, an account name, a privilege level, and a password will be entered. System default values can be entered for L/RET and L/UPD by pressing [RETURN]. If a password is not entered or needs to be modified this can be accomplished later using the PASSWORD command.

Changing the Password

To add or modify a password, the PASSWORD command must be used. Logon to the SYSPROG account and at TCL enter:

```
PASSWORD[RETURN]
```

The system will display:

```
ENTER ACCOUNT NAME?  
WALT[RETURN]  
ENTER NEW PASSWORD?  
TEST[RETURN]  
ENTER NEW ACCOUNT NAME?  
[RETURN]
```

You are prompted to enter another account name. If there are no additional passwords to change pressing [RETURN] will clear the screen and return you to TCL.

To remove an existing password enter:

```
PASSWORD[RETURN]
```

The system will display:

```
ENTER ACCOUNT NAME?  
WALT[RETURN]  
ENTER NEW PASSWORD?  
TEST[RETURN]  
  
ENTER NEW ACCOUNT NAME?  
[RETURN]
```

>

Session 6: CT (Copy to Terminal)

The purpose of this command is to copy or display a list of selected items on the terminal screen or, optionally, the system printer or line printer. In chapter four we provided a few examples on the use of this command. For clarity we will restate the format of the command and provide additional examples of its usage. Before using this command with a printer, we recommend you read the tutorial session in this chapter on the STARTPTR command or be certain there is a printer properly attached to your system. The format of the CT command is

CT filename itemlist (option)

where filename is the name of the file whose contents is to be listed and itemlist is a list of the items to be listed. Option is usually LPTR for line printer.

Using the CT Command

We will illustrate the use of the CT command with two types of examples. One example will show you how to output data to a terminal. The other example will show you how to output data to a printer. The terminal examples will be depicted as they should appear on your screen. The examples specifying the LPTR option will not be shown in a figure because the output generated is voluminous. We recommend that you execute the command, as illustrated, to give you practice using this command option as well as to provide you with a printout of each listing to study. All examples presume you are logged on to the SYSPROG account.

Suppose you were to type:

```
CT MD SYSPROG[RETURN]
```

the system will display:


```
      SYSPROG
001 PQ
002 HTERM ,,,,,,80,60,I
003 P
004 HSET-SYM PSYM
005 P
006 O
007 HWHD
008 P
009 O
010 ODocumentation addenda may be reviewed +
011 Oby typing 'ADDENDA' at TCL.
012 HCOLOR BROWN
013 P
014 HTERM-TYPE
```

Figure 6.3. CT MD SYSPROG display

SYSPROG is a PROC which is executed every time a user logs onto the SYSPROG account. Its function is to check the terminal type, display the user account name, and display the message concerning documentation addenda. As another example, type:

```
CT DOC * (LPTR) [RETURN]
```

Entering this command line will generate a hard copy printout of the supplement to the PICK documentation. When executed, you will find that the printout of this file, in version 1.3 of the PICK operating system, occupies 27 pages. Instead of indicating a specific attribute name, we used the * (asterisk) as a **wildcard character**. The wildcard character means list all items in the file called DOC.

Consider the following example. Type:

```
CT SYSPROG-PL COLDSTART(P)
```

The print out should resemble that depicted in figure 6.4.

```
COLDSTART
001 PQ
002 HTERM-TYPE
003 P
004 HSET-SYM PSYM
005 P
006 O
007 O
008 OPlease enter the time +
009 RI
010 IP
011 IF #A G 10
012 HSET-TIME
013 A
014 P
015 10 C
016 O
017 OPlease enter the date +
018 RI
019 IP
020 IF #A G 20
021 HSET-DATE
022 A1,11
023 A
024 P
025 20 C
026 HSTARTPTR 0,0,1,S1 (S
027 P
028 O
029 OVerifying system modes, Please wait...
030 HVERIFY-SYSTEM
031 P
032 HOFF
033 P
034 (SYSPROG-PL USER-COLDSTART)
```

Figure 6.4. Listing of COLDSTART PROC

This is a PROC which is executed every time the system is booted. It prompts you to enter the correct time and date, automatically starts a printer and spooler, verifies the system, and logs you off. More information will be provided about this PROC in chapter 8. Also note

that the P option can be used interchangeably with LPTR and have identical results.

Session 7: The DELETE-ACCOUNT Command

This command permits you to delete a user account and all of its files from a PICK system. DELETE-ACCOUNT will delete the master dictionary and all data files from an account. The RMDIR (remove sub-directory) MS-DOS command is analogous to DELETE-ACCOUNT. In order to execute the command, you must be logged onto the SYSPROG account. It is also recommended that no other users be logged onto the system at this time since an item in the SYSTEM dictionary will be deleted. To use the command, logon to SYSPROG and enter:

```
DELETE-ACCOUNT[RETURN]
```

The system will display:

```
TO THE PRINTER (P)?
ACCOUNT NAME ?
FILES TO DELETE IN ACCOUNT: WALT TIME DATE
PAGE #
```

FILE	BASE	MOD	SEP
FILE1	3245	29	1
FILE2	3678	29	1
FILE3	4532	29	1

```
,
,
,
```

```
STILL WANT THE ACCOUNT DELETED?
```

If Y[RETURN] is entered in response to the TO THE PRINTER question a list of all files to be deleted will be printed. Otherwise entering N[RETURN] or [RETURN] will suppress the printer listing. For future reference it is a good idea to get a printout of all of the deleted files. Next, an existing account name must be entered. The system will then search the disk and display all data files asso-

ciated with the account until all files owned by this account are displayed. PICK gives you a chance to change your mind about deleting the account. When the message

```
STILL WANT THE ACCOUNT DELETED?
```

is displayed, the only way to delete the account is to enter `Y[RETURN]`. Anything else returns you to TCL.

If there are no data files owned by the account, only the files associated with the account's master dictionary will be deleted. When an account has successfully been deleted, there is no message displayed. To verify that an account has been deleted try logging on to that account by typing:

```
LOGTO accountname[RETURN]
```

If the account has been deleted, the system will display:

```
USER-ID?
```

and you will be returned to TCL.

If you try to delete an account and an error message is displayed such as:

```
filename EXISTS ON FILE
```

This may mean that you will have to use the EDITOR to delete that file before the account can be deleted. *See* chapter seven on the EDITOR.

Session 8: FDISK

The PC/XT allows up to four different operating systems to reside on the hard disk. Each operating system exists within the confines of the disk space allocated for it. The purpose of the FDISK command is to select and determine which operating system is executing. Only one operating system may be executing at a time. The FDISK command functions similarly to the FDISK command in MS-

DOS. This command must be executed from the SYSPROG account on line 0. The format of the command is:

FDISK[RETURN]

To invoke FDISK type:

FDISK[RETURN]

The system will display:

```
IBM PERSONAL COMPUTER
FIXED DISK SETUP PROGRAM VERSION 1.00
(C)COPYRIGHT PICK SYSTEMS 1984
```

FDISK OPTIONS

CHOOSE ONE OF THE FOLLOWING:

1. CREATE PICK PARTITION
2. CHANGE ACTIVE PARTITION
3. DELETE PICK PARTITION
4. DISPLAY PARTITION DATA

ENTER CHOICE: []

Although four choices are displayed on this menu the create PICK partition choice is non-functional because a PICK partition is automatically created during installation. If choice 1 is selected the message:

*PICK PARTITION EXISTS , CANNOT CREATE ,

is displayed.

The most commonly selected option is number 2. When this option is chosen information about each partition is shown and the message:

```
ENTER THE NUMBER OF THE PARTITION YOU WANT
TO MAKE ACTIVE.....[ ]
```

is displayed.

In order to use the newly selected operating system, the system must be rebooted. Once a non-PICK operating system has been activated it is now possible to execute option number 3. If option 3 is chosen **before** another operating system is activated, the message:

```
* ACTIVATE ANOTHER PARTITION BEFORE DELET-  
ING PICK.
```

is displayed.

The purpose of option 4 is to display the number and status of each operating system which resides on the disk. To exit the FDISK menu press the ESC key and you will be returned to TCL.

Session 9: FILE-SAVE

The FILE-SAVE command provides the method for the complete backup of the data files in a PICK system. This command is similar to the BACKUP command in MS-DOS. The media used for backup in a PC/XT system is a floppy disk. Before disks can store data, they must first be formatted using the FFORMAT command described in session 10. In order to ensure continuity, it is recommended that you make frequent backups. No computer system can force you to make backups. That is your responsibility!

In the PICK documentation, backup media is referred to as tape. Each PC/XT floppy disk can store 360 Kbytes of data. Unless you have a limited number of small files, your data files will not fit on a single floppy disk. As additional floppies are required, you will be prompted by PICK to insert them. For this reason the process cannot be left unattended. FILE-SAVE must be run from the SYSPROG account. The format of the FILE-SAVE command is:

FILE-SAVE

The FILE-SAVE Options

The SAVE verb is responsible for executing this command. However, FILE-SAVE is actually a PROC which automates the FILE-SAVE procedure. A partial list of options associated with the SAVE

verb is depicted in figure 6.5. Within the FILE-SAVE PROC, the options specified are S,T, and D. Since a PROC is controlling the backup procedure you cannot specify any additional options. In order to change the options, the PROC itself must be modified.

Options	Function
D	Save data. If not specified files are not saved.
F	Allows file names to be printed as they are saved. If not specified, only the SYSTEM file and account names are displayed.
G	Disk (group format) errors are repaired.
P	Output file name list to printer.
S	Create STAT-FILE (file statistics) list for printing.
T	Output the FILE-SAVE to disk. If this option is not specified nothing will be saved to disk.

Figure 6.5. SAVE verb options

Using the FILE-SAVE Command

To use the FILE-SAVE command follow these steps:

1. Insert a formatted floppy disk in the drive.
2. Log on to SYSPROG and at TCL enter:

```
FILE-SAVE [RETURN]
```

The system will display:

```
TAPE ATTACHED BLOCK SIZE : 500      .  
,  
,  
FILE SAVE BEGINNING AT current date cur-  
rent time  
CONSOLE LISTING TO PRINTER? .....
```

DO YOU WANT A FILE STATISTICS REPORT?

BLOCK SIZE: 4000

ENTRY #1

In order for the FILE-SAVE to continue, the two questions depicted must be answered. Appropriate answers are Y or N followed by [RETURN]. As the procedure progresses the files which are saved are displayed on the screen along with any error messages. ENTRY #1 indicates the listing will be output to the printer upon completion of the FILE-SAVE procedure.

3. When additional disks are required the system displays:

```
load PICK PC DATA FILES #2 then type "C" to
continue
disk LABEL current date data files being
saved #
```

4. After the new disk is inserted enter C to continue.

5. Execution continues until all files have been saved.

6. Upon completion the message:

```
FILE SAVE FINISHED AT current time
```

is displayed.

7. If a file statistics report was specified, it is now output to the printer.

Additional Notes about FILE-SAVE

- Always have plenty of floppy disks ready to insert if you are backing up a lot of files.
- Execution time can be lengthy depending upon the number and size of the files.

- Since FILE-SAVE can be executed in a multi user environment it is possible that other users may need to access data being saved. During the procedure, the frames being saved are temporarily locked. When another user tries to access these frames, that user's terminal "hangs" until the files are unlocked upon completion of the FILE-SAVE procedure.
- The options implemented for the SAVE verb on a standard PC/XT system can be modified by changing line 055 in the FILE-SAVE PROC. See chapter 8 for more information.
- FILE-SAVE overwrites any existing data on the floppy disks.

Session 10: FFORMAT

The purpose of this command is to initialize or format floppy disks on the PC/XT for use by PICK. Before information can be written on any computer disk or tape, formatting must occur. Formatting is the process of creating a "road map" the operating system can follow when storing files, so it knows where information can be written. Files may not be stored on uninitialized disks.

Although FFORMAT is similar to FORMAT in MS-DOS, disks formatted by this program are not readable by PC or MS-DOS. This command must be executed from the SYSPROG account. The format of the FFORMAT command is:

FFORMAT

To format floppy disks insert a PC compatible floppy disk (usually double side, double density) in the drive and enter:

```
FFORMAT[RETURN]
```

at TCL. The system will display:

```
FFORMAT - Pick Systems
Insert diskette in floppy drive (A:)
and strike any key when ready
Formatting...
```

PICK does not allow any disks with bad sectors to be used. When such errors are encountered the screen displays:

```
368640 bytes total disk space
xxxxxx bytes available on disk
yyyyyy bytes in bad sectors
This diskette is unusable by PICK

Format another (Y/N) =N[RETURN]
T-DET
>
```

where xxxxxx is the number of bytes available on disk for storage, and yyyyyy is the number of bytes in bad sectors. The system asks if you want to format another disk. In this case N was entered. The system automatically detaches the floppy disk and returns you to TCL. If the formatting process was successful the screen displays:

```
FFORMAT - Pick Systems
Insert diskette in floppy drive (A:)
and strike any key when ready

Formatting...Format complete.

368640 bytes total disk space
368640 bytes available on disk

Format another (Y/N) =
```

If you respond with Y be sure to remove the first floppy and insert the new one before pressing a key. The system assumes the currently loaded disk is unformatted. As in the previous example when N is entered, the floppy drive is detached and you are returned to TCL.

Notes: Formatting erases any previously stored data on the disk. FFORMAT is used to format new disks or as a means of recycling used disks. If a disk is not in the drive when FFORMAT is executed the disk drive is accessed. The drive will continue to spin until a disk is inserted.

Session 11: Listing Utilities

The commands we will discuss in this session are referred to as listing utilities. The commands are a series of PROCS which automate their assigned task. Each command has a unique function but since they all involve listing various types of system data, we will include all of them in this session. The commands we will discuss are:

- LISTACC
- LISTFILES
- LISTPROCS
- LISTU
- LISTVERBS

When a listing generates more than one page of output to the terminal, the system pauses and waits for you to press [RETURN] to continue the listing. After any screen page of data is displayed, you can stop the listing and exit to TCL by entering CTRL X.

Since these commands require extensive disk access, they take longer to execute compared to some of the commands discussed earlier. This is particularly true during execution of LISTFILES and LISTPROCS. You know that the disk is being accessed by the illumination of the red indicator light on the hard disk drive.

LISTACC

This command displays the contents of the account history file. PICK automatically tracks system usage statistics and stores the data in a file. The data in the account history file is stored as an item for each user account consisting of the ACCount name, DATE and TIME of logon, amount of time CONNected to the system, the number of charge UNITS, and the number of PAGES printed by the system printer. However, the account history file for any account is not updated unless the system dictionary item for that account contains the value "U" in attribute 9. To check the values of attribute 9 for the accounts on your system type:

```
CT SYSTEM ACCOUNTNAME[RETURN]
```

where ACCOUNTNAME is the account name whose attributes you wish to view. Typical attribute values for the system dictionary item SYSPROG on a newly installed system are:

```
SYSPROG
001 D
002 2774
003 11
004 1
005
006
007
008 SYS2
009 L
010 10
```

Since a U is not present in attribute 9 of a newly installed SYSPROG account, you will have to use the EDITOR to place a U in attribute 9. This can be accomplished with the following entries:

```
EDIT SYSTEM SYSPROG[RETURN]
TOP
,G9[RETURN]
009 L
R[RETURN]
009 U[RETURN]
,F[RETURN]
`SYSPROG' FILED,
```

>

The accounting history file allows the system manager to supervise system usage. The format of this command is:

LISTACC accountname#line

where *line is the line to which the user's terminal is attached. For example, entering the command line:

```
LISTACC SYSPROG#1[RETURN]
```

may produce the following display:

```
PAGE 1                                TIME  DATE

ACC.....DATE ..TIME ..CONN .. UNITS PAGES

          *          *          *          *
SYSPROG#1  03/14  01:06 00:02    35
              01:08 00:02    73
              01:12 00:01   312
              01:28 00:15   110

END OF LIST

>
```

Note that the time is indicated in the 24 hour format so that 1:06 would indicate 1:06 AM. Connect time is in hours and minutes.

Notes: This listing cannot be output to the printer. In addition, if the #line is omitted after the accountname or a U is not present in the system dictionary item for any account, the message:

```
[401] NO ITEMS PRESENT,
'accountname' NOT ON FILE,
```

is displayed.

LISTFILES

This command permits each user to list the files present in any dictionary including the system dictionary. LISTFILES is similar to the DIR command in MS-DOS. Logging on to an account is required to execute LISTFILES. File listings may be generated from any account by any account. For convenience, the output generated by this command may be sent to the printer if the LPTR option is specified in the command line. The format of this command is:

LISTFILES dictname (option)

where option is LPTR or line printer. If an option is not specified the file listing is output to the terminal. If a dictionary is not specified, the files in the dictionary of the currently logged account will be displayed. The command line:

```
LISTFILES SYSTEM (LPTR)[RETURN]
```

produces a display similar to that depicted in figure 6.6.

```
PAGE 1 *** SYSTEM *** FILE DEFINING ITEMS 01 NOV 1985
```

SYSTEM.....	CODE	F/BASE.....	F/MOD.....	F/SEP...
ACC	D	3624	1	1
BLOCK-CONVERT	D	2687	11	1
CYCLONE	D	3647	19	1
PROCLIB	D	2742	23	1
SYSPROG	D	2774	11	1
SYSTEM	D	2676	11	1
SYSTEM-ERRORS	D	2709	11	1
TUTOR	D	3549	29	1
WALT	D	4555	29	1
WIZARD	D	5176	29	1
COLDSTART	Q	SYSPROG		

11 ITEMS LISTED.

Figure 6.6. LISTFILES SYSTEM (LPTR)

This command line produces a listing of the files in the SYSTEM dictionary. The command line:

```
LISTFILES (LPTR)[RETURN]
```

produces a display similar to that depicted in figure 6.7.

This command produces a listing of the files in the SYSPROG master dictionary.

LISTPROCS

LISTPROCS sorts all PROCS in a file or dictionary and displays the listing on the terminal or outputs the listing to the printer. Each listing will display the dictionary name and a short description de-

scribing the function of the PROC. The format of this command is:

LISTPROCS dictname option

where option is the LPTR or line printer. If an option is not specified the file listing is output to the terminal. Note that unlike other commands, the LPTR option is **not** enclosed in parenthesis. If parentheses are used the error message “ILLEGAL FORMAT” is displayed and nothing is printed. If a dictionary is not specified, the PROCS in the dictionary of the currently logged account will be displayed.

Assume you are logged onto the SYSPROG account and enter:

```
LISTPROCS LPTR[RETURN]
```

The system will output a list of PROCS similar to that depicted in figure 6.8. Each listing displays the PROC name within the master dictionary (MD) and a description of each PROC.

PAGE	1	*** MD ***	FILE DEFINING ITEMS	14 MAR 1985
MD.....		CODE	F/BASE.....	F/SEP...
DOC	D	2785		I
ERRMSG	D	3432		I
FUNCKEYS	D	3242		I
KEYBOARDS	D	3041		I
NEWAC	D	3510		I
PSYM	D	2988		I
STAT-FILE	D	2946		I
BP	DC	3057		I
POINTER-FILE	DC	2935		I
SYSPROG-PL	DC	3250		I
ACC	Q	ACC		ACC
BLOCK-CONVERT	Q	BLOCK-CONVERT		
M/DICT	Q			
MD	Q			
PROCLIB	Q	PROCLIB		
SYS-ERRS	Q	SYSTEM-ERRORS		SYSTEM-ERRORS
SYSPROG-BP	Q	SYSPROG		BP
SYSTEM	Q	SYSTEM		
SYSTEM-ERRORS	Q	SYSTEM-ERRORS		SYSTEM-ERRORS
WALT	Q	WALT		

20 ITEMS LISTED.

Figure 6.7. LISTFILES (LPTR)

PAGE 1

19:14:25 28 OCT 1985

MD..... Description.....

ACCOUNT-SAVE	(SYSPROG-PL ACCOUNT-SAVE)
ADDENDA	HSORT DOC
ADDENDUM	HCOPY DOC
CAT	(PROCLIB CAT)
CHARACTERS	(SYSPROG-PL CHARACTERS)
CHOO-CHOO	(PROCLIB CHOO-CHOO)
COLDSTART	(SYSPROG-PL)
COLOR	(PROCLIB COLOR)
CREATE-ACCOUNT	(SYSPROG-PL CREATE-ACCOUNT)
CT	(PROCLIB CT)
DELETE	(PROCLIB DELETE)
DELETE-ACCOUNT	(SYSPROG-PL DELETE-ACCOUNT)
EXCHANGE	(PROCLIB EXCHANGE)
FFORMAT	HT-ATT
FILE-SAVE	(SYSPROG-PL FILE-SAVE)
LIST-FILE-STATS	(SYSPROG-PL LIST-FILE-STATS)
LISTACC	(PROCLIB LISTACC)
LISTCONN	(PROCLIB LISTCONN)
LISTDICT	(PROCLIB LISTDICT)
LISTFILES	(PROCLIB LISTFILES)
LISTPROCS	(PROCLIB LISTPROCS)
LISTU	(PROCLIB LISTUSERS)
LISTUSERS	(PROCLIB LISTUSERS)
LISTVERBS	(PROCLIB LISTVERBS)
LOOP-ON	(PROCLIB LOOP-ON)
MONO	(PROCLIB MONO)
POWER-OFF	(PROCLIB POWER-OFF)
REBOOT	(PROCLIB REBOOT)
SET-FILE	(PROCLIB SET-FILE)
SET-FUNC	HRUN BP SET-FUNC
SET-KBRD	HRUN BP SET-KBRD
SET-LPTR	(PROCLIB SET-LPTR)
SNAKE	(PROCLIB SNAKE)
SYSPROG	HTERM-TYPE
T-SPACE	(PROCLIB T-SPACE)
WHICH	(PROCLIB

36 ITEMS LISTED.

Figure 6.8. Listing of PROCS on SYSPROG

In figure 6.8, the master dictionary of SYSPROG contains 36 PROCS. The description column shows that each of the listing PROCS which we are discussing reside in a file called PROCLIB. For example, LISTPROCS is stored as an item in PROCLIB (PROCLIB LISTPROCS). In order to display the contents of the LISTPROCS item, you must type:

CT PROCLIB LISTPROCS[RETURN]

LISTU

This command can be executed when you want to know who is logged onto the system. It generates output to the terminal only. The format of the command is:

LISTU

Entering the command line:

```
LISTU[RETURN]
```

with only one user logged onto the system produces the following display:

```
CH# PCBF NAME..... TIME.. DATE..LOCATION.....  
-16 0200 SYSPROG 17:47 14 MAR 85  
*00 0400 SYSPROG 17:29 01 NOV 85 Memory Mapped Monitor
```

```
2 ITEMS LISTED
```

```
>
```

The line number (CH#) issuing the command is indicated by the asterisk (*). PCBF provides the disk location via the frame number of the account's files. In the case of the PC/XT the memory mapped monitor is only available on line 0. If another user were currently logged onto the system, the display would indicate this fact by showing CH# 01 with the user name. The location would be asynchronous communication port 1. Remember that an addition to the memory mapped monitor (line 0) there is a maximum of two serial ports available on a standard PC/XT. If a serial printer is used, only one port remains available. Since the printer is not treated as a user, its status is not displayed here.

LISTVERBS

The purpose of this command is to output an alphabetically sorted list of verbs available for use by an account. One of the security methods provided by PICK is to limit the presence of certain verbs to the

PAGE 1

17:02:28 01 NOV 1985

M/DICT... *A1..... *A2..... *A3.. *A4..... *A5.....

:ABSLOAD	P	D7			
:STARTSPOOPER		500A	6086	10AF	
:SWD	PA	35	1090	44	
:SWE	PA	35	1090	45	
:SWX	PA	35	1090	58	
:SWZ	PA	35	1090	5A	
ABS-DUMP	P	118E			
ACCOUNT-REPTORE		80D7			
ADD	PA	40A0			
ADDX	PA	A0			
B/ADD	PZ	2	53	57	CS
B/DEL	PD	2	53	57	CS
B/UNLOCK	P	4058			
BASIC	P	2	BE		UP
BATCH	P	208A			
BLOCK-PRINP		500A	41		
CATALOG	P	2	E0		
CHARGE-TO	P	6032			
CHARGES	P	5032			
CHECK-SUM	PX	35	4F		
CLEAR-BASIP-LOCKS		500A	30EE		
CLEAR-FILEPO		8F			
COMPARE	P	2	17D		
COPY	PZ	2	8C		UZ
COPY-LIST	PL	3013			
COUNT	PB	35	1069		
CREATE-FILPO		8D			
DECATALOG	PY	2	E0		N
DEL-ACC	P	E6			
DELETE-FILPO		108F			
DELETE-LISPL		3064			
DIVD	PD	40A0			
DIVX	PD	A0			
DTR	P	10A0			
DTX	P	10A0			
DUMP	PZ	42			
ECHO	P	5080			
ED	PE	2	D		CUPN
EDIT	PE	2	D		CUPN
EDIT-LIST	PL	13			
FDISK	P	E6			
FIND	P	E6			
GET-LIST	PL	1064			
GROUP	P	2	50A0		F
HASH-TEST	PA	35	106A		
INTER	P	308A			
ISTAT	PA	35	6A		
ITEM	P	2	30A0		N
L	PA	35	4D		
LINK-WS	P	40AC			
LIST	PA	35	4D		
LIST-ITEM	PA	35	4D	508E	
LIST-LABELPA		35	4D	7D	

Figure 6.9. LISTVERBS

“master” account, SYSPROG. There is a predetermined set of verbs provided as standard equipment with a PICK system. When application software is added to a system, the list of verbs is likely to increase as each software package may add new verbs of its own to the system. The format of this command is:

LISTVERBS accountname option

where option is the LPTR or line printer. If an option is not specified the file listing is output to the terminal. Note that in this command line the LPTR option is **not** enclosed in parenthesis. If parentheses are used the error message “ILLEGAL FORMAT” is displayed and nothing is printed.

Entering the command line:

```
LISTVERBS LPTR[RETURN]
```

produces a listing similar to that depicted in figure 6.9. Figure 6.9 is a partial listing of verbs contained in the SYSPROG account. If an accountname is not specified, as is the case in our example, the command outputs a list of verbs belonging to the currently logged account.

Session 12: The LOGON and OFF Commands

Once a PICK user account has been established using the CREATE—ACCOUNT command, any files or programs belonging to that account can be only accessed by “logging on” to that account. If there are no user accounts present, these commands have no meaning because there are no accounts to logon to or off from. LOGON and OFF are complementary commands. As such they will be discussed together.

The LOGON/OFF processors perform the following functions:

- Identification of users and their passwords
- Termination of a data entry session
- Accumulation of system usage statistics for billing and/or monitoring employee performance
- Association of each user with their privilege and security code

The LOGON Command

Logging onto a PICK system is accomplished by following these steps:

1. At the LOGON prompt, enter an existing account name and press [RETURN].
2. If a password is associated with this account the

PASSWORD :

message is displayed. The PASSWORD prompt may be bypassed by entering the password after the account name in the following format:

LOGON accountname,password[RETURN]

Note that a comma separates the accountname from the password and is an important part of the command syntax.

3. If no account or password is found matching the description entered the messages:

USER-ID?
LOGON :

are displayed.

4. When a valid account and password have been entered the system will display:

```
<<< THE PICK SYSTEM>>>  
<<< time   release number   date >>>  
  
>
```

When this sign on message displays logon is achieved. The appearance of the TCL prompt character indicates that a PICK command line may now be entered.

Note: In PICK version 1.3 logging onto the SYSPROG account displays a message pertaining to the availability of documentation addenda before returning you to TCL. The documentation addenda provide information about new features implemented on the current version of PICK.

The OFF Command

Logging off the system is accomplished by entering:

```
OFF[RETURN]
```

at TCL. The message:

```
<CONNECT TIME = n MINS,; CHARGE UNITS = m,LPTR PAGES=x  
  <<LOGGED OFF AT time ON date>>
```

is displayed. *n* is the number of minutes CONNected to the system, *m* is charge units of CPU time. Each charge unit is equal to one tenth of a CPU second. *x* is the number of pages generated by the system printer, *time* is the current time, and *date* is the current date.

Note: The OFF command can also be used to exit the DEBUG mode. When the DEBUG prompt (!) is displayed, you can enter:

```
OFF[RETURN]
```

to exit the DEBUG mode and logoff the system. The system will then display the LOGON message and wait for the next user to log onto the system.

The LOGTO Command

LOGTO is a command which facilitates logging directly from one account to another. The format of this command is:

```
LOGTO accountname,password[RETURN]
```

where *accountname* is the name of the new account and *password* is the potential password associated with the account. When this com-

mand is executed successfully, the system displays messages similar to those for LOGON. Note that a comma must separate the account-name from the password.

Session 13: The MSG Command

This command allows accounts to send messages to each other through the system. The format of this command is:

MSG account name message text

Account name is the user to which the message is sent. Message text is the message itself and must be separated from the account name by one character. The message cannot be edited in any way and is limited in length to approximately fifty characters. There are no options associated with this command. After the message is sent, the sender is returned to TCL.

Assume you are logged onto SYSPROG and enter:

```
MSG WALT THIS IS A TEST OF THE PICK MSG  
COMMAND[RETURN]
```

If WALT were logged onto the system, the message text would immediately be displayed in the middle of the screen WALT was working on. Any programs executed by WALT would be unaffected. If this message were sent and WALT was not logged onto the system the message:

```
USER IS NOT LOGGED ON
```

would be displayed on the sender's terminal.

The MSG command also permits users such as SYSPROG, who have a privilege level of two, to send messages to all users by substituting the asterisk (*) for account name. When the * is used the message sent is displayed on the sending as well as the receiving terminal.

The MSG command allows you to send messages to a specific line number as well as to a specific user by preceding the line number

with an exclamation mark (!). Using the command in this manner requires the format:

MSG !line number message text

where line number is a number 0, 1, or 2 depending upon which line you wish a message sent. In addition, it is possible to send messages to all users substituting the * for line number whether the users are logged on or not.

