

THEOS/OASIS

User's Handbook

THEOS HELP SCREEN Version 2 6.1-8

Command	Description
ACCOUNT	Maintain the user account names file.
ARCHIVE	Backup hard disks onto floppies or tapes.
ASSIGN	Assign a file to an I/O channel.
ATTACH	Logically connect a device for future access.
BACKUP	Copy full hard disk to another disk or tape.
BASIC	Interactive THEOS BASIC interpreter.
BULKERA	Erases cartridge tapes.
CACHE	Allows initialization for cache disk operation.
CADVERF	Allows detection and correction of hard disk errors.
CHANGE	Modify privilege level of a command.
CLASS	Customize SYSTEM.CLASSnn CRT files for your terminal.
COMPARE	Compares files and reports any differences.
COPYFILE	Move files from device to device.
CREATE	Allocate disk space for Direct or Indexed file.
CSI	THEOS Command String Interpreter.
DEBUG	Debugger for machine language programs.
DUMPDISK	Dump contents of file in hex and ASCII.
EDIT	System file editor, both line and context.
ERASE	Remove a file from the disk.
ERRMSG	Maintain the SYSTEM.ERRMSG file.
EXEC	THEOS EXECutive Job Control Language.
FILELIST	List contents of file directory.
FILL6080	Translate from Intel to Zilog assembly mnemonics.
FORCE	Cause a user partition to execute a command.
GETFILE	Move files from other systems to THEOS.
HELP	Display function, syntax and options of a command.

**A comprehensive guide to the business applications and functions
of the THEOS operating system.**

by Walter C. Stagner Jr.

THEOS User's Handbook

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 express 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 out of the furnishing, performance, or use of this book and/or its programs.

THEOS™ and THEOS86 are trademarks of THEOS Software Corp.
Z80™ is a trademark of Zilog Corp.
RM Cobol™ is a trademark of Ryan-McFarland
Lotus 1-2-3™ is a trademark of Lotus Development Corp.
Apple DOS™ is a trademark of Apple Computer Corp.
CPM™ is a trademark of Digital Research Inc.
MS-DOS™ is a trademark of Microsoft Corp.
Spellbinder™ is a trademark of Lexisoft Inc.
multiWRITE™ and multiCALC™ are trademarks of Logic Extension Resources
UNIX™ is a trademark of Bell Laboratories

Published by:
Weber Systems, Inc.
8437 Mayfield Road
Chesterland, Ohio 44026

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

THEOS User's Handbook

Copyright© 1985 by Weber Systems, Inc. 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.

Library of Congress Cataloging-in-Publication Data

Stagner, Walter
THEOS User's Handbook.

Includes index.

1. THEOS (Computer operating system) 1. Title.
QA76.76.063 1986 001.64 85-11619
ISBN 0-938862-00-6

Contents

1. Operating Systems And THEOS	12
Introduction	12
What is Software	12
Terminology	12
A Tour of Terminal Keyboard	14
The Operating System	16
Monitor and Utility Programs	17
Operating System Selection	19
History and Development of THEOS	20
Design Philosophy of THEOS	22
The THEOS Components	23
THEOS Hardware Requirements	24
Application Software for THEOS	24
THEOS Statistics and Capabilities	26
Summary	26
2. An Overview of THEOS	27
Security	27
User Accounting	28
Assigning Accounts	28

File Ownership _____	28
Logging on to an Account _____	29
Adding Accounts _____	29
Privilege Levels _____	30
Account Synonyms _____	30
Account Passwords _____	30
Accounting History _____	32
Multi-User THEOS _____	32
The Built-in Help Facility _____	37
Other Special THEOS Features _____	38
THEOS Files _____	40
Data File Formats _____	40
File and Disk Protection _____	41
Disk Space Utilization _____	41
Control and Sort _____	41
R/M COBOL _____	42
File and Disk Management Utilities _____	42
Summary _____	43

3. THEOS Conventions _____ 45

Introduction _____	45
Documentation Conventions _____	46
System Conventions _____	46
Standard Search Order _____	47
Exceptions to Standard Search Order _____	47
Prompting Characters _____	47
Filename, Filetype, Filedisk _____	48
Reserved File Descriptions _____	49
Wildcards and Special Characters _____	50
System Control Keys _____	52
THEOS Command Lines _____	52
Summary _____	54

4. THEOS Commands _____ 55

Introduction _____	55
Session 1: Starting and Stopping a THEOS System _____	56
THEOS Command Summary _____	59

Session 2:	The ACCOUNT Command	61
Session 3:	The RESTORE Command	67
Session 4:	The ARCHIVE Command	73
Session 5:	The RESTORE Command	78
Session 6:	The CHANGE Command	83
Session 7:	The COPYFILE Command	84
Session 8:	The CREATE Command	89
Session 9:	The FILELIST Command	92
Session 10:	The INITDISK Command	102
Session 11:	The INITTAPE Command	107
Session 12:	The LIST Command	109
Session 13:	The LOGOFF/LOGON Commands	114
Session 14:	The MAILBOX/MSG Commands	117
Session 15:	The RENAME Command	120
Session 16:	The ERASE Command	123
Session 17:	The EDIT and SCRIPT Commands	126
Session 18:	The SET/START/STOP Commands	140
Session 19:	The SHOW Command	144
Session 20:	The SPOOLER Command	148
Session 21:	The SYSGEN Command	154
Session 22:	SEND/RECEIVE/TERMINAL Commands	157
Session 23:	How to Install Application Software	168

5. System Files And Diagnostic Commands 173

Introduction	173
SYSTEM.DEVnn Files	174
SYSTEM.CLASSnn Files	174
SYSTEM.ERRMSG	178
SYSTEM.EXECnn Files	182
SYSTEM.SPOOLQUE File	182
System Diagnostics	183

6. The THEOS EXEC Language 189

Introduction	189
System Cancel, IPL, EXEC, and User Logon File	191
System Cancel	191
IPL.EXEC and User Logon	191

The EXECn Files _____	193
Changing An Existing SYSTEM. EXECn File _____	194
The SELECTED.EXEC File _____	195
Components of the EXEC Programming Language _____	197
Constants and Variables _____	197
True Variables _____	198
Reserved Variables _____	199
Labels _____	199
Operators _____	200
Comments _____	201
Keywords _____	201
EXEC Instructions _____	202
&BEGSTACK _____	202
&BEGTYPE _____	203
&CRT _____	204
CSI _____	205
&END _____	206
&ERROR _____	206
&GOTO _____	207
&IF _____	207
&QUIT _____	208
&READ _____	208
&SPACE _____	209
&TYPE _____	209
&WAIT _____	210
Using the EXEC Language _____	211
Using the IPL.EXEC _____	215
Summary _____	219
7. Using Menus Created By the EXEC Language _____	221
System Startup _____	222
Running Application Software from a Menu _____	225
Accounting _____	225
Financial Modeling _____	226
Terminal Program _____	226
Spellbinder Word Processing _____	226

Utility Functions _____	226
Send Messages _____	226
Retrieve Messages _____	228
Logoff/Logon _____	229
MultiWRITE Word Processing _____	229
Quit _____	229
Menu Error Messages _____	230
System Maintenance from a Menu _____	230
Copy Files _____	231
Archive Files _____	233
Errors Encountered During Archive Procedure _____	237
Restore Files _____	237
Account Maintenance _____	238
Report Misallocations _____	240
Format Floppy Discs and Tapes _____	242
Single User Mode _____	244
Multi User Mode _____	245
Multi User Mode Without Spooler _____	246
Erase Files _____	246
List Files _____	248
Return to Master Menu _____	249
Logoff System _____	249
System Cancel _____	249
Summary _____	250
8. A User's Introduction to THEOS Software _____	251
Introduction _____	251
Why You Probably Haven't Heard of THEOS	
Application Software _____	252
Three Types of Software Everyone Can Use _____	253
Word Processors _____	253
Spellbinder _____	253
multiWRITE _____	256
Spreadsheets _____	258
Scratchpad 3.1 and multiCALC 2.0 _____	259

Database Managers _____	261
UserBase _____	262
Control-8 _____	265
General Accounting Software _____	268
Open Systems Accountnign _____	269
Ashley Accounting System _____	272
Vertical Market Software _____	275
Where to Find THEOS Software _____	276
THEOS Application Software Directory _____	276
THEOS Times _____	277
Hardware Distributors _____	277
Summary _____	278
9. THEOS86 _____	279
Introduction _____	279
16 Bit vs. 8 Bit Microprocessors _____	280
Advantages of 16 Bit Computers _____	281
An Overview of THEOS86 _____	281
Number of Users _____	282
Disk Storage _____	283
Multiple Print Spoolers _____	283
More Working Memory _____	283
Graphics _____	284
THEOS86 Hardware Vendors _____	284
Seiko _____	285
Altos _____	285
IBM PC/XT _____	285
Onyx _____	286
Advanced Digital _____	286
THEOS 86 Application Software _____	287
A Comparison of THEOS vs. UNIX _____	287
Design Goals _____	288
Data File Management _____	289
File and Record Locking _____	289
Cost vs. Performance _____	290
Documentation _____	290
Summary _____	291

10. Further Consideration	292
Introduction	292
About THEOS Software Corporation	293
THEOS Version 6.1	293
Version 6.1 Refinements	294
System Cancel	295
Archiving in Multi-User Mode	294
New Help Messages	295
New EXEC Features	295
More Working Memory	295
The THEOS User's Group	296
Appendix A Glossary	297
Appendix B THEOS Command Summary	302
Appendix C Common THEOS Warning and Error Messages	304
Appendix D Using MCI Mail to Transmit Files	310
Appendix E THEOS EXEC Programs	312
Appendix F THEOS Supported Terminals	340

Acknowledgements

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

Kathleen Williams, Oasis Technologies

Fred Bellomy, Oasis User's Group

Jack Adams, Eagle Company

Chris Traut, Kimtron Corporation

Bill Berne, Quantum Information Systems

Rhett Tindall, Logic Extension Resources

Chris Pratt, Business Information Technologies

Chuck Blades, Computec Computers

Brian Ashley, Ashley Computer Systems

Paul Barnaby Jr., BCS Computers

Jeff Fender, Audio Plus

I would especially like to thank my wife Barbara for her proof reading and encouragement and Jeff Weber, for his patience and giving a new author a chance.

The author will be grateful to receive any suggestions from OASIS users for improvements to the book.

1

Introduction

This book is intended to be a non-technical tutorial and reference guide to the THEOS™ operating system. We intend to illustrate a method of creating a “turn-key” THEOS installation, and we wish to provide a supplement to the documentation provided by THEOS Software Corp., the marketers of THEOS. This book can be used by managers who may be considering the purchase of a THEOS based computer system as well as those who are already THEOS users.

The book is divided into two sections, the system manager section and the system user section. The system manager section is designed for the person charged with the responsibility of installing and maintaining the system, while the user section is designed for those who want to learn how to use THEOS in their everyday work.

OPERATING SYSTEMS AND THEOS

Introduction

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

What Is Software?

Of all the phases of computing **software** is one of the most important because all computer systems, regardless of size or cost, need software to operate. Software can be defined as the programs or list of instructions that tell the computer what you want it 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-oriented) task 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.

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. In order to remember the contents of your file folders, you always label them,

right? Similarly, your computer files have labels. Suppose you stored a group of names and addresses as a computerized file with the name, NAMES. The word NAMES is referred to as the **filename**.

RECORDS

Records are the individual items that make up each file. If your NAMES file contains a person's name, address, city, state, and zip code (visualize an index card like the one below) these 5 items would constitute a record.

Name _____
Address _____
City _____
State _____
Zip Code _____ - _____

Figure 1.1. Example of a record or "card"

The number of records (or cards) 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 applications software.

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 from those files which contain your data in that data files contain information entered by the user.

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 the information *you* put into the computer in a format that both you and the computer understand. When you fill in the index card shown previously, you are supplying the data.

A Tour Of Your Terminal Keyboard

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

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 NOT FOUND will be displayed.

SHIFT KEY

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. Some keyboards have a shift lock feature which allows you to generate upper case letters without holding the shift key down. Shift lock only affects the keys that type letters.

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.

ESC KEY

ESC stands for escape, which is a term left over from the days when teletypes were used for computer terminals. The name has stuck even though it is generally not descriptive of the key's function. ESC has a variety of uses which will be described throughout this book.

Unlike the shift and CTRL key, the ESC key is never used by holding it down while pressing another key. ESC is always pressed and released before the next key is pressed and released. This two-key

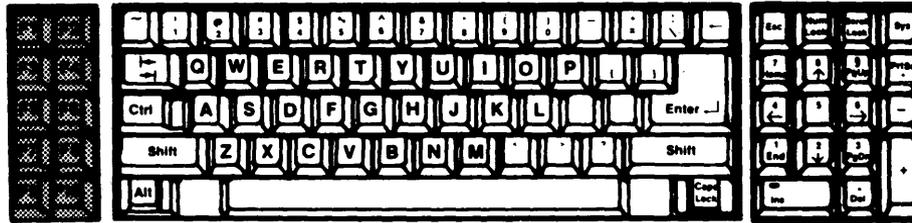


Figure 1.2. Typical terminal keyboard

operation is called an **escape sequence**. Examples of escape sequences will be provided later.

ARROW KEYS

These keys will allow you to move the cursor up, down, to the right and to the left. If you want to retype an entry, use the left arrow key to backup over the characters to be changed and then type over them.

FUNCTION KEYS

These keys are labelled F1 through Fn where n is the number of function keys provided. If your keyboard has 10 function keys, they will be labelled F1 through F10. Function keys are used for specific commands in applications software.

OTHER KEYS

These keys include the letters of the alphabet, the numbers zero through nine, and a set of symbols (which may vary from one keyboard to another).

Many typists do not distinguish between the number zero and the letter O or the number 1 and the lower case letter l. When using computers, you must get into the habit of typing the exact characters you desire. Guard against substituting zeroes (0) and the letter O and 1's and the letter l.

Any other keys you need to know about will be described as they are encountered.

The Operating System

The operating system is a program, or a series of programs, which serves as an interface between the hardware (the computer and its peripherals) and the outside world (the user and applications programs). 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 disk or other storage device and load these into the computer workspace (RAM). When the program has finished executing, control of the computer reverts to the operating system.

Operating systems are generally classified as software. Since an operating system is software, a portion of the operating system must be stored in RAM. Some 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:

- Oversees the transfer of data from RAM to a storage device such as a floppy disk. The operating system ensures that the data is stored exactly as it was entered.
- Allows you to store your data in files, each with a unique name for retrieval at a later time.
- Keeps a directory of all files stored on your disks much like a table of contents in a book.
- Allows you to erase unnecessary files and reuse the disk space.
- Allows you to communicate with the computer via a terminal/keyboard.
- Allows you to output information stored in the computer to a printer.

MONITOR AND UTILITY PROGRAMS

The operating system is generally divided into two components:

monitor	Manages the flow of information within the computer system. Think of it as a supervisor or a police person directing traffic at a busy intersection
utilities	A complete set of programs which manage computer "housekeeping" functions such as copying files, erasing unneeded files, etc.

The monitor provides the interface (or a means of communication) between the hardware, the user, and application software. When the 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). Operating System programs are stored on a disk and transferred to RAM as needed. This conserves space in RAM for application software and data.

Utility programs are operating system programs that allow you to erase files, make backup copies, print files, etc. Since all of these programs occupy RAM thereby taking space away from your data and applications software, they are stored on a disk and only called into RAM when needed. The monitor is the component of the operating system which actually accomplishes the task of "calling-up" the utility programs.

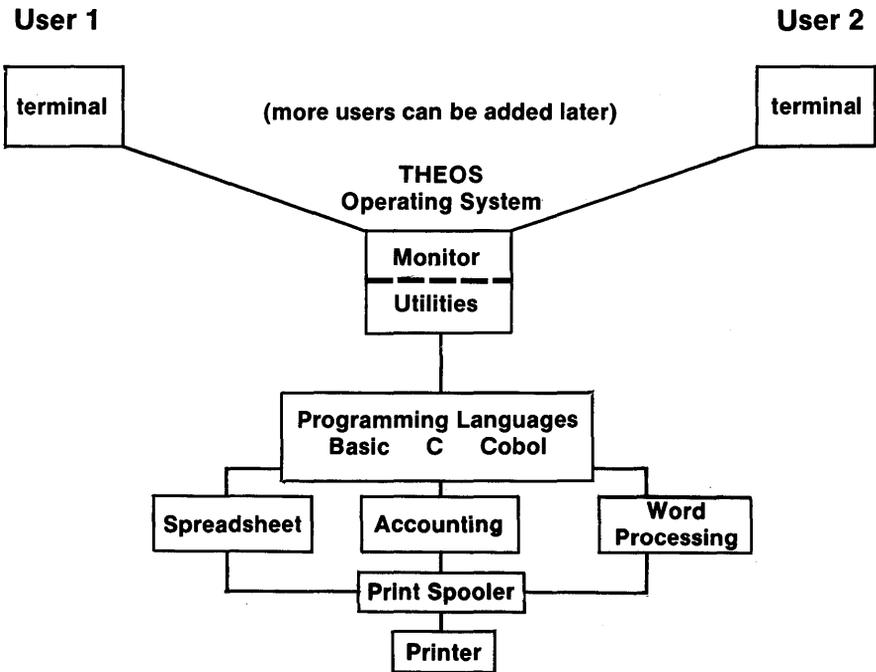


Figure 1.3. Diagram of THEOS operating system and user interaction

Operating System Selection

Choose carefully when selecting an operating system. The more popular the operating system, the more applications software is available. What? You mean you have to take the operating system into account when looking for applications software? The answer is yes!

Selection of the wrong operating system can have a monumental negative effect on the functionality, ease-of-use, and cost of your computer system. In addition, if the operating system cannot accommodate growth and increased "computing power" dictated by the applications 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 applications software as well.

The History and Development of THEOS

THEOS was originally named OASIS, which is an acronym for (O)nline (A)pplication (S)ystem (I)nteractive (S)oftware. Phase One Systems, Inc. of Oakland, California began marketing OASIS in 1978. Effective January 1, 1985, THEOS Software Corp. began exclusive marketing of OASIS, which was renamed THEOS. Aimed primarily at the business and marketplace, THEOS software performs equally well in single and multi-user installations. THEOS provides a cost-effective and convenient upgrade path for a company's growing data processing needs.

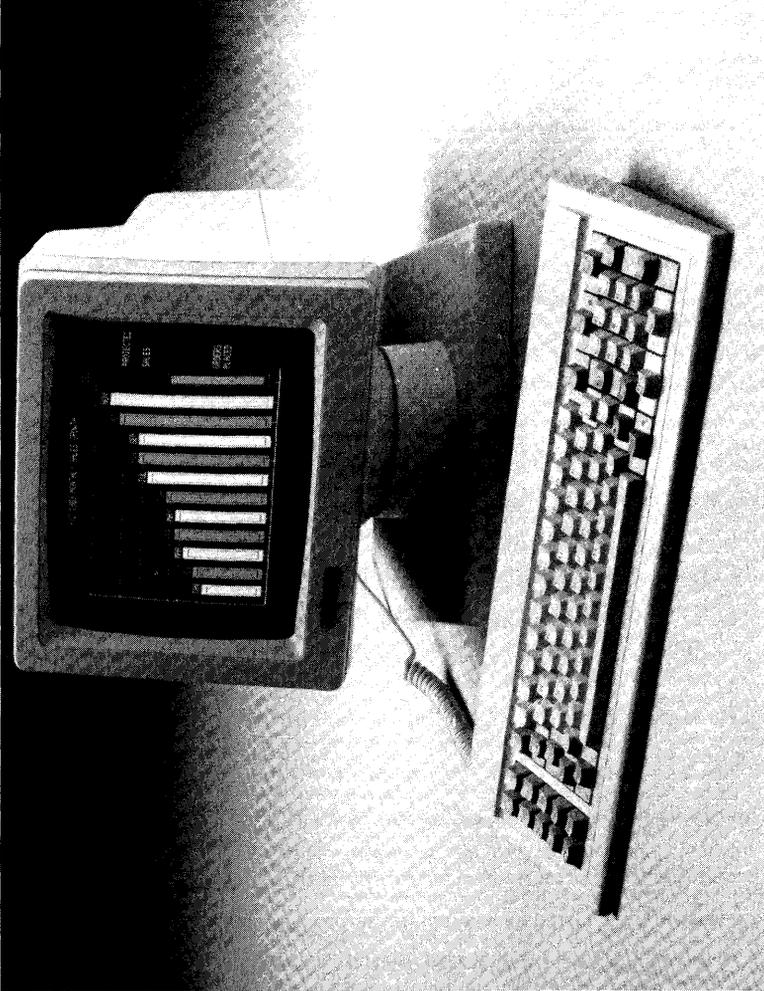


Figure 1.4. Example of Add-on Data Terminal

Although the 16-bit version of THEOS is now commercially available, the majority of THEOS systems in use are 8-bit. Therefore the majority of this book will be devoted to the 8-bit version of THEOS.

The Design Philosophy of THEOS

THEOS was designed to accomplish the following four goals:

1. Implementation on microcomputers with features found only on mini and mainframe computers.
2. Implementation on a wide variety of popular micro computers.
3. Ease of use.
4. To provide an easy and efficient means of writing, developing, and executing application software.

The first goal was easy to accomplish since the background of the THEOS designers had been almost exclusively in mini and mainframe machines. Those of you who are familiar with mini or mainframe computers may see many operating similarities in THEOS.

Implementing THEOS on a variety of machines was a little more difficult. That goal has been accomplished, however. A list of hardware manufacturers who have implemented THEOS is provided in table 1.1. New machines are being added to this list regularly.

Table 1.1. List of hardware manufacturers who have implemented THEOS

Aeon	Altos
Astrocom	California Computer Systems
Canberra Industries	Cannon
Compupro-Godbout	Cromemco
Digital Group	Digital Microsystems
Digitex	Dynabyte
Elmicron	IBC (Integrated Business Computers)
IBM	IBM PC-XT
Intertechnique	Kaypro 10
Kontron	Mercator Business Systems

Micro Universe	Morrow Designs
National Semiconductor	National Semiconductor
NNC Electronics	Northstar
Onyx	Pentel
Quay	SD Systems
Seconsia	Seiko
Sirius (Vector)	Systems Group-Measurement Systems
Tandy-Radio Shack	Televideo
Toshiba	Vector Graphic

Making THEOS easy to use was the most difficult task. A partial solution was to provide built-in help messages, using English words for the various operating system commands. However, the master stroke was the design and implementation of the EXEC Language which allows programmers, system developers, and even end-users to effectively *isolate* users from the operating system. As we will see in a later chapter, it is possible to construct menus to perform any day to day computer task.

The fourth task was accomplished by designing the operating system so that it works *with* application software to maximize speed and efficiency while minimizing memory requirements.

The THEOS Components

THEOS can be divided into the following component parts:

- Nucleus
- Device Drivers
- CSI or Command String Interpreter

The **nucleus** is a special set of instructions which provide the “intelligence” of the system. When a system is started up or “booted” it is the nucleus that is loaded into RAM. In the multi-user version of THEOS, it is the nucleus that controls the time-sharing of the micro-processor for each user and determines how much memory each user receives.

Other programs not directly part of but used in conjunction with the nucleus are the **device drivers**. Device drivers are designed to allow

the use of various types of terminals, printers, modems, etc. with THEOS. These provide a means of maximizing performance of peripherals on a particular computer system.

The **csi** or **command string interpreter** is the center or "brain" of THEOS communications. It is the key that allows you access to system and user programs; it performs all input and output functions.

To summarize, the **nucleus**, **csi**, and **device drivers** together constitute the THEOS operating system.

THEOS Hardware Requirements

The following components are required to run an 8-bit THEOS system:

- A Z80 microprocessor
- An interactive terminal/keyboard
- A minimum of 1 million character (1 megabyte) disk storage. The complete operating system itself takes up 1/2 megabyte of disk space. A hard disk is recommended to maximize storage and ensure speedy operation.
- Minimum of 64K of random access memory. The maximum amount of memory supported by the 8-bit version of THEOS is 784K.

Special Note: floppy disk based systems are not recommended with THEOS because the amount of storage is limited which slows system operation to the point where using such a system in a multi-user environment is impractical.

Application Software For THEOS

A wide variety of applications software is available for THEOS systems from special purpose medical accounting packages to the more common spreadsheet, word processing, and data base management programs. Before buying software, ask vendors if their software takes full advantage of the capabilities of THEOS. Some of the programs may just be modified versions of programs from other operating systems. Phase One Systems publishes an applications software directory quarterly which lists over 1000 business-oriented applications

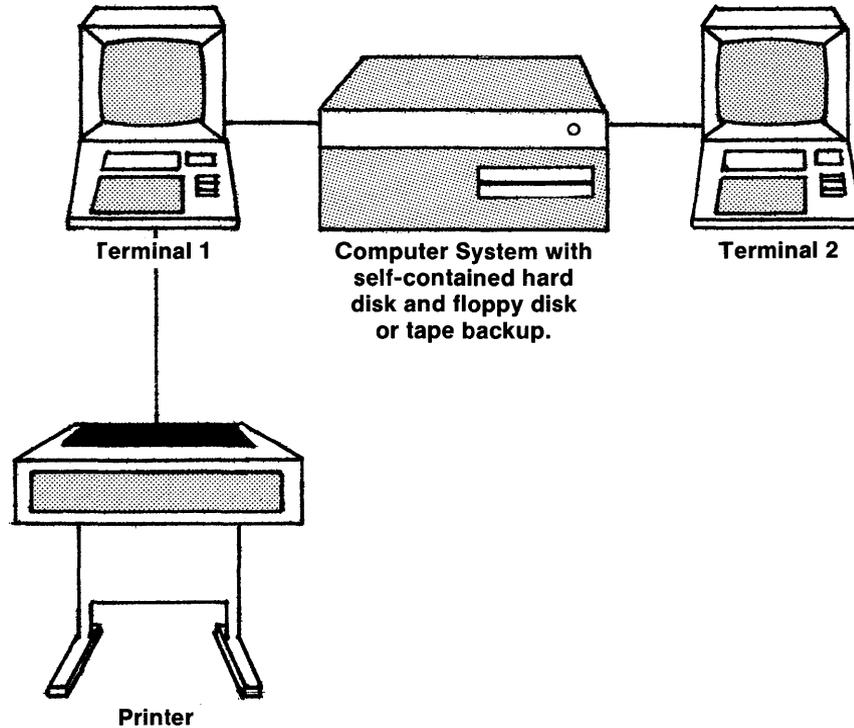


Figure 1.5. Typical THEOS system

packages from more than 200 software houses worldwide. New applications are added to the directory every quarter.

Some THEOS Statistics and Capabilities

As of this writing over 50,000 copies of THEOS have been installed at end user locations world-wide. The popularity of THEOS is due to the following attributes:

- Multi-user THEOS can support up to 16 users simultaneously (assuming the hardware can support that many). Each user appears to have his own completely independent computer system.
- There is built-in security for important files, complete with the capability of assigning passwords.
- There is a built-in facility called EXEC which allows for the creation of "menus". This permits "housekeeping" functions and applications software to be performed by making choices from menus. This facility *eliminates* the need to purchase any extra "help" programs as with other operating systems. A further benefit is that as new software applications are added, the menus can easily be modified to accommodate new programs.
- Print Spooling which allows all system users to share one printer.

All of the above features will be discussed in detail in later chapters.

Summary

THEOS was designed to function as a multi-user system and to run *only* business-oriented software such as accounting, word processing, order entry, etc. If your needs dictate a multi-user business system, you may want to consider THEOS. Even if you need only one terminal at present, you can add additional terminals (which is less expensive than buying individual computers) as the need arises.

2

An Overview of THEOS

With features such as security (provided by user accounting), accounting history, file and record locking, and an on-line help facility, THEOS is a powerful yet easy to use business operating system. This chapter will introduce these features as well as some useful supplementary programs which can improve the speed and operating efficiency of THEOS.

THEOS UNIQUE FEATURES

Security

THEOS controls access to the computer system and its files and prevents unauthorized persons from accessing data or programs while allowing authorized persons to access these same programs and data.

This feature is highly desirable in a business environment where it is necessary to control the use of certain programs and to prevent changing or possible erasure of data files.

USER ACCOUNTING

User accounting is one way THEOS provides data security. The component parts of user accounting are account assignments, file ownership, account log on, new accounts, privilege levels, synonyms, and passwords. These will be discussed here as well as in the tutorial session on the ACCOUNT command.

ASSIGNING ACCOUNTS

Each person who uses the system is assigned an account name which enables him to gain access to the system. This account name is assigned by the system manager and can be changed if necessary. The account can be visualized as a file cabinet; it contains all the data files which must be accessible to the user. Any data files not contained within the account cannot be accessed by that user.

FILE OWNERSHIP

File ownership refers to the fact that each file is "owned" by a specific account. Files or programs that all users have access to are called **public files**. Files or programs that are only accessible by certain individuals are called **private files**. Files that a specific user has given permission to another user to use are called **shared files**. The **owner** of a file is the particular user who has access to a specific file.

Private files may only be accessed, deleted, or changed by the owner of the file. Private files may be *shared* by another user if permission has been given by the owner of the file. The sharing of the file is controlled by the operating system. A shared file may be changed or updated by the non-owner, but it may not be erased by him.

A public file is a special type of shared file. A public file is owned by the "master account" called the **system account**. All users have access to these files; special permission is not needed to access them. Programs required by all system users are generally placed in the

system account. Examples of such programs would be any THEOS operating system command or any application software programs.

The primary benefit of private files is not so much to limit access to programs but rather to prevent or limit access to data files. Obviously certain data should not be available to all users, and only certain authorized individuals should be able to alter or erase important data files.

We will show you how to set up user accounts later in this book. To clarify the concept of file ownership, study the following example.

Files Owned By:

System Account	Walter	Brian
Order Entry Programs	Order Entry Data	
Inventory Programs		Inventory Data
Accounts Receivable Programs	Accounts Rcv Data	
Accounts Payable Programs	Accounts Pyble Data	Accounts Pyble Data
General Ledger Programs		General Ledger Data

Based on what we have discussed thus far, the accounts named Walter and Brian can access all of the accounting programs (system account). But only Walter can access the order entry and accounts receivable files. Only Brian can access the inventory and general ledger. Both Walter and Brian *share* access to the accounts payable file.

LOGGING ON TO AN ACCOUNT

In order to access *any* information, one must first **log on** to an account. Using our preceding example, if one logs on to the system account, the accounting programs can be run but *none* of the data files can be accessed. An example of how to log on to an account will follow shortly.

ADDING ACCOUNTS

Accounts can be added using a special operating system program called the ACCOUNT command. When a THEOS system is first installed, it has no accounts. A special THEOS command called ACCOUNT allows you to add, change, or delete accounts. It is not

mandatory to use the accounting feature in single-user THEOS, but it is *required* in multi-user THEOS. For our purposes, we will assume that user accounting is being utilized.

PRIVILEGE LEVELS

Privilege levels are another built-in protection provided by THEOS which restrict access to the computer and its programs. In the process of setting up new accounts, the system will ask what *privilege level* you wish for an account. The lowest privilege level is 0 while the highest level is 5.

The system account is the master controlling account. Its privilege level should *never* be changed. If the privilege level of this account is changed to a number less than 5, certain programs owned by the system account may *never* again be accessible by anyone! More details about *all* user accounting features will be provided in the section on the ACCOUNT command.

ACCOUNT SYNONYMS

An account synonym is merely another name for the same account. Account synonyms are a useful safeguard because they may provide backup in the event the original account was accidentally deleted. As an example, if you create a synonym account for WALTER called WALTERBA (for WALTER backup) with the same privilege level, you will have two accounts with the same privilege level, which can access the same data files.

ACCOUNT PASSWORDS

Password protection is one additional level of security provided by THEOS. All accounts have a *potential* password associated with them. When a password is assigned to an account, the system will automatically ask for it every time a log on is attempted to that account. When the person logging on enters the password, it is not displayed on the screen. Instead a series of asterisks (*) is displayed, one for every letter typed in. This prevents anyone who does not know your password from discovering it by looking over your shoulder as you enter it.

```
Logon please: SYSTEM<RETURN>
Password?***** (Assume the assigned password is LETMEIN)
Logon at 10:59:30, on 07/27/84.
          TIME    DATE
>
```

Figure 2.1. Successfully logging on to the system

To summarize, user accounting provides data security in an THEOS system. The topics discussed above are part of user accounting and are the means that user accounting uses to provide security. User accounting will be discussed in more detail in the tutorial session on the ACCOUNT command.

Accounting History

Accounting history is a special THEOS feature which allows the system to keep track of the following information:

1. Who logged on?
2. When did they log on?
3. How long were they logged on?
4. When was the system last booted?
5. When was a backup copy last made?
6. General information about the backup disk such as disk name, storage capacity, etc.

This is a useful feature but since this information is cumulative it can occupy large amounts of disk space. For this reason the feature is disabled as supplied with each copy of the operating system. If you decide to use the accounting history feature, you must enable it. (See the tutorial session on the SET command in chapter 4). The system accounting information is contained in a special file called SYSTEM.-HISTORY and must be cleared out periodically by using the ACCOUNT command.

Multi-User THEOS

THEOS versions 5.4 and above offer multi-user features. In other words, these versions of THEOS allow more than one terminal to share access to the computer at one time. In this sort of "time sharing" arrangement, it appears to each user that he or she is the only one using the system.

Although the computer itself only has one "brain" or microprocessor, the operating system allows users to have a "slice of time" allocated to them to perform their tasks. In addition, each user is allocated his or her own portion of RAM to run application software

or use operating system commands. Because of this sharing arrangement, only one copy of the operating system and application software must be purchased for the computer.

Because any microprocessor can only perform one task at a time, it is important to note that the faster the microprocessor, the faster the system will run. As an example, the 8-bit version of THEOS requires a Z80™ microprocessor. There are different versions of Z80's which run at different clock speeds. The faster the clock speed, the faster the microprocessor.

In addition, the microprocessor serves the various users by switching between them at a very high speed. It is the operating system which controls this switching. When each user has exhausted his or her time slice, the next user in line is selected.

In any multi-user arrangement certain problems may arise. First, since there is sharing of a single microprocessor, as more users log on to the system it slows down. Selecting hardware with a fast microprocessor and using a hard disk is important if this slowing effect is to be kept to a minimum. Second, a situation may arise where two users want to update the same disk file. It is also possible that two users will want to make changes to the same record within that file without knowledge of the other user. Some method of file access must be provided to avoid this **contention**. The THEOS BASIC programming language provides a means of both **file** and **record lockout**.

In THEOS, file or record contention can be avoided if the application software follows these rules:

1. Whoever gets to the file *first* automatically locks out anyone else.
2. Whoever gets to the desired record *first* automatically locks out anyone else until all the necessary changes are made.

Contention can also arise as a result of *writing* new information to the file or record. Reading the same file or record is generally not a problem. If no provisions for file or record locking existed, files would end up in a state of chaos.