In order for a user not currently logged onto the system to receive a message the terminal must be powered on with the LOGON message displayed. The system makes no provisions for storing messages. Assuming a terminal on line 01 is active with the LOGON prompt displayed, consider the following command line entered on line 0:

```
MSG !01 THIS IS A TEST OF THE PICK MSG COM-  
MAND[RETURN]
```

The receiver's terminal would display:

```
LOGON  
current time date From: accountname  
THIS IS A TEST OF THE PICK MSG COMMAND
```

In order to redisplay the LOGON prompt [RETURN] must be pressed.

Session 14: The POVf Command

A virtual memory operating system, such as PICK, uses the disk drive as extension of random access memory. PICK divides the disk into a series of 512 byte or 1024 (character) frames. It then assigns each frame a "Frame-ID" or "FID". The frames are numbered in sequence 1, 2, 3....until the disk space allocated to the PICK disk partition is reached. A determinate number of frames is reserved for the PICK operating system files while the remaining frames are used for programs and data.

On a newly installed or restored system, all PICK system files are **contiguous**, or stored one next to the other. The remaining unused disk space, called the **overflow pool**, is also contiguous. As the system is used new files are added. Existing files expand to accommodate new data and contract when data is deleted. Disk space is automatically taken from and returned to the overflow pool. Over time, the disk space becomes fragmented and slows down system operation.

The POVf command is used to display the contiguous disk space. The format of the command is:

POVF (option)

where option is the line printer when specified. Consider the command line:

```
POVF[RETURN]
```

A newly installed or restored system, will display:

```
8757- 10067 : 1311
TOTAL NUMBER OF CONTIGUOUS FRAMES : 1311
>
```

where 8757 is the first frame of a contiguous disk area, 10067 is the last frame of the area, and 1311 is the number of contiguous frames in the PICK partition. The frame numbers you obtain will likely be different depending upon the size of your PICK partition and the number of programs and data files on your system.

As the system is used, the total number of contiguous frames is likely to decrease. When this occurs, or when the system seems to run more slowly than normal, it is time to execute a FILE-RESTORE procedure. A FILE-RESTORE will return the fragmented disk space to the overflow pool and reestablish maximum contiguous space. You may want to incorporate this procedure into normal system maintenance. *See session 17.*

Session 15: The POWER-OFF Command

The POWER-OFF command is provided with PICK PC/XT beginning with version 1.3. Its purpose is to provide an orderly system shutdown and ensure that all frames in memory are written to disk before power-off. Although PICK automatically monitors the virtual memory system, the POWER-OFF command guarantees that any disk frames in memory will be written to disk.

Because POWER-OFF will unconditionally terminate operation of all users, we recommend that you use the MSG command to advise users of imminent system shutdown **before** executing POWER-OFF. This command can only be executed from line 0. The format of the command is:

POWER-OFF[RETURN]

Once POWER-OFF is executed the system must be rebooted by physically powering off/on the computer.

Session 16: The REBOOT Command

The purpose of the this command is to reload the PICK operating system without powering off the system or using the key sequence CTRL-ALT-DEL. In order to use the command you must be logged onto the SYSPROG account on line 0. The format of the command is:

REBOOT[RETURN]

When this command is executed, the system will boot without the memory check executed by the computer during power on.

Session 17: Restoring the System

The PICK PC/XT system software is supplied on five floppy diskettes. Three of these disks contain the system files. The remaining disks contain a minimal set of working data files. As a means of providing security for PICK Systems Inc., it is not possible to make back up copies of the PICK system files (diskettes one through three).

Therefore, when system restoration becomes necessary, the system must be booted from the floppy disk labelled "PICK PC System #1".

There are typically three instances when a restore may be required:

- The software does not verify during the boot process as indicated by the error message "System does not verify."
- The disk space has become excessively fragmented.
- A new release of the PICK software has become available.

Restore Procedures

To restore a PICK system follow these steps:

1. Insert the floppy disk labelled "PICK PC System #1" in the drive and reboot the system by entering CTRL-ALT-DEL. The system will display:

```
Floppy Restore, PICK Systems IBM PC XT, 1985
Copyright (C) 1985 Pick Systems, as an unpublished work.
All rights reserved. This work is the property of, and
embodies trade secrets and confidential information
proprietary to Pick Systems, and may not be reproduced,
copied, used, disclosed, transferred, adapted or modi-
fied without express written approval of Pick Systems.
```

```
Options (A,F,K) =
```

Caution: do not select **any** option until you have read the sections below which discuss each option. You are not given a chance to change your mind or are you informed of the consequences of your selection before it is executed.

Option A

Selection of this option will restore the files on PICK Systems disks one through three. These three diskettes contain the PICK operating system programs. When option A is selected the following events occur:

- The contents of disk #1 will be read and transferred to the hard disk and you will be prompted to insert the second disk.
- The contents of disk #2 will be read and transferred to the hard disk and you will be prompted to insert the third disk.
- The contents of disk #3 will be read and transferred to the hard disk.
- When the transfer of files on disk three is complete, the system is automatically rebooted from the hard disk. The restore of the system files is complete. The data files and user accounts have not been altered.

During this procedure, the only message displayed on the screen is:

```
load PICK PC system #n then type 'C' to con-
tinue.
```

where # n is the number of the next diskette to load. The transfer of the files contained on the three system diskettes to the hard disk requires about five minutes.

Option F

Before selecting this option, we suggest you execute a FILE-SAVE to save all existing data files and user accounts. See session 9 for details on this procedure. Use of the F option has serious implications. This option will:

- Erase all files in the current PICK partition.
- Reinitialize the PICK partition.
- Restore the contents of PICK Systems diskettes one through three.
- Restore the contents of the original PICK data files.

or

- Restore the contents of your data files obtained from a FILE-SAVE procedure. If your data files are being restored, you will be prompted to insert each diskette from your FILE-SAVE, in sequence, until all of your disks have been restored. During the restoration of data files, the file names will be displayed on your screen.
- After the contents of the last data diskette have been transferred, the system will automatically reboot from the hard disk.

At this point, a complete file restore has been accomplished and maximum contiguous disk space is available. This can be verified by executing a POVf command.

Note: If you restore the contents of the data files supplied with the PICK system rather than your FILE-SAVE diskettes, your current accounting structure and data files will be lost.

Option K

The sole purpose of option K is to delete the PICK partition. **Never** select this option unless you have a current set of FILE-SAVE diskettes from which you can execute a FILE-RESTORE. Executing this option has the same effect as the DELETE PICK PARTITION option in the FDISK menu.

Session 18: The RUNOFF Command

This command, when used in conjunction with the EDIT command discussed in chapter 7, provides PICK users with limited word processing capability. The RUNOFF command permits data entered via the EDIT command to be formatted on a printed page according to user specifications. This session will be devoted to a brief description of RUNOFF.

The RUNOFF command is covered thoroughly in the PICK documentation and, although it is powerful in its formatting capabilities,

it is not designed for novice users. Some of the text formatting features included in RUNOFF are:

- Headers
- Footers
- Text justification
- Titling
- Variable line spacing
- Paging control
- Tab stops
- Automatic page numbering
- Underlining and boldface type

The RUNOFF command is used for the formatting of a data file which already exists and was created using the EDITOR.

Embedded Commands

Unlike other PICK commands, RUNOFF consists of a number of **embedded commands** which are part of the file operated upon by the RUNOFF processor. An embedded command is a command that is typed as part of the actual text while the file is being edited. The RUNOFF commands must be preceded by a period which allows PICK to interpret the line as a formatting command. Each command line may contain multiple commands each starting with a period. The format of the RUNOFF command is:

RUNOFF dictname file-name item-list (options)

Consider the following RUNOFF command lines embedded within a text file called BOX previously created using the EDITOR:

```
001 .BP
002 .BOX 5,65,CENTER
003 This is a test of the BOX embedded com-
mand.
004 .BOX
```

Entering the command line:

```
RUNOFF MD BOX[RETURN]
```

will display:

```
!This is a test of the BOX embedded command!
```

>

In the text file there are four embedded commands; the BEGIN PAGE (BP) command (001), the BOX command with its margin parameters, the CENTER command (002) , and a second .BOX command (004). Line 002 is the text to be enclosed within the box. Note that each RUNOFF command is preceded by a dot (.).

Other RUNOFF Features

Multiple short documents can be combined to create a single large document. “Boilerplating” (piecing together a single document with information from other document files) is also possible. Special features such as boldfacing and underlining, are supported by RUNOFF but may not be supported by the printer being used. Check the documentation provided by your printer vendor to see which features are supported and how they are implemented. If extensive word processing is to be performed, using one of the PICK word processing packages is recommended. They are easier to learn, provide task-oriented tutorials, and may have on screen help messages available. If you would like to learn more about RUNOFF, refer to the PICK documentation.

Session 19: The SET-TIME and TIME Commands

The SET-TIME and TIME commands are used to enter a new system time and display the current time. These commands are similar to the TIME command in MS-DOS. In order to use the SET-TIME command you must be logged onto SYSPROG. The format of the SET-TIME command is:

SET-TIME hh:mm:ss[RETURN]

where hh is the current hour, mm is the current minute, and ss is the current second. The seconds may be omitted but the colon (:) must be used as a delimiter between each parameter entered otherwise the error message “[298] FORMAT ERROR IN SPECIFICATIONS.” is displayed. Leading zeroes are not required. PICK indicates time in military or 24 hour format.

For example to set the current system time to 7:05 pm, type:

```
SET-TIME 19:05[RETURN]
```

The system displays:

```
19:05:00 current date
```

Note that entering 19:5[RETURN] would also have set the time to 7:05 pm.

The TIME command can be executed from any account. Its sole function is to display the current system time and date. The format of this command is:

```
TIME[RETURN]
```

To display the current time, type:

```
TIME[RETURN]
```

The system will display:

```
current time current date
```

```
>
```

Session 20: The SET-DATE Command

This command functions similarly to the SET-TIME command described in session 19 and is similar to the DATE command in MS-DOS. The function of the SET-DATE command is to change the current system date. In order to use this command you must be logged onto SYSPROG. The format of this command is:

```
SET-DATE mm-dd-yy[RETURN]
```

where mm indicates the month, dd the day, and yy the year. Leading zeroes are not required. To set the current date to January 1st, 1986 type:

```
SET-DATE 1-1-86[RETURN]
```

The system will display:

```
current time 01 JAN 1986
```

```
>
```

The current date can be displayed at any time via the TIME command.

Session 21: The STARTPTR and Related Commands

Either parallel or serial printers can be used in a PICK system. In order to eliminate a time bottleneck at the printer, on a PC/XT system, a spooling routine is provided to permit the sharing of a single printer among all users. This spooling routine writes files specified for printing in a temporary file on the disk and holds them in a queue. A queue is simply a temporary holding area for a series of files specified for printing. Under the control of PICK, files are printed in a first written, first printed method.

There are three verbs supplied with a PC/XT system which control the use of a printer. The function of these verbs is to start, stop, and delete printers from the system. Our discussion will be confined to the STARTPTR verb which activates the printer and spooler. Under normal circumstances, the spooler and parallel printer are automatically activated when the system is booted. Some application software may allow you to reconfigure the system for a second printer or use of a serial printer instead of a parallel printer.

In order to maximize the number of users, the PICK PC/XT system uses the STARTPTR command to activate a parallel printer. This frees two serial ports for attaching terminals. The simplest form of the STARTPTR command is:

```
STARTPTR n,f,p,Tm, (S
```


where *n* is the printer number (must be a number 0 through 3), *f* is the output queue (usually 0), *p* is the number of pages to skip at the end of each file printed, *T* is the printer type: *P* for parallel, *S* for serial, *m* is the line number of a serial printer (on a PC/XT this must be a number 0 through 3), *S* specifies that the page-eject command issued prior to printing a file is suppressed.

Note: The *S* option will only work with serial printers.

In this session we assume that the proper cabling for your printer has been performed. If you are using a parallel printer all that is required is a Centronics parallel cable. If you are using a serial printer, please refer to the PICK documentation supplement, pages nine and ten, and your serial printer documentation for the pin configurations required for the cable. Since parallel printer set-up is a standard feature of PICK on the XT, the remainder of this session will be devoted to installation of a serial printer.

Serial Printer Startup

During the boot process a PROC called COLDSTART is automatically executed. Upon examining figure 6.10 you will note that line 013 issues a STARTPTR command for a parallel printer.

Suppose we want to execute a STARTPTR command and activate a serial printer. First, we must know which form of the command to use. The command line:

```
STARTPTR 0,0,1,S1 (S[RETURN]
```

will activate printer # 0, print jobs in output queue # 0, eject one page after each job, and initialize a serial printer attached to line 1. The *S* option indicates that a new page will not be ejected prior to the printing of each job.

Note: a new STARTPTR command cannot be issued while another printer or terminal is already active on that line.

```
COLDSTART
001 PQ
002 HTERM-TYPE
003 P
004 HSET-SYM PSYM
005 P
006 HSET-TIME 17:00:00
007 P
008 HSET-DATE 11-01-85
009 A1,11
010 A
011 P
012 20 C
013 HSTARTPTR 0,0,1,P0
014 P
015 D
016 DVerifying system modes, Please wait...
017 HVERIFY-SYSTEM
018 P
019 HOFF
020 P
021 (SYSPROG-PL USER-COLDSTART)
```

Figure 6.10. COLDSTART PROC

If you know that your system will be using a serial printer it is best to execute the STARTPTR command from within the COLDSTART PROC. In chapter 7, you will learn how to modify line 013 of this PROC using the EDITOR. If you would like to begin using a serial printer now follow these steps:

1. Enter the command line:

```
SP-KILL D0[RETURN]
```

This will delete printer 0 which is a parallel printer. The message:

```
PARALLEL PRINTER # 0 HAS BEEN DELETED.
```

is displayed.

2. Enter the command line:

```
STARTPTR 0,0+1,S1 (SIRETURN]
```

The message:

```
The printer control block has been ini-
tialized.
>
```

is displayed. To verify that the startup of a serial printer was successful you can type:

```
SP-STATUS[SIRETURN]
```

The system will display:

```
PRINTER # 0 IS SERIAL, INACTIVE, AND ON
LINE.
THE PRINTER IS RUNNING ON LINE 1.
ASSIGNED OUTPUT QUEUES: 0
THE NUMBER OF INTER-JOB PAGES TO EJECT IS 1.
The SPOOLER is in an unambiguous state.
>
```

Printing Control

Once the printer control block has been initialized, printing documents should be easy. If you were unable to print some of the examples at the beginning of this chapter because you are using a serial rather than a parallel printer, go back and try them now. Since the printer is under the control of the operating system, rather than the application software, it should rarely be necessary to know any commands other than those discussed in this session. Output generated by application software will automatically be output to the specified printer. After the file has been printed, it will automatically be erased from the disk.

Some application software allows you to configure the system printer from a menu. This is a convenient and desirable feature. If you would like to learn about the other spooler commands consult the chapter in the PICK documentation on peripherals.

Note: If for some reason the printer was physically disconnected from the computer at the time a file was specified for printing, the file may not be printed immediately upon reconnection of the printer. The easiest way to correct this problem is to reboot the system. The COLD-START PROC will initialize the printer control block and restart the spooler. If the file does not begin to print, try reissuing the the command originally used to print the file.

Session 22: The TERM, TERM-TYPE, SET-LPTR Commands

The purpose of this group of commands is to establish the terminal or printer characteristics within the system. Once these characteristics are set, it should not be necessary to change them unless peripherals within the system are changed. One of the unique aspects of the PC/XT is the way video is displayed on line 0. On lines 1 and 2 video is displayed serially via an RS232 serial cable. Each character displayed on the screen is transmitted down the connecting cable one character behind the other. As each character is received by the terminal, it becomes the responsibility of the terminal to construct the screen display from the incoming data. Should the terminal be powered off, the screen display will be lost. When the terminal is powered on, the display will remain blank until more data is transmitted from the computer.

In contrast, the data transmitted from the computer down line 0 takes advantage of the **memory mapped video monitor** feature of the PC/XT. The memory mapped video monitor acts as a “window” into the section of the computer’s memory which “paints” the screen with data. The computer, rather than the terminal, is responsible for maintaining the screen display. This provides two advantages over the serial data transmission method:

- Screen updates are faster
- The monitor can be powered off/on without losing displayed data

PICK provides support on line 0 for both a monochrome and color monitor. A color monitor can display data in various colors, but

as of this writing there is no support for graphics. For more information on color see the session on the **COLOR** command.

The TERM/TERM-TYPE Commands

The **TERM** command is used to display or set the parameters for the data terminal and line printer. The **TERM-TYPE** command sets the default parameters for the terminal and printer. The importance of these commands lies in their ability to configure the PICK system easily for use with a variety of peripherals. For example, if an incorrect terminal type is used, on screen data will be improperly displayed, and the editing keys may not function correctly.

There are a limited number of terminals which can be used by PICK. Figure 6.11 depicts those terminals which are currently supported. If you plan to use a terminal which is not on this list, the terminal must be capable of emulating one of those listed. Some of the terminals on this list may not be familiar to you. Not to worry...these terminals were chosen by PICK as the best in terms of quality of display, ease of use, and reliability. More terminals will likely be supported in a later version of PICK.

```
A  ADDS 580
B  AMPEX 210
C  C.ITOH VT52
D  DATAMEDIA
I  PC/XT memory mapped mono/color (line 0 only)
L  LEAR SIEGLER
M  AMPEX 80
P  PERTEC 701
R  ADDS REGENT
V  ADDS VIEWPOINT
W  WYSE 50
```

Figure 6.11. PICK version 1.3 term types

Using the TERM and TERM-TYPE Commands

In this section we will discuss using the **TERM** command to display and change the terminal/printer default values. To use the **TERM**

command, logon to the SYSPROG account and at TCL enter:

```
TERM[RETURN]
```

The system will display a screen similar to Figure 6.12.

	TERMINAL	PRINTER
PAGE WIDTH:	79	80
PAGE DEPTH:	24	60
LINE SKIP:	0	
LF DELAY:	0	
FF DELAY:	2	
BACKSPACE:	8	
TERM TYPE:	I	

Figure 6.12. TERM command used to display default values on line 0.

This figure depicts the current specifications for the terminal and printer on line 0. The displayed parameters are the default values set automatically every time a logon to SYSPROG is executed. The default values are stored in the TERMTYPE item in the SYSPROG-BP file. To view the default values for lines 0, 1, and 2 on a PC/XT system, type:

```
CT SYSPROG-BP TERMTYPE[RETURN]
```

The system will display:

```
TERMTYPE
001 79,24,0,0,2,8,80,60,I

002 79,23,0,0,2,8,80,60,R
003 79,23,0,0,2,8,80,64,R

>
```

001 holds the default values for line 0, the memory mapped video monitor. 002 and 003 hold the default values for terminals attached to lines 1 and 2. The format of the TERM command to change the default values is:

```
TERM a,b,c,d,e,f,g,h,t
```

where a is PAGE WIDTH (the number of characters per line).

b is PAGE DEPTH (the number of lines per screen page).

c is LINE SKIP (the number of blank lines per screen page).

d is LF DELAY (the number of delay characters required by the terminal to keep up with the computer data transmission rate).

e is FF DELAY (the number of delay characters following each top-of-form).

f is BACKSPACE (the decimal number equivalent to the backspace character for the terminal).

g is the printer PAGE WIDTH.

h is the printer PAGE DEPTH.

t is the terminal type chosen from figure 6.11.

Use figure 6.13 to clarify the function of each TERM parameter.

	TERMINAL	PRINTER
(a) PAGE WIDTH:	79	80 (g)
(b) PAGE DEPTH:	24	60 (h)
(c) LINE SKIP:	0	
(d) LF DELAY:	0	
(e) FF DELAY:	2	
(f) BACKSPACE:	8	
(t) TERM TYPE:	I	

Figure 6.13. TERM command labels

Note: When using this command it is not necessary to specify every parameter. However, in order for a parameter to remain unchanged, a comma must be substituted for the parameter. An exception is the t parameter. If this is the only parameter to be changed, no commas need be entered. Entering TERM[RETURN] with no parameter list produces a display of the current TERM parameters. Consider the following example. At TCL enter:

```
TERM , , , , , 56 , [RETURN]
```

Upon entering TERM[RETURN] the screen will produce a display similar to that depicted in figure 6.14.

	TERMINAL	PRINTER
PAGE WIDTH:	79	80
PAGE DEPTH:	24	56
LINE SKIP:	0	
LF DELAY:	0	
FF DELAY:	2	
BACKSPACE:	8	
TERM TYPE:	I	

Figure 6.14. TERM command with PAGE DEPTH changed for printer

TERM-TYPE

In the previous example the default value for PAGE DEPTH was changed. The TERM-TYPE command can be used to restore the default settings on any account and line number for the terminal and printer. Assume you logged onto the SYSPROG account from line 1 and typed:

```
LOGTO SYSPROG[RETURN]  
TERM[RETURN]
```

The system will display:

	TERMINAL	PRINTER
PAGE WIDTH:	79	80
PAGE DEPTH:	24	60
LINE SKIP:	0	
LF DELAY:	0	
FF DELAY:	2	
BACKSPACE:	8	
TERM TYPE:	I	

Note that these values are the same as those in figure 6.12. Now if you enter:

```
TERM-TYPE[RETURN]
```


The system will display:

	TERMINAL	PRINTER
PAGE WIDTH:	79	80
PAGE DEPTH:	23	64
LINE SKIP:	0	
LF DELAY:	0	
FF DELAY:	2	
BACKSPACE:	8	
TERM TYPE:	R	

These are the default settings for a terminal attached to line 1. If you would like to change the default values, this can be accomplished by executing the following command line:

```
ED SYSPROG-BP TERMTYPE
```

and using the EDITOR to modify lines 001 through 003 of the TERMTYPE item.

SET-LPTR

This command automates the setting of the printer characteristics. The command is actually a PROC which prompts you for input then executes the TERM command using your entries to set the printer characteristics. The format of the command is:

```
SET-LPTR[RETURN]
```

To use the SET-LPTR command enter:

```
SET-LPTR[RETURN]
```

The system will display:

```
PRINTER PAGE WIDTH (# COLUMNS)?79
PRINTER PAGE DEPTH (#LINES)?24

      TERMINAL      PRINTER
PAGE WIDTH:         79      your entry
PAGE DEPTH:         24      your entry
LINE SKIP:          0
LF DELAY:           0
FF DELAY:           2
BACKSPACE:          8
TERM TYPE:          I
```

When prompted to enter new values you are not required to make entries. If no entries are made a page width of 140 and a page depth of 60 will automatically be entered as default values.

Note: The values entered using SET-LPTR are not saved. When an OFF command is executed the values for TERM revert to the default settings contained in the TERMTYPE item.

Session 23: The WHO Command

This command allows you to display the account name and line number of user accounts. The format of this command is:

WHO options

where options may be:

- a single line number
- a range of line numbers
- an asterisk (*)
- a specific account name such as SYSPROG

Using the WHO Command

Since the PC/XT has only three potential users, the primary use of this command is to inform the system manager who is currently using the system. Any lines not currently logged on will display as “UNKNOWN”. Entering an option is not required but is useful when

information about a specific user or line number is required. If you would like a reminder of which account and line you are currently logged to, at TCL you can enter:

```
WHO[RETURN]
```

Let's try some examples using options. Logon to SYSPROG and at TCL enter:

```
WHO 0 [RETURN]
```

The screen will display:

```
0 SYSPROG
```

```
>
```

```
WHO 1-2[RETURN]
```

The screen will display:

```
1 UNKNOWN
```

```
2 UNKNOWN
```

```
>
```

Entering:

```
WHO *[RETURN]
```

displays:

```
0 SYSPROG
```

```
1 UNKNOWN
```

```
2 UNKNOWN
```

```
3 UNKNOWN
```

```
4 UNKNOWN
```

Entering:

```
WHO 'SYSPROG'[RETURN]
```

displays:

```
0 SYSPROG
```

```
>
```

This command line produces a display of all lines logged onto the SYSPROG account. If another line were logged onto SYSPROG that line number and account name would be displayed.

Summary

In this chapter we have discussed and demonstrated the most frequently used PICK commands. The commands were discussed alphabetically rather than by their function. Some commands are actually PROC'S which help to automate certain tasks. We will learn more about PROC'S in chapters eight and nine. If you would like to study the commands not covered here, consult the PICK user reference manual.

7

The Editor

The Editor provided with the PICK PC/XT system is a line Editor. That is, it generally accesses and works on only one line at a time. That line is called the current line. The Editor is limited in its scope of capabilities but is easy to learn. Also, this simple Editor is adequate for the tasks of maintaining databases, developing programs, and generating short memos and reports.

The Editor is not designed for word processing applications. Moreover, it seems doubtful that the PICK system would be selected for text processing applications. The PICK system is designed to support information management and retrieval.

We have divided this chapter into three sections. In the first section we discuss how to enter the Editor and how to edit the dictionary and data items. We include examples of how to create both data and dictionary items in this section.

The second section is intended as a tutorial in the use of the Editor. This section provides step-by-step instructions on the use of the Editor's basic features. We encourage the reader to execute the examples in the tutorial at his own terminal.

The final section is a reference guide to the Editor. This section should be used once the reader has a grasp on the fundamentals on using the Editor. The reader should take time to become familiar with the function of each command. Not all of the Editor commands and command options will be discussed in the tutorial section.

Finally, we assume that the reader has a basic knowledge of the PICK file system. Chapter five offers discussion of this topic. Review that chapter before trying to use the Editor. Also, page 90 of the PICK User Reference Manual contains a condensed summary of the Editor commands. The summary is presented in table form.

Entering the Editor

The Editor is accessed from TCL by the Edit command. The Edit command must specify a file to be edited. This file must already exist within the system. The first time a file is edited, it will be empty. The Editor can manipulate items in both the dictionary and data files. We will assume that the file "TEXT-PRACTICE" is a new file. To create the file TEXT-PRACTICE log to SYSPROG and enter: `CREATE-FILE TEXT-PRACTICE 29 I[RETURN]`.

Editing Data Items

The command to edit a data item in the TEXT-PRACTICE file is the following:

```
ED TEXT-PRACTICE [RETURN]
```

The system will respond with the following prompt:

```
ITEM ID:
```

Reply to the prompt by typing the following:

```
COMMENT [RETURN]
```

The system will respond with the following lines:

```
NEW ITEM
TOP
.
```

Note the period. The period is the Editor command prompt. The Editor command prompt signifies that the program is ready for an Editor command. A full listing of the Editor commands is presented at the end of this chapter in the form of a reference guide.

The input command allows additions to be made to the item being edited. Type the following:

```
I [RETURN]
```

The system will respond with the following prompt:

```
001 +
```

This number prompt means that the system is ready to receive input into the item. All keystrokes entered at the keyboard will appear on the screen. A carriage return is used to access another line. A *null line* is used to signal the end of input. A null line produced by pressing the return key at the beginning of a blank line. For example, type the following:

```
THIS FILE IS A PRACTICE FILE. [RETURN]
I AM USING IT TO LEARN THE EDITOR. [RETURN]
[RETURN]
```

Note that the system displayed a number each time a new line was accessed. The final display should appear as follows:

```
001+THIS FILE IS A PRACTICE FILE.
002+I AM USING IT TO LEARN THE EDITOR.
003+
TOP
.
```

The appearance of TOP means that the Editor has set the current line to the first line in the item. The current line can be displayed by

pressing the return key. Pressing the return key will cause the system to display the following:

```
001 THIS FILE IS A PRACTICE FILE ,
,
```

Note that the plus sign no longer appears after the line number. The plus sign is used to indicate that the Editor is ready to receive input.

Press the return key again. The following lines will be displayed to the system:

```
002 I AM USING IT TO LEARN THE EDITOR ,
EOI 002
,
```

The appearance of EOI is used to represent the end of the item. File the item by typing the following:

```
FI <RETURN>
```

The system will display the following:

```
`COMMENT' FILED ,

>
```

The data item COMMENT has been filed in the file TEXT-PRACTICE. The TCL prompt is displayed. Check the existence of the comment item by typing the following command:

```
LIST TEXT-PRACTICE <RETURN>
```

The system will display the following:

```
Page 1                      15:41:37  12 OCT 1984
TEXT-PRACTICE
COMMENT
END OF LIST
```

Editing Dictionary Items

As we discussed earlier, dictionary items have special formats. To illustrate, we will create a dictionary item. The dictionary item will be a synonym definition for the TEXT-PRACTICE file.

We will create the synonym PRAC for the text practice file. To begin this task type the following:

```
ED DICT MD PRAC [RETURN]
```

The system will respond with the following display:

```
NEW ITEM
TOP
*
```

We will create the synonym definition. The synonym definition should appear as follows:

```
001 Q
002
003 TEXT-PRACTICE
004
005
006
007
008
009 L
010 80
```

The significance of such a definition is discussed in the chapter on the file system. Special measures must be taken in order to obtain blank lines within the file. Pressing the return key at the beginning of a blank line causes the input mode to be closed. So, filler material must be placed on each blank line. This filler material can later be changed to a blank line. This point is clarified by the example below:

Start inputting data into the dictionary item by typing the following:

```
I <RETURN>
```

The system will respond with the following prompt:

001

Now type the following:

```
Q [RETURN]
FILLER [RETURN]
TEXT-PRACTICE [RETURN]
FILLER [RETURN]
FILLER [RETURN]
FILLER [RETURN]
FILLER [RETURN]
FILLER [RETURN]
L [RETURN]
80 [RETURN]
[RETURN]
```

Note that each line will be preceded by a number prompt. After the final carriage return, the system will display:

```
TOP
.
```

The occurrences of **FILLER** within the dictionary item must now be changed to blank lines. The universal replace command can be used to accomplish this task. Type the following:

```
RU10/FILLER//[RETURN]
```

The system will respond by displaying the following:

```
002
004
005
006
007
008
EDI 010
```

Now check the file. Type the following:

```
T [RETURN]
L10 [RETURN]
```

The system will respond by displaying the following:

```
001 Q
002 FILLER
003 TEXT-PRACTICE
003 FILLER
004 FILLER
005 FILLER
006 FILLER
007 FILLER
008 FILLER
009 L
010 B0
.
```

What happened? It appears as if the changes were not made. The Editor uses two buffers while editing an item. One buffer holds the current copy of the item. The other buffer holds the changes that have been made in the file. These changes cannot be seen until an F command is issued. Type the following:

```
F [RETURN]
```

The system will respond with the following display:

```
TOP
.
```

The F command always returns the current line to the top of the file. Now, display the file by typing the following:

```
L10 [RETURN]
```

The system will respond with the following display:

```
001 Q
002
003 TEXT-PRACTICE
004
005
006
007
008
009 L
010 80
E01 010
.
```

Exit from the Editor and save the new item by typing the following:

```
FI [RETURN]
```

The system will display the following:

```
`PRAC' FILED.
```

The TCL prompt will return to the screen.

Test the new synonym by typing:

```
CT PRAC COMMENT [RETURN]
```

The system will respond by displaying the following:

```
COMMENT
001 THIS FILE IS USED AS A PRACTICE FILE.
002 I AM USING IT TO LEARN HOW TO USE THE EDITOR.
```

The new synonym, PRAC, for the file TEXT-PRACTICE is now defined and working.

An important point to keep in mind when creating dictionary items is that they must follow a precise format in order to function correctly. If the format is followed closely, no problems will be experienced. The proper formats for different types of dictionary items are detailed in the chapter on the file system.

Tutorial on the Editor

In this section we will present examples on how to use the important features of the Editor. We will start with an item named MAIN-BODY in the file TEXT-PRACTICE. The starting contents of MAIN-BODY are listed below. The examples are meant to be executed sequentially. The file will be modified as the examples are executed.

We encourage the reader to create the original MAIN-BODY item and to execute the examples at his own terminal. To create the original contents of MAIN-BODY, type the following:

```
ED TEXT-PRACTICE MAIN-BODY [RETURN]
I [RETURN]
```

MAIN-BODY

```
This file contains the bulk of the text used in the examples.
The file was created with [RETURN]
some mistakes purposely left in it. Thes [RETURN]
mistooks in the file will be corrected as we go. [RETURN]
Although we are working with a text file, it [RETURN]
should be kept in mind that the PICK system [RETURN]
will most likely be used for data base management. [RETURN]
The items should then be small and easy to [RETURN]
handle because each individual item in the data [RETURN]
base is usually kept relatively simple. [RETURN]
F1 [RETURN]
```

Displaying the Contents of an Item

When the Editor is initially entered by a command in upper case letters such as the following:

```
ED TEXT-PRACTICE MAIN-BODY [RETURN]
```

The system will display the following:

```
TOP
.
```

The contents of the item being edited can be displayed using the LIST command. The LIST command always displays a group of lines starting with the current line. The length of the group can be specified by the user. For example, to display the first three lines of the item, type the following:

```
L3 [RETURN]
```

The system will respond by displaying the following:

```
001 MAIN-BODY
002 This file contains the bulk of the test used in the examples.
003 The file was created with
```

Note that any number of lines to be displayed can be entered. If the item contains fewer lines than was specified, the entire file from the current line to the end of the item will be displayed. For example, typing the following command:

```
L99 [RETURN]
```

will cause the remainder of the main-body item to be displayed. That is, lines four through eleven will appear on the screen. The EOI (End Of Item) message will also appear after line eleven. Note that the List command causes the current line to be set to the line after the last line in the display. So, after the L3 command, the current line was set to line four.

Displaying the Current Line Number

Now that we have displayed the entire file, what line is the current line? The number of the current line can be displayed by typing the following:

```
? [RETURN]
```

The system will respond by displaying the filename, item-id and line number of the current line. In this case, the display would show the following:

TEXT-PRACTICE MAIN-BODY L 011

Suppressing the Line Numbers

The line numbers on the display from the Editor can be suppressed. To accomplish this task type the following:

S [RETURN]

The system will respond by displaying the following:

SUPPRESS ON

.

Now issue an L99 command. Note that the line numbers are not included in the display. Issuing the S command again will allow the numbers to be displayed once again. Type the following:

S [RETURN]

The system will respond by displaying the following:

SUPPRESS OFF

.

The line numbers will once again be displayed in output from the Editor.

Changing the Number of the Current Line

The List command changes the number of the current line. However, it accomplishes the change in a rather indirect manner. The Editor has several commands that allow the current line to be changed directly to any line number.

To the Top

The current line can be changed to the first line in the item. To accomplish this task, type the following command:

T [RETURN]

The system will respond with the following display:

```
TOP
.
```

The current line is now set to the first line in the file.

To the Bottom

The current line can be changed to the last line in the item. To accomplish this task, type the following:

```
B [RETURN]
```

The system will respond by displaying the following:

```
EOI 011
.
```

Note that line 11 is the last line in the file. The current line is now that line.

Moving Up

The number of the current line can be changed so that it is a specific number of lines less than its present value. To make the current line number two less than its present value, type the following:

```
U2 [RETURN]
```

The system will respond by changing the current line from line 11 to line 9 and displaying the contents of line 9, shown below:

```
009 The items should then be small and easy to
.
```

Moving Down

The number of the current line can be changed so that it is a specific number of lines greater than its present value. To increment

the current line number by two, type the following:

```
N2 [RETURN]
```

The system will respond by changing the current line from line 9 to line 11 and displaying the contents of line 11:

```
011 base is usually kept relatively simple.  
EOI 011  
.
```

Note that the EOI message appears only when the new current line is the last line in the item.

To a Specific Line Number

The current line can be changed to any specific line number. To change the current line to line 5, type the following:

```
5 [RETURN]
```

The system will respond by making line 5 the current line and displaying the contents of line 5:

```
005 mistakes in the file will be corrected as we go.  
.
```

Correcting Errors

The Replace command is used to correct errors by substituting a correct string of text for the erroneous data. The REPLACE command can be used on entire lines, a part of the current line, universally, or parts of several lines.

Replacing Data in One Line

If some of the data in the item is erroneous, it must be replaced by the correct data. For example, in the current line (line 5), “mistooks” should be “mistakes”. To make the necessary replacement, type the following:

```
R/mistooks/mistakes/ [RETURN]
```

The system will respond by displaying the new contents of the current line as follows:

```
005 mistakes in the file will be corrected as we go.  
.
```

An F command should now be issued. The F command Combines the buffer containing a copy of the file and the buffer containing the editing changes into one buffer. Issuing an F command after each Replace command will help avoid confusion. To issue an F command type the following:

```
F[RETURN]
```

An F command always returns the current line to the top of the item.

If SEQN? is ever displayed, issue an F command before continuing with the editing process.

Universal Replace

The universal form of the Replace command is used to correct a mistake that appears repeatedly in several lines. For instance, in our example, each occurrence of “file” should read “item” to be strictly accurate. To change “file” to “item”, type the following command:

```
RU11/file/item/[RETURN]
```

The system will respond by displaying the following:

```
002 This item contains the bulk of the text used in the examples.  
003 The item was created with  
005 mistakes in the item will be corrected as we go.  
006 Although we are working with a text item, it  
EOI 011  
.
```

Note that each line that is changed is displayed. The display should be checked carefully for unexpected and unwanted changes. Unexpected and unwanted changes are usually caused by specifying a search string that is too general in nature. Enough characters should be spec-

ified in the Replace command's search string to uniquely describe the data entry of interest. Remember to issue an F command after the REPLACE command.

Finding a Specific Line

The Locate command is used to find the occurrence of a search string in a specific line. The Locate command will cause lines containing the search string to be displayed. For example, the following command:

```
L11/item/[RETURN]
```

would result in the following display:

```
002 This item contains the bulk of the text used in the examples.
003 The item was created with
005 mistakes in the item will be corrected as we go.
006 Although we are working with a text item, it
009 The items should then be small and easy to
010 handle because each individual item in the data.
EOI 011
,
```

Note that the current line is set to line 11, the last line referred to by the Locate command.

Inserting New Lines

New lines can be inserted into the item at any point. The contents of the new lines can be input from the keyboard or read from another item.

Inserting from the Keyboard

To input new lines from the keyboard, the Input command is used. The Input command causes the new line to be inserted after the current line. The Input command always uses the current line number for the number prompt when editing an existing item. This is in contrast to the number prompts that increase by one for each line that is added to a new item.

To insert text into the item after line 5, type the following:

```
5 [RETURN]
```

The system will respond by displaying the following:

```
005 mistakes in the item will be corrected as we go.  
.
```

Now type the Input command:

```
I [RETURN]
```

The system will respond with the following number prompt:

```
005+
```

The text to be inserted can now be typed. Pressing the return key will access a new line. Pressing the return key when the line is empty will cause the end of insertion . To add two lines to the item, type the following:

```
Two lines are being inserted. This is the first.[RETURN]  
And this is the second.[RETURN]  
[RETURN]
```

Note that the following number prompt:

```
005+
```

will appear at the beginning of each of these lines.

To view the result of the INSERT command, type the following commands:

```
F [RETURN]  
L99 [RETURN]
```

The entire item should be displayed. The display should appear as follows:

```
001 MAIN-BODY
002 This item contains the bulk of the test used in the examples.
003 The item was created with
004 some mistakes purposely left in it. These
005 mistakes in the item will be corrected as we go.
006 Two lines are being inserted. This is the first.
007 And this is the second.
008 Although we are working with a text item, it
009 should be kept in mind that the PICK system
010 will most likely be used for database management.
011 The items should then be small and easy to
012 handle because each individual item in the data
013 base is usually kept relatively simple.
EOI 013
```

Inserting from Another Item

The Merge command is used to read data from another item and insert it into the item currently being edited.

To insert the item COMMENT at the end of the MAIN-BODY item, type the following:

```
ME2/COMMENT/ [RETURN]
```

The system will display the following:

```
EOI 002
```

```
.
```

To view the result of the Merge command, type the following:

```
F [RETURN]
L99 [RETURN]
```

The system will display the contents of the MAIN-BODY item. The two lines from the COMMENT item will be inserted at the end of the MAIN-BODY item. There will now be fifteen lines in the MAIN-BODY item.

Deleting Lines

Lines can be removed from the item with the Delete command. Parameters in the Delete command allow lines to be deleted starting with a line that contains a specified search string or they may be deleted beginning with the line that is the current line.

Delete Starting from the Current Line

We will now delete the lines that we added to the middle of the MAIN-BODY item. To delete these lines, start by making the first line to be deleted the current line. Type the following:

```
G [RETURN]
```

The system will respond by displaying the following:

```
006 Two lines are being inserted. This is  
the first.  
*
```

To delete the two new lines, type the following:

```
DE 2 [RETURN]
```

The effect of the Delete command can be checked by typing the following:

```
F [RETURN]  
L99 [RETURN]
```

The system will respond by displaying the modified contents. The two lines that had been inserted earlier will be gone. There will now be 13 lines in the item.

Deleting after a Search

We will now remove the two lines from the COMMENT item that were inserted at the end of the MAIN-BODY item. We will use the Delete command from that operates after a search is conducted.

First, make the first line the current line. Type the following:

```
T [RETURN]
```

Now issue the Delete command. Type the following:

```
DE13/US/ [RETURN]
```

The system will respond by displaying the lines being deleted:

```
012 THIS FILE IS USED AS A PRACTICE FILE ,
013 I AM USING IT TO LEARN THE EDITOR ,
EOI 013
,
```

Note that case is significant in the search. The capital letters US occurred together only in the two lines that we wanted to delete. Also, the 13 in the command specified how many lines were to be searched.

Remember to incorporate the editing changes into the file. Type the following:

```
F [RETURN]
```

Updating the Disk File

The FI command is used to update the permanent copy of the item on the hard disk. The item can be either saved or deleted. Note that the F command does not cause the disk file to be changed. The F command only combines the contents of the two editing buffers. Recall that one of the editing buffers contains a copy of the item while the other buffer is used to store the editing changes. All of the File commands also perform the function of an F command.

File and Continue Editing

The FS command is used to save the changes made in the item to date. This command is useful during long editing sessions. The disk item is updated without exiting from the editor. To issue an FS command, type the following:

```
FS [RETURN]
```


The system will respond by displaying the following:

```
TOP
*
```

File and Exit

The FI command is used to save the updated item and exit to TCL. This command is useful at the end of a successful editing session. To issue an FI command, type the following:

```
FI [RETURN]
```

The system will respond by displaying the following:

```
'MAIN-BODY' FILED,
>
```

Delete Item and Exit

The FD command is used to delete the item being edited. Also, an exit to the TCL level is executed. The FD command destroys the item being edited. To execute an FD command, type the following:

```
FD [RETURN]
```

The system will respond by displaying the following

```
'item-id' DELETED,
>
```

Editor Reference Guide

In this section we present the proper syntax for each of the Editor's commands as well as detailed explanations concerning the functions of these commands. Examples are used within the text to clarify the explanation of these commands.

The following conventions will be used when representing the syntax of each command:

- **Upper Case** Capital letters are used to represent strings that must appear exactly as shown.
- **Lower Case** Lower case letters are used to represent parameters that must be supplied by the user.
- **Braces { }** Braces are used to enclose optional parameters. An optional parameter is not required but is available if the situation warrants its use.

Invoking the Editor

The ED or EDIT command is used to invoke the Editor. The syntax of the ED command is shown below:

```
ED[IT] [DICT] filename [items] [options] [RETURN]
```

- | | |
|-----------------|---|
| DICT | The inclusion of DICT is used to indicate that the dictionary portion of the file should be edited. If DICT is omitted, the data portion of the file will be edited. |
| filename | The filename indicates which file should be edited. A filename must be specified. |
| items | At least one item must be specified. If more than one item is specified, the next item in the list will be loaded into the editing buffer after the current item has been edited. |
| options | There are several options that alter the manner in which the Editor functions. Note that all options must be enclosed in parenthesis. Multiple options must be separated by commas. |

The following options are available for use with the ED command:

- A** This option is used to turn on assembly-code formatting. This option performs the same tasks as the AS Editor command.
- M** This option is used to turn on the macro expansion capabilities. This option performs the same tasks as the M Editor command.
- P** This option is used to obtain a printed copy of all of the system's responses while using the Editor. This option is useful to document changes made in a file during an editing session.
- S** This option is used to turn on the suppression of line numbers. This option performs the same tasks as the S Editor command.
- Z** This option is used to suppress the TOP and EOI messages that usually automatically displayed by the system.

For example, the following command:

```
ED DICT ACCOUNTS "GENERAL HARDWARE" DALTON [RETURN]
```

would cause the Editor to be invoked. The items GENERAL HARDWARE and DALTON in the file accounts are specified for editing. Note the use of double quotes around the entry GENERAL HARDWARE. An item name that contains an internal blank space within it must be set off by delimiters such as the double quotes. If the double quotes had not been used, the system would have interpreted the command as "Edit the items GENERAL, HARDWARE, and DALTON."

The GENERAL HARDWARE item is loaded into the editing buffer first. When the editing of this file is completed, the item DALTON will be loaded into the editing buffer instead of exiting to TCL.

The following Editor commands signal that editing is complete: FI, FD, and EX. Furthermore, the command EXK causes an exit to TCL regardless of how many more items remain in the list to be edited.

Editor Commands

We have arranged the Editor commands in alphabetical order. Each command is listed under a name that generally provides a clue to its function. Note that all Editor commands must be entered after the appearance of the Editor's period prompt. If a period prompt is not displayed, the Editor is not ready to receive a command. Generally, if the Editor is not displaying the period prompt, it is ready to accept data for input into the item. Finally, all parameters that specify a number must be positive integers, unless otherwise specified.

Again (A)

The Again command is used to repeat the search executed by the last Locate command. The syntax of the Again command is shown below.

```
A [RETURN]
```

Assembler Format (AS)

The Assembler Format command is used to turn the Editor's assembly formatting functions on and off. If the functions are currently off, the command will activate them and vice versa. The assembler formatting should only be used on assembler programs. The assembler Format command has the following form:

```
AS [RETURN]
```

When this command is executed, the system will respond by displaying either AS ON or AS OFF, whichever is appropriate.

Bottom (B)

The Bottom command is used to move the current line to the last line in the item. The B command has the following format:

```
B [RETURN]
```

Column Number List (C)

The Column Number List command causes the columns across the screen to be numbered. The display will appear as follows:

```
12345678901234567890.....012345
```

The display is useful when editing data that must conform to a fixed field format and when using the Runoff processor. The syntax of the Column Number List command appears below:

```
C [RETURN]
```

Note that the display will appear on the screen but will not be inserted into the item being edited.

Current Line (?)

The Current Line command is used to find the number of the current line. The format of the current line command follows:

```
? [RETURN]
```

The output from this command will list the filename, item-id, and the current line number in the following form:

filename item-id L line number

The line number will be a three digit number such as 024.

Delete (DE)

The Delete command is used to remove unwanted lines from the item. The Delete command has two forms. The simple Delete command has the following format:

```
DE (number) [RETURN]
```

number The number parameter must be positive integer. The number of lines to be deleted is specified by number. Deletion starts with the current line. If no number is specified, only the current line will be specified, and thus deleted.

For example, the following command:

```
DE 5 [RETURN]
```

would cause five lines to be deleted from the item. The deletion would include the current line as well as the following four lines. Note that if the end of the file occurs before the fourth line, only the lines from the current line to the last line will be deleted.

The complex form of the delete command is as follows:

```
DE {number} "search string" {starting Col-  
umn{-ending column}} [RETURN]
```

number	The value of number determines how many lines should be included in the search. If number is not specified, only the current line will be searched.
search string	The search string specifies the string of characters being sought. All lines in the range specified by the number that contain a match to the search string will be deleted. Note that the search string must be enclosed in double quotes.
Starting Column	The value of the starting column number specifies the column number where the search should begin. If the starting column is not specified, the search will be started at the first column.
ending column	The value of the ending column number specifies the column where the search should end. If the ending column is not specified, the search will end at the last column.

For example, the following command:

```
DE12"COM"1-3 [RETURN]
```

would cause the twelve lines starting with the current line to be searched. Any lines with **COM** appearing in the first three columns would be deleted from the item. Note that the system will display each line that is being deleted.

Exit (EX)

The Exit command is used to abandon the edited item. The permanent copy of the item on disk will retain its contents when the Editor was invoked or when the last FS command was executed. The Exit command causes the current copy of the editing buffer to be thrown away. An Exit command causes an exit to TCL or to the next file in the item list. The form of the Exit command is shown below:

```
EX{K} [RETURN]
```

The K parameter causes an exit to TCL even if an item list exists.

File Commands (F, FI, FS, and FD)

The editor uses two buffers in order to accomplish editing tasks. One of the buffers is used to store the current copy of the item. The other buffer is used to store editing changes in the current item. The editing changes must always be in sequence according to their line numbers. Therefore, a change in line 1, then line 7, and finally line 22 is legal. However, a change in line 1, then line 22, and finally line 7 is not permitted. The error message SEQN? will be displayed on the screen if an attempt is made to edit a line out of sequence.

To edit such a line, the two buffers must first be merged. After the merge, one buffer will contain the updated version of the item while the other buffer will be empty and ready to accept more editing changes. It is important to note that any editing changes made will not appear in the current copy of the item until after a merge has been executed.

The file commands are used to handle the Editor's two buffers. These commands are also used to control writing of the current copy of the item to the disk. The following list details the tasks accomplished by each of the file commands:

- **F** The F command causes the two editing buffers to be merged. The current line will always be returned to the top of the item by an F command. The F command does not cause the item being edited to be written to the disk.
- **FI** The FI command causes the two editing buffers to be merged. The item being edited is then written to the disk. All of the changes made during the editing session will be contained in the updated version of the item on the disk. After the item has been written to the disk, an exit from the editor to TCL is executed.
- **FS** The FS command causes the two editing buffers to be merged. The item being edited is then written to the disk. All of the changes made during the preceding portion of the editing session will be contained in the updated version of the item on the disk. After the item has been written to the disk, the current line will be set to the top of the item, editing continues.
- **FD** The FD command causes the item being edited to be removed from the disk. After the item has been removed, an exit to TCL will be executed.

Note that if an editing list has been specified, the above commands will act as described. But, instead of exiting from the Editor, the next item in the editing list will be loaded into the editing buffer. Also, if the Editor was called by a PROC, the exit will return control to that PROC instead of the TCL level. Finally, to abandon the copy of the item in the editing buffer, see the Exit command, which we described earlier.

Goto (G)

The Goto command is used to change the current line to a specific line number. The Goto command has the following form:

```
{G}number [RETURN]
```

where number represents the number of the line that is to become the current line.

For example, the following command:

```
G 25
```

would change the current line to line 25. The system will display the contents of the new current line.

Input (I)

The Input command is used to insert data into a new item or add data to an existing item. The form of the Input command is the same in either case. This form is shown below:

```
I [RETURN]
```

The system will then respond with a number prompt. The number prompt is a three digit number followed by a plus sign, such as 008+. Data can be entered after the number prompt. A new line is accessed by pressing the return key. Pressing the return key when no data is present on the line causes input to be terminated.

With new items, the number prompt will start with a value of 001+ and will be incremented by one for each new line. The data that is input in such a manner will comprise the current copy of the item.

With existing items, the line number of the current line will be used repeatedly as the number prompt. The data input in such a manner will be stored in the editing changes buffer. Before the new lines can be viewed, an F command must be issued. The Editor will re-number the lines in the item when an F command is executed.

List (L)

The List command is used to display the current contents of the item in the buffer. Note that the List command will not reflect changes stored in the editing changes buffer. The List command has the following form:

```
L {number} [RETURN]
```

where number represents the number of lines that are to be listed. The listing always starts at the current line.

For example, the following command:

```
L 22 [RETURN]
```

would cause 22 lines to be listed, starting with the current line. Note that if the item does not contain that many lines, the lines from the current line to the end of the file will be listed.

The List command can be used in this manner to display the entire contents of the file. The following sequence of commands will accomplish this task for all short files:

```
T [RETURN]  
L99 [RETURN]
```

Locate (L)

The Locate command is used to find the occurrence of a search string in an item. The form of the Locate command is shown below:

```
L {number} "search string" {starting col-  
umn{-ending column}} [RETURN]
```

number

The value of the number specifies how many lines should be included in the search. If the number is omitted, the next occurrence of the search string will be found. That line will be displayed and made the current line.

search string	The search string represents the character string being sought. Note the double quotes used to delimit the search string.
----------------------	---

Note that starting column and ending column take on the same significance as with the Delete command as we discussed on page 181.

For example, the following command:

```
L10 "COM" [RETURN]
```

would cause the 10 lines following the current line to be searched for the character string COM and then display them. Note that the current line would be changed to the last (tenth) line included in the command.

As a further example, the following command:

```
L "COM" [RETURN]
```

would cause the next line containing the character string COM to be displayed. The current line would be set to that line.

As a final note, recall that the Again command can be used to repeat the last Locate command.

Merge (ME)

The Merge command is used to read lines into the item currently in the editing buffer. The source of the lines read in can be another item in the same file, another item in a different file, or the item currently in the editing buffer. The form of the merge command to read from a different file is shown below:

```
ME{number} ({DICT}filename  source  item)  
{start line} [RETURN]
```

number	The value of the number represents how many lines should be merged into the current item from the source item. If no number is specified, only one line will be merged.
DICT	If DICT is included, the source item will be taken from the dictionary portion of the specified item. If DICT is excluded, the data portion will be used.
filename	The filename is used to specify the file from which the source item is to be taken. If the filename is excluded, the source item will be taken from the file that contains the item currently being edited.
source item	The source item is used to specify the item from which the lines to be merged will be taken.
start line	The value of start line is used to specify at which beginning line number in the source item the lines are to be taken. If start line is excluded, the lines will be taken beginning from the first line in the source item.

For example, the following command:

```
ME2(NEW PRODUCT)3 [RETURN]
```

would cause the two lines starting at line 3 of the item **PRODUCT** in the file **NEW** to be inserted after the current line in the item presently in the editing buffer. Remember that an **F** command must be issued before the new lines will appear in a listing of the item. Take special note of the use of parentheses. The parentheses are required only when the source item is in a different file than the receiving item. Also note that the source item must not be enclosed in double quotes in this example.

The form of the merge command to read from the same file is shown below:

```
ME {number}" "source item' '{start line} [RETURN]
```

where number, source item, and start line all have the same meaning as previously defined. Note the use of double quotes to set off the source item. Parentheses cannot be used in this form of the Merge command.

For example, the following command:

```
ME2"MAIN-BODY"3 [RETURN]
```

would cause the two lines beginning at the third line of the item MAIN-BODY in the same file as the current item to be placed after the current line.

As a final example, the following command:

```
ME3" "2 [RETURN]
```

would cause the three lines beginning at line 2 of the current line to be placed after the current line.

Next (N)

The Next command is used to increment the current line by a specified amount. The Next command has the following form:

```
N{number} [RETURN]
```

where number represents the number of lines by which the current line will be incremented. If no number is specified, the current line will be displayed but not changed. Pressing [RETURN] is equivalent to an N1 command. Note that the Next command causes the new current line to be displayed.

Replace (R)

The Replace command can be used to replace entire lines with new ones. Alternatively, the Replace command can be used to substitute a replacement string for a search string.

The form of the Replace command to replace entire lines is shown below:

```
R{number}
```

where the value of number specifies the number of lines to be replaced. If no number is specified, one line will be replaced. When this version of the Replace command is issued, the current line number will be displayed. The new data for that line can then be entered. When [RETURN] is pressed, the next line number will be displayed. This process continues until the number of lines specified has been changed or [RETURN] is pressed when a line is empty. The period prompt will then be displayed. If the second method is used to exit from the change, all lines after and including the line that was exited, will remain as they were before the command.

For example, the following command:

```
R5 [RETURN]
```

would cause the current line number to be displayed. If data was input on two lines and the editing process exited, the third, fourth and fifth line would remain unchanged.

The form of the Replace command to substitute a replacement string for a search string is shown below:

```
R{U}{number}"search string"replacement  
string"{start column{-end column}} [RETURN]
```

U

The U parameter specifies that all occurrences of the search string should be replaced with the replacement string. If the U parameter is not specified, only the first occurrence of the search string will be replaced.

number	The value of the number specifies how many lines will be scanned for the search string. The scan always starts with the current line. If no number is specified, only the current line will be scanned.
search string	The search string is the character string being sought.
replacement string	The replacement string is the character string that will be substituted for the search string.

Note that the start column and end column take on the same significance as with the Delete command. We discussed the Delete command on page 181.

For example, the following command:

```
RU16"rong"wrong" [RETURN]
```

would cause all occurrences of rong in the 16 lines starting with the current line to be charged to wrong. Each line that is changed will be displayed.

Size (S?)

The Size command is used to find the length of the item in the editing buffer. The length will be reported as a number of bytes. The form of the size command is shown below:

```
S? [RETURN]
```

The output from the Size command has the following form:

```
ITEM LENGTH IS 'X' BYTES
```

Suppress (S)

The suppress command is used to turn the display of line numbers on and off. Initially, the line numbers will be displayed (suppress is off). The form of the Suppress command is listed below:

```
S [RETURN]
```

The system will respond by displaying either SUPPRESS ON or SUPPRESS OFF. The SUPPRESS ON message indicates that the line numbers will not be displayed. The SUPPRESS OFF message indicates line numbers will be displayed.

Tab (TB)

The Tab command is used to set tab stops. The Tab command has the following format:

```
TB number{number,...,number} [RETURN]
```

where each number represents the column on the screen at which a tab stop should be set. Up to 15 tab stops can be set. The tab stops must be arranged in ascending order.

For example, the following command:

```
TB 20,40,60 [RETURN]
```

would cause tab stops to be set at columns 20, 40, and 60.

A tab is executed by pressing either control-I or the tab key. The tab key is labelled $\Rightarrow$ on the IBM PC.

It should be noted that the tab stops set by the TB command do not coincide with the scale produced by the Column Number List command. To make the two coincide, add three to each tab stop. So, if tab stops were required at columns labelled 20, 40, and 60 by the Column Number List command, the following Tab command should be used:

```
TB 23,43,63 [RETURN]
```


Top (T)

The Top command causes the current line to be set to the first line of the item in the editing buffer. The Top command has the following form:

```
T [RETURN]
```

Up (U)

The Up command is used to decrement the number of the current line. The Up command has the following format:

```
U(number) [RETURN]
```

where the value of number represents the number of lines by which the current line number will be decremented. If no line number is specified, the current line will be displayed, but the current line number will remain unchanged. Note that the new current line is displayed as a result of this command.

Wildcard (^)

The wildcard command enables or disables the significance of the caret (^) as a wildcard. A wildcard will match any character when it is used in a search string. When the Editor is first entered, the wildcard is disabled. The wildcard command has the following format:

```
^ [RETURN]
```

The system will respond by displaying either ^ ON or ^ OFF. The ON message means that the caret is wild. The OFF message means that the caret carries no special significance.

An example of the use of the wildcard in the Locate command follows:

```
L10"H^S" [RETURN]
```

If the caret is wild, the Locate command will find character strings such as HAS, HBS, H9S, and HS. If the caret is not wild, the Locate command will only find the character string HS.

X

The X command is used to cancel the effect of the last editing command. Editing commands include: Input, Delete, and Replace. These commands display the results of their execution. If these results are unsatisfactory, they can be discarded with the X command. The X command has the following form:

```
XCF) [RETURN]
```

where the F parameter is used to specify that all editing changes since the last F command should be discarded.

By using the X command judiciously, editing errors can be avoided.

Messages Displayed by the Editor

The following table lists the messages displayed by the Editor and provides an explanation of each message's meaning.

Table 7.1. Editor diagnostics.

Message	Explanation
CMND?	The Editor does not recognize the command entered at the period prompt. Check for typographical errors.
COL#?	There are unneeded characters after the end of the command, or the column specification is wrong. Check command syntax.
EOI number	The end of the item occurred at the line number specified.
L number	Printed after X command to show update is deleted.

NOT ON FILE	The item specified by a merge command cannot be found on the disk file.
SEQN?	Editing change made out of sequential order. Issue an F command and redo the editing command.
STRING?	String specification is illegal or missing. Check command syntax.
TOP	The current line has been set to the top of the item (line 000).
'item'DELETED	The item has been deleted from the disk file using the FD command.
'item'EXITED'	The item has been abandoned with an EX command.
'item'FILED	The item has been updated on the disk file using the FI command.