Special Note: this locking mechanism is not an automatic feature of THEOS. Programs must be *written* to utilize these features that are

Type	Clock Speed (Measured in Mhz)
Z80A	4 Mhz
Z80B	6 Mhz
Z80H	8 Mhz

Figure 2.2. Chart of currently available Z80 microprocessors

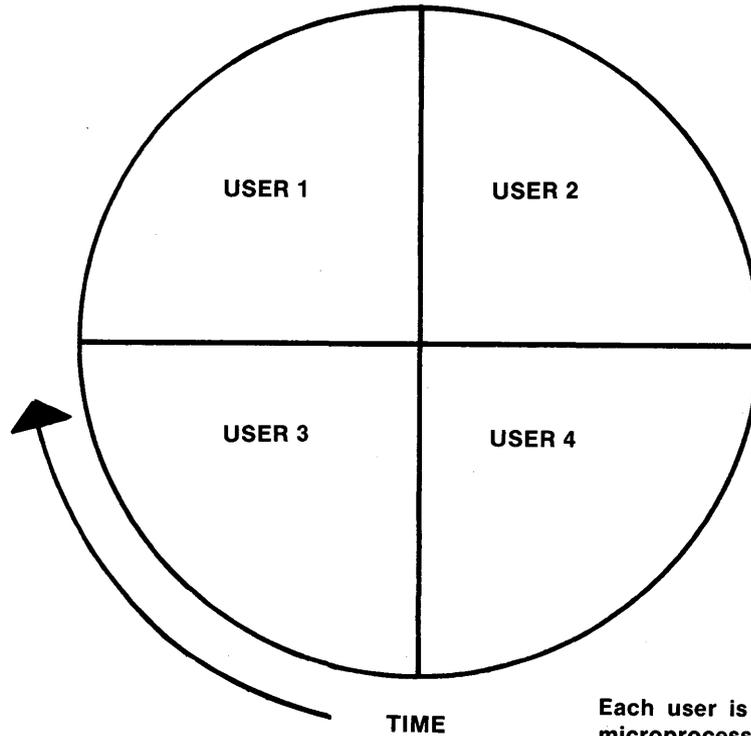
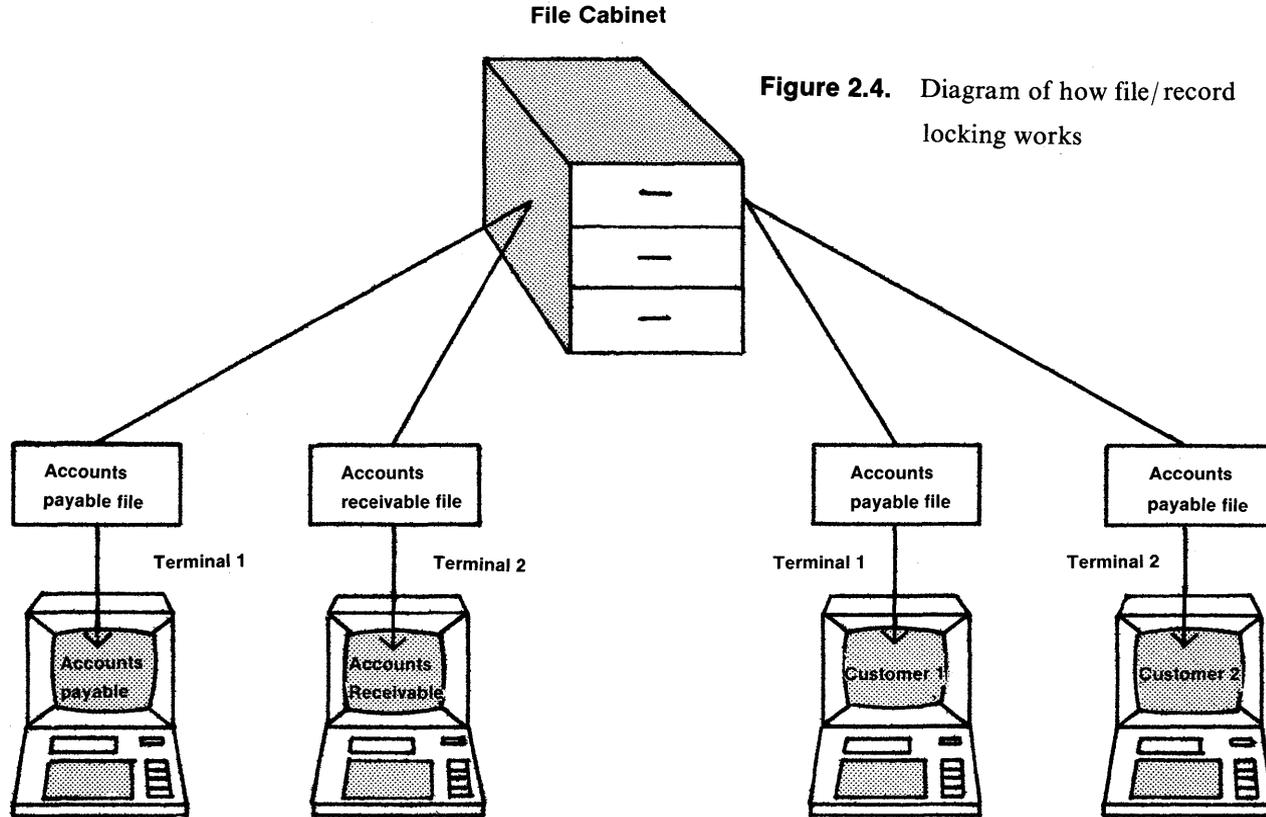


Figure 2.3. Time slice arrangement

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



In a file locking system, two users may access different files. In this example, if both users attempted access to accounts payable, the system would likely "hang."

In a record lock system, two users may access the same file, but cannot access the same record simultaneously.

part of THEOS BASIC. In addition, care must be taken when writing programs to avoid **deadlocking**. This occurs when any user is locked out of any file or record for an excessive amount of time. When looking for THEOS software, be certain you ask for these important features.

The Built In Help Facility

The built-in help feature is unique to THEOS. Whenever you see the THEOS prompt (>) on the screen, if you type the word HELP a complete list of all the operating system commands along with what they do will appear in front of you. For example if you type:

>HELP[RETURN]

A screen resembling that depicted below will appear:

Command	Function
ACCOUNT	Maintain user accounting feature
ARCHIVE	To make backup copies of hard disks
ASSIGN	Assign files or devices to I/O channels
ATTACH	Make a disk or device available to THEOS
BACKUP	Copy the entire contents of a THEOS disk to another disk
BASIC	Execute or maintain a THEOS BASIC language program

Figure 2.5. General THEOS HELP screen

All THEOS commands along with their functions will be listed in alphabetical order.

Since all of the THEOS commands will not fit on the screen simultaneously, you will see an up-arrow (↑) in the lower left-hand corner of the screen indicating that more information follows. Press <RETURN> to see the remainder of the commands. When you have

reviewed all of the commands, you will be returned to the operating system (CSI).

The CSI prompt, `>`, will then appear.

If you would like help with a specific command you can type:

`>HELP XYZ[RETURN]` where XYZ is the name of the specific THEOS command

The following information is displayed:

1. The general function of the command
2. The syntax (how to use it)
3. Any special options or functions available for use with that command

OTHER SPECIAL FEATURES OF THEOS

Some of THEOS's other features include:

- The ability to send messages from one user to another.
- The ability to share a single printer among the various users via print spooling.
- The ability to communicate and transfer data or programs via telephone modem or directly from one THEOS based computer to another.
- Built-in diagnostic programs that test memory and verify media.
- The ability to see what another user is doing via your own terminal.
- Serialized programs. Each copy of THEOS has a serial number. Most high quality application software has the serial number of the operating system entered into it. If a user attempts to use the software on a THEOS system with a different serial number, the error message "Incorrect serial number" appears on the screen. This prevents unauthorized copying of software, but allows the owner/user of the software to make an unlimited number of backup copies for archival purposes.

These features will be covered in detail in later chapters.

HELP FILELIST

Function: To list directory entries for selected or all files.

Syntax: FILELIST [fn.ft] [:fd] [(options)]

Options:

PRINTER N	Display list on printer n
EXEC	Create SELECTED.EXEC:S file
FILE	Create SELECTED.FILES:S
APPEND	Append to SELECTED.EXEC or .FILES
FN	Only list File Names
FT	Only list File Name and File Type
FD	List up through File Drive
DATE	List up through date & time
ALLOC	List up through allocation
FORMAT	List up through format
LABEL	List up through disk label (default)
SORT	Sort the list (default)
NOSORT	Do not sort the list
OWNER	List selected files from all accounts
HEAD	Display page heading (default)
NOHEAD	Do not output heading
PUBLIC	List public as well as private files
mm/dd/yy	Only include if date >= to requested
*	Include all information

>

Figure 2.6. Sample specific HELP screen

THE THEOS FILING SYSTEM

THEOS Files

THEOS files are stored on the disk drives. Each drive may contain a mixture of both program and data files. THEOS has the ability to handle up to eight drives. Each disk is maintained independently which means that a file may not start on one disk and continue on another. Each disk may handle any number of individual files up to the capacity of the disk being used.

Like other operating systems, before *any* disk can be used it must first be initialized or FORMatted using the INITDISK command (discussed in chapter 4). Disks, as received from the manufacturer, are blank and contain no information. It is this formatting process that creates a "road map" on the disk, including a directory (table of contents), disk label (to identify the various disks), and instructions which tell the computer that the disk is available for storing information and how much disk space is available. In THEOS the smallest amount of disk space any file can occupy is one data block. Each THEOS data block is equal to 1K or 1024 characters. After formatting, the disk can then be used for adding, erasing, or changing files.

Special Note: there is no limit on how many times a disk can be initialized. However if you reinitialize a disk with existing data, that data is automatically erased and cannot be recovered.

DATA AND FILE FORMATS IN THEOS

The following data and file formats are available in THEOS:

1. Sequential Files
2. Direct Files
3. Indexed Files
4. Keyed Files
5. Absolute Files
6. Relocatable Files

The first four file formats are used with data files while the final two are reserved for program and operating system commands.

PROTECTING FILES AND DISKS

THEOS's `RENAME` command can be used to prevent accidental erasure of unauthorized changes in files and disks. The `RENAME` command will allow you to:

- Prevent any program from erasing a file
- Prevent writing information to an existing file
- Prevent reading of files
- Prevent any writing on an entire disk

As we will see, since using the `RENAME` command has serious implications, it must be used cautiously. Details on `RENAME` are provided in chapter 4.

DISK SPACE UTILIZATION

THEOS is designed so that *continuous or contiguous* space on a disk is required to write files. Single files *cannot* be spread across multiple disks and file space for *most* data files is preallocated. This means that, with the exception of sequential files, empty data files are created before any application software is run.

When a new THEOS system is brought up for the first time, all disk space will be continuous. As the system is used and files are added and deleted, the file space occupied by those deleted files may not be easily reclaimed. The only way this space may be reclaimed is by first making a backup copy of that disk and then reinitializing (which will automatically *erase* existing data) on the disk. This will reclaim the space and *squeeze* everything back together. All of your files will then have to be recopied onto the newly initialized disk. There is an easier way to reclaim unusable disk space (*see* File Management Utilities at the end of this chapter).

SOME USEFUL ADDITIONS TO THEOS

Control and Sort

Control and Sort are two pieces of application software offered separately from THEOS by Phase One Systems that transform your

computer system into a powerful data base manager. Control allows you to create data entry screens that are easy to use. You can use it to create card files, inventory files, or almost any type of data entry system. Once you have created the data entry screens, fill in the data and store it. You can then design any report format you prefer. The sort program allows sorting of the information created with the control program. Sort can also be used with other application software that lacks sorting capability or sorts too slowly.

R/M Cobol™

R/M Cobol is a THEOS module available separately that is required to run programs written in the COBOL language. If you are looking at application software in the THEOS software directory, it will always indicate when R/M Cobol is required to run the software. Although COBOL is a language preferred by many programmers, some experienced THEOS users (and programmers) have found that application software written in COBOL runs substantially slower than software written in THEOS BASIC.

File and Disk Management Utilities

The special programs called file and disk management utilities save considerable time and effort. In particular three programs from Quantum Information Systems are quite helpful. These programs are REORG, SQUEEZE, and QCOPY. They are described below.

The purpose of file and disk management utilities is to provide a simple solution to problems that may occur in managing THEOS files, such as maximizing disk space, and ensuring faster disk access.

The size of THEOS BASIC data files must be preallocated, and substantial degradation in the speed with which records in the file are accessed occurs when the files become seventy to eighty per cent full. This creates a need to be able to *resize* these files as the user's requirements change. Also, when records are deleted from these files, they are not truly removed but are simply delete coded. While records that have been delete coded may not be reused, they have the same effect on access time as if the record was not deleted. This creates the need to periodically "clear out the dead weight" so that the file contains only active records.

A similar problem arises with the disk directory. Time required to access the directory will degrade when it becomes seventy to eighty percent full. At this point there is a need to reorganize the directory and remove deleted entries in order to optimize file access time. A further problem may occur as files are added and deleted from the disk; the available space on the disk may become fragmented providing very little usable space in one location (remember, THEOS requires continuous space to write files). This creates the need to be able to compress the unused space on the disk into one contiguous area at the end of the disk where it is more useful.

REORG is a program that handles changing file sizes without losing data in the file. SQUEEZE is a program that handles the disk directory and disk space reorganization. Although it is possible to reorganize a disk by reinitializing it, SQUEEZE can perform the same operation and more, in less time, without reinitializing the disk.

QCOPY is a high speed copy program which reduces the time necessary to copy files. The QCOPY program is able to copy files two to four times faster than the COPYFILE program provided by THEOS, and eliminates the problem of copying files to multiple diskettes. It is not necessary to determine *in advance* how many files will fit on a disk. The program will copy files until no more will fit. The user is then prompted to insert another diskette. The procedure is repeated until all files have been copied. Before this particular program is purchased, it is recommended that the user work with the COPYFILE command that is part of THEOS. Many users will find that COPYFILE does the job.

The Quantum utilities should be an integral part of every THEOS installation. REORG and SQUEEZE in particular can improve system speed and operating efficiency considerably.

Summary

Hardware manufacturers are constantly upgrading their computers and trying to outperform their competition. In addition, they may provide certain enhancements to THEOS that will further improve their competitive edge. As a result, it is generally required that you buy

THEOS from the same vendor as the hardware. Some hardware manufacturers supply THEOS with some additional software such as the packages described above. However, as long as the screens on your system resemble those we have described, you can be sure that the operating system you are using is THEOS.

THEOS includes features, many of which are found only on mainframe computers, that make it ideal for use in business. We have described some of these features and their benefits. You may find that certain features appeal to you more than others. As you begin to use THEOS in the next chapters, our goal is to help you feel comfortable using it and help you appreciate its advantages.

3

THEOS Conventions

Introduction

This chapter will provide the background necessary to begin using the THEOS operating system. We will begin by defining what a documentation convention is. Next, THEOS system conventions such as how disks are “searched,” how to assign names to files, using wildcards, and system control keys will be discussed. We will conclude by defining the concept of a command line and giving a few examples of how to enter THEOS commands.

In this chapter a number of screen displays will be included in the discussion. In these displays, items to be entered via the keyboard by the reader are shown in **boldface**. <RETURN> will indicate that the Return key is to be pressed.

If you have access to a computer with THEOS software, study this chapter while seated at a terminal. Enter the examples and compare the screen displays on your terminal with the illustrations provided.

THEOS Documentation

In order to use THEOS certain rules or instructions must be followed when typing commands for the computer system to interpret. The rules governing the correct way to type a THEOS command are called **documentation conventions**. It is important to follow these documentation conventions because they provide a means of communicating with the operating system. Some examples of documentation conventions are:

- Abbreviations
- Delimiters and special characters such as asterisks (*), commas (,), equal signs (=), parentheses(), periods (.), spaces (), and colons (:).

Abbreviations are the shortest form of a THEOS command. Available THEOS command abbreviations will be indicated in the tutorial section on commands.

Delimiters and special characters in THEOS are used much like punctuation is used in languages to clarify thoughts and expressions. The asterisk (*) is called a **wildcard** character and will be discussed in the section of this chapter titled wildcards. The other special characters will be discussed as required.

THEOS System Conventions

THEOS system conventions are the set of rules that the operating system uses to perform tasks such as finding files, erasing files, storing files, determining when each user can access the printer, etc. Every time THEOS is directed to perform a task, these rules are strictly followed. The conventions cannot be changed but as you will see, they can be used to improve the speed and efficiency of computer operations.

STANDARD SEARCH ORDER

Standard search order is the order in which all disk drives on the computer system are searched when a disk operation is performed. Since the access time required to locate files on disk is affected by their location, this makes standard search order an important convention. When searching for data files and programs, disks are searched in alphabetical order: A,B,C,D,E,F,G,S. If you want to maximize access time, program files and data files should be located on the disks that will be searched first!

The disk labelled S is a special disk in that it is designated as the **system** disk. The system disk can be either a floppy disk or a hard disk. It is used to boot or start the system and contains all of the operating system commands. It may contain other files as well depending upon its storage capacity.

EXCEPTIONS TO STANDARD SEARCH ORDER

One exception to the standard search order is operating system commands or special programs called **execs**. The standard search order for commands or execs is disk S followed by disks A through G.

Another exception occurs when a command is executed, a data file is written, and no disk drive is specified. In such an instance the system disk is examined for blank space. If insufficient room is available there, the other disks will be searched in alphabetical order until sufficient space is found to write the file.

PROMPTING CHARACTERS

Prompting characters are printed on the screen when the computer is waiting for an entry. This book will only be concerned with the following THEOS prompts:

- > The command string interpreter is waiting for a command.
- * TEXTEDIT or EDIT is waiting for a command.
- : An EXEC is waiting for an entry.
- ↑ A full screen of data has been displayed and the system is ready to display the next data screen. Press <RETURN>.

The > is the primary THEOS prompting character and is the one most frequently encountered by users. Examples of entries to be made in response to these other prompting characters will be given later.

FILENAME, FILETYPE, FILEDISK

A **filename** is a unique name that identifies a program or data file. A filename can consist of from one to eight characters. The first character must be a letter. A filename can only contain letters, digits, or a dollar sign.

Filetype is used to further describe the filename and indicates its function. A filetype of DATA could indicate a data file, while a filetype of COMMAND implies a THEOS command file. A filetype can consist of from one to eight characters. The first character must be a letter and like the filename, only letters, digits, or dollar signs are allowed for the remaining characters.

Filedisk is a letter assigned to a particular drive S, or any letter A through G. Generally, one need not specify a filedisk unless a file is to be written to a specific disk or file access time is to be reduced. If a filedisk is not specified, the system will search all available disks according to the rules of standard search order.

Filename, filetype, filedisk always appear in this order separated by a space () or by a period (.). Filedisk can be omitted. In such instances, the proper order is filename.filetype. The combination of filename, filetype, and filedisk together are referred to as a **file description**. Uppercase letters are always used to indicate file descriptions. Whenever the THEOS prompt is on the screen, any letters typed automatically appear in upper case.* It is not necessary to use the shift key.

* THEOS versions 6.0 and above now support either upper or lower case file descriptions.

Table 3.1. Examples of THEOS file descriptions

SAMPLE.DATA S
ASSIST.EXEC G
TERMINAL.COMMAND S
AR0IA0.DF C
LETTER1.WPF D
LETTER1.BACKUP B
ACCTING DATA A
ACCTING BACKUP

Note that either a period or a space can be used to separate the filename, filetype, and filedisk.

THEOS Reserved File Descriptions

THEOS has a number of **reserved file descriptions**. When a file description is reserved for THEOS, this means that it cannot be used to name a data or program file. Table 3.2 indicates the file descriptions reserved for THEOS.

Table 3.2. THEOS Reserved File Descriptions

IPL.EXEC
SYSTEM.NUCLEUS
SYSTEM.CSI
SYSTEM.EXEC
SYSTEM.DEV NAMES
SYSTEM.ACCOUNT
SYSTEM.HISTORY
SYSTEM.ERRMSG
SYSTEM.EXEC
SYSTEM.SPOOLER
SYSTEM.SPOOLQUE
SYSTEM.DEVn
SYSTEM.CLASSn
HELP.COMMAND

In addition there are certain filetypes reserved for use by THEOS. File descriptions with these filetypes may not be used to name data or program files.

Table 3.3. THEOS Reserved Filetypes

ASSEMBLE
BACKUP
BASIC
BASICCOM
BASICOBJ
BASICUSR
C
COBOL
COMMAND
COPY
EXEC
FORTH
FORTRAN
LINK
LOADFILE
MACLIB
MACRO
OBJECT
PASCAL
SCRIPT
SYNONYM

WILDCARDS AND OTHER SPECIAL CHARACTERS

There are a number of characters which have special meaning to THEOS. These special characters are called **wildcards**.

They are:

* ? @ # =

Because of their special use you should not use these characters in file descriptions. In fact, avoid the use of any symbols on the keyboard when naming files.

In most of the examples provided we indicate files using a specific file description. You may also use **wildcards** to specify a file or a group of files to eliminate keystrokes. Consider the following wildcard characters:

* ? @ # =

When THEOS encounters these characters as part of a filename, it replaces the filename with a sorted list of matching filenames from the disk directory.

WHAT THE WILDCARD CHARACTERS DO

- The * matches any string of alphanumeric characters in the file description.
- The ? indicates that a single character in this position will match.
- The @ indicates that a single alphabetic character in this position will match.
- The # indicates that a single numeric character in this position will match.
- The = is used only in the COPYFILE and RENAME THEOS commands and will be discussed in those sections.

The use of wildcards and special characters is best illustrated with examples.

Type the following entries shown in **boldface**.

```
>FILELIST ATTACH.COMMAND <RETURN>
```

The system will display information about the file ATTACH.-COMMAND on the screen.

```
>FILELIST A*.COMMAND <RETURN>
```

The system will display a list of all filenames that begin with the letter A and have a filetype of COMMAND.

Wildcard characters can be used in combination with each other. Consider the following examples:

```
>TEST*.F??????# S
```

This file description indicates that all files whose filename starts with the letters TEST, whose filetype starts with the letter F, has a number in the eighth position, and is on the S drive are to be included.

```
>@@@@?#*.?? *
```

This file description indicates that all files should be included whose name is at least seven characters long, of which the first four characters are alphabetic (four @'s), the next two may be any character

(??), the seventh character is numeric (#) and the eighth character may be anything including a space (*). The filetype must be two alphanumeric characters (??) and may be located on any disk (*).

To summarize, wildcard characters are a useful addition to THEOS because it is not always possible to remember an exact file description. Wildcards can be used in place of an exact file description. They provide users with a method of finding files when a memory lapse occurs, and help eliminate the burden of memorizing numerous file descriptions.

SYSTEM CONTROL KEYS

System control keys are special key sequences which perform specific functions under THEOS. There are a total of 12 key sequences that are a part of THEOS but for now we will only deal with the **system cancel key**. The key sequence ESC Q (this means hit the ESCape key and then press Q) is called the **system cancel key**. Its function is to terminate the execution of any system operation in progress, thereby returning the user to the operating system prompt >.

Caution: try not to get into the habit of using ESC Q. It should only be used when an operation can be cancelled without the risk of losing data.

THE THEOS COMMAND LINE

In the next chapter, we will present a series of tutorials designed to teach you to use the operating system. Throughout the tutorials we will consistently use the symbols and procedures outlined below:

- The ">" indicates the THEOS operating system prompt. THEOS automatically prints this character on an otherwise empty line to tell you it is ready to accept your next entry.
- A THEOS system command is a program which performs some desired function. For example, SHOW DISK is a command that displays information about the disk drives in a THEOS system. THEOS commands are shown in uppercase **boldface** type.
- A **command line** includes all the characters typed following the THEOS prompt until you press the return key to indicate you wish to terminate the entry.

- A **command line** consists of one or more distinct elements. Each element is a sequence of alphanumeric characters separated from other elements by either a space () or a period (.).
- The first element immediately after the > prompt is always the name of the command. For simple commands (such as HELP) you type the command and then terminate the command line by pressing the return key.
- For more complex commands, you type the command name, a space or a period, followed by one or more elements, which specify additional information.
- **Command operands** are keyword and variable symbols of not more than eight alphanumeric characters. These keywords and variables specify the information on which THEOS operates when it performs the command function. Examples of command operands are filename, filetype, and filedisk. Some commands require no operands while others require several.
- **Command options** are keywords used to control execution of the command. The most frequently used command options will be listed with each command in the next chapter. The first option must be preceded by a left parenthesis — a closing parenthesis is not required. Each THEOS command may have a number of options associated with it.
- **Default options** are certain command options that are performed automatically unless otherwise specified. When default options are used, it is not necessary to use a left parenthesis unless you wish to include some non-default options.
- All input to the keyboard, when typing command lines, is automatically “folded” into upper case. It is not necessary to use the shift key.

How To Enter Command Lines

Command lines are always entered in the following format:

command name operands (options...

Consider the following THEOS command line:

```
>COPYFILE FILENAME.FILETYPE S = = G <RETURN>
```

Here, ">" is the THEOS prompt; "COPYFILE" is the command name; "FILENAME, FILETYPE, and S" are operands; the "=" signs are special characters which indicate that the **same** filename and file-type are to be used when the file is copied to disk drive G.

Summary

In this chapter we have provided you with the necessary concepts to continue your study of THEOS. The best way to learn about THEOS is to use it. In the next chapter we will present a series of tutorials designed to teach you to use the operating system.

4

THEOS Commands

Introduction

The purpose of this chapter is to provide a series of tutorial sessions that will allow you to begin using THEOS. The first session will show you how to start and stop a THEOS system. The remainder of the sessions will concentrate on the other THEOS commands. In order to maximize comprehension and hold your attention, we have divided each THEOS command into a working **session**. The commands and sessions will be described in alphabetical order. 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. The chapter will conclude with a tutorial on how to load, install, and run application software.

Session 1: How to Start and Stop a THEOS System

THEOS startup is an easy procedure but may vary with the type of system you have. Please consult your computer system manual and note any special requirements for your system and how to transfer THEOS to the hard disk. The operating system is generally supplied on eight or five and one quarter inch floppy diskettes, or cartridge tape. For our purposes we will assume that you are using a hard disk system and that THEOS has been transferred to the hard disk.

Special Notes:

- Put the THEOS floppy away and store it in a safe place.
- Make extra copies of **all** of your operating system files. We'll show you how in the COPYFILE session.
- Some hardware may have unique startup instructions. Check your documentation.

The steps for starting a THEOS system are:

1. Turn on your computer, printer, and terminal. If the computer is already on, press the reset button (some computers may require pressing the reset button after power on).
2. Press the ESC key on your terminal.
3. You may see the following on your terminal:

```
XXXX Loader Prom Version 2
```

```
*
```

This message comes from the Bootstrap Loader ROM chip where the startup instructions are stored. XXXX will be replaced by the name of the hardware manufacturer. The * (asterisk) is a **prompt** from the computer indicating you are to enter information via the keyboard.

4. Since we are using a hard disk, respond to the * by typing BS (for boot system).
5. In a few seconds, the following message will be displayed on your screen:

```
Single-user THEOS version v.r -nnk  
or  
Multi-user THEOS version v.r -nnk
```

The **v** will be replaced by the current **version number**.

The **r** will be replaced by the current **release number**.

For example if Multi-User THEOS 5.6 appears on your screen, you have version 5 release number 6 of the operating system.

The **nn** will be replaced by the amount of memory on your particular machine. If the message is not displayed after a reasonable period of time after pressing BS (5-10 seconds) press the reset key and return to step 2.

6. The system will then display:

Time (HH:MM:SS)

THEOS keeps track of the time of day, but you must enter it in 24 hour format. For example if you are starting the system at 6:30 PM, you would type 18:30 (24 hour time indicates PM by adding 12 to the number of hours after 12 noon). It is not necessary to enter any leading zeroes or seconds. If keeping track of the correct time of day is not important to you, you may press the Return key in response to the time prompt. Doing so causes the system to use the time of the last system start-up.

Special notes: Not all THEOS systems support the time of day feature. In addition, some newer systems on the market have built-in real time clocks and automatically enter the correct time for you.

7. The screen will then display:

Date (MM/DD/YY)

THEOS keeps track of the date but you must first type it in. Enter the correct date. It is not necessary to enter leading zeroes. The date you enter must be a valid date and must be greater than 01/01/77. The date may be entered in any reasonable format such as July 12th, 1984. As with the time, you may press RETURN and the last date of system start-up will automatically be entered.

If your system has a built-in real time clock, the date (in addition to the time) will be automatically entered for you. If for some reason

you need to change the system time and date, consult the tutorial in this chapter on the SET command. It is permissible to omit the year. If you omit the year, the system will use the year number of the latest start-up.

8. After you have entered the date and time, the system will display the values you have entered along with the month and day of the week. The system will then look for the files it needs to turn control of the system over to you. If you have started the system successfully, you may see the following on your screen:

```
Multi-User THEOS version 5.6 — 256K
Time (HH:MM:SS) 16:30
Date (MM/DD/YY) 7/22
16:30:07 Sunday, July 22, 1984
>
```

This right facing carat is the prompt from the CSI (Command String Interpreter). Whenever you see this prompt, the system is waiting for a command. If the system was unable to find the THEOS files for startup, an error message will appear telling you which file or group of files was not found. Before re-starting the system, make sure the necessary files are in fact on disk. A complete list of THEOS operating system files can be found in your THEOS operating system manual.

To summarize, starting a THEOS system is accomplished by:

1. Turning on the terminal and computer.
2. Pressing the reset switch.
3. Pressing the ESC key.
4. Typing BS.
5. Entering the current time.
6. Entering the current date.

TURNING OFF THE SYSTEM

A THEOS system can be turned off by powering off your terminal, computer, and other components but it is *not* advisable to do this until all programs have been exited and you have logged off the system. Turning off a THEOS system in this fashion will minimize possible data loss or damage to the hardware components.

To log off the system simply type:

```
>LOGOFF <RETURN>
```

The system will respond with:

```
Logoff at TIME, on DATE, ESTIMATED TIME ON SYSTEM  
Logon please:
```

Where **TIME** is the exact time of logoff; **DATE** is the current date; **ESTIMATED TIME** is the time (in minutes) you were logged on the system.

To summarize, turning off a THEOS system is accomplished by:

1. Exiting an applications software package.
2. Logging off the system.
3. After logging off the system, turning off the computer and any peripheral equipment.

More information about logging off and exiting application software will be provided later in this chapter.

THEOS COMMANDS

As discussed in the previous chapter, the Command String Interpreter (CSI) provides the means of communication between the user and the computer system. After a successful system start is accomplished, control of the system is turned over to the CSI, and the operating system prompt (>) appears on the left-hand side of your terminal screen. It is the CSI that allows the entering of the THEOS command names and allows immediate execution of those commands when the <RETURN> key is pressed.

The <RETURN> key must be pressed after entering any THEOS command. If you forget to press the <RETURN> key, the system will not respond to your command.

THEOS Command Summary

There are a number of commands associated with THEOS, but it is beyond the scope of this user's guide to cover all of them. All THEOS

commands can be found in the THEOS documentation provided by THEOS Software Corp. All of the commands we will cover are listed in figure 4.1.

Command Name	Command Function
ACCOUNT	Maintains the user accounts
ARCHIVE	Makes back-up copies of hard disks in volumes on removable media
ATTACH	Allows precise matching of peripherals to a computer system
CHANGE	Provides a means of changing the privilege level of THEOS commands
COPYFILE	Makes copies of files on an individual file basis
CREATE	Allows creation of indexed, keyed or direct files
EDIT	Allows on-screen editing of sequential files
FILELIST	Allows on-screen or printer display of all directory entries
FORCE	Provides a means of forcing a user to perform a certain task or execute a THEOS command
INITDISK/INITTAPE	Performs initialization of all media
LIST	Allows on-screen or printer display of data file contents
LOGOFF/LOGON	Provides the only means of sign-on to user accounts
MAILBOX/MSG (MESSAGE)	Built-in message facility for all users on the same computer system
PEEK	Provides a means of viewing what another user is doing via your screen
RENAME	Provides a means of changing filename, filetype, or protection status of files
SCRIPT	Provides a means of formatting sequential files for customized printout...owner's manuals, letters, etc.
SET	Allows the changing of important system parameters like memory allocation, time, date, etc.
SHOW	Provides a means of displaying the system parameter dictated by the SET command
SPOOLER	Controls who gets to use a certain printer in a multi-user situation
START	Activates a user's portion of memory
STOP	Deactivates a user's portion of memory

SYSGEN	Allows saving of important system parameters such as peripheral attachments, time, date, etc. for future system starts
TERMINAL	Provides a means of transferring files from one THEOS system to another and provides a means of communicating with another computer via telephone lines with a modem

Figure 4.1. THEOS command summary

In each of the following sessions the use of a command with one or two options will be illustrated in detail. We recommend that you try each command with the various options. Note that some commands allow you to use more than one option at a time.

Many THEOS commands can be abbreviated. Using abbreviations, of course, facilitates keyboard entry. We'll indicate these abbreviations using bold in the following sections.

One final point to keep in mind is that THEOS commands can be entered in either upper or lower case. Lower case entries are automatically translated into upper case.

Most THEOS command options have **defaults** associated with them. This means that if you do not choose a specific command option, the computer automatically executes the option THEOS designates as default. The benefit of default options is that once you know what they are, you can save time and effort by not having to type them as part of the command line. The default options are indicated in the figures along with a description of what the options do.

Session 2: The Account Command

This command provides the only means of maintaining the SYSTEM.ACCOUNT file, which contains all of the user account information necessary for proper system operation. The SYSTEM.ACCOUNT file is not owned by any account, and it can only be accessed from the SYSTEM account.

The format of the ACCOUNT command line is:

ACCOUNT (options

Option	Function
TYPE	The computer will display the current user account file on the screen
PRINTER	The computer prints the user account file on the specified printer
CLEAR	The computer transfers the account history file to the file ACCOUNT.HISTORY. The previously existing ACCOUNT.HISTORY file is erased.
KILL	The computer disables the user accounting function
COPY=fd	Allows the computer to copy the current user account file to another system disk.

Figure 4.2. Account Options

If the ACCOUNT command is used without any options, interactive account maintenance is automatically invoked. Account maintenance allows addition, deletion, changing of account names, passwords, and the assigning of privilege levels.

Before setting up an accounting structure, it is desirable to use the ACCOUNT command with the TYPE option to examine any existing accounting structure. Type the following:

```
>ACCOUNT (T <RETURN>
```

Figure 4.3 shows a typical display of user account data.

Account is the user account name you assign. Password is the password you assign. Priv is the privilege level you assign. Id is the identification number assigned by the system. You have no direct control over how this number is assigned.

ID NUMBERS

Id numbers are the means THEOS uses to determine file ownership. The SYSTEM account and any synonyms of the SYSTEM account all have an Id of 0. In the list above, IPL, KISH, PAUL, SYSTEM, and SYSTEMBA have an Id of 0. SYSTEM is the main account; the other accounts with an Id of 0 are all **synonyms** of the account called SYSTEM.

Account	Password	Priv	Id
BARBARA		5	1
DDEMO		5	5
DEMO		5	12
IPL		5	0
KISH	KISH	3	0
MED2		5	2
MEDVII		5	2
NEW		5	13
PAUL	PAUL	3	0
SANDY		1	1
SYSTEM		5	0
SYSTEMBA		5	0
USER1		5	1
USER10	USER10	5	10
USER11		5	11
USER12		5	12
USER2		5	2
USER3		5	3
USER4		5	4
USER5		5	5
USER6		5	6
USER7		5	7
USER8		5	8
USER9		5	9
WALT		5	1

Figure 4.3. Display of user account data

In a new THEOS system the first Id number assigned, after the SYSTEM account which has an Id of 0, is 1. The next Id number assigned is 2, 3, 4, etc. up to the number of account names on the system. In the next section where setting up accounts is discussed, we will provide more insight into the Id numbering system.

SYNONYMS

In figure 4.3, IPL, KISH, PAUL, and STSTEMBA are unique account names, *but* they are *synonymous* with SYSTEM. If you want unique account names and passwords when creating new accounts, but you also want those accounts to share data or programs with another account, make them **synonym accounts**. The system will automatically assign them the SAME Id number as the existing account. Refer to the following examples.

PRIVILEGE LEVELS

Privilege levels for each user account are assigned by the system manager during account maintenance. These levels range from zero to five with zero being the lowest and five being the highest. All valid THEOS commands require that the user has a certain privilege level in order for them to be executed. Erasing files for example requires a privilege level of 1. If a privilege level of 0 is assigned to a user and that user tries to use the ERASE command, the "Insufficient privilege" error message will appear on the screen. Privilege levels can be changed by the system manager. *See* the section on the CHANGE command.

MAINTAINING USER ACCOUNTS

A privilege level of 5 is required to maintain user accounts. To maintain user accounts type:

>ACCOUNT or AC <RETURN>

The type of maintenance performed is determined by the answer to the question:

Mode (Add/Change/Delete/End)?

If you want to add an account type A for add.

If you want to change an account type C for change.

If you want to delete an account type D for delete.

If you want to end account maintenance type E for end.

Let's assume we are setting up a new THEOS system and we want to add some accounts. There are no accounts on the system other than

the system account and its synonyms (all have Id numbers of 0). Choose A for add accounts.

Mode (Add/Change/Delete/End)? **A** <RETURN>

The following will be displayed on your screen. Note the entries that you should type in response to the questions. (User entries are displayed in bold.)

Account name? **TEST** <RETURN>
Synonym name? <RETURN>
Password? **LETMEIN** <RETURN>
Privilege? **0** <RETURN>

Mode (Add/Change/Delete/End)? **A** <RETURN>
Account name? **TEST1** <RETURN>
Synonym name? **TEST** <RETURN>
Password? **LETYOUIN** <RETURN>
Privilege? **1** <RETURN>

We have now created two user accounts. The first account called TEST is not a synonym of any other account; it has a password of LETMEIN and a privilege level of 0. Since it was the first account added that is not a synonym of the system account, THEOS assigned it an Id of 1.

The second account we called TEST1. Because TEST1 is a synonym of TEST, it was assigned the same Id, 1.

Note: Synonym accounts and Id numbers are related in that synonym accounts have the *same* Id number as their synonym. If a new account has no synonym, it is assigned the next sequential Id number.

We assigned the account called TEST1 a password of LETYOUIN and a privilege level of 1. It is possible to have different passwords and privilege levels for synonym accounts.

Next, let's see what we need to do to change an account. After we have finished adding all of our new accounts, the same question will appear on your screen:

Mode (Add/Change/Delete/End)?

This time type C for change and <RETURN>. The following will appear on your screen:

```
Account Name? TEST1 <RETURN>
Password? # <RETURN>
Privilege? <RETURN>
```

First type in the name of the account to change, in this case TEST1. In response to the Password question, we typed the pound sign. This special character means that we want to delete the original password. To change the password, we could have typed a new password and then pressed <RETURN>. In response to the privilege question, we press <RETURN> unless we want to change the privilege level. When we finished making our entries, we are returned to:

```
Mode (Add/Change/Delete/End)?
```

DELETING AN ACCOUNT

If you wish to delete an account, you can type D and the computer will respond with:

```
Account name?
```

Enter the name of the account you wish to delete, press <RETURN> and you are returned to:

```
Mode (Add/Change/Delete/End)?
```

Special Note: Careless deleting of accounts may result in loss of access to certain files. Be sure that a synonym account exists (with the same Id number as the deleted account), *or* that no files belong to the deleted account. If you are not certain, do not delete the account.