8

The PROC Language/Stored Procedures

Introduction

One of the most powerful features of the PICK operating system is the PROC language. PROC is a process control language used with user-written PROC routines to perform repetitive functions of the operating system. PROCs which accomplish specific tasks can be developed and saved for re-use. PROCs can be invoked simply by typing the name of the PROC at the TCL prompt.

PROCs can store frequently used sequences of commands and can save a great deal of time. In order to write these routines we must first learn the language. It is similar to the JCL ~Job Control Language used on mini and mainframe computers. It has the additional

features of being able to interact with the operator and execute commands on a conditional basis. PROCs can:

- isolate the user from the operating system via a series of menus for program selection and system maintenance
- minimize operator error
- minimize the knowledge required to use PICK commands

For example, a PROC might be written to perform daily backups. The operator would simply type the name of the PROC and the program would provide information on the loading of disks, where they can be found, what to do with them later and would perform the actual backup. This chapter will instruct you in the use of the PROC language. By the end of this chapter you will be able to begin writing your own PROCs which will help to automate your system.

PROC's Defined

A PROC is a file which is created and maintained using the EDITOR. A PROC is stored as an item in a user's master dictionary. The PROC may contain:

- TCL commands
- PROC commands
- ACCESS commands
- calls to a user program
- calls to other PROCs
- EDITOR commands

It is the job of the PROC processor to read and interpret the commands in a PROC file and execute them. These commands may be in the form of PICK command lines or PROC commands. A PROC is considered a high level language program. Writing PROC programs is accomplished using PROC commands. Some commands are mnemonic and easy to remember. For example, C specifies a comment line, O specifies output of text to the terminal. The PROC commands we will use throughout the remainder of this book are discussed in subsequent sections.

In this chapter we assume you are familiar with the EDITOR since PROCS are constructed using the EDITOR. It may be helpful to review chapter seven before you continue reading. Some of the examples used here were created with the help of a program called INSTANT WIZARD. This program is described in detail in chapter twelve.

Storing a PROC

A PROC is stored as an item in the user's master dictionary. Attribute one of the PROC must contain the code PQ. Assume we are logged onto the SYSPROG account. The following sequence of commands will create an item in the master dictionary with the code PQ in attribute one:

```
ED MD TEST[RETURN]
```

The system will display:

```
NEW ITEM
TOP
, I[RETURN]
001+PQ[RETURN]
002+C CLEAR SCREEN[RETURN]
003+T C[RETURN]
004+[RETURN]
TOP
, F I[RETURN]
`TEST' FILED

>
```

where TEST is the name of the PROC. The function of this PROC is to clear the screen and home the cursor.

After PQ has been entered in attribute one, the substance of the PROC is entered beginning with attribute two. Use attribute two as a comment line describing the function of the PROC. When the LIST-PROCS command is executed, the value of attribute two will be placed in the description column, informing users of the function of each PROC. Or, if the PROC calls another PROC and consists of only a

few lines, place the link command in attribute two. This will provide users with the name of the PROC being linked to.

Note: If the system responds to the ED command by simply displaying the message TOP, the name selected for the PROC is already in use. If such a situation occurs, issue an EX command at the EDITOR's period prompt. This command causes an exit from the EDITOR without changing the item which was accessed. Another ED command with a different name for the PROC can then be issued.

Termination of PROC Control

Normally, once a PROC is invoked it remains in control of the system until all lines in the PROC have been executed or until an exit to TCL is called for by the PROC itself. Occasionally, it may become necessary to terminate execution prematurely. To terminate PROC execution on the PC/XT enter the sequence:

```
CTRL Break
```

The system will display:

```
I x.d  
!
```

where x.d is the location of the interruption. The ! is the prompt indicating you have entered the DEBUG mode. To exit the DEBUG mode, type END to return to TCL, or OFF to logoff the system.

Logon PROC's

One of the most "user friendly" features of PICK is the automatic execution of a PROC when the system is first booted or when a user logs onto a new account. After the system is booted and a user enters an account name, PICK will automatically search for and execute a PROC whose item-id is the same as the user's account name. This PROC is called a logon PROC and should contain the commands to be executed following booting of the system or account logon. These commands might include setting up the terminal and printer charac-

teristics and subsequent transfer to another PROC used as a menu to select other programs.

Note: If no PROC exists matching the account name, the user is logged on to the specified account and placed at TCL.

Let's look at an example. Assuming the Logon: prompt is displayed on your screen type:

```
SYSPROC[RETURN]
```

The system will display:

```
<<< THE PICK SYSTEM >>>
<<< time PC/XT version n.n date >>>

0 SYSPROC

Documentation addenda may be reviewed by
typing 'ADDENDA' at TCL.

>
```

A PROC called SYSPROG exists in the master dictionary of the SYSPROG account. Whenever logon is requested to SYSPROG this PROC is automatically executed. Its three major functions are to:

- Set the terminal and printer characteristics for line 0.
- Display the name and line of the currently logged account.
- Display the message concerning documentation addenda.

After we have discussed the basics of the PROC programming language and a few of the PROC commands we will list the SYSPROG PROC and explain how it works.

Components of the PROC Programming Language

There are many components of the PROC language. These components consist of:

- input/output buffers
- numeric labels for control of branching or looping
- operators and relational testing capability
- pattern testing
- comments
- PROC commands

There are a number of PROC commands. Our discussion will be confined to those commands depicted in figure 8.1.

Command	Function
A	Moves data from input to output buffers
IF	Executes commands on a conditional basis
C	Specifies comments for documentation purposes
T	Provides cursor control and screen attributes
H	Builds TCL command line in the output buffer
P	Causes execution of command line within a PROC
PW	Used for debugging PROCs
X	Causes exit to TCL or calling PROC
GO	Controls branching to other parts of a PROC
RI	Empties the input buffer
IP:	Accepts input from terminal with : as prompt
O	Outputs specified text string to terminal
()	Link to specified PROC

Figure 8.1. Summary of PROC commands and their functions

For information on other PROC commands consult the PICK User Reference Manual.

PROC Components and Commands

Although there is no limit to the number of lines or commands within a PROC, only one command per line is permitted. The first line of every PROC must contain the code PQ which identifies the item as a PROC. In order to expedite execution time, each PROC should be limited in length to one hundred lines. Complex tasks can be executed by linking a number of PROCs together. In this section the PROC components and commands are discussed.

The Input/Output Buffers

The PROC input/output buffers are temporary data storage areas. The function of a PROC is to move data between the input/output buffers. When this is accomplished, TCL commands and other processor commands can be executed. The data is found in lines in the PROC or entered and stored in the buffers when the PROC prompts the user for input. Although the passing of data from one buffer to another is transparent to the user, there are commands required to execute the passing of data between the buffers which we will discuss.

Labels

PROC labels are used as reference points for branching to another area of the PROC or executing repetitive loops. A label is indicated by any number of numeric characters. When using a label, the PROC command must begin one space after the number. You must be careful to avoid using duplicate labels since only the first occurrence of the label is used to transfer control. Although there is no limit to the number of numeric characters or special sequence required, if you are writing a number of PROCs it would be helpful to establish a consistent labelling system within your PROCs for clarity. Examples of using consistent labelling are:

- 1 start of PROC
- 999 logoff system

```
      INTRO
001 PQ
002 C!  INTRO
003 1 0
004 T C,+
005 C Display section of the proc.
006 0
007 0
008 0      -----
009 0              INTRO
010 0      -----
011 0
012 0      1.  WORD PROCESSING
013 0
014 0      2.  SPREADSHEET/FINANCIAL MODELING
015 0
016 0
017 0
018 0
019 0
020 0
021 0
022 0
023 0
024 0
025 0
026 0
027 0
028 0      Select one of the above, or ESC, or 'OFF' +
029 RI
030 IP:
031 C Branching section of the proc.
032 IF A = EX X
033 IF A = [ X
034 IF A = . X
035 IF A = OFF GO 999
036 IF A = 1 GO 10
037 IF A = 2 GO 20
038 GO 1
039 C Program calling section of the proc.
040 10 C Run word processing program
041 [LINK TO WORD PROCESSING PROGRAM]
042 GO 1
043 20 C Run spreadsheet program
044 [LINK TO SPREADSHEET PROGRAM]
045 GO 1
046 999 C LOGOFF SYSTEM
047 HOFF
048 P
```

Figure 8.2. INTRO PROC

- 99 terminate PROC execution and exit to TCL
- Using 10, 20, and other increments of 10 as branch points to other locations within the same PROC.

Consider the PROC in figure 8.2.

Each line in the PROC is numbered beginning with 001 and ending with 048. Note the labels 10 and 20. Each label is on a line by itself. The line following the label executes the command to run a program. The labels in this PROC are in lines 003, 040, 043, 046. Figure 8.3 depicts the function of each label.

Label	Function
1	start of PROC
10	word processing program branch point
20	spreadsheet program branch point
999	logoff the system branch point

Figure 8.3. Function of labels in INTRO PROC

Relational Operators

Relational operators are used to indicate a relationship between parameters. There are six relational operators.

Symbol	Function
=	Indicates equality
#	Indicates not equal
<	Indicates less than
>	Indicates greater than
[	Indicates less than or equal to
]	Indicates greater than or equal to

Figure 8.4. Relational operators

Examples of how to use relational operators will be provided in the section on the IF command.

Moving Data between Buffers: The A Command

This command is used to move data from the input buffer to the output buffer. In our PROC examples the A command will be used primarily to temporarily store user entered data. It will be used in conjunction with the IF command.

Relational Testing: The IF Command

This command permits conditional execution of specified PROC commands. The general format of this command is:

IF a-command operator string proc-cmnd

where a-command is A, the command which moves data between buffers, operator is any of the relational operators depicted in figure 8.4, string is any string of characters which the parameter is to be compared against, and proc-cmnd is any valid PROC command.

The IF command evaluates the relation and, if the relation is true, executes the PROC command specified. If the relation is false the PROC command following the relation (operator) is ignored and PROC execution continues with the command on the next line. Consider these lines from a PROC called INTRO:

```
028 D Select one of the above, or ESC, or 'OFF'+
029 RI
030 IP:
031 C Branching section of the PROC
032 IF A = EX X
033 IF A = [ X
034 IF A = . X
035 IF A = OFF GO 999
036 IF A = 1 GO 10
037 IF A = 2 GO 20
038 GO 1
```

The only operator in use is =. Lines 032 through 037 use the IF command to test the value of A. Let's interpret the purpose of each of these lines.

```
032 If A equals the string EX execute the X command.  
033 If A equals the [ character execute the X command.  
034 If A equals . execute the X command.  
035 If A equals the character string OFF go to the label  
999.  
036 If A equals 1 go to the label 10.  
037 If A equals 2 go to the label 20.
```

Line 038 is important because it transfers control to the label 1 in the event that A doesn't equal any of the strings tested for. Label 1 is the beginning of the PROC and the the command lines at the beginning of the PROC cause the screen to be repainted. Lines 032 through 037 are designed to restrict operator input. If any values are entered other than those permitted, control is transferred to the line labeled 1.

The C Command: Comments

When writing a PROC, it is helpful to use comment lines at various points to indicate the function of a particular portion of the PROC. Comments are always preceded by a capital C. Comments have no effect on PROC execution. Consider the first few lines in the PROC below:

```
001 PQ  
002 C THIS IS A PROC CALLED INTRO  
003 1 0  
004 T C,+  
005 C THE NEXT FEW LINES ARE PART OF SCREEN  
DISPLAY
```

Lines 002 and 005 contain the C command. Comment lines are placed in the PROC to clarify the function of certain areas of the PROC.

Screen Control Commands

The screen control commands allow you to format a screen display within a PROC. The T command is used to specify cursor positioning, position text on the screen, clear the screen, set screen at-

tributes (such as blinking and protect), or cause a bell code (terminal beep) to be output. The format of the T command is:

T function, function, ...

where function may be:

- a text string
- output a B (BELL) code
- an x,y coordinate
- C or CLEAR to clear the screen
- an attribute-controlling value as depicted in figure 8.5

Value	Function
(-1)	Clears screen and positions cursor to upper left corner of the screen ("home" position)
(-3)	Clears screen from cursor position to end of screen
(-4)	Clears screen from cursor position to end of line
(-5)	Turns blinking attribute on
(-6)	Turns blinking attribute off
(-7)	Starts protected field...data cannot be overwritten
(-8)	Turns off protect mode
(-9)	Backspace the cursor one character
(-10)	Move the cursor up one line

Figure 8.5. Screen attribute controlling values

Screen controlling values and x,y coordinates **must be** enclosed in parenthesis. Before using any of the screen control functions shown in figure 8.5, be certain that they are supported by your terminal.

More than one function on a line, each separated by a comma, is possible. It is also possible to continue a T command on the next line **if** the preceding line ends with a comma. For clarity, it is best to specify CLEAR or BELL instead of issuing a C or B command. Some examples of T commands and their functions are:

Command	Function
T (12,13)	positions cursor to column 12, line 13
T (7,6)	positions cursor to column 7, line 6
T B	sounds the terminal bell/buzzer/ beeper
T C	clears the screen
T C,(-5), "MENU", (-6)	clears screen, starts a blinking field, prints the word "MENU" and ends the blinking field.
T C,B,(9,9), "MENU", (9,8) comment, "MASTER"	clears screen, sounds the terminal bell, positions the cursor to column 9, line 9 and prints the word "MENU", positions the cursor to column 9 line 8 leaves room for a comment (not displayed) and prints the word "MASTER."

Constructing TCL Commands and Clearing Buffers

PICK commands and application software can be executed from within a PROC program by using the H command. The H command is used to specify that the following data should be stored in the active input/output buffers. The active input/output buffers are used to build TCL command lines. The RI command is used to reset the input buffer to an empty condition. The format of the H command is:

H(text)

Text can specify:

- a PICK command
- the name of the program to be run

H command examples are depicted in figure 8.6. When executed from within a PROC program it is as if the command were typed at the PICK prompt. Note that there is **no space** between the H and the command.

Command	Result
HSTARTPTR 0,0,1,S1	Builds STARTPTR command line for a serial printer
HVERIFY-SYSTEM	Builds VERIFY-SYSTEM command line
HOFF	Builds an OFF command line

Figure 8.6. H command examples

PROC Execution/Debugging and Termination

Three commands are described in this section. They are:

- P
- PW
- X

The P command causes a PROC to execute a TCL command. The PW command is used as a debugging tool during the writing of a PROC. The X command is used to unconditionally terminate PROC execution from within the PROC itself. Since a PROC automatically terminates after the last line of the PROC the purpose of the X command is to terminate execution prior to the final PROC statement. Study the SYSPROG PROC depicted in figure 8.7.

```
SYSPROG
001 PQ
002 HTERM 79,24,0,0,2,8,80,60,I
003 P
004 HSET-SYM PSYM
005 P
006 Q
007 HWHO
008 P
009 Q
010 QDocumentation addenda may be reviewed +
011 Qby typing 'ADDENDA' at TCL.
012 HCOLOR BROWN
013 P
```

Figure 8.7. SYSPROG PROC

Lines 002, 004, 007, and 012 each contain TCL commands placed in the primary output buffer by the H command. The P commands in lines 003, 005, 008, and 013 cause the commands to be executed. If the P commands were replaced by PW commands, each TCL command would be displayed on the screen before execution with a ? prompt. If S[RETURN] is entered the TCL command displayed is skipped (not executed) and PROC execution continues until the next H command line is encountered. If X[RETURN] is entered, PROC execution is terminated. Pressing [RETURN] allows the TCL command displayed to be executed and PROC execution to continue. If an X command replaced the P command in line 008, the WHO command would not be executed and an exit to TCL would occur.

The GO Command

The GO command allows branching to another area of the PROC. The format of the GO command is:

GO n

where n is a valid PROC numeric label. Some examples of the GO command are:

- GO 1
- GO 10
- GO 20

GO is often used with other PROC commands as we saw in the section on the IF command.

Clearing Buffers and Inputting Data

The RI command is used to empty the contents of the input buffers. After the RI command is issued, it can be followed by an IP command. IP causes a prompting character to be displayed and reads data input from the terminal keyboard and places it in the currently active input buffer. The format of the RI command is:

RI

The format of the IP command is:

IP(r)

where r indicates the character to be used for prompting. If r is not specified the default prompting character (>) is used. In all of our PROCs we will use the command IP: which will display the : (colon) prompting character. When the prompting character is displayed an entry must be made before PROC execution resumes.

The O Command

This command is used to display characters on the screen for user prompts and to output blank lines to the screen. The format of the O command is:

Otext (+)

where text can be any string of characters. If the last character entered on a line is the + sign, a [RETURN] will not be output by the PROC processor. This is most useful when terminal input is requested. The + sign indicates that data is to be placed in the input buffer and that [RETURN] will be entered by the operator. For example, the screen display output by the INTRO PROC in figure 8.2 was produced using O commands. Note the use of the + sign at the end of line 028. This is followed by an RI command in line 029 and an IP: command in line 030. When the O command is on a line by itself, a blank line is output. The + sign can also be used to cause the PROC to pause until the operator types a key. This allows the operator time to view a screen of data. Consider the following line in a report generating PROC:

```
0End of Report. Press RETURN to continue. +
```

When this line is output, the PROC will pause until the [RETURN] key is pressed allowing the last screen page of the report to be viewed.

The Link Command

Earlier we said that it is advisable to limit the length of PROCs to no more than one hundred lines in order to speed their execution.

The link command makes it possible for one PROC to “call” another PROC. The link command can be placed anywhere in a PROC, although it is usually found at the end of a PROC or following a numeric label used as branch point. The format of the link command is:

(file-name item-id)

where file-name is the name of the file and item-id specifies the name of the PROC stored within the file. Consider the following example:

```
W
001 PQ
002 (WPL W)
```

This two line PROC may be stored as item W in a user’s master dictionary. Its sole function is to execute another PROC stored as item W in a file called WPL. The W PROC is executed when W[RETURN] is entered from the keyboard. Later in this chapter we will look at some examples of the link command used as branch points.

Using the PROC Language

We will now consider a series of PROCs which can be used as models for writing other PROCs. In some examples you may be asked to type in the examples and try them while in others only the PROC operation and function will be discussed. Feel free to use any of these PROCs on your system or to modify them as your needs require. Complete listings of all PROCs can be found in the appendix. We will start with some simple examples.

Example 1

This PROC is called WHICH. It is stored as the item WHICH in the PROCLIB file. Its function is to use the BLOCK-PRINT verb to print the message: IBM PC-XT to the screen as depicted in figure 8.8.

```

      IIII  BBBB  MM    MM
      II   BB   BB   MMM  MMM
      II   BB   BB   MM  MMM MM
      II   BBBB  MM  MMM MM
      II   BB   BB   MM  M  MM
      II   BB   BB   MM    MM
      IIII  BBBB  MM    MM

PPPPPP      CCCC      XX      XX  TTTTTTTT
PP  PP  CC  CC      XX  XX      TT
PP  PP  CC      XX  XX      TT
PPPPPP  CC      ----  XXX      TT
PP      CC      ----  XX  XX      TT
PP      CC  CC      XX  XX      TT
PP      CCCC      XX      XX      TT

```

...the Pick system for

Figure 8.8. Screen display for BLOCK-PRINT

Let's look at the WHICH PROC.

```

WHICH
001 PQ
002 HBLOCK-PRINT IBM PC-XT
003 P
004 O ...The PICK system for

```

Line 001 is the code PQ. This specifies this item as a PROC. Line 002 uses the H command to build the TCL command BLOCK-PRINT. IBM PC-XT is the text string which will be block printed. Line 003 uses the P command to cause execution of the PROC. Line 004 uses the O command to output a text string beginning with three periods to the terminal.

If we had wanted the output from the BLOCK-PRINT verb directed to the printer, line 002 would have to be modified using the EDITOR to add the printer option:

```
002 HBLOCK-PRINT IBM PC-XT (P)
```

Note that only the text specified for block printing is output to the printer. The O command only outputs text to the terminal.

Example 2

The purpose of the PLISTPROCS PROC is to list all of the PROCS on the currently logged account and give the operator a choice of routing the data to the screen or printer. Upon termination of the PROC control is passed to another PROC stored as item UTILITY in a file called WPL. The PROC illustrates:

- the use of the A command
- how to link to another PROC
- execution of an ACCESS command from within a PROC
- the use of the T command
- the use of the C command

```
PLISTPROCS
001 PQ
002 C MENU LISTPROCS
003 T CLEAR
004 OOutput to the (T)erminal or (P)rinter? +
005 RI
006 IP:
007 HSORT DICT MD WITH D/CODE "PQ" *P2
008 IF A1 = P HLPTR
009 P
010 O
011 O
012 OEnd of report. Press RETURN to con-
    tinue. +
013 IP:
014 (WPL UTILITY)
```

Figure 8.9. The PLISTPROCS PROC

MASTER

001 PQ
002 C! MASTER
003 1 0
004 T C,+
005 C Display section of the proc.

006 0

007 0

008 0

MASTER MENU

009 0

010 0

Your Company Name

011 0

012 0

1. ACCOUNTING/ORDER ENTRY/INVENTORY CONTROL

013 0

014 0

2. FINANCIAL MODELING/SPREADSHEET

015 0

016 0

3. JET WORD PROCESSING

017 0

018 0

4. DATA BASE MANAGEMENT (WIZARD)

019 0

020 0

5. CYCLONE III WORD PROCESSING

021 0

022 0

6. TOTALWARE

023 0

024 0

7. SYSTEM UTILITIES

025 0

026 0

8. HELP ON PICK COMMANDS

027 0

028 0

Select one of the above, or ESC, or 'OFF' +

029 RI

030 IP:

031 C Branching section of the proc.

032 IF A = EX X

033 IF A = [X

034 IF A = . X

035 IF A = OFF GO 999

036 IF A = 1 GO 10

037 IF A = 2 GO 20

038 IF A = 3 GO 30

039 IF A = 4 GO 40

040 IF A = 5 GO 50

041 IF A = 6 GO 60

042 IF A = 7 GO 70

043 IF A = 8 GO 80

044 GO 1

045 C Program calling section of the proc.

```
046 10 C Run accounting programs
047 [WPL ACCOUNTING]
048 GO 1
049 20 C Run spreadsheet programs
050 [WPL SPREADSHEET]
051 GO 1
052 30 C Run JET word processing programs
053 [WPL JET]
054 GO 1
055 40 C Run WIZARD programs
056 [WPL W]
057 GO 1
058 50 C Run Cyclone word processor
059 [WPL CYCLONE]
060 GO 1
061 60 C Run TotalWare office automation system
062 [WPL TOTALWARE]
063 GO 1
064 70 C Run utility proc menu
065 [WPL UTIL]
066 GO 1
067 80 C Run help proc
068 [WPL HELP]
069 GO 1
070 999 C Logoff system
072 HOFF
073 P
```

Figure 8.10. The MASTER PROC

Example 3

The MASTER PROC as depicted in figure 8.10 displays a series of menu choices. This PROC was written using the instant WIZARD program which will be discussed in chapters ten and twelve. WIZARD writes PROCs which follow a standard format. The format consists of:

- a display section
- the reading of terminal input
- a branching section
- a program/PROC linking section

Consider the MASTER PROC as depicted in figure 8.10.

If we analyze the PROC, we see that line 001 specifies the item as a PROC. Line 002 is a comment line identifying the item. Line 003 is the label one. Line 004 clears the screen. Lines 005 through 028 use the O command to output text to the screen. Lines 029 and 030 reset the input buffer and prompt the user for input. Lines 031 through 043 are the branching section of the PROC. Depending upon the input the following may occur:

- an exit to TCL
- the execution of an OFF command
- branching to any of the other labels specified by the value of A.

For example, if $A = 1$ the PROC branches to label 10 (046). At this point the PROC calls a program called **WPL ACCOUNTING**. Line 044 causes execution to continue at the label 1 (line 003) in the event that A does not equal any of the values specified.

The MASTER PROC, as depicted, is non-functional, with the exception of the ESC and 'OFF' selections, because the lines which specify linking to other PROCs or programs must be modified using the EDITOR. The names of the programs on **your system** must be specified in the link command. If the PROC attempts to link unsuccessfully to a program the [3]VERB? error message will be displayed and you will be returned to TCL. If linking to another PROC is unsuccessful the message :

```
[267] PROC TRANSFER TO 'Procname' CANNOT  
BE COMPLETED
```

is displayed and you are returned to TCL.

To illustrate how the PROC can be modified, let's activate menu selection 8. Using the EDITOR type:

```
ED WPL MASTER[RETURN]
```

The system will display:

```
TOP
*
G043[RETURN]
.R[RETURN]
043G0 999[RETURN]
.FI[RETURN]
'MASTER' FILED.

>
```

Now when choice 8 is selected, the result is the same as if 'OFF' was entered at the prompt. If you would like to clear the screen display, you can insert a new line in the PROC as follows:

```
ED WPL MASTER[RETURN]
.G070[RETURN]
.I[RETURN]
T C[RETURN]
FI
```

This inserts a new line which will clear the screen and home the cursor before execution of the OFF command.

Getting Help

The PICK system has no built in provision for on-screen help. However, a menu could be written using the PROC language to either provide instruction for using frequently used commands, or provide help and actually execute system maintenance commands such as FILE-SAVE and FILE-RESTORE. In addition menus or help screens could be constructed for each account which would direct each user to a menu of software choices for that account. This would virtually isolate novice users from the operating system. Example four illustrates such a help screen.

Example 4

This example illustrates a help PROC which provides a summary of the PICK commands and their functions which were discussed in

chapter six. The link command may be used to link this PROC to a utility menu which provides specific help on or executes PICK system commands. This PROC is most helpful to novice users.

```
HELP
001 PQ
002 T C PICK Version 1.3 Help Screen
003 O          PICK Version 1.3 Help Screen
004 O
005 OCommand/Proc      Function
006 OACCOUNT-SAVE      Backup individual accounts
007 OACCOUNT-RESTORE   Restore backup data to an account
008 OCOLOR             Set primary color of main terminal
009 OCREATE-ACCOUNT    Create new accounts
010 OCT               Display file contents on terminal
011 ODELETE-ACCOUNT    Deletes an account and its files
012 OFDISK            Creates/changes active disk partition
013 OFILE-SAVE         Performs complete system backup
014 OFFORMAT           Formats floppy disks under PICK
015 OLISTACC           Lists logon data for each account
016 OLISTCONN          Lists ACCESS connectives
017 OLISTFILES         Lists the files on an account
018 OLISTPROCS         Lists the PROCs for currently logged account
019 OLISTU             Lists all currently logged on users
020 OLISTVERBS         Lists the verbs for an account
021 OLOGTO             Logon to another account
022 OMSG              Send messages to other users
023 OOFF              Logoff system
024 OPOVF             Display contiguous disk space
025 OPOWER-OFF        Flushes RAM to disk for system shutdown
026 O
027 O...press RETURN for next screen page+
028 IP:
029 T C
030 O          PICK Version 1.3 Help Screen
031 O
032 OCommand           Function
033 ORESTORE SYSTEM    Restores operating system/data files from floppy
034 ORUNOFF            Formats edited text for printing
035 OSET-TIME          Set current system time
036 OSET-DATE          Set current system date
037 OSTARTPTR          Activate port/spooler for serial/parallel printer
038 OTERM             Set up terminal/printer parameters
039 OTERM-TYPE         Set terminal default values
040 OTIME              Outputs system time and date
041 OMHAT             Displays system configuration
042 OMHO              Outputs current account status
```

Figure 8.11. The HELP PROC

To create the PROC on your system type:

```
ED filename HELP[RETURN]
```

and use the EDITOR to input the lines as depicted in figure 8.11. As is illustrated by figure 8.11, the majority of this PROC is devoted to using the O command to output text to the screen. The other commands used are the T C and IP: commands.

Summary

The purpose of this chapter was to familiarize the system manager with the PROC language so that PROCs can be written to automate operating system functions or to isolate users from the operating system. Sample PROCs have been provided to illustrate the use of the PROC language and may be used in their present form or modified to accommodate your system. More PROCs will be described in chapter ten.

When writing PROCs either use an existing filename and enter the PROC as an item in that file, or create a new file using CREATE-FILE and enter the PROC as an item in the new file. Then, using the editor, create an item with the name of the PROC to be executed such as:

```
ED MD HELP[RETURN]  
I  
PQ[RETURN]  
CH HELP[RETURN]
```


9

PICK System Maintenance

Introduction

This chapter is intended to acquaint the PICK user with the tools necessary to support and maintain a properly functioning system. Some of the topics were discussed in the tutorials in chapter six and are worthy of review. The topics we will cover in this chapter are:

- peripheral attachment
- restoring contiguous disk space
- the SYSTEM file and system-level files
- the ERRMSG file
- accounting history file
- error handling and Group Format Errors
- deleting items from a file (DELETE PROC)
- sharing program and data files
- a review of system backup procedures

In this chapter when you are asked to type examples, particularly those using specific account names, remember that these accounts might not be part of your system. When typing these examples, you may want to use an account name present on your system other than SYS-PROG or TUTOR.

Peripheral Attachment

The most frequently used peripherals in a PICK system are:

- the floppy disk drive
- terminal
- printer

The Floppy Disk Drive

The floppy disk drive only need be attached when performing backup procedures or installing software. In version 1.3 of PICK the commands required for attaching and detaching the floppy disk drive are incorporated in the FILE-SAVE PROC. During selective file restores, it is necessary to use the T-ATT command to attach the floppy and the T-DET command to detach it. Remember that in a PICK PC/XT system the floppy disk is referred to as **tape**.

The Terminal

The terminal attached to the system must be of a type supported by PICK. Otherwise characters will not be displayed on the screen properly. The currently supported XT terminal types are depicted in figure 9.1. Some application software packages, such as WIZARD, may provide facilities for supporting and creating new terminal types. For simplicity we recommend using a PICK-supported terminal.

Term Type	Manufacturer/Model
A	Adds 580
B	Ampex 210
C	C. Itoh VT52
D	Datamedia
I	PC/XT color or monochrome (line 0 only)

L	Lear Siegler
M	Ampex 80
P	Pertec 701
R	Adds Regent
V	Adds Viewpoint
W	Wyse 50

Figure 9.1. PICK-supported XT terminal types

If your terminal is capable of emulating one of those depicted in figure 9.1, consult your terminal documentation for instructions on placing the terminal in emulation mode.