ENDING ACCOUNT MAINTENANCE

If you type E <RETURN> in response to the mode question, all changes made to the SYSTEM.ACCOUNT files are saved to disk, and you are returned to CSI. If you type the system cancel key (ESC Q)

before you type E, none of the additions or modifications you made . will be saved.

USE OF OTHER ACCOUNT COMMAND OPTIONS

This command does not permit the use of multiple options. To output the user account data to a printer type:

>ACCOUNT (PRINTER <RETURN>

To copy the user account data to a backup disk type:

>ACCOUNT (COPY=fd<RETURN>

In place of fd, type the drive name (A through G) that the file is being copied to.

To clear out the ACCOUNT.HISTORY file type:

>ACCOUNT (CLEAR <RETURN>

SUMMARY

The sole purpose of the ACCOUNT command is to maintain the user account file.

In order to use this command, you must be logged onto the SYSTEM account. This command has few options associated with it because most of the work is done in the interactive mode of updating and changing the contents of the file. Exercise caution when using this command and do not delete any accounts unless you are certain at least one synonym account exists.

Session 3: The ATTACH Command

The ATTACH command is used to “connect” peripherals such as disk drives, printers, terminals, modems, etc. to the computer system. It is this command that allows precise matching of these hardware accessories to the computer. If one of these peripherals is not *attached*, the computer will not be aware of its presence, and it cannot be used until it is *attached* via this command.

The ATTACH command completes the “connection” between the microprocessor and each peripheral. The only permanent attachments illustrated here are the terminal on Port 1 and HARD DISK PLATTER S.

The format of the ATTACH command is:

ATTACH logdev phydev (options)

logdev is the logical device name of the peripheral being attached. phydev is the actual name of peripheral such as COMM (for a modem) or CONSOLE (for a terminal).

THEOS has a reference table of terms to use for peripherals. The most commonly used terms are shown in figure 4.4.

LOGDEV	PHYDEV
A thru G or S	DISK1 thru 8 if hard disk or FLOPPY1/FLOPPY2 if floppy disk
CONSOLE	PORT1, PORT2, PORT3, etc.
PRINTER1	
PRINTER2	
PRINTER3	
PRINTER4	
COMM1	
COMM2	
COMM3	
COMM4	
TAPE1	CART or REEL
TAPE2	
TAPE3	
TAPE4	

Figure 4.4. THEOS reference table of peripheral terms

Each computer system is supplied with a special file where the ‘logdev’ names are stored. The name of this file is SYSTEM.DEV-NAMES. The contents of this file may vary slightly from one system to another due to hardware differences.

```
SYSTEM.DEVNAMES:HARDISK1  
  
FLOPPY1 1 D 1 2  
FLOPPY2 2 D 1 2  
DISK1 3 D 3 4 5 6 7 8  
DISK2 4 D 3 4 5 6 7 8  
DISK3 5 D 3 4 5 6 7 8  
DISK4 6 D 3 4 5 6 7 8  
DISK5 7 D 3 4 5 6 7 8  
DISK6 8 D 3 4 5 6 7 8  
CART 25 T  
REEL 26 T  
PORT1 18 CPSIO  
PORT2 17 CPSIO  
PORT3 16 CPSIO  
PORT4 15 CPSIO  
PORT5 14 CPSIO  
PORT6 13 CPSIO  
PORT7 12 CPSIO  
PORT8 11 CPSIO  
PORT9 10 CPSIO  
PORT10 9 CPSIO  
BISYNC 27 CSIO  
CENT 28 PO
```

Figure 4.5. Sample listing of SYSTEM.DEVNAMES

Note: It is possible to *permanently* attach peripherals via the SYSGEN command described later. The SYSGEN command eliminates the need to *attach* peripherals that are regularly used such as disk drives and printers.

Other ATTACH command options are available besides these listed here. We will limit ourselves to the ones used most frequently. Refer to the THEOS System Reference Manual for other options.

Option		Function
Bnnnn	n is the baud rate*	Sets the baud rate * of the serial device being attached. When the baud rate is not specified, the default value is 9600.
Cn	n is the class code**	Determines the class of the terminal or printer being used. Requires the use of a special program called a device driver .
ENABn	n is a number 1-4	Determines how characters are transmitted through the serial ports.
Lnnn	n is line width	Determines the width of a line displayed on a terminal or printed on a printer.
Pnnn	n is page length	Determines the page size of the printer or terminal being attached.
PE		Indicates that the attached serial device requires even parity for communication.
PO		Indicates that the attached serial device requires odd parity for communication.
PUBLIC		Specifies that the disk being attached can be accessed by anyone using the system.

Figure 4.6. ATTACH Options

To see what peripherals are currently attached in a THEOS system type:

>ATTACH <RETURN>

The system will display the currently attached peripherals.

```
ATTACH
Name      Device  Options
S         DISK1   "D1      ",avail=6621K,dirsize=1944,Public
CONSOLE   PORT1   L79,P23,C31,LF0,FF0
PRINTER1  PORT6   L80,P56,C0,LF0,FF0
```

Figure 4.7. THEOS system attachment table

* Baud rate is the speed at which the computer transmits data to a peripheral or vice versa. Some computers cannot use THEOS to determine the baud rate. In those cases, the baud rate is determined by switch settings inside the computer and THEOS ignores the baud rate option.

** Class code is an example of a device driver that precisely matches the characteristics of a printer or terminal to the computer.

In the figure on page 70, the logdev is S; the phydev is DISK1. avail means that there are 6976 Kilobytes of available disk space. dirsiz means that there is room in the disk directory for a total of 1,944 files. Public means that the S drive is accessible by any user logged onto the system. Console is the logdev, PORT1 is the phydev. L79 is the width of a line (79 characters). P23 is the page length (23 lines to a terminal screen page). C31 is the class code required for the terminal in use, in this case a Televideo 950 or its equivalent.

THEOS CLASS CODE

Special care must be taken when selecting a terminal to use with a THEOS system. It is best to select a terminal that has a predefined THEOS class code. If such a terminal is not available, the next best choice is a terminal that is capable of *emulating* a terminal that has a THEOS class code. A partial list of THEOS class codes can be found in appendix F.

ATTACHING PERIPHERALS

The following example illustrates the attachment of a peripheral, specifically a terminal, to port 1 on the computer.

```
>ATTACH CONSOLE PORT1 (C7, L79, P23 <RETURN>
```

Since the C (Class) option was used, THEOS knows that you are attaching a terminal. Since the baud rate was not specified, the default value of 9600 was used. Most terminals require a line width of 79 characters and a page length of 23 lines.

As another example, type:

```
>ATTACH PRINTER1 PORT6 (L80, P56<RETURN>
```

This example illustrates the attachment of PRINTER1 to PORT6. PRINTER1 is the logdev, PORT6 is the phydev. L80 is the width of a line printed, P56 is the page length (56 lines to a printed page). No special class code was specified for the attached printer. The options

indicate that the printer will print a line that is 80 characters wide and a page that is 56 lines long.

THE DETACHING OF PERIPHERALS

Detaching a peripheral is accomplished by using the ATTACH command with no phydev specified. For example to detach the printer in the preceding example type:

```
>ATTACH PRINTER1 <RETURN>
```

The system will then return you to CSI. To reattach the printer type:

```
>ATTACH PRINTER1 PORT6 <RETURN>
```

Again you will be returned to CSI.

SPECIAL NOTES REGARDING THE ATTACH COMMAND

- Disk drives may be attached out of sequence. In other words if you have one drive currently attached (S) and you want to attach a floppy drive (G), you may attach G without first attaching A through F.
- The system disk may not be detached. If you type:

```
>ATTACH S <RETURN>
```

The message “system disk cannot be detached” will appear on your screen. It is possible to attach a floppy disk as the system disk. To do so you must insert the new system disk in the drive you wish to use. Otherwise the “Disk not ready” error message will appear on your screen when you attempt to access the drive. After you type,

```
>ATTACH S FLOPPY1 <RETURN>
```

The computer will search the newly attached disk for all operating system files. A SYSTEM.NUCLEUS file of the same version number as the previous system disk must exist on this new system disk.

- When a device is attached in multi-user mode, it is considered a private device. Only the user attaching the

device (and its account synonyms) may access it, unless the device was attached using the PUBLIC option.

- All peripheral device attachments are SYSGENable.
- Check your computer system manual to see what ATTACHments are SYSGENed as standard equipment on your computer system. Also check the SYSTEM.DEV-NAMES file to check the logdev names for your attachments.

SUMMARY

The purpose of the ATTACH command is to notify THEOS that peripherals are being attached to the computer. It is important to remember that peripherals may be physically attached to the computer but unless the ATTACH command is used, none of the peripherals can be used.

Session 4: The Archive Command

The ARCHIVE command provides the means of making back-up copies of files that are too large to fit on a single disk or are too numerous to fit in the disk directory of a floppy disk. In short, it provides the only effective means of backing up important files on removable media such as floppy disk or tape cassettes. The ARCHIVE command has no privilege value associated with it and no abbreviated form is available.

The format of the ARCHIVE command is:

ARCHIVE FILENAME FILETYPE DRIVE1 DRIVE2 (options)

FILENAME is the name of the file to be archived and FILETYPE is the type of the file to be archived. Wild cards are allowed for both FILENAME and FILETYPE.

DRIVE1 is the drive that contains the files to be archived. DRIVE2 is the drive that is to receive the archived files. It is not permissible to use wildcards for DRIVE1 and DRIVE2. Both DRIVE1 and DRIVE2 must be specified.

Option	Function
ACCOUNT	Files owned by the account you are logged onto are archived. This is the default option when FILENAME and FILETYPE are specified.
VOLUME	Files of all accounts are selected for archiving. This is the default when FILENAME and FILETYPE are omitted.
TYPE	The computer displays the file descriptions of the files being archived. This is a default when archiving to floppy disk. In addition, the sector number (location on the disk) is displayed next to each file.
NOTYPE	File descriptions are not displayed during the archive process.
VERIFY	Computer verifies that files were written on the media used. This is a default when archiving to disk.
NOVERIFY	No verification is performed.
QUERY	The system manager is asked on a file-by-file basis if a file is to be included in the archive. A reply of Y is required to archive the file. A reply of N or <RETURN> means that the file is <i>not</i> included in the archive.
NOQUERY	Files specified are automatically archived without asking.
date1	The first date specified as an option means that those files whose date was on or after the date specified will be archived.
date2	A second date specified after the first date means that the files whose date was on or before the specified date will be archived.
ASK	The system manager will be asked prompting questions before archiving begins.
NOASK	The system manager will not be asked prompting questions before archiving begins.

Figure 4.8. Archive options

Note: Media being used for archiving should not contain information you wish to save, as archiving erases all files on the media including the directory. It is required that the media be initialized before being used for archiving. If the media contains files previously

archived, the media can be reused as long as you are aware that those previously archived files will be erased.

Since the ARCHIVE command is specifically designed to make back-up copies of hard disks onto removable media such as floppy disks or tapes, it will automatically create as many volumes of removable media as required to complete the archive. This means that you should always have enough initialized floppy disks or tape on hand to complete the archive.

ESTIMATING THE NUMBER OF VOLUMES

There is no specific formula for determining the number of volumes required for an archive. Therefore accurately estimating the number of volumes requires some first-hand experience. A good rule of thumb is to have enough removable media available to equal the amount of disk space occupied by the files you are archiving.

As an example, to archive a five megabyte hard disk, prepare five one megabyte floppy disks (or ten 1/2 megabyte floppies).

Archive Command/Options	End Result
>ARCHIVE S G	Archives all of drive S to drive G and displays the files as each is archived. Verification is performed if archiving to floppy disk.
>ARCHIVE *.ACCTDATA S G (08/13/84)	Archives * (all) ACCTDATA files on drive S to drive G that were updated on or after August 13th, 1984.
>ARCHIVE *.WPF S G (QUERY)	All (*) files with a filetype of wpf will be archived to drive G. The system will ask if each file with a filetype of wpf is to be included.
>ARCHIVE *.ACCTDATA S G (08/13/84 NOQ)	Archives * (all) ACCTDATA files on drive S to drive G that were updated on or after August 13th, 1984. The NOQ option means that files are automatically archived without asking.

Figure 4.9. Using the ARCHIVE command and its options

To make a complete archive of the S drive, type:

>ARCHIVE S G <RETURN>

The computer will respond:

Source on drive S(3)
Destination on drive G(1)
Mount volumes now — <RETURN> or Y

You must now insert some type of removable media (floppy diskette or tape) and type Y or <RETURN>.

The computer will respond:

Source disk is labeled "D1 "
Destination is labeled "TEST "
Okay to start archive (Y/N)? Y
Selecting and sorting files
1303 files selected

The preceding display indicates that there are 1303 files on the disk to be archived. The files are sorted alphabetically within each account. If you have over 500 files on disk, a few minutes will elapse between the "Okay to start archive" message and the "Selecting and sorting files" message. Don't become impatient. It takes time to alphabetize that many files.

```
ARCHIVE S G

Source on drive S(3)
Destination on drive G(1)
Mount volumes now -

Source disk is labeled "D1"
Destination is labeled "TEST"
OK to start archive (Y/N)? Y

Selecting and sorting files.

1357 files selected.

BOV: TEST      10/30/84 11:54:36 from disk D1      volume 01
Account: SYSTEM (0)
Account: SYSTEMBA (0)
Account: IPL (0)
Account: PAUL (0)
File: $IPL.EXEC,          at sector 5
File: ACCOUNT.COMMAND,   at sector 11
File: ARCHIVE.COMMAND,   at sector 24
File: ASM.EXEC,          at sector 52
File: ASSIGN.COMMAND,    at sector 62
File: ASSIST.BACKUP,      at sector 65
File: ASSIST.EXEC,        at sector 66
File: ATTACH.COMMAND,    at sector 67
File: BACKUP.COMMAND,    at sector 89
File: BARBARA.EXEC,      at sector 112
File: BASCONV.COMMAND,   at sector 125
File: BASIC.COMMAND,     at sector 144
File: BASIC.LOADFILE,    at sector 148
File: BASIC.OVERLAY1,    at sector 256
File: BASIC.OVERLAY2,    at sector 358
File: BASIC.OVERLAY3,    at sector 392
File: BASIC.OVERLAY4,    at sector 431
File: BASIC2.LOADFILE,   at sector 557
File: BISUB.EXEC,        at sector 651
File: CES.BACKUP,        at sector 674
File: CES.EXEC,          at sector 681
File: CHANGE.COMMAND,    at sector 691
File: CLASS.MACLIB,      at sector 693
File: COPYBOOT.COMMAND,  at sector 727
File: COPYFILE.COMMAND,  at sector 729
File: CPY.EXEC,          at sector 754
File: CRAA.EXEC,         at sector 764
File: CREATE.COMMAND,    at sector 770
File: CRT.COMMAND,       at sector 776
```

Figure 4.10. Example of ARCHIVE screen

Even though the TYPE option was not used in the preceding ARCHIVE command, it is a default value and the file descriptions (FILENAME.FILETYPE) along with the file sector numbers (the physical locations on the removable media) will be displayed on the screen.

BOV: TEST is the back-up volume disk name. The system provides the date, time, and disk being archived from as well as which volume of media is currently being used, in this case volume 01. As more volumes are required for back-up, you will be prompted to insert the next volume and asked if you wish to continue the archive process.

In a complete back-up the system will back-up all files belonging to all accounts beginning with those accounts whose Id is 0. When all files with an Id of 0 have been archived, the system archives files of all Id numbers until all accounts have been archived. When the archive is complete, you will be returned to CSI.

SPECIAL NOTES ABOUT ARCHIVING

- Archives must generally be performed in the single-user mode.
- Archives may not be performed unattended. Someone must be present to insert volumes of removable media when prompted, or in case errors occur during the archive process.
- The directory of each archived volume contains only one entry.
- It is a good idea to have more than one archive. In other words, make a second backup.

SUMMARY

The purpose of the ARCHIVE command is to make backup copies of files onto removable media. ARCHIVE has a number of options associated with it that allow you to backup files on a global or selective basis. It is suggested that you practice making archives and feel comfortable with the procedure before installing application software or creating data files.

Session 5: The RESTORE Command

This command is the mirror image of the ARCHIVE command. The RESTORE command provides the ability to retrieve archived files. Files may be restored individually, in a group, by account, or using other options as described in figure 4.11. Like the ARCHIVE

command, RESTORE has no privilege value associated with it. In order to use RESTORE, you must be logged onto the SYSTEM account and be operating the computer in single-user mode.

The format of the RESTORE command is:

```
RESTORE FILENAME.FILETYPE DISK1 DISK2 (options
```

FILENAME.FILETYPE are the files to be restored (wildcards are allowed). DISK1 is the disk the files are being restored from. DISK2 is the disk which is to receive the restored files. DISK1 and DISK2 must be specified and wildcards are not allowed.

Option	Function
ACCOUNT	Files owned by the currently logged account will be restored. This is the default option when FILENAME.-FILETYPE is specified.
ASK	Allows the system manager to be prompted before the restore process begins. This is the default.
NEWACC xxx	Only those files owned by account xxx are to be restored to the currently logged account.
NEWFILE	Files are to be restored only when there are no existing files of the same name in the owning account. This is a default.
NOASK	Suppresses questions specified by ASK above.
NOQUERY	System manager is not asked if it is OK to include files in the restore process. This is a default.
NOTYPE	File descriptions being restored are not displayed on the screen during the restore.
NOVERIFY	No verification of files is performed during the restore procedure.
OLDFILE	Files will only be restored when a file already exists with the same name.
QUERY	System manager is queried on a file-by-file basis during restore. A response of Y is required to restore a file. A response of N or <RETURN> advances to the next file query.
REPLACE	Restored files are to replace any existing files of the same name.
TYPE	File descriptions of each restored file are displayed on the screen during the restore operation. This is a default.

VERIFY	Verification of the restored files is performed. This is a default.
VOLUME	All files without regard to the owning account are restored. This is a default when no FILENAME and FILETYPE is specified.

Figure 4.11. Restore options

Note: When restoring from a multiple volume set of disks, each archive volume is restored one at a time *in sequence*. You are prompted on the screen during this procedure as with the ARCHIVE command. You cannot restore volume 01 first and then skip to volume 04. Even when restoring individual files, all volumes are searched in sequence until they have been examined for the files to be restored.

Let's try some RESTORE examples. Assume we have previously archived some accounting data files to a floppy disk drive labelled G, which we now wish to restore. Since our floppy is only used for backup purposes, it is not always attached. Type:

>ATTACH G FLOPPY1 <RETURN>

Now type:

>RESTORE G S (NOQ <RETURN>

The system will respond and you will type:

Source on drive G(1)
 Destination on drive S(3)
 Mount volumes now — **Y or <RETURN>**

The system will respond:

Source disk is labeled "TEST "
 Destination is labeled "D1 "
 Archive is from disk "D1 " on 08/15/84 (or current
 date) at 14:14:49 (or current time)
 Ok to start restore (Y/N)? **Y <RETURN>**

The system will display the following:

```

Account:User10 (10)
File 1 not restored because file exists
File 2 not restored because file exists
File 3 not restored because file exists
File 4 not restored because file exists
File 5 not restored because file exists
.
.
.
File 20 not restored because file exists
No files restored

```

Figure 4.12. Screen display of files being restored

```

>RESTORE G S (NOQ REP
Source on drive G(1)
Destination on drive S(3)
Mount volumes now -

Source disk is labeled "TEST      "
Destination is labeled "D1      "
Archive is from disk "D1      " on 08/15/84
Ok to start restore (Y/N)? Y
Account:  USER10 (10)
      File: PAUL.EXEC:S
      File: WALT.EXEC:S
      File: APOIAO.PM:S
      File: OASIS.MWRITE:S
      File: LEDGER.BAL$BARN:S
      File: SELECTX.COMMAND:S
      File: MWRITE.COMMAND:S
      File: DATAFILE.A:S
      File: DATAFILE.B:S
      File: OMOIM1.GL:S
      File: SPAD.COMMAND:S
      File: ORGANIZE.EXEC:S
12 files restored.
>

```

Figure 4.13. Screen display of files being restored (NOQ REP options

In the preceding example we wanted to restore all files on disk G to drive S, and we did not want to be QUERied on a file-by-file basis. In this case we happened to be logged onto an account called USER10. Because the ACCOUNT option is a default, only files belonging to USER10 were restored. The reason being that those were the ONLY archived files on drive G.

As you can see from the screen display, no files were actually restored. The reason for this is that files of the same name already exist on the S drive. If one of those files became damaged or unusable for some reason, how could you restore that file from the backup without the "file already exists" message being displayed.

Try this:

>RESTORE G S (NOQ REP

This time, all files were successfully restored because we used the REPLACE option. This means that even though the files were already present, the REPLACE option tells THEOS to replace them anyway. If we wanted to be asked if it was Ok to replace each file, we could have typed:

>RESTORE G S (REP

Since QUERY is a default, we would have been asked on a file-by-file basis if it is Ok to restore the files.

MORE ON THE RESTORE COMMAND AND OPTIONS

RESTORE

Command/Options	End Result
RESTORE G S (NOQ NEWFILE	Files on drive G will be restored to drive S only when there are no files of the same name (NEWFILE) existing on S. NOQ tells THEOS not to ask on a file-by-file basis if it is Ok to restore.
RESTORE G S (NOQ NEWA USER1	Files on drive G will be restored to drive S which belong to the account called USER1 (NEWA option).

NOQ tells THEOS not to ask on a file-by-file if it is Ok to restore.

SUMMARY

To summarize, the RESTORE command is the mirror image complement of the ARCHIVE command. RESTORE and its options allow archived files to be retrieved on a global or selective basis. It is suggested that before a THEOS system is loaded with application software or data files, one spend time practicing the ARCHIVE/RESTORE procedures.

If a tape drive is used for backup purposes instead of a floppy disk, TAPE1 can be substituted for the floppy we have designated as drive G.

Session 6: The CHANGE Command

This command provides you with the ability to change the privilege level required to use THEOS commands. We have said that every THEOS command has a privilege level associated with it. Privilege levels are designed to keep unauthorized users from executing THEOS commands.

The system manager is automatically assigned a privilege level of 5 so that any THEOS command can be executed. The system manager can change the privilege level required to execute any command. In order to use the CHANGE command and change privilege levels, your privilege level must be at least 3 and equal to or greater than the command privilege level you are trying to change.

The format of the CHANGE command is:

CHANGE FILENAME.FILETYPE (PRIV n)

FILENAME is the THEOS command name to be changed. FILETYPE is COMMAND. PRIV is the option. n is the new privilege level for that command.

Option	Function
PRIVILEGE	To change the privilege level of the THEOS system commands.

Figure 4.14. The CHANGE command option

As you can see there is only one option for the CHANGE command. This option be used in order to execute the command. If this option is not used, the error message "Syntax error" will appear on your screen.

To change the privilege level of a THEOS command such as COPYFILE, which normally has a privilege level of 1, type:

```
>CHANGE COPYFILE.COMMAND (PRIV 2<RETURN>
```

COPYFILE is the command whose privilege level is to be changed. It is not necessary to specify a drive. THEOS will search all of the attached drives until the COPYFILE command is found. PRIV is the required option. 2 is the new privilege level required to execute the command.

Specifying a drive is optional. THEOS will search all attached drives for the file COPYFILE.COMMAND.

SUMMARY

The CHANGE command is intended for use by the system manager. Its sole purpose is to change the privilege levels required to execute THEOS commands.

Session 7: The COPYFILE Command

This command provides you with the ability to make copies of all files stored on a disk drive. Unlike the ARCHIVE command, which copies files in compressed form and creates multiple volumes, COPYFILE copies files one at a time in their entirety. When the COPYFILE command is used to copy a file, the new file occupies the *same* amount

of disk space as the original file. The new file also becomes part of the disk directory on which the file is located.

COPYFILE provides a means of copying one file to another, groups of files from one disk to another, as well as selected records of one file to a new file. A privilege level of 1 is required to use COPYFILE. The "Insufficient privilege" message will appear if execution of the COPYFILE command is attempted by a user with a privilege level less than 1.

The format of the COPYFILE command is:

COPYFILE Inputfile Outputfile (options)

The Inputfile is the original file to be copied, including the FILENAME.FILETYPE and drive. The Outputfile is the new file copied by the COPYFILE command (the destination file), including the FILENAME.FILETYPE and drive.

Option	Function
NEWDATE	The Outputfile will use the current system date as the date of its directory entry.
NEWFILE	The Outputfile will not replace any preexisting file on disk of the same name. This is a default option.
NOQUERY	The system operator will not be asked if it is ok to copy a file. This is only a default option when the Inputfile is specific and not a wildcard such as *.
NOTYPE	This option tells the computer not to display the results of each successful copy operation on the screen.
NOVERIFY	Tells the COPYFILE command not to verify the copy question. This is a default option.
OLDDATE	Tells the system to use the same date for the directory entry of the Outputfile as the Inputfile.
QUERY	The system operator is asked on a file-by-file basis if it is ok to copy a file. A reply of Y is required if the file is to be copied. This is a default if wildcards are used.
REPLACE	The Outputfile can replace any preexisting file of the same description. However, any files that are protected from deletion may not be replaced. Refer to the section on the RENAME command.

TYPE	The system will display the results of a successful copy on the terminal screen.
VERIFY	Tell COPYFILE to verify the copy operation by reading what was just written to disk and comparing the contents of the Outputfile with the Inputfile.

Figure 4.15. The COPYFILE command options

USING THE COPYFILE COMMAND AND ITS OPTIONS

Let's try a few COPYFILE examples. To copy a data file (such as a word processing data file) onto a floppy disk, insert a floppy disk in the drive and type:

```
>ATTACH G FLOPPY1<RETURN>
```

```
>COPYFILE FIRST.WPF S = G<RETURN>
```

If the file was successfully copied, the following message should appear on the screen:

```
FIRST.WPF S copied to FIRST.WPF G
```

In the above example, we began by making sure that the floppy disk was attached. In this case drive G was attached as FLOPPY1. We then typed the specific FILENAME and FILETYPE of the Inputfile and the drive where the file was located and pressed the <RETURN> key.

THE EQUAL SIGN WILDCARD

Notice in the preceding COPYFILE example that we used two equal signs. The first sign tells the computer to use FIRST as the FILENAME for the Outputfile. The second sign tells the computer to use WPF as the FILETYPE for the Outputfile.

We then specified the drive where the new file is to be located, in this case drive G. Equal signs can be used as wildcard substitutes for the FILENAME and FILETYPE of the Outputfile.

If you are through copying files to drive G and you have no further need to use the drive at this time, the drive can be detached by typing:

>ATTACH G<RETURN>

Had we wished to make a second or backup copy of FIRST.WPF and put it on the S drive, we could have typed:

>COPYFILE FIRST.WPF S FIRST.BACKUP S<RETURN>

If the copy was successful, you would see the following on your screen:

FIRST.WPF S copied to FIRST.BACKUP S

In this case our Inputfile was FIRST.WPF. Since we wanted a different name for the file (because we wanted a backup copy) we typed a specific FILENAME and FILETYPE for our Outputfile, in this case FIRST.BACKUP.

THEOS always tries to prevent operator error. Had we typed:

COPYFILE FIRST.WPF S = S <RETURN>

the following error message would have appeared on the screen:

FIRST.WPF S not copied to FIRST.WPF S because file exists

The COPYFILE command will not “write over” a file that already exists unless you use the (REPLACE option. Using this option would, of course, replace any existing file of the same name with the new file.

Note: In all of the preceding examples, the only options in effect are the ones listed in the command option table as defaults.

One option of considerable importance that should be used whenever possible is the VERIFY option. Using this option will ensure that an accurate copy of the desired file was made.

Practice using the VERIFY option by typing:

>ATTACH G FLOPPY1 <RETURN>

>COPYFILE FIRST.WPF S = G (VERIFY <RETURN>

First the floppy disk G is attached. We then want to copy FIRST.WPF located on drive S to a file of the same name on drive G.

Because the VERIFY option was specified, the computer will automatically verify that an accurate copy of the file was made. If you are finished with the G drive, type:

>ATTACH G <RETURN>

USING * AS WILDCARDS

The * is most often used when copying a group of files. For example if you wished to copy all files with a FILETYPE of WPF, you could do so by typing the following command:

>COPYFILE *.WPF S = G (VERIFY <RETURN>

In this case, the command line says copy all files (*) with a FILETYPE of WPF to drive G. As each file matching this description is found, you will be asked if it is ok to copy that particular file. If you answer Y, the file will be copied. If you answer N, that file will not be copied and the system will proceed to copy the next file matching that description until all files matching the description have been copied and all "OK to copy" questions have been answered Y or N.

Copyfile Command/Options	End Result
COPYFILE MYFIRST.TIME S NEWFILE.TIME S (NOQ	A file called MYFIRST.TIME located on drive S is copied to a different file called NEWFILE.-TIME S. Because the NOQ option was used, the operator was not asked if it is ok to copy the file. The contents of the two files are identical; they just have different file descriptions.
COPYFILE SECOND.TRY S = G (NEWD	A file called SECOND.TRY located on drive S is copied to a file of the same name located on drive G. The NEWD option gives the file on drive G a date in its directory that is the current date.

Figure 4.16. Other examples of using the COPYFILE command and options

SUMMARY

The COPYFILE command is used for making copies of files in their entirety. It has many useful default options which ensure that files will be copied accurately and minimize possible operator errors. It is not possible to use COPYFILE to copy files that are larger than the available disk space on the Outputfile drive.

Session 8: The CREATE Command

This command provides you with the ability to create new data files to be used by application software. THEOS utilizes four different types of files for storing data; indexed, keyed, direct, and sequential. The application software determines the type of file required. Before any application software can be used, the data files must be created either manually, by you typing in the commands to create them, or automatically via a program which is part of the application software. In this session we will illustrate how to create each of the file types manually using the CREATE command.

In THEOS all data files, with the exception of sequential files, must be created before data can be entered using application software. Sequential files are generated automatically by application software or THEOS programs. As an example the files created by most word processing programs or text editors are sequential. When these files are created, THEOS allocates disk space for them automatically.

When new files are originated by the CREATE command, the disk space the files is to occupy must be contiguous. There can be no other files interspaced. In a system with multiple disk drives, a single file may not start on one disk and end on another disk. If there is not enough contiguous space available to hold the file being created, the error message "Disk full" is displayed and file creation is aborted.

A privilege level of 1 is required to use the CREATE command. The format of the CREATE command is:

CREATE FILENAME.FILETYPE DRIVNAME (options)

FILENAME is the name of the file to be created. FILETYPE is the type of the file to be created. DRIVNAME is the name of the drive where the file is to be located.

In order to create a file, all of the above parameters must be specified. Specific names must be assigned to the files as well as the drive. Wildcards are not allowed.

Option	Function
DIRECT	The file created is to be a direct file.
FILESIZE nn	The number nn indicates the number of records the file is to contain. The maximum number of records permitted in a THEOS file is 65535.
INDEXED	The file created is to be an indexed file.
KEYED	The file created is to be a keyed file.
KEYLEN nn	The number nn indicates the length of the index key. KEYLEN may be specified only for indexed or keyed files. The maximum keylength allowed is 128.
RECLLEN nn	The number nn indicates the physical length of each record in the file. The maximum record length allowed is 512.

Figure 4.17. The CREATE command options

Depending upon the size of the file that is to be created, the CREATE command may appear slow to execute. The larger the file created, the longer the execution time.

USING THE CREATE COMMAND AND ITS OPTIONS

As you will see below, the type of file to be created is generally specified first, followed by the other options of how many records the file is to store (filesize), keylength, and record length.

Let's try a few CREATE examples. Type:

```
>CREATE IC0DM0.IM S (DIRECT FILESIZE 15 RECLLEN 350 <RETURN>
```

This example creates a direct file called IC0DM0.IM which is located on drive S. It can hold 15 records (FILESIZE 15) and has a maximum record length of 350.

Now type:

```
>CREATE IC0DA0.IM S (INDEXED FILESIZE 21 RECLLEN 3 KEYLEN 12<RETURN>
```

This example creates an indexed file on drive S which can hold 21 records with a record length of 3 and a maximum keylength of 12. Try typing:

```
>CREATE APOIM0.GD S (KEYED RECL 350 KEYL 15 FILE 300 <RETURN>
```

This example creates a keyed file on drive S with a maximum record length of 350, a keylength of 15 and space for 300 records.

If each file was successfully created, you will be returned to CSI after each CREATE command has been executed.

The amount of disk space occupied by the created files is determined by the options FILESIZE, RECLEN, and KEYLEN. If you would like to approximate the amount of disk space required for given files before the CREATE command is executed, you can do so using the following formulae:

If the file is an indexed file, the amount of disk space allocated will be:

$2 * \text{FILESIZE} / 1024$, rounded up +

$(2 + \text{RECLEN} + \text{KEYLEN}) * \text{FILESIZE} / 1024$, rounded up.

If the file is a keyed file, the amount of disk space allocated will be:

$(\text{RECLEN} + \text{KEYLEN}) * \text{FILESIZE} / 1024$, rounded up

If the file is a direct file, the amount of disk space allocated will be:

$(\text{RECLEN} * \text{FILESIZE}) / 1024$, rounded up.

Using our formula for keyed files and example 3 above, we can determine the amount of disk space required to accommodate the file:

$$(350 + 15) * 300 / 1024 = 102 \text{ disk blocks}$$

One disk block occupies 1 Kilobyte of disk space. So our file will require 102K of contiguous disk space in order to be created.

Now that you know how to perform these calculations, use the other formulae and examples 1 and 2 to determine the amount of disk space each file will require.

Now check your answers. Example 1 requires 6 disk blocks. Example 2 requires 1 disk block.

SUMMARY

The CREATE command is the only means supplied by THEOS for creating application software data files. These files can be created by following the steps outlined above in conjunction with a particular application software installation guide. We have outlined all of the steps required to create files manually without the aid of programs that automate the process. The automatic file creation procedure will be examined in a later chapter.

Session 9: The FILELIST Command

The purpose of the FILELIST command is to list the disk directory entries in your account. It provides a means of displaying the disk drive's *table of contents*. The THEOS disk directory is quite comprehensive and is capable of displaying the following headings:

Fname Ftype Drv Date Time Recs Blks Format Label

Fname is the filename. Ftype is the filetype. Drv is the drive where the files are located. Date is the current system date assigned to the file whenever the file was changed, created, or modified by a THEOS command or your software. Time is the current system time assigned to the file in conjunction with the date.

Note: Some THEOS systems do not support the time function.

Recs indicates the current number of records in the file. Blks indicates the number of 1K disk blocks the file occupies. Format consists of two parts. Part one shows how the file is stored as well as its protection status:

- I for indexed file
- D for direct file
- S for sequential file
- K for keyed file
- A for absolute command or THEOS command file

Protection status is indicated by:

- D for delete protected
- W for write protected
- R for read protected

In part two the record length of the file is displayed. In the case of sequential files, the length will be the length of the longest record stored in the file (sequential files have variable record length). For direct and indexed files the record length is that specified at the time the file was created.

Label is the disk label assigned to the disk during the formatting process. This is not the same as the LABEL option described under default options.

The FILELIST command has no privilege level associated with it. The format of the FILELIST command is:

```
FILELIST FILENAME.FILETYPE DRIVNAME (options
```

FILENAME is the name of the file to be listed. FILETYPE is the type of the file to be listed. DRIVNAME is the name of the drive where the file is located.

When using this command, wildcards are allowed for FILENAME, FILETYPE, and DRIVNAME. The use of wildcards is provided as a convenience and slightly slows the time required to execute the command.

DEFAULT OPTIONS

The four default options in effect during the FILELIST command are: TYPE, HEADER, SORT, and LABEL. The TYPE option will automatically display the list of files to the screen. Unless the NOHEADER option is specified, all column headings will be displayed. Reference the list of FILELIST options in figure 4.18 that will allow you to selectively display column headings.

Option	Function
FNAME	The disk directory will only include the filename column.
FTYPE	The disk directory will include both the filename and file type column.
FDISK	The disk directory will include the first three columns: filename, file type, and disk drive.
DATE	All columns will be included up to and including the time column.
ALLOC	All columns will be shown up to and including the Blks column.
FORMAT	All columns will be shown up to and including the format.
LABEL	All columns will be included up to and including the disk label. This option is the same as executing the FILELIST command with no options. All headings will be displayed provided the line length of the screen is at least 79 columns.
TYPE	The file listing will be displayed on the screen. This is a default option.
PRINTER n	The file listing will be displayed on the printer number designated by the letter n. If no printer number is specified, the listing will print to the printer attached as PRINTER1.
EXEC	The list of files will automatically be written in a sequential file on the disk and automatically assigned the name of SELECTED.EXEC. The use of this option will be described briefly in an example below and discussed in detail in the chapter on EXECs.
NOSORT	The list of disk files will appear on the screen in the order in which they appear on the disk; they will not be sorted alphabetically. If this option is not specified, the files are sorted in alphabetical order.
mm/dd/yy	When this option is used, only those files whose date is later than the date specified will be included in the list. The date must be specified in the format shown.
OWNER	This option provides the only means of listing all disk files regardless of the account you are currently logged onto. In order to use this option, you must have a privilege level of 5 or be logged onto the system account.