The Printer

A PICK system is capable of supporting both serial and parallel printers. In order to maximize the number of users, the COLDSTART PROC automatically configures the PC/XT for a parallel printer. If you plan to use a serial printer you must follow the procedures outlined in chapter six, session twenty-one. In addition, we recommend you study the serial cabling requirements for your printer and the serial cabling diagram depicted in figure 9.2.

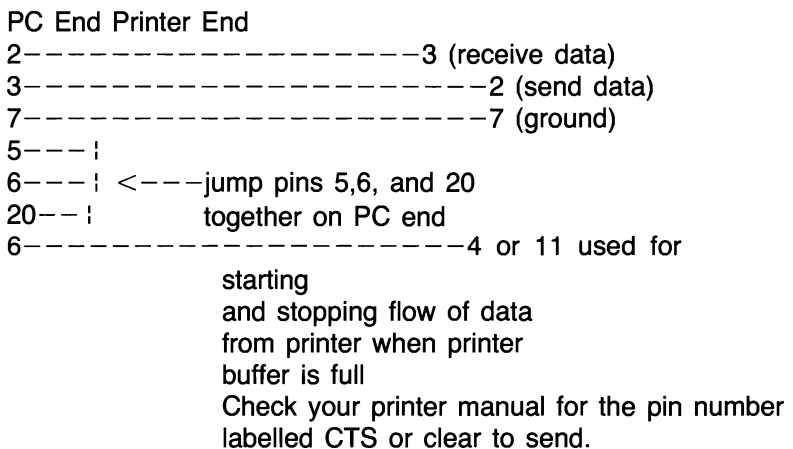


Figure 9.2. Serial printer pin assignment cabling schematic

Note: the PC/XT is capable of supporting two serial printers if only the line 0 terminal is used exclusively.

Restoring Contiguous Disk Space

As programs and data files are added to and deleted from the hard disk, the once contiguous disk space becomes fragmented. The POVf command and the FILE-RESTORE procedure, as described in chapter six session fourteen, can be used to determine the amount of contiguous disk space and restore contiguity. In order to execute a FILE-RESTORE you must have a current set of FILE-SAVE disks from which to restore your files.

Note: Executing this procedure has serious implications because **everything** in the PICK partition is erased and the PICK partition is reinitialized.

Let's briefly review the procedure which restores contiguous disk space.

Step 1. Insert the disk labelled PICK PC SYSTEM #1 in the drive reboot the system by pressing **CTRL-ALT-DEL**.

Step 2. The message:

```
OPTIONS (A,F,K)
```

will display. Be careful in making your selection. Formatting of the PICK partition begins immediately...you are not given a chance to change your mind. Select option F. When selected, the system displays:

```
One moment, initializing Hard Disk drive  
0.
```

Step 3. When the hard disk has been initialized, you will be prompted to insert PICK PC SYSTEM diskettes two and three in sequence. When the data on these disks has been transferred the PICK operating system files have been restored. The next step is to load your data files.

Step 4. When prompted to load the data file diskettes, begin with disk #1 from your FILE-SAVE instead of using the PICK data files diskettes. Insert each of the remaining FILE-SAVE diskettes **in sequence** when

prompted. As each file is restored, the file name is displayed. Use the hard copy list of file names generated during your last FILE-SAVE procedure as a reference. You can compare the files on the list with each file restored to be certain all of the files were restored.

Step 5. After the data from the last diskette has been transferred, control is passed to the COLDSTART PROC as if you had executed a system boot. The last data files diskette can now be removed. Control will pass to TCL.

If a POVf command is executed after a FILE-RESTORE, you will find that all disk space is now contiguous. The amount of time required to execute a system restore should be equal to the amount of time it took for the FILE-SAVE plus an additional two to three minutes for initializing the PICK partition. A FILE-SAVE of ten megabytes worth of data takes thirty to forty-five minutes.

What to Do About Errors During FILE-SAVE and Restore

During the FILE-SAVE and restore procedures it is possible that errors will be encountered. We will discuss some possible responses to these error messages. However, you may want to consider performing two FILE-SAVE procedures. This way you'll always have a second backup should the primary backup become damaged. Three of the most common FILE-SAVE/RESTORE error messages are:

- Parity Error
- Obj Data Error
- Group Format Error

Parity Errors

If parity errors are encountered, the messages:

```
PARITY ERROR
ACCEPT/RETRY/QUIT (A/R/Q)?
```

are displayed. This usually means that an error has been encountered due to bad media or missing characters. Your options are (A)ccept,

(R)etry, (Q)uit. If you select A, the system will attempt to ignore the error and restore the file anyway. However some of the items in the file may be lost. When R is selected the system will try to “correct” the error. If this is unsuccessful, you are reprompted and must make another choice. If you select Q, the FILE-RESTORE procedure is exited and you are returned to TCL. If the message:

```
REEL# WAS LABELED  
#
```

is displayed, this means that for some reason the system is having difficulty reading the disk label. Try typing C or O. If this doesn't work try the next disk of your FILE-SAVE and reenter C or O. If this doesn't work enter Q for quit and restart the FILE-RESTORE using your second set of backup disks. This message usually occurs as a result of trying to restore FILE-SAVE disks out of sequence.

Object Data Errors

Should the message:

```
OBJ DATA ERR @ location CROSS-REFERENCING
```

be displayed, it may be in conjunction with the parity error described above. PICK is attempting to find where the last disk or file left off and continue with the restore procedure.

Note: When making an entry in response to an error message such as PARITY error, entering a Q will abort the FILE-RESTORE procedure. The restore cannot continue past this point or be restarted. If any accounts other than SYSPROG were not restored, they may be restored individually using the ACCOUNT-RESTORE procedure. SYSPROG and its files **must** be restored in order for the system to be rebooted.

Group Format Errors (GFE's)

Group format errors are generally encountered while accessing a program or a data file stored on the disk. When such an error is encountered the message:

GROUP FORMAT ERROR @XXX

is displayed. The specific cause of these errors is unknown but is believed to be the result of a hardware related problem such as static electricity or faulty disk drive. Do not suspect a faulty disk drive unless these error messages are encountered regularly. When the group format error message is displayed you can make one of the following entries:

Entry	Function
D	transfers control to the system debugger
E	transfers control to TCL.
F	specifies that the GFE error handler is to try to repair the error and continue execution.

In most cases you will want to enter 'F' to try to fix the error. However, be aware that the fixing process may cause some items in the file, when the GFE was encountered, to be lost. If this occurs you must determine which item was lost and execute a selective restore as soon as possible. If 'E' is entered, control immediately passes to TCL. You are now free to retry the procedure. If GFE's are encountered again, try the fix procedure.

The SYSTEM File and System-level Files

The SYSTEM file is used to define user master dictionaries. The system file also contains pointers which are used to access important PICK files. These files are:

- ACC (accounting file)
- BLOCK-CONVERT (used for block-printing)
- PROCLIB (library of frequently used PROCS)
- SYSTEM-ERRORS (stores disc error messages)

A brief description of each file follows with specific attention focused on the PROCLIB file.

ACC

This file is the accounting history file. It keeps track of the actively logged users and the system usage statistics for each user.

BLOCK-CONVERT

The purpose of this file is to store the items which define the procedures for displaying the characters output during execution of the BLOCK-PRINT command.

PROCLIB

This file stores the commonly used PROCS such as CT, LIST-PROCS, FILE-SAVE, etc. Each PROC is stored as an item in this file. To obtain a listing of any PROC stored in this file type:

```
CT PROCLIB itemname[RETURN]
```

where CT is the copy to terminal command, PROCLIB is the file where the PROC is stored, itemname is the item-id of the PROC. To generate a listing of all items in the PROCLIB file type:

```
CT PROCLIB *[RETURN]
```

The asterisk is used as a wildcard character specifying all items in the PROCLIB file are to be listed. The listing could have been output to the printer if (P were specified as an option in the command line.

SYSTEM-ERRORS

As of this writing, this file is only used to store disk errors.

The ERRMSG File

The ERRMSG file contains the error messages displayed when a PICK processor encounters an error. Each PICK system comes with a standard set of approximately 250 error messages. Each error message is assigned a number which is its item-id. When a PICK processor calls for an error message the number and its associated error message are displayed. Although it is beyond the scope of this book

to illustrate methods of altering this file we will demonstrate the usage of the PRINT-ERR verb. The PRINT-ERR verb is used to selectively display PICK error messages for possible modification. The syntax for the ERRMSG command line is:

```
PRINT-ERR ERRMSG item-id or list (option
```

where PRINT-ERR is the verb, ERRMSG is the file containing the error messages, item-id or list is the error message number or range of error message numbers, and option is P if the output is to go to the printer. If more than one error message is to be displayed, each item-id must be specified. For example to list error messages 335 and 336 type:

```
PRINT-ERR ERRMSG 335 336[RETURN]
```

Accounting History File

The accounting history file is an integral part of a PICK system. It contains the data pertaining to system usage. The account name, date and time of logon, connect time, cpu units, and the number of LPTR pages generated during each logon are tracked. Accounting history is not automatic for every account. In order to use accounting history, the SYSTEM dictionary item for each user account must have a 'U' in attribute 9 of the account. If you type:

```
CT SYSTEM SYSPROG[RETURN]
```

the system will display:

```
SYSPROG
001 D
002 4557
003 11
004 1
005
006
007
008 SYS2
009 U
010 10
```

In this case, the account history file will be updated because U is the value assigned to attribute 009. If you were to type:

```
CT SYSTEM WALT[RETURN]
```

the system will display:

```
WALT
001 D
002 6228
003 29
004 1
005
006
007
008 SYS2
009 R
```

In this case the account history file will not be updated because U is not the value in attribute 009. This file can be modified by using the EDITOR to place a U in attribute 009 of the item WALT in the SYSTEM file.

Deleting Items from a File (DELETE PROC)

The DELETE PROC is used to delete items from a file. To use the PROC type:

```
DELETE filename item[RETURN]
```

If the item is not specified, the system displays:

```
NEXT ITEM
```

You are then required to type the name of the item you wish to delete. To exit the PROC type [RETURN] at the NEXT ITEM prompt. This PROC will function on any account.

SYSPROG-PL (SYSPROG PROC LIBRARY)

This file is a special library of PROCS and programs which resides in the SYSPROG account. Entering the following command line

while logged onto the SYSPROG account will produce a listing of PROCs contained in the file. This list is depicted in figure 9.3. To obtain the list of PROCs in this file type:

```
SORT SYSPROG-PL WITH D/CODE = "PQ"[RETURN]
```

The system will display:

```
SYSPROG-PL

ACCOUNT-SAVE
CHARACTERS
COLDSTART
CREATE-ACCOUNT
DELETE-ACCOUNT
FILE-SAVE
FORMAT
LIST-FILE-STATS
USER-COLDSTART

9 ITEMS LISTED

>
```

Figure 9.3. List of PROCs in the file SYSPROG-PL

Because this file resides in the SYSPROG dictionary, it is not possible for any other account to execute these PROCs. This is why the message [3] VERB? message is displayed when other accounts try to execute one of these PROCs.

Using PROCs on Another Account

In order to access PROCs in other accounts, file synonym definition items are required. This section describes a procedure for creating file definition items.

File Synonym Definition Items and Q-pointers

During account creation, a file residing in the SYSPROG account called NEWAC is automatically copied to the newly created account

using the CREATE-ACCOUNT PROC. NEWAC becomes the master dictionary for the new account. Each item id in the NEWAC file determines the verbs (commands), PROCS, or ACCESS commands the account may execute. The items in NEWAC which point to PROC related files are called file synonym definition items and contain Q pointers. A Q-pointer is similar to the PATH command in MS-DOS. A Q-pointer allows files to be shared among the accounts in a PICK system.

Q pointers are sometimes referred to as “Q-items.” In order to access PROCS in the PROCLIB file, each user account master dictionary must contain Q-items which point to the PROCLIB file and the items within it. The items in the NEWAC file which contain Q pointers are:

- BLOCK-CONVERT
- SYSPROG-BP
- PROCLIB
- MD
- POINTER-FILE
- M/DICT
- SYSTEM
- ERRMSG
- ACC

Each account must have a Q pointer which references a file and a second dictionary entry for each PROC (stored as an item in that file) to be executed. As we will see, an item pointing to a PROC file uses a PQ as a pointer instead of a Q in attribute 001. This identifies the item as a PROC. The items in NEWAC which reference PROCS are:

- LISTACC
- LISTPROCS
- DELETE
- EXCHANGE
- LISTUSERS
- LISTVERBS
- SET-LPTR
- LISTFILES
- CT

- CHOO-CHOO
- LOOP-ON
- LISTDICT
- SET-FILE
- CAT
- LISTU
- LISTCONN
- T-SPACE

To clarify this discussion consider the following command line entered from the SYSPROG account:

```
CT MD PROCLIB[RETURN]
```

The system will display:

```
PROCLIB
001 Q
002 PROCLIB
003
004
005
006
007
008
009 L
010 10
```

>

This is the master dictionary entry for PROCLIB. Note that there are ten attributes for this item. PROCLIB is the item id, the Q in attribute 1 defines the item as a pointer, attribute 002 (PROCLIB) tells Q which file to point to. We'll show you how to create this type of pointer later using the SET-FILE PROC. Suppose you were to type:

```
CT MD LISTFILES[RETURN]
```

The system will display:

```
LISTFILES
001 PQ
002 (PROCLIB LISTFILES)

>
```

This listing depicts a specific kind of Q pointer which points to the LISTFILES PROC stored as an item in PROCLIB. LISTFILES is stored as an item in the master dictionary (MD). Note the PQ in attribute 001. On new accounts, this item can be created using the EDITOR. As we already know, LISTFILES is a PROC which, when executed, lists all of the PROCS residing in the currently logged account. The body of the LISTFILES PROC resides in the PROCLIB file. If it were not for pointers, each account would have to have its own copy of every PROC and program. This would be an inefficient means of access and require excessive amounts of disk storage for the individual items and files.

Pointers also provide an additional means of security. If no pointer exists in the master dictionary on an account for a command, PROC, or program, that command, PROC, or program may not be executed.

Two Methods for Sharing PROC'S among Accounts

Method 1

If all newly created accounts are to share the same PROCS you can use the EDITOR to add items to the NEWAC file. This will automatically place pointers in the account's master dictionary to each of the items in a file you wish an account to access. For example, let's assume that you have added a new item called HELP to the existing PROCLIB file. Log on to SYSPROG and enter the following:

```
ED NEWAC HELP[RETURN]
```

The system will display:

```
NEW ITEM
TOP
,I[RETURN]
001+PQ[RETURN]
002+(PROCLIB HELP)[RETURN]
003+[RETURN]
TOP
,FI[RETURN]
'HELP' FILED.
```

>

Every time a new account is created the new account will have access to a PROC called HELP. If you are using this method, you must create an item in the NEWAC file for each new PROC to be accessed. If you forget one, you must use method two to add the item directly to the account's master dictionary.

Method 2

This method of sharing PROCS utilizes the SET-FILE PROC. The purpose of the SET-FILE PROC is to quickly set up a Q pointer file which will be used to point to the desired PROCS. Let's assume that the PROCS we wish an account to access reside in the SYSPROG account in the TESTPROCS file. To use SET-FILE, follow these steps:

Step 1. Logon to the account from which you want to access the programs.

Step 2. Create a "Q" pointer to the SYSPROG account for TESTPROCS by typing:

```
SET-FILE SYSPROG TESTPROCS[RETURN]
```

the system will display:

```
'QFILE' ADDED
```

>

Step 3. Type:

```
COPY MD QFILE[RETURN]
```

the system will display:

```
TD:TESTPROCS[RETURN]
```

Step 4. Create a pointer to the TESTPROCS file by entering:

```
ED MD HELP[RETURN]
```

The system will display:

```
NEW ITEM
TOP
.I[RETURN]
001+PQ[RETURN]
002+(TESTPROCS HELP)[RETURN]
003+[RETURN]
TOP
.FI[RETURN]
'HELP' FILED.
```

>

In chapter ten we will see how other PROCS stored in this file are accessed. In this section we have demonstrated the procedures for setting up Q-pointers to PROCS. If the body of the PROC does not exist and an attempt is made to run a PROC, the error message:

```
[267] PROC TRANSFER TO 'PROCNAME CANNOT BE
COMPLETED.
```

is displayed.

```
      FILE-SAVE
001 PQ
002 HT-ATT
003 P
004 IF E = 93 X
005 HT-REW
006 P
007 HT-WEOF
008 P
009 HT-REW
010 P
011 OFILE SAVE BEGINNING AT +
012 HTIME
013 P
014 OCONSOLE LISTING TO PRINTER+
015 RI
016 IHN
017 IP?
018 S2
019 IH4000
020 S3
021 IH1
022 COTAPE LABEL+
023 S4
024 CIP?
025 IHFILE-SAVE
026 O
027 ODO YOU WANT FILE STATISTICS REPORT+
028 S5
029 IHN
030 IP?
031 G 87
032 C *****
033 C TEMPORARY INSERT OF "COLD" PROC
034 C WRITE BOOT & ABS EVERY DAY
035 C
036 HT-ATT (512
037 P
038 HQSELECT DICT SM COLD-LIST
039 STON
040 HCOLDDUMP SM (E,I) (
041 P
```

```
042 HT-ATT (4000
043 P
044 HQSELECT DICT SM ABS-LIST
045 STON
046 HABS-FROM-OBJ SM<
047 HR771E<
048 P
049 C
050 C *****
051 87 C
052 HT-ATT (4000)
053 P
054 S1
055 HSAVE (S,T,D
056 IF A(,1) = Y H,P
057 H)
058 STON
059 S4
060 A(,100)
061 H<
062 P
063 OSTAT-FILE BEING DUMPED TO TAPE
064 HT-DUMP STAT-FILE ID-SUPP
065 P
066 HT-REW
067 P
068 OFILE SAVE FINISHED AT +
069 HTIME
070 P
071 HT-DET
072 P
073 C CHECK TO SEE IF WE NEED STAT REPORT
074 S5
075 IF A(,1) # Y GO 999
076 OGENERATE FILE STATISTICS REPORT
077 HTERM ,,,,,,132
078 P
079 HSORT STAT-FILE
080 H BY USER BY NAME
081 H ID-SUPP
082 H BREAK-ON USER "*** TOTALS FOR USER: 'UVD'" ID
083 H NAME BMS
```

```

084 H TOTAL SIZE
085 H TOTAL XITEMS
086 H TOTAL FRAMES
087 H HEADING "PAGE 'P'      * * * *   F I L E      S T A T
      T      * * * * ' "
088 H AV/ITM ITM/GP FRMS/GP TOTAL %UT
089 H TOTAL PAD TOTAL GFE
090 H LPTR
091 P
092 999 HT-DET
093 P

```

page 3 of 3

Figure 9.4. Listing of the FILE-SAVE PROC

System Backup

The FILE-SAVE PROC is stored as the item FILE-SAVE in the file SYSPROG-PL (SYSPROG PROC LIBRARY). Since it is only present in the SYSPROG master dictionary, it can only be executed from the SYSPROG account. If you are of a curious nature you can examine the FILE-SAVE PROC in detail by entering the command line:

```
CT SYSPROG-PL FILE-SAVE (PIRETURN)
```

The listing should resemble the one depicted in figure 9.4.

The function of the FILE-SAVE PROC is to execute the SAVE verb with the S,T,D,F,G options. To briefly review, figure 9.5 depicts each of the SAVE options and their functions.

Option	Function
S	Displays important statistics about each file
T	Outputs the FILE-SAVE to floppy disk
D	Saves the data portion of the file
F	Displays all of the file names being saved
G	An attempt is made to repair any group format errors should any occur.

Figure 9.5. FILE-SAVE options and functions

No one can force you to backup your system. However the FILE-SAVE procedure is simple enough that it should be performed at least weekly...more often if you enter data frequently. To execute a FILE-SAVE follow these steps:

Step 1. Have formatted floppy disks ready to insert in the drive when prompted.

Step 2. Logto SYSPROG and enter:

```
FILE-SAVE[RETURN]
```

The system will display:

```
BLOCK SIZE:500  
FILE SAVE BEGINNING AT current time cur-  
rent date  
CONSOLE LISTING TO PRINTER?Y[RETURN]  
DO YOU WANT A FILE STATISTICS RE-  
PORT?Y[RETURN]  
BLOCK SIZE: 4000
```

```
ENTRY #1
```

We recommend that the listing of files be sent to the printer along with a file statistics report. Entry #1 indicates that the file list and file statistics report will be output to the printer at the end of the FILE-SAVE procedure. Since most FILE-SAVEs will require insertion of additional disks the system cannot be left unattended. You will be prompted to insert disks as required. When the new disk is inserted the expected entry to continue the FILE-SAVE is "C" for continue.

Step 3. When the FILE-SAVE is complete, the list of saved files is output to the printer, the disk drive is detached, and you are returned to TCL.

Note: if the FILE-SAVE procedure is aborted and the reports are not generated, a spooled but non-printed file may exist on disk. This file can be removed automatically by rebooting the system.

Summary

In this chapter we have described the concepts and procedures necessary to maintain a PICK PC/XT installation. In addition we have outlined methods for sharing frequently used PROC'S among all the accounts on the system. The procedures described in the section on sharing PROC'S are necessary to utilize the menu PROC'S which we will discuss in the next chapter.

10

Using PROC'S as Menus

Introduction

This chapter is designed to illustrate the use of PROCS in a turnkey PICK installation. A turnkey installation is a complete computer system where minimal training is required to begin using the system. Time is spent learning how to put the computer to work rather than learning how to operate the computer.

The PROCS we will discuss are designed to minimize or eliminate the need to enter PICK command lines. The discussion that follows assumes that the PROCS have been installed by the system manager and that they are accessible on all user accounts. We also assume that:

- The PICK operating system has been installed.
- Application software has been installed.
- User accounts have been set-up, possibly including passwords.
- Q pointers have been set up in each account's master dictionary file allowing access to the PROCS.

The PICK PC/XT system used by the author had only the PICK supplied PROCS such as COLDSTART, FILE-SAVE, etc. to work with. The PROCs discussed here were written to facilitate system use, particularly for novices. The use of PROCs to help automate a PICK XT system will be examined in detail. The PROCs we will examine in detail are:

- A system startup PROC called COLDSTART
- An account logon PROC
- A help PROC called HELP
- A menu PROC called MASTER

The construction of a PROC designed to execute utility functions such as FILE-SAVE and FILE-RESTORE will be examined briefly. This PROC could be called UTILITY because it performs utility functions.

The Functions of PROC

In addition to its menu capabilities, PROC is the primary interface between the PICK BASIC programming language and TCL. There is no **direct** access to TCL commands from BASIC. This is important because anytime a BASIC program requires a TCL command to be executed, the program must exit to a PROC, have a PROC execute the command, and then reenter the BASIC program. This procedure, which slows down the system, will no longer be required when the version of PICK with the Execute Statement becomes available. *See chapter 13.*

Earlier we discussed the means of security provided by passwords and privilege levels. PROC'S used as menus provide an additional means of security by isolating users from the operating system and preventing unauthorized access to programs and data files. Only

the desired programs are placed on the menu. Anyone unfamiliar with PICK and TCL will be unable to gain access to other programs and files.

Starting the System: The COLDSTART PROC

When a PC/XT is powered on the system automatically boots from the hard disk. A special PROC called COLDSTART is executed which:

- Sets the default terminal type for lines 0, 1, 2.
- Prompts the operator to enter the current time and date.
- Activates the system printer, usually parallel.
- Verifies the integrity of the system software.
- Executes a logoff.

After the OFF command has been executed the system displays:

```
Logon
```

You must enter your account name in order to use the system. In response to the Logon prompt, type the name of the account to which you wish to logon. User account names acceptable by the system are determined by the account names entered by the system manager. The account name is usually the name of the user or a specific program making it easy to remember. It is not necessary to enter any command lines or know anything about PICK itself. All that is required is to know your user account name.

Logon PROC's

Upon logon PICK attempts to execute a PROC with an item-id identical to the user account name. The master dictionary of the logged account is searched for a PROC name matching that of the account name. If one is found it is automatically executed. Typically, the primary functions of logon PROCs are to insure that the proper terminal and printer characteristics are set for a line and to exit to TCL or link to another PROC or menu such as the MASTER PROC discussed later. Consider the WALT PROC depicted in figure 10.1.

```
WALT
001 PQ
002 HTERM 79,24,0,0,2,8,80,56,I
003 P
004 HWIZARD
```

Figure 10.1. Example of a logon PROC

We assume this PROC is stored as the item WALT in the master dictionary of the WALT account. The PROC has only two functions:

- To set the terminal and printer characteristics for the WALT account via the TERM command
- To automatically execute the program WIZARD

If the WIZARD programs are accessible from the WALT account, you will be taken to the WIZARD menu.

The HELP PROC

We assume that this PROC is stored as the item HELP in a file called TESTPROCS and that the appropriate Q pointer has been placed in the master dictionary of the WALT account. The PROC can be executed from TCL by typing:

```
HELP[RETURN]
```

or by selecting choice 8 from the MASTER PROC menu which we will discuss later. In either case a screen display is produced similar to that depicted in figure 10.2.

PICK Version 1.3 Help Screen	
Command	Function
ACCOUNT-SAVE	Backup individual accounts
ACCOUNT-RESTORE	Restore backup data to an account
COLOR	Set primary color of main terminal
CREATE-ACCOUNT	Create new accounts

CT	Display file contents on terminal
DELETE-ACCOUNT	Deletes an account and its files
FDISK	Creates/changes active disk partition
FILE-SAVE	Performs complete sys- tem backup
FFORMAT	Formats floppy disks under PICK
LISTACC	Lists logon data for each account
LISTCONN	Lists ACCESS connec- tives
LISTFILES	Lists the files on an account
LISTPROCS	Lists the PROCs for currently logged ac- count
LISTU	Lists all currently logged on users
LISTVERBS	Lists the verbs for an account
LOGTO	Logon to another ac- count
MSG	Send messages to other users
OFF	Logoff system
POVF	Display contiguous disk space
POWER-OFF	Flushes RAM to disk for system shutdown

...Press RETURN for next screen page

Figure 10.2. Screen display produced by HELP PROC

Since only twenty-four lines can fit on a screen page the [RE-
TURN] key must be pressed to see the next data screen. If the HELP
command was entered at TCL, rather than from the MASTER MENU,
you will be returned to TCL when the HELP PROC terminates. If

the command was executed via the MASTER PROC, you will be returned to the MASTER MENU upon termination of the HELP PROC.

The primary benefit of the help PROC is to provide a convenient reminder to new system managers of the most commonly used PICK commands and their functions. A further extension of the help concept would be to provide a PROC which, when requested, specific help for commands would be displayed. This help could consist of the command name, options, correct command line syntax, and a few examples. The item ADDENDA and the file DOC, supplied with every copy of PICK version 1.3, might be used as a model by the system manager to set up such a system. To view the contents of the ADDENDA item and DOC file type:

```
CT MD ADDENDA[RETURN]
```

and

```
CT DOC *[RETURN]
```

The printer option can be used in both cases if you would like a hard copy printout.

Running Application Software from a Menu: The MASTER PROC

This PROC was written using the WIZARD program described in chapter twelve. Because it is designed to operate with specific software packages, you must be certain the system manager has selected and installed the software to be used on your system and has modified the MASTER PROC to operate with the software selected. The MASTER PROC must also be accessible from all accounts. A complete listing of the MASTER PROC can be found in the appendix.

The purpose of the MASTER PROC is to isolate users from the operating system and to provide users with meaningful menu choices. All that is necessary to run a program is to select a menu choice and press [RETURN], the PROC does the rest. Figure 10.3 depicts the screen display provided by the MASTER PROC.

MASTER MENU

Your Company Name

1. ACCOUNTING/ORDER/ENTRY/INVENTORY CONTROL
2. FINANCIAL MODELING (SPREADSHEET)
3. JET WORD PROCESSING
4. DATA BASE MANAGEMENT (WIZARD)
5. CYCLONE III WORD PROCESSING
6. TOTALWARE
7. SYSTEM UTILITIES
8. HELP ON PICK COMMANDS

Select one of the above, or ESC, or 'OFF':

Figure 10.3. The MASTER PROC

Let's examine the function of each option.

Option 1: Accounting Programs

This option automatically executes an accounting program. Most businesses require some type of accounts receivable, accounts payable or general ledger package. A more specialized software package for doctors, lawyers, or accountants may be executed from this menu if installed on the system. When selected, the user is taken to the main menu of the accounting program.

Option 2: Financial Modeling

The financial modeling option is designed to execute a spreadsheet program. Some spreadsheet software, such as the Cyclone III package, may have their own PROC menu which this selection may transfer control to. We will review two PICK spreadsheet packages in chapter twelve.

Option 3: JET Word Processing

The JET word processing package is selected with this option. JET has become the most popular word processing program for the PICK environment. It has been integrated with many other software packages such as the TotalWare office automation system.

Option 4: Database Management (WIZARD)

For our purposes, database management is synonymous with the WIZARD programs. We will discuss WIZARD in detail in chapter twelve. When selected this option causes the WIZARD main menu, which is another PROC menu, to be displayed. Selecting this option will allow you to begin using the powerful PICK database system without learning complex commands.

Option 5: Cyclone III Word Processing

Selecting option five presumes that a word processing package called Cyclone III is installed on your system. Why two word processing packages on the same system? Each package has certain strengths the other lacks. See chapter twelve for further explanation on this point.

Option 6: TotalWare

The TotalWare office automation system is selected with this option. TotalWare is discussed in detail in chapter twelve.

Option 7: System Utilities

When this option is selected, the user is taken to a utility menu where system maintenance functions such as formatting floppies, FILE-SAVES and FILE-RESTORES and deleting files can be performed. This option is provided as a convenience and must be implemented by the system manager. It may also be password protected. In addition, most of the functions on this utility menu require that the user be logged onto the SYSPROG account. The use of the utility menu is described in detail later in this chapter.

Option 8: Help on PICK Commands

When selected this option automatically executes the HELP PROC.

Using the Menu to Logoff or Exit to TCL

No special menu choices have been provided for executing a logoff or exiting to TCL. In order to exit to TCL, press the ESC key then press [RETURN]. To logoff the system, enter OFF [RETURN] instead of selecting a menu choice. You will be logged off the system, ready to logon to another account.

Note: when the ESC key is pressed the character [is displayed. This is customary. Ignore this and press [RETURN].

A Word about Master Menu Error Messages

PROCs have limited error handling capability. If the MASTER PROC calls another PROC and can't find it an error message such as:

```
[267] PROC TRANSFER TO 'PROCNAME' CANNOT  
BE COMPLETED
```

is displayed. If the MASTER PROC causes a program to be executed and the program is not found an error message is displayed. When this occurs you are returned to TCL. In order to reenter the MASTER PROC simply type:

```
MASTER[RETURN],
```

If error messages are displayed regularly when attempting access to a program or PROC, consult your system manager.

Performing System Maintenance Functions: The UTILITY PROC

When option five is selected from the master menu, the user may be taken to the system utilities menu. Because this menu is important in the maintenance of a turnkey PICK system, we will consider the

implementation of such a menu. Some systems houses and value added resellers may have such menus already implemented on their systems.

The programming of such a menu is a complex task and is best left to a programmer. Unlike the MASTER PROC, which can be used as soon as the PROC or program names required are inserted into the appropriate line numbers, our discussion of the UTILITY PROC is limited to providing suggestions for menu functions and suggestions on how these functions might be implemented. It is hoped that the ideas presented here will help make your job easier when considering the purchase of a PICK system. We recommend you purchase a system from a vendor who has installed a utility menu to perform “housekeeping” functions. A utility menu might appear as depicted in figure 10.4.

```

                                SYSTEM UTILITIES MENU

                                Your Company Name

1. TAPE/FLOPPY MENU   7. LOGTO NEW AC-
                        COUNT

2. FILE-SAVE          8. RESET TIME (S)

3. FILE-RESTORE       9. RESET DATE (S)

4. ACCOUNT-SAVE (S) 10. CREATE NEW AC-
                        COUNTS (S)

5. ACCOUNT-RESTORE 11. DELETE ACCOUNTS
    (S)              (S)

6. CHECK DISK SPACE 12. RETURN TO MASTER
                        MENU

13. POWER-OFF (S/line 0 only)

Select one of the above, or ESC, or 'OFF':
```

Figure 10.4. The SYSTEM UTILITIES MENU

Note that some choices on this menu have an S or other instruction after them. If the choice has an S this means that you must be logged onto the SYSPROG account to execute the function. The POWER-OFF selection only works on line 0 **and** must be executed from the SYSPROG account. If choice 1 is selected, you may be taken to a sub-menu which allows you to format disks or tapes under PICK and change the active disk partition. Such a sub-menu might appear as depicted in figure 10.5.

```
DISK/TAPE UTILITY MENU

Your Company Name

1. FORMAT FLOPPY DISKS UNDER PICK
2. CHANGE ACTIVE PARTITION
3. DISPLAY CONTIGUOUS DISK SPACE
4. SP-STATUS (spooler/Printer status)
5. LISTPEQS (check status of spooled files)
6. RETURN TO UTILITY MENU

Select one of the above, or ESC, or 'OFF':
```

Figure 10.5. UTILITY PROC sub-menu

Designing the UTILITY PROC Menu

There are two methods of designing a utility PROC menu:

Method 1

Allow each command to be executed from within the PROC.

Method 2

Provide on screen help for each menu selection but an exit to TCL must be performed and the command executed manually.

The easiest method to implement is method two. However, this method leaves much room for operator error. In addition, a reasonable knowledge of PICK commands is required even when help messages are provided.

The more difficult method (have a PICK programmer do it) to implement is method one. Assuming method one is used, here are some points to consider in the design of a “bullet proof” utility menu system:

- A means of backing up or changing an entry should be provided
- No critical operation such as deleting a file or an account should be performed unless a message is first displayed warning the user of the consequences and provide a choice to exit the operation.
- The user should be returned to the menu upon successful command execution
- The user should be reminded to press [RETURN] in order to execute the command, if required.

Note: A good example of where a lot of help should be provided is during execution of the FILE-RESTORE procedure, since no help and no opportunity to change an entry is provided once the FILE-RESTORE screen is displayed.

Some Examples of Utility Menu Choices

In this section we will consider the following PROC menu choices:

- FILE-RESTORE (3)
- CHECK DISK SPACE (6)
- RESET-TIME (8)
- RESET-DATE (9)

With the exception of FILE-RESTORE, functions of the utility menu will require either linking to an existing PROC (such as CREATE-ACCOUNT) or using a PICK verb.

Example 1

Let's look at the FILE-RESTORE procedure. FILE-RESTORE is initiated by booting the system from the PICK-supplied floppy disk. After the sign-on message is displayed, the system displays:

```
OPTIONS (A,F,K)
```

unfortunately, you are not told what each of the options do. Ideally it would be best if the following were displayed when the RESTORE choice is selected:

```
PICK RESTORE PROCEDURES
```

```
A. RESTORE MONITOR AND ABS ONLY
```

```
F. INITIALIZE DISK/FULL FILE-RESTORE
```

```
K. DELETE PICK PARTITION
```

```
Insert PICK SYSTEM disk #1 CTL-ALT-DEL to  
reboot:
```

This screen would provide you with a reminder as to the functions of each selection and instruct you on what to do next.

Example 2

Selecting option 6 from the utility menu will execute a POVf command. Executed from a PROC the system might display:

```
Checking disk space allows you to see the number of  
contiguous frames of disk space. A single number indi-  
cates that one large block is available...this is the most  
desirable condition.
```


If a large number of fragmented blocks is displayed we recommend you execute a FILE-RESTORE procedure.

PRESS [RETURN] TO CHECK DISK SPACE:

Upon pressing the [RETURN] key the POVf command is automatically executed and the total number of contiguous frames is displayed. Another message might then be displayed informing you how to return to the utility menu.

Example 3

Setting the system time is accomplished using the SET-TIME verb on the SYSPROG account. The SET-TIME verb requires that the time be input in a specific format. Therefore selecting option 8 might display the following:

Setting the system time requires that you be logged onto the SYSPROG account and that the time be entered in the format:

hh:mm:ss including the colons.

Enter M to return to menu or T to set time:T

Please enter the correct time in the format
hh:mm:ss :

If you are logged onto the SYSPROG account and T is entered the time will be set and displayed and you will be returned to the UTILITY menu. If T is entered and you are not logged onto SYSPROG the message [3] VERB? will be displayed and you will be returned to TCL.

Example 4

Setting the system date is accomplished in a similar manner as the time. The primary difference is that the date must be entered in the format dd-mm-yy including the hyphens. If the hyphens are not included, the date remains unchanged. As in the SET-TIME example, if you are not logged onto SYSPROG the message: [3] VERB? is displayed when executing SET-TIME.

Errors Encountered during Utility Command Execution

On occasion errors may be encountered during command execution. If an error occurs you may be transferred to the debug mode and the ! prompting character displayed. If you enter:

```
END[RETURN]
```

you should be returned to TCL where you can reenter the UTILITY PROC by typing the name of the PROC.

Summary

This chapter has described the use of PROC menus in a turnkey PICK installation. The PROCs discussed were created using a combination of the PICK EDITOR and the WIZARD program. The data portion of each PROC must be stored as an item in a file, such as the WPL file in the WIZARD program. In addition, each PROC must have an associated dictionary item. For example since the MASTER PROC was generated using the WIZARD program it was stored as the item MASTER in a file called WPL (WIZARD PROC library). In order to function, a dictionary item called MASTER was also created. The data portions and dictionary portions of each PROC discussed in this chapter can be found in the appendix.

The concept of a utility menu was introduced to perform system maintenance functions such as FILE-SAVEs and FILE-RESTOREs. Due to the importance and complexity of such a menu, it is recommended that the ideas presented here be implemented by an experienced PICK programmer.

11

The PICK ACCESS Query Language

Introduction

Quick and easy data retrieval is a problem plaguing microcomputers. Once data is entered, there must be a way to retrieve the data in a meaningful way. Other microcomputer operating systems, such as MS-DOS and UNIX, have no built-in facility for data manipulation and reporting. Since the heart of PICK is its data file structure, retrieving data is possible by using the PICK ACCESS query language.

In order to extract information from a PICK database, a powerful query language called ACCESS was developed. In PICK circles this language is sometimes called ENGLISH because it uses common words such as SORT and LIST for its commands which are descriptive of their functions. This chapter will provide an introduction to the basic operations of ACCESS. In chapter twelve we will see how an appli-

cation program called INSTANT WIZARD can execute simple and complex ACCESS commands with minimal ACCESS knowledge.

This chapter presumes you are logged onto the PICK-supplied TUTOR account.

Data Files

Before ACCESS can be used, data files must exist. To briefly review, data files are created using the CREATE-FILE verb. The syntax for the CREATE-FILE command is:

```
CREATE-FILE filename m1 m2
```

where CREATE-FILE is the TCL command for file creation, filename is the name of the file to be created, m1 and m2 (modulo 1 and modulo 2) are the number of frames to be allocated for the dictionary and data portions of the file respectively. If file creation is successful, the messages:

```
[417] FILE 'filename' CREATED; BASE = x,  
MODULO = y, SEPAR = 1,  
[417] FILE 'filename' CREATED; BASE = x,  
MODULO = y, SEPAR = 1,
```

are displayed. The first message is for the dictionary portion of the file. The second message is for the data portion of the file. Since data files occupy more disk space, m2 is generally larger than m1. BASE x is the starting frame number, MODULO y is the number of frames specified for allocation, and separation is always one.

After a file has been created, the attributes must be specified. On the PICK-supplied TUTOR (tutorial) account this is accomplished using the BUILD.DICT program.

Note: The BUILD.DICT program only functions on the TUTOR account.

ACCESS Commands and Sentences

ACCESS commands are entered at TCL or via a PROC or program which prompts a user for input. Each ACCESS command is a

command line or **sentence** consisting of:

- An ACCESS verb
- The name of the file to access
- The file items to access
- Sorting criteria, selection criteria and modifiers
- Printer formatting commands or parameters

The tutorial discussed in the next section will instruct you in the construction of some ACCESS sentences.

PICK ACCESS Tutorial

A newly implemented feature of PICK version 1.3 is a tutorial. The tutorial documentation can be found in the last chapter of the PICK User Reference Manual. It is intended to familiarize novice users with the basic operation of the PICK system and provide an introduction to ACCESS. In this section we will illustrate the use of the tutorial as well as add a few new examples to build your knowledge.

ACCESS only understands upper case commands. Leave the caps lock key on while executing ACCESS commands. If an ACCESS command line is entered in lower case letters the error message:

```
[3] VERB?
```

is displayed. The name of the account used in the tutorial is TUTOR. To enter the tutorial account enter:

```
TUTOR[RETURN]
```

at TCL. The screen will display:

```
PASSWORD:
```

The password for this account has been predetermined. Enter:

```
LEARN[RETURN]
```

When a password is entered, the system does not display it on the screen. This provides a means of security. If the password is entered incorrectly, the system responds with the message:

PASSWORD?

and reprompts you with the Logon message. If you made a mistake entering the password, enter TUTOR again and carefully reenter the password:

LEARN[RETURN]

When logged onto the TUTOR account you will automatically be placed at TCL. You are now ready to use ACCESS.

Using ACCESS

The ACCESS processor is considered a database retrieval language. Previously entered and stored data can be sorted, listed, and printed with only a few keystrokes. A data file called “CUST” exists in the TUTOR account and is used in the tutorial examples. Before we try some ACCESS examples, we should examine the contents of the sample data file called CUST.

The CUST file consists of ten attributes numbered one through ten. The CUST # is the unique item-id, or sort key, automatically assigned to each file item. The CUST file attribute definitions are depicted in figure 11.1.

- COMPANY (1)
- CONTACT (2)
- ADDRESS (3)
- CITY (4)
- STATE (5)
- ZIP (6)
- TELEPHONE (7)
- INV# (8)
- AMT (9)
- DATE1 (10)

Figure 11.1. CUST file attribute definitions

Each attribute can be referenced either by its name or attribute number. The attribute number for the CUST file is shown in parenthesis next to each attribute definition in figure 11.1

ACCESS Syntax

In all of our examples, ACCESS commands will be entered at TCL. In order to use ACCESS **command sentences** must be constructed. The sentences tell the ACCESS processor which data elements and commands you wish to use. The proper syntax for an ACCESS sentence is:

ACCESS verb file name parameters (option

where ACCESS verb is an action oriented word invoking a specific process such as LISTing or SORTing, file name is the name of the data file you wish to use, parameters are selection criteria/modifiers, sort keys, or customized printing instructions, option is (P or LPTR which is used to direct output to the system printer.

ACCESS Error Messages

It is possible that you will encounter error messages while using ACCESS. Error messages usually occur as a result of improper command syntax. Typical error messages are :

```
[10] FILE NAME MISSING
[24] THE WORD 'ABXY' CANNOT BE IDENTIFIED
```

where ABXY is a word which ACCESS does not recognize. If you encounter one of these messages check the command syntax and try reentering the command line.

ACCESS Verbs

Some commonly used ACCESS verbs and their functions are depicted in figure 11.2.

Verb	Function
LIST	Lists the data in a file
SORT	Sorts the data in a file (used with modifiers)
LIST-LABEL SORT-LABEL	Each of these verbs is used to print mailing labels and other specialized lists where it is necessary to print multiple items on a line in close proximity.

Figure 11.2. Commonly used ACCESS verbs

Selection Criteria, Modifiers, Options

These parameters are used to extend the capabilities of ACCESS. Selection criteria, modifiers, and options are always used in conjunction with an ACCESS verb such as LIST or SORT. Figure 11.3 depicts some ACCESS modifiers and their functions.

Modifier	Function
BREAK-ON	Causes output to be broken into categories according to the sorting criteria specified.
BY	Designates the attribute name immediately following as a sort key. Default value of a sort operation is in ascending order.
BY-DSND	Sorts in descending order instead of ascending order.
COL-HDR-SUPP	Suppresses the output of: column headings, time, date, and end of list message. This modifier can be abbreviated as (C).
ID-SUPP	Suppresses the output of item-ids in a listing. This modifier can be abbreviated as (I). It is usually used in conjunction with the LIST-LABEL and SORT-LABEL verbs.

LPTR	Directs output to line printer. Can be abbreviated as (P).
TOTAL	Causes totals to be calculated for the attributes specified.

Figure 11.3. Sampling of ACCESS modifiers and functions.

Using ACCESS Verbs

The LIST Verb

This verb will list data in the file in the order in which it was entered. Using the data in the CUST file, consider the following entries:

```
LIST CUST COMPANY CITY STATE[RETURN]
```

o r

```
LIST CUST 1 4 5[RETURN]
```

The screen will display:

```
PAGE      1                               19:40:46  01 NOV 1985

CUST..... COMPANY..... CITY..... STATE

1006      TRACK AUTOMOTIVE      NEWARK      NJ
1007      MESA TRAVEL AGENCY    HUNTINGTON BEACH  CA
1000      ACME HARDWARE COMPANY IRVINE      CA
1001      NEWTON DEVELOPMENT    HUNTINGTON BEACH  CA
1008      WORD ALBEGRA         CHICAGO      IL
1002      UPTOWN PRINTERS       LOS ANGELES    CA
1009      MY TIMES MAGAZINE     NEWARK      NJ
1003      RITE-WAY DRUGS        CHICAGO      IL
1010      PICK SYSTEMS         IRVINE      CA
1004      LIKE-NU UPHOLSTERY    CHICAGO      IL
1005      A-I APPLIANCES        NEWARK      NJ

11 ITEMS LISTED.
```

Figure 11.4. LIST CUST 1 4 5 command line output

In this example, LIST is the verb, COMPANY CITY STATE are selection criteria. This command line provides us with a list of all the customers in the file, displays their customer number, company name, city, and state where they are located. Note that we could have specified the (P option and directed the output to the system printer. Either command line depicted in figure 11.4 will generate the same output. Using the attribute numbers in place of the attribute names is a kind of ACCESS shorthand and can be a timesaver.

The report is displayed in a columnar format. Since the screens on most terminals, and many printers, can only display or print 80 columns, ACCESS will determine (by looking at how the printer parameters are set via the TERM verb) whether or not a report will be printed in columns. If the system printer does not have a specified line length greater than 80 columns, the report will be displayed horizontally with one item to a line.

The SORT Verb

The SORT verb displays a specified list of items in alphabetically sorted order. SORT is most often used with selection criteria and modifiers to produce specific data lists. If no modifiers or selection criteria are used, the file will be sorted by the item-id (sort key) of the file. In the case of the sample file CUST entering the command line:

```
SORT CUST COMPANY[RETURN]
```

The system will produce a display as depicted in figure 11.5.

PAGE 1

12:23:15 31 DEC 1967

CUST..... COMPANY.....

1000	ACME HARDWARE COMPANY
1001	NEWTON DEVELOPMENT
1002	UPTOWN PRINTERS
1003	RITE-WAY DRUGS
1004	LIKE-NU UPHOLSTERY
1005	A-1 APPLIANCES
1006	TRACK AUTOMOTIVE
1007	MESA TRAVEL AGENCY
1008	WORD ALBEGRA
1009	MY TIMES MAGAZINE
1010	PICK SYSTEMS

11 ITEMS LISTED.

Figure 11.5. SORT CUST COMPANY display

If you type:

SORT CUST BY CITY COMPANY BREAK-ON CITY[RETURN]

This generates the following output:

PAGE 1

19:45:38 01 NOV 1985

CUST..... COMPANY..... CITY.....

1003	RITE-WAY DRUGS	CHICAGO
1004	LIKE-NU UPHOLSTERY	CHICAGO
1008	WORD ALBEGRA	CHICAGO

1001	NEWTON DEVELOPMENT	HUNTINGTON BEACH
1007	MESA TRAVEL AGENCY	HUNTINGTON BEACH

1000	ACME HARDWARE COMPANY	IRVINE
1010	PICK SYSTEMS	IRVINE

1002 UPTOWN PRINTERS

1005	A-1 APPLIANCES	NEWARK
1006	TRACK AUTOMOTIVE	NEWARK
1009	MY TIMES MAGAZINE	NEWARK

11 ITEMS LISTED.

Figure 11.6. SORT CUST BY CITY COMPANY BREAK-ON CITY
output

where SORT is the verb, CUST is the filename BY CITY is the sort parameter, COMPANY and CITY cause the company name and city where the company is located to be displayed. This example provides us with a list of all customers in the file sorted alphabetically by city.

The LIST-LABEL and SORT-LABEL Verbs

These are the verbs provided by ACCESS to print mailing labels and other specialized lists where it is desirable to print more than one data item on a line. LIST-LABEL and SORT-LABEL function identically to the LIST and SORT verbs with the exception that parameters which specify how data is to be arranged on the labels must be entered. These parameters determine the label printing format such as one across, two across, three across, etc. The LIST-LABEL and SORT-LABEL parameters and functions are depicted in figure 11.7.

Parameter	Function
count	Determines the number of labels printed across each page.
rows	Determines the height of each label in rows.
skip	Determines the number of blank lines to skip between labels.
indent	Determines the number of spaces to indent from the left margin.

size	Determines the width of each label in columns.
space	Determines horizontal spacing between labels when more than one label across is printed across a page.
C	When specified C inhibits printing of attributes which are empty. Otherwise, blanks will be printed in place of empty values.

Figure 11.7. LIST-LABEL/SORT-LABEL parameters and functions

Using LIST-LABEL and SORT-LABEL

Printing mailing labels is a common function in a business. In this section we will illustrate the use of LIST-LABEL and SORT-LABEL using the data in the CUST file on the TUTOR account. In order to use these verbs two steps must be followed:

Step 1. Enter the desired ACCESS command line and press [RETURN].

Step 2. When the ? (question mark) prompt appears, specify the display or printing format using the label parameters and press [RETURN].

The format of the LIST-LABEL and SORT-LABEL commands is considered in the following examples.

Example 1

Entering the ACCESS command line:

```
LIST-LABEL CUST COMPANY CITY STATE
(P[RETURN]
```

and entering the parameters:

```
?3,6,0,0,20,3,C[RETURN]
```

produces the output depicted in Figure 11.8.

```

PAGE      1                                20:39:06  05 MAR 1986

1006      1007      1000
TRACK AUTOMOTIVE  MESA TRAVEL AGENCY  ACME HARDWARE COMPAN
NEWARK        HUNTINGTON BEACH  IRVINE
NJ            CA                CA

1001      1008      1002
NEWTON DEVELOPMENT  WORD ALBEGRA  UPTOWN PRINTERS
HUNTINGTON BEACH  CHICAGO    LOS ANGELES
CA                IL        CA

1009      1003      1010
MY TIMES MAGAZINE  RITE-WAY DRUGS  PICK SYSTEMS
NEWARK        CHICAGO    IRVINE
NJ            IL        CA

1004      1005
LIKE-NU UPHOLSTERY  A-1 APPLIANCES
CHICAGO        NEWARK
IL            NJ
  
```

11 ITEMS LISTED.

Figure 11.8. LIST-LABEL COMPANY CITY STATE (P

Example 2

If you type:

```

SORT-LABEL CUST BY CITY COMPANY CITY
(P[RETURN]
  
```

Enter the parameters:

```

?3,6,0,0,20,3,C[RETURN]
  
```

The system will generate the output depicted in figure 11.9.

In examples one and two we generated a display which included the item-ids a page header with time and date, and the number of items listed. When printing mailing labels it is desirable to suppress these features. We will illustrate this feature in example three.

```

PAGE      1                                20:40:47  05 MAR 1986

1003      1004      1008
RITE-WAY DRUGS    LIKE-NU UPHOLSTERY    WORD ALBEGRA
CHICAGO          CHICAGO                CHICAGO

1001      1007      1000
NEWTON DEVELOPMENT MESA TRAVEL AGENCY    ACME HARDWARE COMPAN
HUNTINGTON BEACH  HUNTINGTON BEACH      IRVINE

1010      1002      1005
PICK SYSTEMS      UPTOWN PRINTERS      A-1 APPLIANCES
IRVINE            LOS ANGELES           NEWARK

1006      1009
TRACK AUTOMOTIVE  MY TIMES MAGAZINE
NEWARK           NEWARK

```

11 ITEMS LISTED.

Figure 11.9. SORT-LABEL CUST BY CITY COMPANY CITY (P

Example 3

At TCL enter:

```

LIST-LABEL ID-SUPP CUST COMPANY CITY STATE
(C P[RETURN]

```

Type the parameters:

```

?2,6,0,0,20,3,C[RETURN]

```

This ACCESS command line produces the output depicted in figure 11.10.

This example illustrates the use of the ID-SUPP option which suppresses output of the item-id. The (C option suppresses output of the column headers.

TRACK AUTOMOTIVE
NEWARK
NJ

MESA TRAVEL AGENCY
HUNTINGTON BEACH
CA

ACME HARDWARE COMPAN
IRVINE
CA

NEWTON DEVELOPMENT
HUNTINGTON BEACH
CA

WORD ALBEGRA
CHICAGO
IL

UPTOWN PRINTERS
LOS ANGELES
CA

MY TIMES MAGAZINE
NEWARK
NJ

RITE-WAY DRUGS
CHICAGO
IL

PICK SYSTEMS
IRVINE
CA

LIKE-NU UPHOLSTERY
CHICAGO
IL

A-1 APPLIANCES
NEWARK
NJ

Figure 11.10. LIST-LABEL ID-SUPP CUST COMPANY CITY STATE
(C P.

Using ACCESS with Your Own Database

It is important to remember that ACCESS is a data retrieval language. Creating a database and entering data are distinct processes. PICK is known for its database **capabilities**. PICK provides a method for creating and entering data (via the EDITOR) and a powerful report generator (ACCESS). What is not provided is a user interface which ties the two functions together in an easy to understand way. In the next chapter we will discuss a program called INSTANT WIZARD. This program, now supplied as standard equipment with every copy of PICK for the PC/XT, provides the user interface and transforms a PC/XT into a powerful database management system. The tutorial supplied with the WIZARD program will guide you step by step from building a database through constructing and formatting reports. If you would like to study the intricacies of the ACCESS processor on your own, we refer you to the PICK User Reference Manual. If you use the WIZARD program, practice building databases and creating reports using the interactive tutorial.

Storing ACCESS Command Lines

As you have seen, some ACCESS command lines can be quite lengthy. For this reason, you may want to consider using a PROC to store these frequently used command lines and printing parameters. There is no direct means provided by PICK for storing command lines other than placing them in a PROC. The INSTANT WIZARD program provides the easiest methods to construct and maintain your database. As we will see in chapter twelve, once a report has been constructed using INSTANT WIZARD, it can easily be recalled from a menu. No complex command lines or exit to TCL is required.

Summary

In this chapter we have discussed the basic operation of the PICK ACCESS processor. ACCESS is an English-like language which allows you to retrieve data and write reports. We have discussed only the simplest ACCESS commands. As we will see in our discussion

of application software in the next chapter, most of the more complicated commands can be executed without learning them from within application software built around the PICK file system.

In chapter twelve, we will discuss some popular PICK application software which takes full advantage of the PICK file structure and ACCESS.

12

A User's Guide to PICK Software

Introduction

A computer is likely to be of little value without good application software. Application software provides the computer with instructions allowing it to perform tasks such as word processing, accounting, mailing list management, etc. Much of the software you need will depend upon your applications: video rental management for video retailers, medical/dental practice management for members of the health care field, printer estimating for printers, and so on.

Chances are that if you are reading this book, you have determined that a microcomputer, like the PC/XT, might suit your needs. You might also have an idea about the type of software you require. In this chapter, we will examine the general categories of software. Where possible some subjective comments from the author will be made about the strengths and weaknesses of specific software packages.

As with other microcomputer operating systems there exists a wide variety of PICK software for almost any application. Since the objective of this book is not primarily to provide evaluations of software, discussions of each software category are brief. Where possible the comments on the software are based on actual hands-on use.

First we will look at some facts you should know before examining specific software packages. Next, we will discuss the following software categories:

- Database managers
- Word processors
- Spreadsheets
- Office automation
- Accounting

The chapter concludes with a brief discussion of vertical market software and PICK software directories.

PICK and Application Software

In this section we will discuss some features of PICK which illustrate how it is ideal for use in business.

Reentrancy

The feature of PICK called **reentrancy** speeds processing time and minimizes memory requirements. The reentrancy feature is important because, in most instances, only one copy of the program being executed must reside in RAM. This speeds data entry because microprocessor time is not wasted managing memory. For example, if there are two users simultaneously using the same word processing program, they share the program in memory.

Task Prioritization

Another feature which speeds processing time is **task prioritization**. PICK automatically allocates microprocessor time depending upon the task being executed. Interactive data entry tasks are allocated the greatest time slice while batch processing tasks such as posting

invoices, printing reports, etc. are allocated the least amount of time. Prioritization minimizes the slowdown of the system as more users log onto the system.

Upward Compatibility

PICK users have, from the beginning, enjoyed upward compatibility. This means that if you start out with a PC/XT and after a time you determine that you need a larger system, everything is easily transferred to the new system. Even a FILE-SAVE can be restored on another PICK system assuming that the media formats are compatible. If you graduate to a new larger and faster system it operates exactly as before. No retraining or reprogramming expense is required.

Note: It is also interesting that downward compatibility is an advantage too. This means that any business running PICK on a mini computer can, with a little help from a programmer, bridge the gap between the mini computer and the PC/XT.

What Types of Businesses Use PICK

Studies have shown that the majority of the business using PICK are in the product distribution business. PICK and its data file structure are particularly suited to inventory control, accounts receivable, order management and many other tasks common to the wholesale business. As we will see, there are the more common types of software such as word processing and spreadsheets too.

Why You Probably Haven't Heard of These Software Packages

The PICK operating system was designed to be a multi-user operating system. Most of the "off the shelf" software packages that are heavily advertised and promoted were written for use with Apple DOS, CP/M, or MS-DOS, all of which are single-user operating systems. The companies that market such packages feel that it would be impractical to take their existing software and make it PICK compatible. To take advantage of the power of PICK it would be best to

start from scratch and write a new package. For this reason, new companies have formed solely for the purpose of writing and developing software for the PICK environment. Some software vendors make demonstration packages available at a nominal charge in order to allow a user to examine the capability of the software.

Documentation

Unfortunately, some of the documentation supplied with PICK software leaves much to be desired. Some documentation is definitely not designed for novices and as such you should consider paying your dealer for training on specific applications. The WIZARD program is a noteworthy exception to this rule. The documentation for WIZARD can be used on-line, including a series of how to get started interactive tutorials, or printed out. WIZARD can easily be used by someone who has never operated a computer. Other software, such as the CompuSheet + spreadsheet software has on-line help available by pressing ?.

Disk Storage Limitations

A standard PC/XT is limited to 10 MB of storage. In order to have enough room for your data files and growth we recommend that you limit yourself to installing only those software packages you will regularly use. A PICK system will operate more efficiently if there is an abundance of contiguous disk space. When you have decided upon the software, it is best to install it on a newly initialized system. This will create a maximum of contiguous disk space.

Three Types of Software Everyone Can Use

Whether you own your own business, are setting up an office in your home, whether you are a retailer, wholesaler, or publisher, there are three programs you will find invaluable. These programs are:

- A database manager/interface
- A word processor
- A spreadsheet

We will discuss each software category in this section.

Database Managers (Interfaces)

Database management software for PICK permits users to design a customized data entry program tailored to specific needs. Screens can be designed to simulate any form currently being used by a business. After data is entered and stored it is possible to sort the data and generate customized reports listing the data previously entered using ACCESS. Database managers should not require programming knowledge but do require easy to follow and comprehend documentation to be used successfully. The best database managers use tutorials provided with sample databases to guide the new user through the steps involved in designing a database.

When a new database is constructed it is composed of three parts: files, items, and attributes. First the file is defined and created. In PICK, it is not necessary to precisely specify the number of items contained in the file since the file will vary in size as items are added and deleted from the file. Finally the attributes of each item are defined. The attributes are the data comprising each item.