HEADER	The listing will include a heading on each page of data displayed.
NOHEADER	The listing will not include a heading on each page. This is a default option when the EXEC option is specified or when anything less than the date column is listed.

Figure 4.18. The FILELIST command options

USING THE FILELIST COMMAND AND ITS OPTIONS

Although the options FNAME, FTYPE, and FDISK are available, they are not used in most instances because most THEOS users want as much information displayed about the files as possible. Our first example depicts the most general use of the FILELIST command which is its execution with no options in effect other than the defaults: LABEL, TYPE, HEADING and SORT.

Note: The following examples are for illustration only. You may type the commands as indicated, but the file lists that you obtain will display different files, since these files are probably not included in your system.

Assuming you were logged onto an account called USER3 and typed:

```
>FILELIST S<RETURN>
```

```

FILELIST *.*:HARDISK1                               12/07/84 09:28 Page 1
Fname--- Ftype-- Dr --Date-- Time- --Recs Blks Format- Label---
BARB      BACKUP  S 12/03/84 17:30    50    1 S    63 HARDISK1
BARB      MWRITE   S 12/06/84 20:25    50    1 S    63 HARDISK1
BILL      BACKUP  S 12/05/84 19:59    25    1 S   126 HARDISK1
BILL      MWRITE   S 12/05/84 20:44    61    2 S   126 HARDISK1
COMMENTS BACKUP  S 11/26/84 18:48    71    3 S   126 HARDISK1
COMMENTS MWRITE   S 11/26/84 18:54    72    3 S   126 HARDISK1
CREATE    BACKUP  S 12/04/84 20:30   165    7 S   126 HARDISK1
CREATE    MWRITE   S 12/04/84 20:31   165    7 S   126 HARDISK1
CREATE    WPF      S 11/05/84 19:25    54    7 S   128 HARDISK1
EXER1     BACKUP  S 11/27/84 22:25    24    1 S    60 HARDISK1
EXER1     MWRITE   S 11/27/84 22:34    38    1 S   126 HARDISK1
FILELIST  BACKUP  S 12/06/84 21:42   144    6 S   126 HARDISK1
FILELIST  MWRITE   S 12/06/84 22:45   147    6 S   126 HARDISK1
OAK       BACKUP  S 11/29/84 22:33    47    2 S    64 HARDISK1
OAK       MWRITE   S 12/03/84 09:31    47    2 S    64 HARDISK1
XMASLET   BACKUP  S 11/27/84 16:56    63    2 S    76 HARDISK1
XMASLET   MWRITE   S 11/27/84 17:06    64    2 S    76 HARDISK1
17 files, 54 blocks.

```

Figure 4.19. List of all files on drive S belonging to USER3 account

a display similar to that depicted in figure 4.19 would appear. This display indicates all of the directory fields up to and including the disk label, and a listing in alphabetical order of all files files belonging to the account called USER3. Although in this case the file listing was sent to the screen, had the PRINTER option been used the same display would have been sent to the printer. When a printer is used the title of the listing requested is displayed in the upper left hand corner. For convenience, the date, time and page number of the listing is placed in the upper right hand corner of the page. The total number of files listed and the total number of disk blocks the files occupy is placed at the end of the file listing.

For our next example, suppose you had typed:

>FILELIST S (DATE<RETURN>

```
FILELIST *.*:HARDISK1

Fname--- Ftype-- Dr --Date-- Time-
BARB      BACKUP   S 12/03/84 17:30
BARB      MWRITE    S 12/06/84 20:25
BILL      BACKUP   S 12/05/84 19:59
BILL      MWRITE    S 12/05/84 20:44
COMMENTS  BACKUP   S 11/26/84 18:48
COMMENTS  MWRITE    S 11/26/84 18:54
CREATE    BACKUP   S 12/04/84 20:30
CREATE    MWRITE    S 12/04/84 20:31
CREATE    WPF      S 11/05/84 19:25
EXER1     BACKUP   S 11/27/84 22:25
EXER1     MWRITE    S 11/27/84 22:34
FILELIST  BACKUP   S 12/06/84 21:42
FILELIST  MWRITE    S 12/06/84 22:45
OAK       BACKUP   S 11/29/84 22:33
OAK       MWRITE    S 12/03/84 09:31
XMASLET   BACKUP   S 11/27/84 16:56
XMASLET   MWRITE    S 11/27/84 17:06
17 files, 54 blocks.
```

Figure 4.20. List of all files on drive S belonging to USER3

A display similar to that shown in figure 4.20 would appear. Here, the display is the same as shown in figure 4.19 except that the only headings printed are those up to and including the current time.

If you type:

```
>FILELIST S (OWN<RETURN>
```

A listing may appear on the screen similar to that shown in figure 4.21.

```

FILELIST *.*:TEST

Files owned by account: USER10 (10)

Fname--- Ftype-- Dr --Date-- Time-- --Recs Blks Format- Label--
OM0507   BASICCOM G 08/15/83 18:46    7    2 SR   254 TEST
WALT     TEXT      G 03/08/84 10:40    5    1 S    4 TEST
    
```

```

FILELIST *.*:TEST

Files owned by account: USER2 (2)

Fname--- Ftype-- Dr --Date-- Time-- --Recs Blks Format- Label--
ADDIT    DEM      G 10/13/83 12:00    4    1 S   128 TEST
ADDIT    WPM      G 10/13/83 11:59   19    3 S   128 TEST
ADVERT   B        G 10/13/83 12:01    7    1 S   128 TEST
ALPHA    WPM      G 10/13/83 11:58   30    4 S   128 TEST
BATCH    WPM      G 10/13/83 11:58    9    2 S   128 TEST
BOILER   MAS      G 10/13/83 12:00    5    1 S   128 TEST
BOILER   TEM      G 10/13/83 12:00    1    1 S   128 TEST
BOILER   WPM      G 10/18/83 04:06   18    3 S   128 TEST
CALC     WPM      G 10/13/83 11:57   50    7 S   128 TEST
CLETTER  A        G 10/13/83 12:00    5    1 S   128 TEST
CUESORT  WPM      G 10/13/83 11:57   20    3 S   128 TEST
CUSLIST  CMA      G 10/13/83 12:00    6    1 S   128 TEST
CUSLIST  DBF      G 10/13/83 11:58    9    2 S   128 TEST
CUSLIST  DEM      G 10/13/83 12:00    6    1 S   128 TEST
DRIVER   A        G 10/13/83 12:00    6    1 S   128 TEST
FORMS    WPM      G 10/13/83 11:59   47    6 S   128 TEST
INSTALL  WPM      G 10/13/83 11:59   11    2 S   128 TEST
INVOICE  TEM      G 10/13/83 12:00    9    2 S   128 TEST
KEYS     A        G 10/13/83 12:00    2    1 S   128 TEST
KPHRASE  WPM      G 10/13/83 12:01   27    4 S   128 TEST
LABEL    TEM      G 10/13/83 12:00    1    1 S   128 TEST
LETTER   DEM      G 10/13/83 12:00    5    1 S   128 TEST
LETTER   TEM      G 10/13/83 12:00    4    1 S   128 TEST
LINENB   WPM      G 10/13/83 11:57   24    4 S   128 TEST
MLABEL   WPM      G 10/13/83 12:00   37    5 S   128 TEST
MMERGE   WPM      G 10/18/83 04:04   51    7 S   128 TEST
MOVEIT   WPM      G 10/13/83 11:58   48    7 S   128 TEST
ORDER    TEM      G 10/13/83 12:00    6    1 S   128 TEST
PAGENUM  WPM      G 10/13/83 11:59   36    5 S   128 TEST
TCPRNT   WPM      G 10/13/83 11:58   32    5 S   128 TEST
32 files, 87 blocks.
    
```

Figure 4.21. List of files on drive S sorted by owner

In order to use this option, remember that your privilege level must be 5 or you must be logged onto the SYSTEM account. In this case you see that there is a complete list of files for each account. THEOS indicates which account the files belong to in addition to providing the information we have discussed above. The appearance of each page in this listing is similar to the previous listings except that the total number of files and the total number of disk blocks the files occupy does not appear until all files have been listed.

Note: Depending upon the type of hardware, its configuration and the number of files stored on disk, an "Insufficient memory" message may appear when executing FILELIST with the OWNER option. This error message isn't serious and means that your particular system does not have sufficient memory to execute this THEOS command option.

If you type:

```
> FILELIST S (NOSORT<RETURN>
```

A listing similar to figure 4.22 may appear on the screen:

```

FILELIST *.*:HARDISK1                12/07/84 09:33 Page 1
Fname--- Ftype-- Dr --Date-- Time-- -Recs Blks Format- Label---
OAK      BACKUP  S 11/29/84 22:33   47   2 S    64 HARDISK1
COMMENTS BACKUP  S 11/26/84 18:48   71   3 S   126 HARDISK1
XMASLET  MWRITE  S 11/27/84 17:06   64   2 S    76 HARDISK1
CREATE   BACKUP  S 12/04/84 20:30  165   7 S   126 HARDISK1
BARB     MWRITE  S 12/06/84 20:25   50   1 S    63 HARDISK1
CREATE   WPF     S 11/05/84 19:25   54   7 S   128 HARDISK1
EXER1    BACKUP  S 11/27/84 22:25   24   1 S    60 HARDISK1
OAK      MWRITE  S 12/03/84 09:31   47   2 S    64 HARDISK1
FILELIST BACKUP  S 12/06/84 21:42  144   6 S   126 HARDISK1
COMMENTS MWRITE  S 11/26/84 18:54   72   3 S   126 HARDISK1
BILL     BACKUP  S 12/05/84 19:59   25   1 S   126 HARDISK1
EXER1    MWRITE  S 11/27/84 22:34   38   1 S   126 HARDISK1
XMASLET  BACKUP  S 11/27/84 16:56   63   2 S    76 HARDISK1
BARB     BACKUP  S 12/03/84 17:30   50   1 S    63 HARDISK1
CREATE   MWRITE  S 12/04/84 20:31  165   7 S   126 HARDISK1
FILELIST MWRITE  S 12/06/84 22:45  147   6 S   126 HARDISK1
BILL     MWRITE  S 12/05/84 20:44   61   2 S   126 HARDISK1
17 files, 54 blocks.

```

Figure 4.22. List of files on drive S unsorted

Here the file list is identical to that in figure 4.19 except that the list is not sorted alphabetically. This list shows the files in the order in which they are actually stored on the disk.

Note: It is possible to use a number of options in combination with each other. For example if you were to type,

```
>FILELIST S (NOSORT DATE<RETURN>
```

the listing would be identical to figure 4.22 but will only display the headings through the current system time.

THE FILELIST COMMAND (EXEC OPTION)

The EXEC option when used with the FILELIST command provides a very powerful tool for the THEOS user who would like to automate execution of particular THEOS commands. Suppose you were to type:

```
>FILELIST *.EXEC S (EXEC<RETURN>
```

This command line is interpreted as “provide a list of all files with FILETYPE of EXEC located on drive S and store them in a special file called SELECTED.EXEC located on drive S.” Once this file has been created, it can be used to automate certain THEOS commands such as listing the contents of the EXEC files. You could list each EXEC file manually by typing,

```
LIST ONE.EXEC S<RETURN>  
LIST TWO.EXEC S<RETURN>  
LIST THREE.EXEC S<RETURN>
```

and so on. You can eliminate this extra work by executing the FILELIST command with the EXEC option as follows:

```
>EXEC SELECTED LIST (PRINT<RETURN>
```

After you press RETURN, THEOS takes over. The contents of the files with a FILETYPE of EXEC are listed to the printer. We’ve not yet discussed the LIST command which allows the listing of the contents of files, not just the filenames. The LIST command will be covered in session 11. More examples illustrating the use of the (EXEC option will be provided in the chapter on EXECS.

SUMMARY

The FILELIST command affords the THEOS user the ability to view the table of contents of all disks attached to the computer. It has many flexible options which permit the selective display of essential information about the files listed such as the time and date they were last updated, how much space they occupy, and the protection status of

each file. In addition, a unique option called EXEC allows lists of files to be stored in a special file called SELECTED EXEC for automating THEOS commands.

Session 10: The INITDISK Command

The purpose of this command is to initialize or format disks attached to a THEOS system. 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.

Unlike other operating systems, the THEOS INITDISK command will initialize disks which can be used with any THEOS software program. Special formats do not have to be provided with individual application programs.

A privilege level of 3 is required to use the INITDISK command. The format of the INITDISK command is:

INITDISK DRIVNAME (options)

DRIVNAME is the disk to be initialized. If formatting is to occur, at least one option is required as described below.

Option	Function
FORMAT	This option is used when a new disk is to be formatted for use by THEOS or when a previously used disk is to be reformatted. This option cannot be used to initialize the system disk.
HEAD	This option specifies the number of disk surfaces for formatting. In the case of a double-sided floppy disk the HEAD option would be specified with the number 2 after it.
SIZE	This option allows you to specify the number of entries allowed in the disk directory. If this option is not specified, THEOS will automatically assign a directory size in accordance with the capacity of the disk. The number of entries is indicated by the number following SIZE.

Figure 4.23. INITDISK command options

DEFAULT OPTIONS

When a disk is formatted, it is not mandatory to specify the HEAD and SIZE options. Before formatting begins, a series of default values appear on the screen, and THEOS asks if these values are acceptable. If they are not, a new INITDISK command line must be entered and new options specified. The default values displayed on the screen when no options are determined by the manufacturer of the computer.

USING THE INITDISK COMMAND AND ITS OPTIONS

Before any disk can be initialized it must be attached to the system. As mentioned above, it is not possible to use the FORMAT option on the system disk. Since all THEOS programs are stored on the system disk, an attempt to format it would result in chaos since all programs including the INITDISK program itself would be erased! We will illustrate the correct procedure for formatting a system disk below. Since the type of disk to be formatted most often is a floppy disk, all examples, with the exception of initialization of a system disk, will utilize a floppy disk attached as drive G.

Example 1 type:

```

ATTACH G FLOPPY1 <RETURN>
INITDISK G (FORMAT <RETURN>

THEOS will respond:
INITDISK will erase all files on drive G(1)
Do you wish to continue (Y/N)?Y <RETURN>

Enter disk Label: TEST<RETURN>
Number of directory entries = 80
Increment between adjacent sectors = 2
Offset between adjacent tracks = 0
Number of surfaces = 1
Number of tracks per surface = 77
Number of sectors per track = 26

Double Density

Is above information correct (Y/N)? N <RETURN>
>

```

Although a great deal of information is provided here, the elements of primary concern to us are:

- Assigned disk label
- Number of directory entries
- Number of surfaces

For our purposes, we can ignore the other entries.

In this example we responded with an 'N' to the final prompt because we want to store data on both disk surfaces so as to have a capacity of more than 80 directory entries. Since we responded to the final prompt with an 'N', we are returned to the THEOS prompt where we can proceed with our next example.

Example 2 type:

```
>INITDISK G (FORMAT H 2 SIZE 200 <RETURN>
INITDISK will erase all files on drive G(1)
Do you wish to continue (Y/N)? Y:<RETURN>

Enter disk Label: TEST <RETURN>

Number of directory entries = 208
Increment between adjacent sectors = 2
Offset between adjacent tracks = 0
Number of surfaces = 2
Number of tracks per surface = 77
Number of sectors per track = 26

Double Density, Double Side

Is above information correct (Y/N)? Y <RETURN>
```

At this point THEOS will begin the formatting process. You will see information on the screen regarding which track is being formatted. This information will continue to be displayed until all tracks, in this case 77, have been formatted. When formatting is complete, you will be returned to the THEOS prompt.

HOW TO FORMAT A SYSTEM DISK

The process of formatting a system disk differs from our preceding example because the disk to be formatted may not be currently attached as a system disk. Let's look at an example.

Assume we have a new THEOS system which has one floppy disk drive and one hard disk platter. Under normal conditions we would boot the system from the hard disk which would automatically be designated as the system disk. However, the hard disk has been formatted by the manufacturer for only 1400 directory entries, and we want to increase the directory size to 2000 entries. Unfortunately, the only way to increase the directory size is to reformat the entire disk. Remember, the hard disk (since it is the system disk) currently contains all of the THEOS programs, no application programs or data files yet. We cannot reformat the hard disk as long as it is currently attached as the system disk, so we must execute the following steps to initialize the hard disk:

- Step 1: Reboot the system with a THEOS floppy disk (now designated as the system disk).
- Step 2: Type ATTACH <RETURN> to determine what the current attachments look like.
- Step 3: Assuming that no disk drive is attached other than S (the system disk) type: ATTACH A DISK1 <RETURN>
This attaches the hard disk.
- Step 4: Now you are ready to begin the initializing process.
Type:

INITDISK A (FORMAT SIZE 2000 CMI <RETURN>

A screen display will appear similar to the preceding example except the number of directory entries will be 1944 and the other entries will correspond to the default values preprogrammed by your hardware manufacturer.

In addition, you may also be asked to input what is called a flaw map. A flaw map is a special code which tells THEOS which areas of the hard disk may be defective. Most 20 MB or smaller disk drives have no flaw map and as such you may just press RETURN in response to this question. If your disk drives have flaw maps, they are located

inside the computer on top of the drives or written in the hard disk section of your owner's manual. Consult your owner's manual for further information on this topic.

Note: Although you are able to specify the number of directory entries desired, THEOS will readjust the number of entries to optimize disk space.

Depending upon the number you type, you may end up with a few more or a few less entries. Also, when formatting a hard disk, some hardware manufacturers may have special options to optimize the formatting process. In the example above, the special option was CMI. This option was preprogrammed into THEOS by a particular hardware manufacturer. Consult your hardware manual for any special options your hard disk may require.

When the formatting process is complete, you will be returned to the THEOS prompt. You now have a hard disk that is ready to store information, but it is not yet a system disk. In order to complete the creation of a system disk, you must transfer the THEOS operating system files from the current system disk to the new disk.

In order to maximize the speed of a THEOS system, the operating system files should be placed on the system disk in a certain order. To transfer the operating system files from the current system disk (the floppy) to the new disk (the hard disk) type:

```
>COPYFILE SYSTEM.NUCLEUS S == A (NOQ<RETURN>
```

When you are returned to the THEOS prompt, copy the remaining system files by typing:

```
>COPYFILE SYSTEM.* S == A (NOQ<RETURN>
```

All files THEOS requires to operate have the filename of SYSTEM. One important file which cannot be copied by this method is SYSTEM.ACCOUNT. If you wish to have the same accounting structure as on the old system disk, this accounting structure can be transferred to the old system disk by typing:

>ACCOUNT (COPY=A:<RETURN>

If there is no accounting structure on the floppy, you will have to use the ACCOUNT command to set one up for the new system disk.