For example, if you were creating a database for mailing labels you might create a file called "mail." Suppose you want the file to store a hundred items. The attributes might be:

```
Name Address City State Zip Code
```

After data is entered, use ACCESS to sort the file alphabetically by name or by zip code. When sorting is completed, a report can be designed. The report format might resemble the following:

```
Name  
Address  
City State Zip Code
```

Because we are using PICK on a hard disk based system, there are few limitations in constructing a database. Figure 12.1 depicts a comparison of the PICK system database capabilities and the popular dBASE III software package. Note that for this description we use the more conventional terminology of fields (attributes) and records (items).

File Characteristics	PICK	DbaseIII
Records per file	unlimited	unlimited
Files per database	unlimited	unlimited
Fields per database	unlimited	unlimited
Fields per record	32,267	128
Characters per record	32,267	4000
Values per field	32,267	1
Data types (money,date,SS#)	unlimited	5
Variable length field types	yes	no
Add new fields without restructuring the database	yes	no
Files expand/contract automatically	yes	no
Report Writer Features:		
Fields per sort	unlimited	100 characters total in fields
Unlimited breaks and subtotals	yes	no
Files opened at any one time	unlimited	10
Special Features:		
Print spooler	yes	no
Levels of password security	4	0
Multiple Users (PC/XT)	3	only 1
Mini/mainframe compatible	yes	no
conversion of files required for upgrades to new software release	no	yes
Copy protection	no	yes

Figure 12.1. PICK/dBASE III comparison

For the person who has used the DbaseIII program, its limitations may be all too familiar. For the benefit of those not familiar with DbaseIII, we will now discuss some of the important advantages of PICK relevant to its database characteristics.

File Characteristics

In PICK, the primary limitation on files is the 32,267 character limit on item size. Files automatically grow in size as new items and attributes are added within the confines of the available disk space. PICK also allows multi-valued fields, variable length data in fields and dynamic file sizing. Since most businesses are concerned about saving time and money, users will appreciate the fact that files do not have to be reorganized or reconfigured when new fields are added to the database. New fields can be added easily at any time.

Report Writer Features

The ACCESS processor is the key to generating any report from the data stored in a PICK system. Data from many files may be merged into a single report without creating additional files or sorting indices. Reports can be created using information contained in existing files. New values can be calculated from the existing ones without creating a new file.

Other Special Features

Perhaps the most important benefits of PICK over conventional database software are the following:

- No special floppy disks or "key disks" are required to run PICK software
- The print spooler allows jobs to be stored on the hard disk and printed when the system printer is unused.
- PICK is a multi-user operating system which means that users on the system can share a common database using multiple workstations
- PICK is the only database system available which runs on microcomputers, minicomputers, and mainframe computers and no reprogramming or reentering of data is required after upgrading to a new version of the system.

A Word about Growing Files

Data files will expand and contract as needed. It is your responsibility to keep track of disk space availability. You wouldn't want to be in the middle of an important data entry section only to see a message such as "Disk Full" displayed.

The WIZARD Program

As we have just discussed, database management is the hub of the PICK system. For all of its power, one characteristic PICK shares with Dbase III is that it is not designed for use by novices without training. Advocates of PICK tout the database facilities of PICK usually without regard for a user's level of expertise. Fortunately there exists a software package called WIZARD which allows anyone who has used a computer to begin using PICK without expensive and time consuming training. WIZARD consists of three separate modules:

- INSTANT WIZARD
- CUSTOM WIZARD
- ADVANCED WIZARD

INSTANT WIZARD is an application development system for use by PICK novices. It creates data entry screens, reports, forms, menus, and complete documentation. INSTANT WIZARD includes easy to follow interactive tutorials, and on-line help.

CUSTOM WIZARD makes it possible for a non-programmer to create sophisticated applications. This is easily done by customizing the data entry screens, reports, forms, menus, and documentation that were previously designed with INSTANT WIZARD. Some additional features include:

- Calculations
 - Titles
 - Multiple files
 - Multi-valued fields
 - Split screens
 - Enhanced report writer
 - Customized user manuals
-

ADVANCED WIZARD is strictly for programmers. It allows a programmer to add modules written in the PICK BASIC programming language to an existing WIZARD program. This feature gives WIZARD tremendous flexibility while increasing a programmer's productivity by a factor of 10 to 30 times.

This author believes WIZARD to be an indispensable working tool in any PICK system. The people at PICK systems obviously agree since instant WIZARD is now supplied as standard equipment with every copy of PICK for the PC/XT. WIZARD is a program designed to facilitate the use of the PICK database system and is designed for use in a multi-user environment. No special programming experience is required to begin using the software. Help is available at each step including examples of possible data entries or defaults. Help can be requested almost anytime by entering ? or ?? in response to the on-screen prompts. It is also possible to "back up" a previous step or out of a program altogether using only a few keystrokes. INSTANT WIZARD includes interactive on-screen tutorials which demonstrate how the system works and how to construct a database and write reports.

WIZARD Installation

INSTANT WIZARD for the PC/XT is supplied on four floppy diskettes. Complete installation instructions are provided as well as directions on terminal configuration. The program itself requires a minimum of 2200 contiguous frames (approximately 1.2 MB of disk space) in order to be installed. As installation of the program proceeds you will sometimes see two files restored with the same name. This is because files have a dictionary and a data portion associated with them. Installation time is approximately thirty minutes.

The WIZARD programs are installed on a new account called WIZARD. When the software is installed, from TCL you can type:

```
LOGTO WIZARD[RETURN]
```

and the WIZARD main menu will be displayed. The WIZARD main menu is depicted in figure 12.2.

```
=====
== == == W I Z A R D I M A I N M E N U == == ==
=====

1. How to get          6. Generate your
   started,            documentation

2. Tutorials,          7. Run your Pro-
                        gram,

3. Instant screens/    8. Browse a report,
   reports,

4. Custom screens/     9. Create/maintain
   reports              files,

5. Design your menu, 10. Maintain Program
                        definitions

Select one of the above, or ESC, or 'OFF' :
```

Figure 12.2. The WIZARD main menu.

WIZARD Menu Functions

The menu depicted in figure 12.2 is the master controlling menu for the WIZARD programs. Selecting any option will take you to another menu of options. Pressing [RETURN] is required to complete an entry. Pressing a single key is not sufficient. Let's briefly discuss the function of each menu option.

Option 1 How to Get Started.

Selecting this option takes you to another menu called the master help menu. This menu is depicted in figure 12.3.

WIZARD II MASTER HELP MENU

- | | |
|------------------------------|-----------------------------------|
| 1. Introduction | 6. Generate your documentation |
| 2. Tutorials. | 7. Run your program. |
| 3. Instant screens/ reports. | 8. Browse a report. |
| 4. Custom screens/ reports | 9. Create/maintain files. |
| 5. Design your menu. | 10. Maintain Program definitions. |
| 11. Utilities | |

Enter a selection number or ESC to exit :

Figure 12.3. WIZARD master help menu

This menu provides you with on screen help explaining each aspect of the program. When you have been helped, you can press ESC[RETURN] to return to the main menu. One additional selection on this menu is the utilities option. This option will tutor you on how to install WIZARD on another account and how to configure terminals attached to other lines on the system.

Option 2 Tutorials

This selection demonstrates how WIZARD is used by guiding you through a number of simple examples. There are tutorials for all aspects of program operation including creating data files and menu design.

Option 3 Instant Screens and Reports

This selection allows you to design simple data entry screens, forms-type reports, and columnar reports.

Option 4 Custom Screens and Reports

Selecting this option requires that CUSTOM WIZARD be installed on your system. CUSTOM WIZARD allows you to customize a screen or report you have previously designed with INSTANT WIZARD. CUSTOM WIZARD allows a screen design to match almost any form you currently use in your office.

Option 5 Design Your Menu

This option is used to create simple menus in the PROC language. The MASTER and UTILITY PROCs discussed in chapter ten were created using this option.

Option 6 Generate Your Documentation

Complete documentation for WIZARD programs can be printed when this option is selected. In addition, if you have created an application using WIZARD, you can print a user's manual using this choice. This user's manual will include a printout of any menus or screen displays as well as an explanation of how data is to be entered.

Option 7 Run Your Program

Select this option when you want to run a WIZARD generated program or menu. This helps to isolate users from the operating system. Enter the name of the program or menu press [RETURN] and the program is automatically executed.

Option 8 Browse a Report

Once a report has been created, this handy option allows you to scroll backwards and forwards through the report on the terminal. Since most terminals are not wider than 80 columns and many reports are 132 columns or wider, you can even scroll from left to right to view the data without using a printer.

Option 9 Create and Maintain Files

Use this option to create new data files or display a list of already existing files.

Option 10 Maintain Program Definitions

This selection allows you to:

- Display a sorted list of all program definitions
- Allow program definitions to be used on other accounts
- Delete program definitions

Comments

The WIZARD program should be an integral part of every PICK installation. In fact, it is one of the easiest to use software packages, for any microcomputer. In addition to its database capabilities, it can help in the writing of PROCS for a menu driven PICK system. The software includes complete documentation, which can be printed at your convenience, and an on-line tutorial which takes you step-by-step through the basics of constructing a database and writing reports. CUSTOM WIZARD is recommended if you want to tailor the screens or reports to closely match your current business forms. ADVANCED WIZARD, is recommended for programmers. ADVANCED WIZARD makes it possible to create very complex applications much faster and bug-free than conventional programming methods. Contact Automatic Programming directly for a list of application packages written with CUSTOM or ADVANCED WIZARD.

Program:	WIZARD
Manufacturer:	Automatic Programming 32 Esplanade Irvine, CA 92715
Contact:	Ken Simms, 714/786-1111
User support available:	through WIZARD dealers or by telephone directly from Automatic Programming at user's expense.

Prices:	INSTANT WIZARD...free with PICK for the PC/XT and AT CUSTOM WIZARD... \$700.00 ADVANCED WIZARD...\$700.00 Pricing for computers other than the PC/XT is available upon request.
Distribution Medium:	5-1/4 inch floppy disk for the PC/XT.
Special Requirements:	Any PICK based system.

Word Processors

Word processing software allows you to type mailing lists, letters and legal documents, forms or any document that can be typewritten. The advantages of using a word processor are:

- You can concentrate on the content of the document and worry about the format later.
- Errors are corrected on-screen.
- When the document is error-free, it can be saved on disk for future use.
- The document can be formatted and printed to your specifications at any time.

The primary benefit of word processing software is that once its use is mastered, it is a real time saver. We will examine two of the currently available word processing packages for PICK: JET and CYCLONE III. Although the purpose of the two programs is identical, the operation and features of each differ.

Note: word processing software may sometimes require lengthy disk access times as well as a great deal of microprocessor time for “re-painting” the screen. In order to avoid excessive slowdown of the system, it is recommended that no more than two users perform word processing tasks simultaneously on a PC/XT system.

The JET Word Processing System

Version 1.3D of the JET Word Processing system was made available to us for evaluation. This version is referred to as "The Works." Some of the features of "The Works" are:

- help during editing by entering a ?
- ability to print mailing labels
- create forms and forms fill-in capability
- support of over 40 different terminal types
- TAB and RETURN key output is visible on the screen facilitate editing
- user can "cut out" a section of text and then "paste" it into a different file or different account
- support of "hard copy" function for terminals with printer port
- editing for documents larger than 32,000 characters is supported
- automatic utilization of the PICK spooler
- spell checking during text entry

JET Installation

Clear and easy to follow installation instructions are provided with JET via the PROC menu depicted in figure 12.4. The software is supplied on three floppy diskettes two of which contain the software while the third contains the spelling dictionary. Installation and configuration of the programs requires about 30 minutes. The spelling dictionary occupies 1400 frames of disk space so if space is at a premium on your system, you may decide not to install the dictionary. There is no menu selection provided for installing the software on other accounts.

Using JET

Unless the software has been installed on other accounts, you must be logged onto SYSPROG in order to run the JET programs. Entering:

```
JET-WP[RETURN]
```

at TCL takes you to the menu depicted in figure 12.4.

```
JET system maintenance menu

1. Read the JET software from floppy
   "ONE"                                necessary!

2. Read CRT drivers/menus from floppy
   "TWO"                                optional

3. Read spelling dictionary from floppy
   "THREE"                              optional

4. Load JET code to ABS area            necessary

5. Verify JET code in ABS area           optional

6. Assign monitor or CRT type necessary!

7. Go to the JET application menu

Please choose a number from 1 to 7. . . [ ]
```

Figure 12.4. JET software menu

Entry to the software is accomplished by selecting option seven from this menu. Pressing [RETURN] instead of selecting an option will return you to TCL.

Note: Upon entering the program you must remember to press the caps lock key to enter both upper and lower case letters. Press the caps lock key upon exiting the program so that TCL commands can be entered.

JET treats a user account as a file cabinet. Before the software can be used a “file drawer” must already exist or be added to the cabinet. After a drawer is added or selected each document created is analogous to a file folder in the drawer. When a file folder is created (given a document name) or selected the word processor is executed. Each time the word processor is executed you are asked to type the name of the drawer where the file folders are stored. After

selecting a drawer the screen displays file folder numbers with the document names previously assigned to them. The system will then display information about the document such as author's name, when the document was created, when it was last edited, and how many characters are in the document.

You are then given three choices:

1 = Revise 2 = Print 3 = EXIT

Revising a document allows you to enter the word processor. Printing a document will allow you to print the document to the spooler, terminal screen or printer attached to the screen if available. Choosing to EXIT will not return you to TCL or the JET menu. An OFF command is automatically executed.

JET Documentation

Learning any new software package always takes time. At the time of review, version 1.3D of JET was a brand new version of the software. The features we have described, allowing the system to be menu driven, are new. The documentation supplied with the software did not cover these new features. The documentation supplied with version 1.3D covers only the basic features of program operation. A task oriented tutorial would be an asset to the program for novice users. In addition, it would be helpful if sample documents were provided as part of the tutorial. The existing JET documentation is 59 pages of single-spaced type-written pages. There is an index but no table of contents.

Version 1.4 of the software was due to be released in January of 1986. If this new version of JET does not have improved documentation we recommend taking a training class from a JET dealer.

JET Configuration and Modes of Operation

It is the job of the system manager to configure JET to work with your system. If improperly configured, lines of text will not be displayed on the screen properly. The program is designed to work with the most popular terminals such as Televideo, Wyse, the PC/

XT monochrome and color monitors, etc. If you are using any terminal other than the ones supported contact your dealer or JET. Function key support for the XT keyboard is not presently available.

The program operates in two modes: a command mode and edit mode. In command mode certain keys are pressed to execute a particular function such as F to save a file, ? to display the help screen, U to form the spelling checker on or off, etc. Text is entered in the edit mode. The ESC key toggles between the edit and command modes. The editing portion of the program is separate from the portion of the program that executes commands. It is possible to insert, use tabs, backspace and correct errors, perform deletions, move across, up or down through the document by toggling between the edit and command modes.

Comments

JET is a very flexible word processing program which can transform a PICK system into a powerful word processing tool. The main weaknesses of the program are its documentation and the amount of time required for novices to learn to use the program. Taking a training class from a JET dealer is recommended.

Some features that would make JET easier to use are:

- incorporate a menu selection to allow the program to be used on more than one account
- be able to select a new folder to edit without being logged off the system
- be able to use the PC/XT function keys and possibly reconfigure the JET commands to resemble popular MS-DOS word processors
- provide on-screen help by entering a ? any time during program execution
- display a summary of commands on-screen during editing
- provide a means of deleting the entire program and its associated files in the event of a serious problem via a menu selection

Note: Some of these features may already be implemented in version 1.4 of JET.

Program:	JET version 1.3D
Manufacturer:	JET Software, 17632 Irvine Blvd. Suite T Tustin, California 92680
Contact:	John Treankler or Sherry Elrod
User support available:	by telephone at user's expense or from JET dealer.
Price:	\$299.00
Distribution medium:	5-1/4 inch floppy disk
Special requirements:	any PICK based system

Cyclone III Word Processing

This program is sold in conjunction with the Cyclone III financial modeling package which we will discuss later in this chapter. One advantage of using a word processing program and a spreadsheet program written by a single vendor is that they usually share the same command sequences such as pressing ESC-H for help or ESC-X to quit, etc. This saves you time when learning to use the software. Some of the features of Cyclone III are:

- PROC menu-driven operation
- single mode operation with on-screen help
- vertical and horizontal screen scrolling for documents wider than 80 columns
- built-in math functions
- on-line spelling checker
- automatic control of the PICK spooler

Note: if you are using version 1.3 of PICK, you must send your operating system diskettes to Lazarov software for modification before using either the Cyclone word processor or spreadsheet. This modification may not be necessary when version 2.0 of PICK becomes available.

Installation

Installation of Cyclone is relatively simple. In addition, the program occupies less disk space, including the spelling dictionary, than JET. The programs are installed under the Cyclone account which is automatically created by the installation procedures. In order to use the software you must log on to the Cyclone account. Logging on to the Cyclone account causes the menu depicted in figure 12.5 to be displayed:

```
CYCLONE II SELECTION SCREEN
current time_date

01) WORD PROCESSING
02) LISTING OF ALL DOCUMENTS ON FILE
03) SPREADSHEET PROCESSING
04) LISTING OF ALL SPREADSHEETS ON FILE
05) INSTALL SYSTEM
06) INITIALIZE FUNCTION KEYS

'EX' TO EXIT

ENTER SELECTION>
```

Figure 12.5. Cyclone main menu

Using Cyclone III

Perhaps the most important feature of Cyclone is that it is easy to learn. Anyone who has previous experience with other word processing software can use the basic features of the program such as entering text and moving around the document, but in order to use some of the other features, referring to the manual is necessary. Cyclone is capable of recognizing function keys to execute some of its commands. However, the Cyclone menu selection which configures function keys, is not designed for the PC/XT. Cyclone, therefore, uses a combination of ESC sequences and CTRL key sequences to execute commands within the program. figure 12.6 depicts a few of the Cyclone commands.

Key Sequence	Function
CTRL-D	delete the character under the cursor
CTRL-I	insert a character under the cursor
ESC-F	file the document
ESC-X	exit the Cyclone program
ESC-DL	delete one line
ESC-DW	delete current word
CTRL-T	move cursor to next tab stop
ESC-T	define a tab stop at current cursor position
ESC-O	print the document

Figure 12.6. Some of the Cyclone word processing commands

Sample documents are provided with the program. This is helpful for observing how lines of text can be typed and formatted and for practicing moving around the document. As with most word processing software, print formatting commands can be embedded in the text to provide underlining, boldface type, and superscript or subscript type, provided these attributes are supported by your printer.

In Cyclone III, as an edited document grows in size, the program automatically writes it to disk. The size of a single document is limited to 32,000 characters. This should not be a serious problem because working with files larger than 15,000 characters may slow down system operation considerably. You can use the insert command to string a series of smaller documents together.

Printing a Document

Since the PICK print spooler is important for sharing a printer in a multi-user environment, it is equally important that the software use the spooler correctly. When using Cyclone, no PICK commands are required to use the spooler. However, the program does require that the software be properly configured for the type of printer you are using. The documentation supplies brief instruction on how to configure the program for specific printers.

Note: All printing specifications including margins, spacing, tabs and text justification must be embedded in the document. When the ESC-O command is executed to print a document it is immediately sent to the spooler. No facility is provided for printing selected pages or printing multiple copies.

Comments

Like its competitor, JET, Cyclone III lacks well structured and organized documentation and a task oriented tutorial. There is no section in the documentation devoted to trouble-shooting the program or handling errors. Familiarity with other word processing packages is helpful when getting started even though Cyclone shares virtually no features or command structure in common with the mainstream MS-DOS word processors. We recommend the package be installed and configured to match your system by a Cyclone dealer. As with JET, we recommend that a few hours of training be purchased from your dealer. The major advantage to this package lies in the fact that it is supplied with a companion spreadsheet. This will save some training time because the spreadsheet and word processor have identical command structures.

Program:	Cyclone III Word Processing
Manufacturer:	Lazarov Software Systems 14201 E. 4th Ave. Suite 344 Aurora, CO 80011 303/363-7204
User support available:	Through Cyclone dealers or directly from Lazarov by telephone at user's expense
Price:	\$495 includes word processor and spreadsheet
Contact:	Steve Lazarov/Mike Sutkowski

Spreadsheets

Much of the current popularity of personal computers can be attributed to spreadsheet programs. It is not surprising that spreadsheet programs have been implemented for the rapidly growing multi-user

PICK market. Calculations possible with paper, pencil, a calculator, and a huge eraser can be accomplished with a spreadsheet. A spreadsheet program allows you to create a worksheet set up in a row/column format to your specifications. Numbers, labels, or formulas can be entered. The program can even be instructed to duplicate previous entries to save you work. Once the worksheet is set up to your specifications, it can be saved on disk for re-use or it can be printed.

Spreadsheet applications include ledgers, loan amortization, budget planning, sales forecasting, sales analysis, break even analysis, and pricing strategies. We will examine two popular PICK spreadsheets: Cyclone III Financial Modeling and CompuSheet +.

Cyclone III and CompuSheet +

These programs will be discussed together as they are quite similar. Both are both fine programs which accomplish the same goals but they differ slightly in features, screen displays, and ease of operation. The programs are designed along the lines of popular MS-DOS spreadsheet programs such as Supercalc or Multiplan. CompuSheet + strongly resembles Multiplan in its design. Although some on-screen help is available for Cyclone, CompuSheet + provides specific help messages about any command being executed whenever the ? key is pressed. Both program's basic features are quite similar. They are listed below:

- Work sheet size is memory independent.
- Easy and fast insertion of rows and columns
- Retrieve data from any files in your database
- Interface with graphics packages is possible.
- In-cell editor makes it easy to make changes without retyping.
- Spreadsheets may be merged with each other.
- User-defined calculation order is possible.
- A wide variety of operators and functions are available including:
ABS, AVG, CHOOSE, COUNT, IF THEN, ELSE, AND,
OR, INT, LOOKUP, MAX, MIN, PI, SUM, SQR, SIN,
COS, TAN, ARCTAN, LOG, EXP, FAC, NPV, +, -,
*, /, ^ (exponentiation), <, ≤, >, ≥, < >.
- Spreadsheets can be interfaced to a word processor.

Both programs include enough features to satisfy the needs of the most demanding spreadsheet user.

Unique Features

One unique feature of Cyclone is the inclusion of templates. Templates are pre-constructed spreadsheets ready for data to be entered. They are particularly helpful for the uninitiated spreadsheet user because they provide examples from which new spreadsheet ideas may be constructed. They can be used to input sample data to get a better idea of how spreadsheet programs work.

Since Cyclone keeps video attributes such as reverse video, half intensity, etc. to a minimum, it requires no special terminal configuration program. Because CompuSheet + makes extensive use of video attributes, including support of the PC/XT color monitor, it is supplied with various terminal configuration utilities. Comprehensive documentation and menus are supplied with CompuSheet + to assist you in configuring your terminal.

Note: When using CompuSheet +, if it appears that color is not being displayed on your screen try adjusting the brightness or contrast controls before changing the configuration files. During installation CompuSheet + automatically configures the system for either the monochrome or color monitor.

CompuSheet + is designed to integrate with ACCU/PLOT II, a graphics package available on many of the larger PICK machines. ACCU/PLOT II will allow you to graph data directly from your spreadsheet, producing bar charts, line charts, scatter diagrams and pie charts.

Comments

Both spreadsheet programs perform adequately, and, in terms of features alone, it would be difficult to recommend one program over the other. The documentation provided in Cyclone is adequate but more material could be provided for the hands-on tutorial approach. The templates provided by Cyclone are certainly a step in that direction.

When it comes to ease of operation, installation, and documentation CompuSheet + is in a class with the WIZARD database package. The documentation is presented in a typeset three ring binder complete with many examples, screen displays, and a comprehensive index. The program is menu (PROC) driven and help can be obtained at time by typing a ?. There is a utility menu which:

- Allows terminal characteristics to be defined
- Allows for creation of new spreadsheet files
- Allows you CompuSheet + to be used on another account
- Permits the account containing the CompuSheet files to be deleted

Interactive Systems should develop a companion word processing package to CompuSheet +. If it were as easy to use and install as CompuSheet + it would become very popular.

Program:	Cyclone III Financial Modeling
Manufacturer:	Lazarov Software Systems, Inc. 14201 E. 4th Ave. Suite 344 Aurora, CO 80011 303/363-7204
Contact:	Mike Sutkowski/Steve Lazarov
User support available:	from Cyclone dealers or by telephone at user's expense.
Price:	\$495.00 includes the spreadsheet and word processor.
Distribution Medium:	3 5-1/4 inch floppy disks
Special Requirements:	any PICK based system. PICK diskettes may have to be modified by Lazarov to the 1024 ABS format.
Program:	CompuSheet +

Manufacturer:	Interactive Systems 635 Southpoint Ct. Suite # 110 Colorado Springs, CO 80906 303/579-6800
Contact:	Susan Eubanks, Carol Lovercio or John Brandon
User support available:	through dealers or from manufacturer by telephone at user's expense
Price:	\$295 for PC/XT version, \$495 for PC-AT version \$795 for all other PICK based systems
Distribution Medium:	2 5-1/4 inch floppy disks
Special Requirements:	Disk space required is 952 frames for the PC/XT and PC- AT versions.

Office Automation Software: TotalWare

The seven module TotalWare software transforms a PC/XT into an office manager's assistant. The seven TotalWare modules and their functions are:

- CommWare which provides an interface between a PC/XT and Western Union's Easylink.
- EasyWare which provides for integrated electronic mail, action lists, and automatic scheduling. On-line help is also available for this function. Menus are displayed in "window" format.
- JET.Menu which provides an easy to understand interface to the JET word processing program.
- MarketWare provides client/customer tracking information.
- PlanWare which is used for appointment scheduling.
- ScheduleWare which is used to graphically display schedules created by PlanWare.
- TeleWare which is a telephone message manager.

TotalWare Installation

The seven module TotalWare system is supplied on seven 5 1/4 inch floppy diskettes. Installation time required is forty-five minutes to an hour. Complete instructions are provided for installing the software including directions for installing the software on different accounts.

Using TotalWare

Much thought and care has obviously gone into designing the software. More importantly, the documentation supplied for each module is written in an easy to follow tutorial format. There are numerous screen displays as well as a complete table of contents. The manuals are written in a task-oriented style. The simple descriptions of each module we have provided cannot begin to describe the convenience provided by this software in managing office information. We recommend you get a demonstration of this software from a TotalWare dealer in order to best evaluate its true capabilities.

A TotalWare module of great interest is the JET word processing interface. This interface provides an easily mastered link to unlocking the power of the JET word processor. The TotalWare JET.MENU is depicted in Figure 12.7.

```

                                JET
MAIN                               WORD PROCESSOR SYSTEM                               HH:MM:SS
date
*****

JET ROUTINES                                FILE MAINTENANCE
CODE      DESCRIPTION                        CODE      DESCRIPTION
CD  CREATE DOCUMENT                         CF  CREATE FILE
CP  CREATE PROGRAM                         DF  DELETE FILE

ED  EDIT DOCUMENT                         DI  DELETE ITEM
EP  EDIT PROGRAM                         RI  RENAME ITEM

PR  PRINT DOCUMENT/PROGRAM                CI  COPY ITEM (SAME FILE)
LI  LIST DOCUMENTS/PROGRAMS              MI  MOVE ITEM (DIFFERENT FILE)

EX  EXIT TO TCL                           SP  SET-UP PRINTERS
LO  LOG OFF                              CA  CHANGE PRINTER ASSIGNMT

PLEASE ENTER DESIRED CODE FROM ABOVE

```

Figure 12.7. TotalWare JET.MENU

Program:	TotalWare
Manufacturer:	Automation Technology 873 Township Line Road P.O. Box 7137 Elkins Park, PA 19117 215/885-4880
Contact:	Stephan Setzman
Disk Storage	PlanWare 450 frames
Requirements:	ScheduleWare 450 frames TeleWare 350 frames EasyWare 500 frames MarketWare 800 frames CommWare 650 frames Jet.Menu 450 frames
User support available:	from the installing dealer. Where no dealer is locally available, Automation Tech- nology will support the user by telephone at the user's expense.
Pricing for PC class systems:	PlanWare \$550 TeleWare \$995 ScheduleWare \$479...All three \$1197 EasyWare \$1197 MarketWare \$1797...EasyWare and Mar- ketWare \$2397 Jet.Menu \$210 CommWare \$597 Individual modules may be purchased separately. Prices quoted upon request.
Distribution	5-1/4 inch floppy disk
Medium:	
Special	any PICK based system
Requirements:	

General Accounting Software

Two options are available to those considering computerizing their business's accounting functions:

- construct your own accounting package using a database program or application generator such as WIZARD
- use one of the readily available PICK-based accounting software packages

The primary drawback to constructing your own package is time. Writing programs is time consuming and your time is probably used more profitably elsewhere. The accounting software readily available for PICK varies widely in its flexibility. Some packages adhere to a strict set of guidelines which you must follow in order to use the software while others are flexible enough to accommodate your needs.

Business accounting software can be categorized as follows:

- Accounts receivable
- Accounts payable
- Order entry
- Inventory control
- Sales analysis
- Payroll
- General ledger
- Purchase order tracking
- Job costing

Accounting software can be packaged as separate modules (accounts receivable is one module). Since PICK systems are hard disk based systems with plentiful storage space, the most basic accounting systems sold usually have a minimum of four of the above modules. In fact, the primary benefit of an integrated accounting system is that all modules interact with each other. Data need be entered only once.

There are a number of general accounting systems available for the PICK environment. Consult the PICK HITS and the IDBMA software guide for a listing of general accounting software vendors.

Vertical Market Software

Vertical market software addresses problems encountered by specific types of businesses such as accountants, lawyers, printers, oil well drillers, chiropractors, audio/video retailers and so on. PICK offers many vertical market packages some of which are depicted in figure 12.8.

Legal Timekeeping & Billing
Church Donation Tracking System
Commercial & Residential Rental Property Management
Fund Raising/Contributor Tracking
Travel Agency Management
Hotel/Resort/Condo Management
Book Store Management
Insurance Agency Management
Retail Pharmacy and Retail Nursing Home
Library Management
Non-profit Funds Manager
Vending Machine Management
Trust Fund Management
Petroleum Exploration and Accounting
Radio Station Management
Video Rental Software

Figure 12.8. Examples of PICK vertical market software

Where to Find PICK Software

Software Directories

The multi-user computer market is quite different from the personal computer market. The cost of the least sophisticated PICK system with software is over \$5000.00. Not only are most computer stores unable to deal with such high price points, their in-store sales personnel also have little knowledge or experience with such highly sophisticated and powerful systems.

The best places to begin looking for software are two publications. Let's examine each and determine how they might help you.

IDBMA Application Software Directory

This publication is not strictly a software directory. It contains:

- A list of over 620 individual application software packages in 93 classifications. 173 companies are listed and cross referenced.

- Names and addresses of 37 local PICK user groups around the world.
- A list of 58 consultants for the PICK environment.
- International information about PICK.

Each vendor listing contains the name, address, and phone number as well as a listing of the type of software being offered for sale. The directory provides an application cross reference which lists each vendor by category of software. If you are looking for something specific this cross reference saves you the time of looking through the entire directory. The publication is available directly from:

IDBMA
9740 Appaloosa Road, Suite 210
San Diego, CA 92131
619/578-3152

The cover price is \$30.00 U.S. plus postage and handling.

Pick Hits

Pick Hits is a new publication distributed by PICK Systems Inc. The publication is continually being updated and provides up-to-date listings of PICK software packages. Software categories are listed alphabetically with company name, package title, and a brief description of the software. An alphabetical vendor listing complete with phone numbers is provided as a cross reference. The publication is available from:

PICK Systems, Inc.
1691 Browning
Irvine, CA 92714
1-800/FOR-PICK

Systems Integrators/Value Added Resellers

Many of the developers of PICK software are systems integrators. They specialize in writing software but they package computer hardware with software and provide service, training and support. At

present there is no special listing for these companies in either the IDBMA software directory or PICK HITS. Some of the software developers listed in either the IDBMA Software Directory or Pick Hits may prefer to sell you a complete system or refer you to one of their dealers where you can get local training and support.

Summary

PICK was conceived as a business-oriented operating system. Therefore it is no surprise that there are over 1000 varied business application programs available. No piece of software is perfect so don't expect to find software that will exactly match the way you conduct your business.

13

Further Considerations

Introduction

The PICK operating system was conceived in the late 1960's and has matured into a powerful multi-user operating system. Nevertheless, the people at PICK continue to make improvements. A new version of PICK for the PC/XT called Open Architecture Release 1.0 is due to be released in early 1987. In addition, PICK for the IBM PC/AT will be released. In this chapter we will briefly discuss Open Architecture Release 1.0 of PICK as well as consider PICK for the PC/AT. Other topics we will cover are Pickworld magazine, the IDBMA, PICK user's groups, newly supported PICK hardware in a variety of price/performance ranges, where to buy a PICK system and what to expect in the way of training and support.

PICK Open Architecture Release 1.0

The term “open architecture” was coined as a result of PICK Systems’ desire to revise the operating system. The PICK operating system was stripped down to its roots and redesigned to be more accommodating to enhancements and to provide a set of standards which can be followed by all PICK licensees. The essence of Open Architecture is to improve the features, speed, and convenience of PICK for users and programmers. A few of the features of PICK open architecture are:

- Frame size increased from 512 to 1024K
- Unlimited item size
- Direct access to TCL and ACCESS from BASIC
- Improved mathematical accuracy of calculations
- Performance hardware monitoring capability
- Support of additional terminal types

A Brief Discussion of Open Architecture Features

Transparency

Under open architecture, most of the changes and improvements made to the operating system are invisible or “transparent” to users. In addition, unlike some new releases of popular MS-DOS software packages, no changes need be made to any data files when a new version of PICK is installed. All of the basic PICK functions, including TCL commands, the EDITOR, and ACCESS operate the same as in any version of PICK prior to open architecture. Compatibility with all data files and previous versions of PICK is ensured.

The EXECUTE Statement

One of the features of open architecture PICK programmers have been clamoring for is the facility to execute TCL and ACCESS commands directly from within a BASIC program. The EXECUTE statement provides this capability. This is important to programmers because many times their programs require using the ACCESS processor or TCL commands. The only means of executing either an ACCESS

command or TCL command has been to activate a PROC, let the PROC execute the TCL command or ACCESS statement, exit the PROC and return to BASIC. The EXECUTE statement will allow programs to run faster.

The EXECUTE statement at the user level is important because some of the newer versions of application software, such as CompuSheet +, capitalize on this feature. For example within the CompuSheet + program there is a command called “XEq” which allows you to leave a spreadsheet and temporarily exit to a TCL environment. You can then execute any TCL command except an OFF or LOGTO command. You may execute an ACCESS command, a PROC, use the EDITOR, run another program, or run a program to check on inventory status, etc. When you want to return to the spreadsheet enter the word “RETURN” or “RE” and you will be returned to the point in the spreadsheet where you were prior to exiting to TCL.

Note: Executing either an OFF or LOGTO command will remove you from the currently logged account. You will not be able to return to the spreadsheet and any recent changes made to the spreadsheet will be lost.

Open Architecture on the PC/XT

As of this writing the open architecture version of PICK on the XT has yet to be released. In the PICK trade publications the new version of PICK for the XT is referred to as PICK Version 2.0. The list of new features to be implemented in PICK 2.0 for the XT is depicted in figure 13.1.

Feature	PC/XT version 1.3	XT 2.0	PC/AT
three users	yes	yes	yes
six users	no	no	yes
MS-DOS bridge	no	yes	yes
tape backup	no	yes	yes
additional hard disk support	yes	yes	yes
High density floppy disk	no	no	yes
EXECUTE statement	no	yes	yes

Figure 13.1. Summary of PICK PC/XT and PC-AT features

PICK on the IBM PC-AT

Shortly after the release of the first version of PICK for the PC/XT, development and implementation of PICK for the PC-AT was begun. The commitment to this project demonstrates that PICK Systems is dedicated to offering a high performance, low cost operating system on the most up to date high performance hardware. All of the power and sophistication of PICK is available on one of the most powerful and widely available microcomputers on the market. Some of the features of PICK on the PC-AT are:

- All of the PC/XT existing features
- Increased number of users (up to six)
- PICK/DOS bridge
- EXECUTE statement (part of open architecture)
- Cartridge tape backup facility
- Multiple hard disk drive support
- High density (1.2 MB) floppy disk drive support

In its basic configuration a PC-AT is capable of supporting three users, has 20 MB of disk space and 512K of RAM and retails for under \$6000.00. The PICK software for this configuration retails for \$795.00. All of the PC/XT features have been implemented in this version plus some new features which we will discuss shortly. The primary benefit of the AT is that it runs three times faster than the XT.

Unique Features of the AT Implementation

Number of Users

The standard AT is configured for three users but can support a maximum of six if a special adapter board manufactured by AST is added. Currently the PICK system for the AT supports a maximum of six users.

PICK/MS-DOS Bridge

Under the fixed disk partitioning concept it is possible to have as many as four different operating systems on the hard disk, each

one isolated from the other. Usually, it is only possible to access files in the currently active partition. The MS-DOS bridge is a special utility which allows files residing in the DOS partition to be accessed. In addition this utility allows you to access files in the PICK partition while DOS is the active partition.

Note: This feature does not allow you to run a DOS program while PICK is the active partition.

Cartridge Tape Backup

On systems with large data files, multiple floppy disk backup procedures are cumbersome and time consuming. An external tape drive can be added to a PC-AT system for under \$2500. The tape drive utilizes 1/4 inch tape cartridges which can store up to 60 MB of data. Time is not wasted changing floppy disks during a FILE-SAVE procedure.

High Density Floppy Disk Drive

A single high density (HD) floppy disk drive comes as standard equipment with every PC-AT. High density refers to the fact that each diskette has 1.2 MB capacity as opposed to the usual 360 KB on the PC/XT. However, the drive can utilize both diskette formats. This helps to ensure compatibility from one PICK machine to another.

The advantage of the high density disk is that one disk can store three times the amount of data of a conventional disk. Fewer disks are required for a FILE-SAVE and less time will be spent swapping disks. If you don't want to spend the money for a tape drive, using the HD disks is a good interim alternative.

Note: High density disks are available from a variety of vendors such as Maxell, TDK, Fuji, and BASF. They should be readily available at any store selling the PC-AT or a PICK dealer. A single HD disk retails for \$4.00-\$5.00.

PC-AT Versions

There are two versions of the PC-AT software. Version one supports three users and retails for \$795. Version two supports up to six users and retails for \$995. Like other versions of PICK, the AT version will continue to evolve. Future versions will likely accommodate additional users, support a variety of AT compatible computers, and support additional hardware peripherals.

PICK Wish List

The majority of this book has focused on PICK for the XT. However, PICK has traditionally been categorized as follows:

- Generic PICK, including PC/XT, AT, and all IBM PC compatibles
- Ultimate
- Microdata

Although all these versions are compatible, because Ultimate and Microdata are companies which have implemented PICK on their mini and mainframe computers, they have added certain refinements which may not be part of generic PICK. One feature they have recently implemented an incremental FILE-SAVE procedure. This feature can be a real time saver when large amounts of data are involved. This feature allows you to back up only those data files which have been changed since the last FILE-SAVE.

For this reason we have compiled a PICK wish list consisting of those features which the PICK community would like to see implemented. Some of these features are:

- Improved telecommunications and networking capability
- A CP/M or MS-DOS emulator
- A C language program compiler to help bridge the UNIX-PICK gap
- A TCL command line editor
- Multitasking capability
- Extensive graphics capability
- A full screen text editor
- On-line help for PICK commands with examples of command syntax

There is, of course, no assurance that some or any of these features will ever be implemented. Any news of these developments would likely be announced in Pickworld, the quarterly magazine published by PICK Systems.

The Future of PICK

The future of PICK appears to be a bright one. Much of the work at PICK systems over the last few years has been devoted to the implementation of open architecture and PICK on the PC/AT. This reaffirms the commitment which PICK Systems has to micro computers, such as the PC/XT and PC/AT, which offer a high price/performance ratio. In addition, the sheer number of IBM PCs and their look alikes offer PICK a vast untapped market. The marketing people at PICK feel that the time is right to begin marketing their system in earnest. To illustrate, consider the following:

- PICK estimates 60,000 installations worldwide averaging six users per system
- There are approximately 10,000 PC system users and growing
- PICK is cooperating with its licensees to put together a curriculum for colleges and universities to teach programming in PICK BASIC
- Hardware companies such as Ultimate Corp. are being encouraged to lend hardware to universities with the stipulation that a class on the PICK operating system be taught.
- According to IDBMA more than \$1 billion in PICK-based hardware and software is installed world wide
- The growth rate of the PICK marketplace is estimated at 50% per year
- A few of the newer companies joining the PICK community are: Fujitsu Microsystems America Inc., Nixdorf Computer Corp., CIE Systems Inc.

PICK Hardware Compatibility

PICK goes to great lengths to ensure compatibility among various machines. This means that if you already own or are considering

upgrading to a popular microcomputer other than the PC/XT you may be in for a pleasant surprise. The following computers have all been successfully tested with PICK XT version 1.3:

- Compaq Deskpro/Compac Plus
- AT&T 6300
- ITT-XTRA
- NCR PC4
- The Ericsson
- AST boards and Ampex 210 terminal

PICK is installed by following the instructions as if you were using a PC/XT. If you have chosen to use an AT&T 6300 you will find it will perform just like the PC/XT with the exception that it will run two to three times faster than an XT! There are other computer vendors, such as General Automation, which offer smaller entry level systems. An example of such a system is the Zebra. Its current price is under \$10,000.

Growing with PICK

One of the major benefits of PICK is its portability. It is possible for a user to learn PICK using a PC/XT, develop applications using the WIZARD software, use word processing, spreadsheet or vertical market applications. As one's business prospers and the need for more computing power arises all existing software and data can be easily "ported" (transferred) to the new system. There is no need to throw everything out and start over. A PC/XT system has good resale value. The money obtained from the sale of the XT can be applied toward the new system. No time or money need be spent on new software or retraining since the system will operate exactly the same way as the old one. Your PICK investment is protected. PICK Systems has regular meetings with its licensees to ensure that 100% portability and compatibility can be maintained. No other microcomputer operating system can presently make this claim.

PICK Longevity

A major concern any business has about computer companies today is their longevity. In fact one of the primary reasons most people purchase an IBM computer is because IBM is likely to be in business for a long time. So is PICK Systems. This is not to imply that PICK Systems is a multibillion dollar company like IBM, but rather that PICK has a huge installed customer base and many major mini and mainframe computer companies who are licensees. PICK is well entrenched in the computer market, and has for many years occupied a unique niche.

Now that PICK is available on popular microcomputers, the awareness of PICK will increase. As more microcomputer users discover the limitations of a single-user system, PICK provides a natural upgrade path. If you already own a PC/XT or PC compatible machine you may want to consider keeping it and investing in PICK and some application software. Your PICK system can grow with your business.

Keeping in Touch with the PICK Community

There are two informative magazines published for the PICK community. Each magazine has a unique format and both should be read regularly to stay informed. In this section we will briefly discuss the two magazines and their publishers.

Pickworld Magazine

Pickworld is published quarterly and is an affiliate of PICK Systems. The magazine is comprised of four sections:

- feature articles
- technical reports
- general news in the PICK community
- regular departments such as letters to the editor, an editorial by Dick Pick, classified advertising, list of PICK dealers, etc.

Each edition features three or four articles of general interest, questions and answers about PICK, user testimonials, how businesses use PICK, PICK dealer listings, user's group listings, etc. The magazine format is attractive and easy to read. In the past PICK and Pickworld have sponsored a trade show called Pickfair. More recently Pickfair has been discontinued in favor of supporting specific licensee shows and user group organizations. A one year subscription to Pickworld in the U.S. costs \$25.00. However, a new PICK user is entitled to a free one year subscription to Pickworld when the PICK software license agreement is signed and returned to PICK Systems. For more information contact:

Jeanne Sweeney
P.O. Box 9539
Fountain Valley, CA 92728-9539
714/839-6708

The IDBMA (International Database Management Association)

The IDBMA is an organization which has as its objectives increasing the awareness of the PICK operating system and providing a means of educating new and existing PICK users via trade shows, meetings and the IDBMA International Spectrum magazine. International Spectrum magazine follows a similar format to Pickworld but is cosmetically different. It features the following sections:

- point of view (editorial dialogue)
- Spectrum U.S.A.-news relating to PICK-based companies
- Spectrum International-news from foreign PICK-based companies
- product updates
- technical aspects
- cover story or the main focus of a particular issue
- PICK people in the news

International Spectrum is published six times per year and has an annual subscription price of \$40.00.

The International Spectrum Trade Show

The IDBMA produces three PICK-oriented conference/exhibitions each year; one in the U.S., one in Europe, and one in the Pacific Basin. Each exhibition is 100% devoted to PICK and features hardware exhibits, software exhibits, and a wide range of seminar topics. A good alternative to attending an exhibition is to select an audio cassette on a topic of interest to you from one of the convention sessions. A list of session topics is printed in the issue of International Spectrum following the conference. For more information contact:

IDBMA, Inc.
9740 Appaloosa Road Suite 104
San Diego, CA 92131
619/578-3152

PICK User's Groups

User's groups are important to any operating system. They are usually formed in response to a need for a number of PICK users in a specific geographic area to meet to discuss a variety of PICK-related topics. A detailed user group directory and calendar of events is printed in each issue of Pickworld. There is no national user's group because local user's groups can provide more immediate benefits to users. It is likely that there is a user's group in most major metropolitan areas of the U.S.

Where to Buy PICK Systems

If you are interested in obtaining a demonstration of a PICK PC/XT computer system, you can obtain a list of PICK PC dealers directly from PICK systems. There are dealers in 37 states from Alaska to Wyoming. Don't expect to find a fancy storefront and sales personnel with shirts and ties. Many PICK dealers have worked with PICK for many years with the large companies such as Ultimate and General Automation. They have started their own company and are devoted to selling PICK turnkey systems. The dealers are PICK experts and can usually help you find the application your business needs. In addition, they can provide the necessary local training and support.

Fujitsu Microsystems of America

Fujitsu Microsystems of America markets and manufactures high performance business micro computers using the PICK operating system. The model 2000 is based on the 8086 (PC/XT) microprocessor, while the model 2020 is based on the 80286 (PC-AT) microprocessor. While it is possible to use MS-DOS programs on these systems (using the Fujitsu version of MS-DOS) these systems are optimized for PICK.

Recalling our discussion of PROCs in chapter 10, we said that a menu-driven system could and should be implemented to facilitate system use by novices. Fujitsu has implemented such a system. All Fujitsu micro systems include:

- easy to read and professionally packaged documentation
- fully menu-driven operation
- on-line help for all menu functions
- on-line "getting started" tutorials
- a color monitor for line 0
- support for telecommunications
- support for graphics

If you are considering the purchase of a new PICK system for 3 to 6 users look into Fujitsu. Contact:

Fujitsu Microsystems of America, Inc.
3025 Orchard Parkway
San Jose, CA 95134
(408) 434-1160

for a list of Fujitsu dealers.

Appendix A: Sample PROC Listings

This appendix contains listings of some of the PROCs discussed in chapters 8 and 10. These PROCs are not a part of a standard PICK XT system and must be manually input and stored on the disk if they are to be used. If you are using one of the menu PROCs to execute software, the lines of the PROCs “calling” the software packages must be modified to include the names of the programs installed on your system. The functions of a menu PROC are discussed in chapter 10.

In order to execute a dictionary and a data portion of each PROC must exist.

The dictionary portion of each PROC follows the format:

```
ITEMNAME
001 PQ
002 (filename)
```

where ITEMNAME is the name of the PROC, PQ identifies the item as a PROC, and (filename) is the name of file where the PROC is stored.

The PROC listings depicted are the data portions of each PROC.

001	PQ	
002	T C PICK Version 1.3	Help Screen
003	O	PICK Version 1.3 Help Screen
004	O	
005	OCommand/Proc	Function
006	OACCOUNT-SAVE	Backup individual accounts
007	OACCOUNT-RESTORE	Restore backup data to an account
008	OCOLOR	Set primary color of main terminal
009	OCREATE-ACCOUNT	Create new accounts
010	OCT	Display file contents on terminal
011	ODELETE-ACCOUNT	Deletes an account and its files
012	OFDISK	Creates/changes active disk partition
013	OFFILE-SAVE	Performs complete system backup
014	OFFORMAT	Formats floppy disks under PICK
015	OLISTACC	Lists logon data for each account
016	OLISTCONN	Lists ACCESS connectives
017	OLISTFILES	Lists the files on an account
018	OLISTPROCS	Lists the PROCs for currently logged account
019	OLISTU	Lists all currently logged on users
020	OLISTVERBS	Lists the verbs for an account
021	OLOGTO	Logon to another account
022	OMSG	Send messages to other users
023	OOFF	Logoff system
024	OPOVF	Display contiguous disk space
025	OPOWER-OFF	Flushes RAM to disk for system shutdown
026	O	
027	O...press RETURN for next screen page+	
028	IP:	
029	T C	
030	O	PICK Version 1.3 Help Screen
031	O	
032	OCommand	Function
033	ORESTORE SYSTEM	Restores operating system/data files from floppy
034	ORUNOFF	Formats edited text for printing
035	OSET-TIME	Set current system time
036	OSET-DATE	Set current system date
037	OSTARTPTR	Activate port/spooler for serial/parallel printer
038	OTERM	Set up terminal/printer parameters
039	OTERM-TYPE	Set terminal default values
040	OTIME	Outputs system time and date
041	OWHAT	Displays system configuration
042	OWHO	Outputs current account status

Intro PROC

```

      INTRO
001  PQ
002  C! INTRO
003  1 O
004  T C, +
005  C Display section of the proc.
006  O
007  O
008  O
009  O
010  O
011  O
012  O
013  O
014  O
015  O
016  O
017  O
018  O
019  O
020  O
021  O
022  O
023  O
024  O
025  O
026  O
027  O
028  O
029  RI
030  IP:
031  C Branching section of the proc.
032  IF A = EX X
033  IF A = [ X
034  IF A = . X
035  IF A = OFF GO 999
036  IF A = 1 GO 10
037  IF A = 2 GO 20
038  GO 1
039  C Program calling section of the proc.
040  10 C Run word processing program
041  [LINK TO WORD PROCESSING PROGRAM]
042  GO 1
043  20 C Run spreadsheet program

```

INTRO

1. WORD PROCESSING

2. SPREADSHEET/FINANCIAL MODELING

Select one of the above, or ESC, or 'OFF' +

044 [LINK TO SPREADSHEET PROGRAM]
045 GO 1
046 999 C LOGOFF SYSTEM
047 HOFF
048 P

MASTER PROC

MASTER
001 PQ
002 C! MASTER
003 1 0
004 T C, +
005 C Display section of the proc.
006 O
007 O
008 O MASTER MENU
009 O
010 O Your Company Name
011 O
012 O 1. ACCOUNTING/ORDER ENTRY/INVENTORY CONTROL
013 O
014 O 2. FINANCIAL MODELING/SPREADSHEET
015 O
016 O 3. JET WORD PROCESSING
017 O
018 O 4. DATA BASE MANAGEMENT (WIZARD)
019 O
020 O 5. CYCLONE III WORD PROCESSING
021 O
022 O 6. TOTALWARE
023 O
024 O 7. SYSTEM UTILITIES
025 O
026 O 8. HELP ON PICK COMMANDS
027 O
028 O Select one of the above, or ESC, or 'OFF' +
029 RI
030 IP:
031 C Branching section of the proc.
032 IF A = EX X
033 IF A = [X
034 IF A = . X
035 IF A = OFF GO 999
036 IF A = 1 GO 10
037 IF A = 2 GO 20

```
038 IF A = 3 GO 30
039 IF A = 4 GO 40
040 IF A = 5 GO 50
041 IF A = 6 GO 60
042 IF A = 7 GO 70
043 IF A = 8 GO 80
044 GO 1
045 C Prgram calling section of the proc.
046 10 C Run accounting programs
047 [WPL ACCOUNTING]
048 GO 1
049 20 C Run spreadsheet programs
050 [WPL SPREADSHEET]
051 GO 1
052 30 C Run JET word processing programs
053 [WPL JET]
054 GO 1
055 40 C Run WIZARD programs
056 [WPL W]
057 GO 1
058 50 C Run Cyclone word processor
059 [WPL CYCLONE]
060 GO 1
061 60 C Run TotalWare office automation system
062 [WPL TOTALWARE]
063 GO 1
064 70 C Run utility proc menu
065 [WPL UTIL]
066 GO 1
067 80 C Run help proc
068 [WPL HELP]
069 GO 1
070 999 C Logoff system
072 HOFF
073 P
```

Index

A

Acc file 227-228
ACCESS 24, 259-274
ACCESS error messages 263
ACCESS LIST-LABEL/SORT-
LABEL verbs 268-272
ACCESS LIST verb 265-266
ACCESS modifiers 264-265
ACCESS options 264-265
ACCESS selection criteria 264-265
ACCESS sentence 261
ACCESS sort verb 266-268
ACCESS syntax 263
ACCESS time 84
ACCESS verbs 263-264, 265-272
ACCOUNT-RESTORE 39, 105-106
ACCOUNT-SAVE 39, 106-107
account 46, 100-101
Accounting history file 229
Addenda 40
Asterisk wildcard 89
Attribute switches 107
Attribute 14, 46

B

Backtabs 17
Block-convert file 228
BLOCK-PRINT 61
Bootstrap loader 19
Break key 98

C

Caps lock 58
Cartridge tape backup 311
Clearing files 86-87
COLDSTART PROC 245
Color 102, 107-109
Command 46
Command line 46, 48-49
Convention 48
Copy to another file 92
Copy to printer 89-90
Copy to same file 90-91
Copy to screen 89-90
CREATE-ACCOUNT 109-110
ct 62, 111-114
Ctrl break 198
Ctrl key 49
Cylinders 33

D

Data 15
Data item 80
Default value 46
Delete PROC 230
DELETE-ACCOUNT 114-115
Deleting files 87-89
Delimiter 46, 60
Diagnostics 19
Dictionary 46

Dictionary item 82
Disk partitions 32-36

E

EDITOR Commands
 (B) bottom command 168, 179
 (U) up command 168, 192
 ? command 166, 180
 A 179
 AS 179 C 180
 Change current line number 167
 EDITOR Command Reference 176-193
 edit data items 158-160
 EDITOR delete 174-175
 Edit DE 180-182
 Edit dictionary items 161-164
 Down command 168
 EX 182
 F 183
 FD 183
 FI 183
 EDITOR file commands 182-183
 FS 183
 G 184
 I 172, 184
 Insert 171-173
 L (list) 166, 185
 L (locate) 185-186
 ME 186-188
 N 188
 R 169-171, 188-190
 S (suppress) 167, 191
 S? 190
 Save file to disk 175-176
 TB 191
 Top (T) command 167, 192
 Wildcard + 192
 X 193
 Editor tutorial 165-176
 Editor 46, 157-194
 Embedded commands 140-141
 ERRMSG file 228-229
 Error messages 100
 EXECUTE statement 244, 308

F

FDISK 33-36, 41-42, 115-117
 FFORMAT 120-121
 Field 14
 File 14, 46

File contention 24
 File creation 83 File data 79

File dictionary 79
 File ownership 93
 File structure 76
 File system 76
 FILE-SAVE 117-120, 239-240
 Floppy disk drive 222
 Frame 28, 46
 Fujitsu 318

G

Group Format Errors 226-227
 Group ID 84

H

HELP PROC 217-219, 246-248
 Hierarchical files 51
 High density floppy 311

I

IDBMA 316-317
 Initialize 37
 Installation 36-38
 Installation menu 37
 Item 14, 47, 80
 Item-ID 47

K

Keyboard 15

L

Line 42, 47
 LISTACC 122-124
 LISTDICT 63
 LISTFILES 64, 124-125
 LISTPROCS 65, 125-127
 LISTU 66, 128
 LISTVERBS 66, 128-130
 Logon PROC'S 72-24, 198-199, 245
 Logon prompt 57
 Logon 131-132
 Logto 132-133

M

Mailing label printing 268-272
 Master dictionary 47, 78

master proc 214-217, 248-251
Memory mapped video monitor 147

O

Object data errors 226
Off 67, 132
Open architecture 308-309
Operating system 17-21
Options 62
Overflow pool 135

P

Parity errors 225-226
password 58, 110
PC-AT versions 312
PICK BASIC 24
PICK Hits 26
PICK user's groups 317
Pickworld magazine 315-316
PLISTPROCS 213
POVF 134-135
POWER-OFF 136
PQ 97
Print queues 43
Print screen (Prt Sc) 17
Print spooler 43
Printer connections 223
Printers 43
Printing control 146-147
PROC 25, 47
PROC Programming 195-219
 A command 204
 C command 205
 GO command 209
 H command 207-208
 IF command 204-205
 Input/output buffers 201
 Labels 201-203
 LINK command 210-211
 O command 210
 Relational operators 203
 RI and IP commands 209-210
 Screen control 205-207
PROC components 200-211
PROC errors 251, 257
PROC execution and termination 208-209
PROCLIB file 228
Programs 15
Protection violation 36

Q

Q-pointers 230-239, 246

R

REBOOT 136
Records 14
Restore system 136-139
Restoring contiguous disk space 224-225
RUNOFF 139-141

S

Security 57, 93-95
Serial connector 43
Serial printer 128
Serial printer startup 144-146
SET-DATE 52, 142-143
SET-FILE PROC 233
SET-FUNC 17
SET-LPTR 152-153
SET-TIME 52, 141-142
Software 13, 275-306
CompuSheet 278, 297-300
 cyclone III word processor 293-296
 cyclone III spreadsheet 297-300
 database interface 279
 dBASEIII 280-281
 IDBMA 304
 JET 289-293
 PICK Hits 305
 Reentrancy 276
 Report writer 281
 Task prioritization 276
 TotalWare office automation 300-302
 Upward compatibility 277
 Vertical market software 303-304
STARTPR 143-146
Startup 52
Stopping 53
Storing ACCESS command lines 273
SYSPROG-PL 230-231
SYSPROG 39, 47, 100-101
System dictionary 77
System ID 55-56

T

Tab stops 67-68
TCL command format 59

TCL prompt 59
TCL 25, 48, 58, 98-100
TERM-TYPE 148-152
TERM 68-72, 148-151
Terminal types 42
Terminals 42, 222-224
THEOS 29
TIME 72, 141-142
Transparency 308

U

UNIX 29, 259
Upgrades 39
Utility proc 251-257
Utility 19

V

Value added resellers 306
Value 48
Verb 48, 59
Virtual memory 27-28, 134

W

WHICH PROC 212
WHO 153-155
Wish list 312
WIZARD 197, 257, 273, 278, 282-287

X

XENIX 29

Working With Pick

By Walt Stagner

Working With PICK is a clear, practical guide to the PICK operating system for microcomputers including the IBM PC/XT, IBM PC-AT, and compatibles. This book benefits the system manager and the novice user. First, it is a "hands-on" guide to the use of the commands (verbs) required to maintain a PICK system for the system manager. Second it describes everything the novice needs to know in order to put PICK to work. Many examples and tutorials are included to clarify discussions.

Two chapters are devoted exclusively to the PICK PROC language. PROC is a powerful, yet easy to use language which allows the user to construct menus making any PICK system virtually menu-driven!

The following topics are covered in Working With PICK:

- System installation
- Virtual memory management
- System security
- Filing system
- Database management capability
- System maintenance
- PROC language
- Examples of PROCs
- Creation of a "turn-key system"
- Software availability
- Reviews of software
- Future enhancements

This book is required reading for anyone using the PICK system, or anyone considering a multi-user micro-computer operating system for business applications.

WEBER SYSTEMS INC.
8437 Mayfield Road, Chesterland, Ohio 44026