Your new system disk is ready for use by THEOS. If you wish to use the hard disk as the new system disk at this point, you must reboot the system from the hard disk.

SUMMARY

Before information of any sort can be written on a new disk, it must first be initialized. The INITDISK command accomplishes this task. Disks can be easily reformatted at any time for reuse, with the exception of the system disk. The special procedures required to format a system disk were described in this session.

Session 11: The INITTAPE Command

The INITTAPE command formats tapes for use by THEOS. In a THEOS system, tapes are generally used as backup media or for loading new programs purchased from an applications software vendor. Therefore, tapes are formatted only for the purpose of making backup copies of the data stored in the computer's disk drives.

The format of the INITTAPE command is:

INITTAPE TAPEn (option

TAPEn indicates the number of the tape drive where the tape to be initialized is located. In THEOS the number *n* is the number of the tape drive as specified by the ATTACH command. The number *n* must be a number ranging from one through four. If you have only one tape drive attached, you may eliminate TAPEn. THEOS will assume a default value of TAPE1.

This command only has one option associated with it. That option is the TENSION option. The purpose of this option is to *fast forward* the tape to the end-of-tape marker and then *rewind* the tape to its beginning. In order to maximize tape performance, it is advisable to execute the command with this option.

USING THE INITTAPE COMMAND

Before using this command, you must make certain that your system has a tape drive that was attached by the ATTACH command. Type:

>ATTACH<RETURN>

Note what is currently attached to your system. If you have a tape drive and it is not attached, type:

>ATTACH TAPE 1 CART<RETURN>

This command line attaches your tape as TAPE1 CART, since most tape drives for microcomputers are of the CARTridge variety. Insert a tape in the drive and type:

>INITTAPE (TENSION<RETURN>

At this time, you will hear the tape being *tensioned* but no prompts will appear on the screen. When the formatting process is ready to begin, the following prompts will appear on your screen:

Enter volume id (8 chars) ?**BACKUP<RETURN>**
Enter owner id (14 chars) ?**WALTSDATA<RETURN>**

The volume id (in this case you typed BACKUP) is a required entry and is used by THEOS to later provide on screen information regarding any restore, archive, or backup procedures. The volume id (in this case you typed WALTSDATA) is an optional entry and may be left blank if you desire. The volume id may be a maximum of eight characters long. The owner id may be a maximum of 14 characters long.

Note: If you try to execute this command without the tape drive physically connected to your computer, the following error message "TAPEn not ready" will appear. You can press ESC Q to return to the THEOS prompt.

Since you have no further need for the tape drive at this time, you may detach it by typing:

>ATTACH TAPE 1 <RETURN>

SUMMARY

The INITTAPE command is designed to format blank tapes to be used by THEOS. In order to use this command, you must have a tape drive physically attached to your system and attached via the ATTACH command.

Session 12: The LIST Command

The LIST command allows the listing of the contents of data files either on the terminal screen or printer. This command should be used only with sequential, indexed, keyed, or direct files. Trying to list other types of files will cause unpredictable results or, in the case of files that are read protected (see the RENAME tutorial), listing the file may not be possible. The type of file most often listed is the sequential file. As we will see in the tutorial session on the EDIT command, the LIST command is most useful when editing a sequential file.

There is no privilege level associated with this command, but files can only be listed that belong to the currently logged account or the system account. For example, if you are logged onto an account called USER1, the contents of files cannot be listed that belong to an account called USER2 unless USER1 and USER2 are synonymous.

The format of the LIST command is:

LIST FILENAME FILETYPE DRIVNAME (options)

FILENAME is the name of the file to be listed. FILETYPE is the type of the file to be listed. DRIVNAME is the name of the drive where the file is located. The DRIVNAME does not have to be specified. However, when the name of drive where the file is located is known, specifying the drive will reduce, by a few seconds, the time required to locate the file.

The use of wildcards is not permitted with this command. If an error message such as "Syntax error" appears on the screen after a command line has been entered, it is usually a result of leaving out either the FILENAME or FILETYPE.

Option	Function
HEADING	Provides your file list with a heading. If this option is not specified, THEOS provides it as the default. The heading includes the file name, file type, disk drive, disk label, current system time, date, and page number. When the list is sent to the screen, the length of the screen page is the attached length of the screen designated by ATTACH
NOHEAD	Tells THEOS not to provide a page heading.
PAGE nn	The listing of the file will not begin until the page number specified by nn is found.
PRINTERn	The list is sent to the printer. n is the number of the printer to which you wish the listing sent.

Figure 4.24. The LIST command options

Default Options

The default options for this command are the **HEADING** option and the listing of the file automatically output to the screen.

USING THE LIST COMMAND

This command is most helpful for listing sequential files. For listing short files (files no longer than two screen pages) outputting the list to the screen may be adequate. When listing longer files, printing the list is more convenient because you may want to refer to it regularly.

The examples below assume that the files are owned by your account. Since these files are not now part of your system, do not attempt to actually type these examples. They are for illustrative purposes only.

Example 1:

```
>LIST ASSIST EXEC S.<RETURN>
```

```
ASSIST.EXEC:HARDISK1                                03/07/86 17:41 Page 1
&CONTROL OFF
-START
&CRT CLR
&TYPE PLEASE TYPE THE OASIS COMMAND YOU WISH HELP ON OR [RET] TO LIST COMMANDS \
&READ &SEL
&CRT CLR
HELP &SEL
&WAIT
&CRT CLR
&CRT 30,14
&TYPE MORE HELP? Y/N \
&READ &SEL
&IF &SEL = Y &GOTO -START
&IF &SEL = N &CRT CLR
&IF &SEL <> Y &IF &SEL <> N &GOTO -START
&QUIT
&WAIT
```

Figure 4.25. Listing of ASSIST.EXEC output to screen

This example displays a listing of the sequential file called ASSIST.EXEC located on drive S on the terminal. All of the heading information is displayed on the screen.

Example 2:

LIST ASSIST EXEC S (PRINTER1 <RETURN>)

```
ASSIST.EXEC:HARDISK1                                     12/17/84 09:37 Page 1
&CONTROL OFF
-START
&CRT CLR
&TYPE PLEASE TYPE THE OASIS COMMAND YOU WISH HELP ON OR [RET] TO LIST COMMANDS \
&READ &SEL
&CRT CLR
HELP &SEL
&WAIT
&CRT CLR
&CRT 30,14
&TYPE MORE HELP? Y/N \
&READ &SEL
&IF &SEL = Y &GOTO -START
&IF &SEL = N &CRT CLR
&QUIT
&WAIT
```

Figure 4.26. Listing of ASSIST.EXEC output to printer

This example illustrates the use of the PRINTER option. The file listing shown on the screen in example one is printed on the printer.

Example 3:

LIST ASSIST EXEC S (NO HEAD<RETURN>

```

&CONTROL OFF
-START
&CRT CLR
&TYPE PLEASE TYPE THE OASIS COMMAND YOU WISH HELP ON OR [RET] TO LIST COMMANDS
&READ &SEL
&CRT CLR
HELP &SEL
&WAIT
&CRT CLR
&CRT 30,14
&TYPE MORE HELP? Y/N \
&READ &SEL
&IF &SEL = Y &GOTO -START
&IF &SEL = N &CRT CLR
&IF &SEL <> Y &IF &SEL <> N &GOTO -START
&QUIT
&WAIT

```

Figure 4.27. Listing of ASSIST.EXEC output to screen with no heading

The file listing is output to the screen, but this time no heading is displayed.

Example 4:

LIST AP0IM0.PDS (PRINTER<RETURN>)

```

APOIM0.PD:TEST                                03/07/86 18:10 Page 1

 2   22:      -478.24,-9.76,05/10/83,print,,0,0,0
 2   52:      -7300,0,07/07/83,test,,0,0,0
 2   56:      -2896,-34,07/31/83,,,0,0,0
 2   57:      0,0,08/25/83,test data,,0,0,0
 2   58:      0,0,09/06/83,test,,0,0,0
 2   61:      0,0,09/19/83,test,,0,0,0
 4   60:      -500,0,09/08/83,tools,,0,0,0
 4   100:     -1000,-10,05/23/83,Payment,,0,0,0
 5   50:      -6573.68,-60,06/29/83,Paymnt on acct,,,0,0,0
 5   51:      -100,0,06/29/83,,,200,1,0
 5   300:     0,0,08/16/83,travel,,0,0,0
100  59:      -5565,0,09/06/83,test check,,0,0,0
100 1234:     -500,0,09/22/83,Computer parts,,0,0,0

```

Figure 4.28. Listing of indexed file AR0IM0.CM output to printer

The preceding example illustrates the appearance of an indexed file listing when it is output to a printer. This file contains data that references paid invoices in an accounts payable program.

SUMMARY

The LIST command provides the only method of listing the contents of THEOS files. Although most THEOS files can be listed, the listing of sequential files is performed most frequently. More detailed examples illustrating the use of the LIST command will be provided in the TEXTEDIT tutorial session.

Session 13: The LOGOFF/LOGON Commands

Once a THEOS user accounting structure has been established, the only way that any files or programs can be accessed is by *logging on* to an account. If no accounting structure is present, these commands have no meaning because no accounts exist to LOGON to or LOGOFF from. LOGON/LOGOFF are complementary commands, and they will be discussed together. These commands have no privilege level associated with them.

The LOGOFF Command

When the LOGOFF command is executed, a series of events occur:

- If accounting history is being used, a history record is written to the ACCOUNT.HISTORY file.
- Any peripherals attached (via the ATTACH command) to the system by the currently logged account are detached.
- Control of the system is transferred to the LOGON command.

The format of the LOGOFF command is:

LOGOFF

The LOGON Command

When a user accounting structure exists in an THEOS system, you will be prompted to "Logon please:" after booting the system or after logging off of a particular account.

The format of the LOGON command is:

LOGON ACCOUNT NAME

In order to accomplish a successful logon, type the name of the specific account name you wish to logon to. When this command is executed, a series of events occurs:

- You are automatically logged off any account you are currently logged on to.
- THEOS searches the SYSTEM.ACCOUNT file for the account name specified. If the account name typed cannot be found, you are returned to the "Logon please:" message.
- When the account name is found in the SYSTEM.ACCOUNT file and a password has been specified, THEOS will respond: "Password?" When a password is entered, it is not displayed on the screen. A series of asterisks is used to prevent others from discovering the password.
- If the password is incorrect you are returned to the "Logon please:" message.
- If accounting history is being used, a record indicating the logon is written to the ACCOUNT.HISTORY file.
- The THEOS SYSTEM.ERRMSG file is searched for messages 99 through 102 which are displayed if found. These are usually messages entered by the hardware manufacturer which describe the type of hardware being used. These messages can be changed. See chapter 5 for instructions on how to enter these messages.
- The system disk is searched for a file called SYSTEM.-LOGON; its contents are displayed if found. These messages may include user instructions on what to do after successful logon. See chapter 5 for details how to enter these messages.
- The logon date and time are displayed on the screen.

At this point, successful logon has been accomplished. Although a number of steps are involved in the logon process, all steps are executed in less than two or three seconds.

After a successful logon has been accomplished, a search of all attached disks and their directories is made for an EXEC program with the file description of ACCOUNTNAME.EXEC, where the ACCOUNTNAME is the account onto which logon was requested. For example, if you requested to logon to USER1, THEOS would conduct a search for an EXEC called USER1. This special program called USER1 might take you to a menu (another EXEC) which will make your THEOS system menu driven. More examples are provided in the EXEC tutorial.

Note: If a successful logon has not been accomplished in three tries, the system automatically *hangs* and must be rebooted. This is a built in safety feature of THEOS designed to prevent users from gaining unauthorized access to the computer.

USING THE LOGON/LOGOFF COMMANDS

Using these commands is simple and straightforward. No options are associated with them.

Example 1:

```
LOGOFF<RETURN>
Logoff at 13:45:21, on 12/17/84
Logon please:
```

This example illustrates the logoff procedure from the THEOS prompt. As shown, the user is automatically returned to the "Logon please:" message.

Example 2:

```
Logon please:USER2<RETURN>
Password?*****<RETURN>
Logon at 13:46:00, on 12/17/84
>
```

This example shows how to logon to an account with a password. Note the series of asterisks THEOS displays in place of the actual password.

Example 3:

```
>L USER4<RETURN>
Logon at 14:00:00, on 12/17/84
>
```

In this case when a request is made to logon to a new account, USER4, THEOS performs an automatic logoff before logging on to

the new account. If an account called USER4 is not found, an “account not found” message is displayed on the screen, and the user is returned to the prompt logged onto the original account.

Example 4:

```
Logon please:USER7,***** <RETURN>
Logon at 15:45:32, on 12/17/84
>
```

This example illustrates logging on to a new account which has a password. Note that the Password? prompt is bypassed because the password was entered as part of the command line. This syntax only works if a logoff has been performed first and the “Logon please:” message is displayed on the screen.

Summary

The LOGON/LOGOFF commands are the only means of accessing THEOS programs that belong to user accounts. After an accounting structure has been established, THEOS will request the user to “Logon please:” after the system has been booted. If the user does not type in an account that exists after three tries, THEOS assumes that the user is unauthorized and *hangs* the system. When a user is finished with the system, it is recommended that the LOGOFF command be executed to ensure an orderly exit from the system.

Session 14: The MAILBOX/MSG Commands

These commands apply only to the multi user version of THEOS since messages must be sent to or received by more than one system user. Neither command has a privilege level associated with it.

THE MAILBOX COMMAND

This command allows the retrieval of messages sent to a user while he was logged off the system. The format of the MAILBOX command is:

MAILBOX

MAILBOX displays the message sent to the account name the user is currently logged onto. The message sent to this account is automatically removed from the system once this command has been executed.

HOW TO USE THE MAILBOX COMMAND

This command is most frequently executed at the start of a new business day. The system manager is probably in the office before everyone else and may have messages for all users. MAILBOX is executed at the time of first LOGON as follows:

```
>LOGON WALT <RETURN>
```

```
You have messages waiting. To retrieve, execute MAILBOX
```

```
>MAILBOX <RETURN>
```

```
From: SYSTEM, 08:30 12/18/84
```

```
This is a test of the THEOS message system....Bye.
```

THE MSG (MESSAGE) COMMAND

This command provides a means of sending a message to another user. The format of the MSG command is:

```
MSG ACCOUNTNAME TEXT
```

ACCOUNTNAME is the account name to which the message is to be sent. An account name of * (asterisk) tells THEOS that the message is to be sent to all users, including the sender. In order to send a message to all users, you must have a privilege level of 5.

TEXT is the actual message which is to be sent. If the text does not follow the ACCOUNTNAME, THEOS allows multiple lines of text to be entered. The end of the message is indicated by a blank line and pressing <RETURN>.

When the MSG command is executed, the message sent will immediately be displayed on the screens of currently logged on users.

The message does not interfere with any application software currently in use. If the message "User is not logged on." is displayed on the sender's screen, the MSG command will ask: "Do you wish to deposit into the mailbox?" You can choose Y or N and press <RETURN>.

HOW TO USE THE MSG COMMAND

The MSG command serves as a convenient method of communication in an office where the terminals are separated by large distances. It also serves as an instant prompt or reminder of functions to be performed by the system manager or user.

Example 1:

```
>MSG USER1 Staff meeting at 9:30...Send message when you  
arrive.<RETURN>  
User is not logged on  
Do you wish to deposit into the mailbox? Y<RETURN>  
>
```

The preceding example illustrates the entry of a short message text directly following the account name which is to receive the message. Since the user is not currently logged on, you are asked if the message is to be deposited in the mailbox. After the message is deposited, you are returned to the THEOS prompt.

Example 2:

```
>MSG * <RETURN>  
Enter message text, terminate input by empty line.  
?Single user mode in 10 minutes.<RETURN>  
?Please finish what you are doing and logoff<RETURN>  
?the system as soon as possible.<RETURN>  
?<RETURN>  
>
```

The preceding example illustrates the procedure for sending a message to all logged on users. Since this message is to be a multi-line message, no text is entered after the *, and THEOS prompts you to begin entering text. After the message is entered, the RETURN key is pressed on an empty line indicating the end of the message.

SUMMARY

The MAILBOX/MSG commands provide a convenient method of sending messages to THEOS users. MAILBOX is executed when a user wishes to retrieve messages deposited in the mailbox. MSG is executed when a user wishes to send messages to another user.

Session 15: The RENAME Command

On occasion, it may be necessary to change the filename or filetype of a file, or change its **protection status**. The RENAME command allows the THEOS user to change a file's description (filename and filetype) and/or protection status. This command only operates on files on the currently logged account.

A file's protection status is defined as follows:

- delete protection
- write protection
- read protection

Delete protection means that a file may not be either deleted (erased) from the disk or renamed without first changing its protection status.

Write protection means that a file may not be updated without first changing its protection status. Write protected files can be erased. A **read protected** file cannot be listed, edited, removed or manipulated in any manner. This type of protection status is generally used only by THEOS programmers to protect their programs.

THEOS files are generally assigned some type of protection, usually delete and write protection, so as to avoid accidental erasure of important files and to prevent changing (updating) of certain files by inexperienced users. Protection status is usually assigned by the system manager. A privilege level of 1 is required to use this command.

The format of the RENAME command is:

RENAME FILENAME.FILETYPE DRIVNAME NEWFILE.NEWTYPE DRIVNAME (options

FILENAME is the current file name. FILETYPE is the current file type. DRIVNAME is the current drive. NEWFILE is the new

filename. NEWTYPE is the file type. DRIVNAME is the name of the drive where the new file is to be located.

Although the use of wildcard characters is allowed when using RENAME, it is best not to use them to avoid any ambiguity. We will provide examples of how to use the RENAME command with and without wildcards.

Option	Function
DELETE	The file is to be protected from erasure. The file cannot be deleted or renamed without changing its protection status.
NODELETE	This option removes a file's delete protection status.
NOQUERY	When this option is specified, the system manager is not asked on a file-by-file basis if it is ok to rename the file.
NOWRITE	Write protection is to be removed from a file.
QUERY	The system manager is asked on a file-by-file basis if it is ok to rename a file. This is a default option when any wildcard characters are used.

Figure 4.29. The RENAME command options

DEFAULT OPTIONS

- NOQUERY is a default option when the FILENAME and FILETYPE are specified.
- QUERY is a default option unless FILENAME, FILETYPE, NEWFILE, and NEWTYPE are specified.
- All operations are automatically displayed on the screen.

USING THE RENAME COMMAND

Suppose you entered the following RENAME command:

```
>RENAME SYSTEM.ERRMSG S (NOW<RETURN>
SYSTEM.ERRMSG:S protection: DELETE, NOWRITE
>
```

Here we have changed the protection status of a file called SYSTEM.ERRMSG. As supplied by the THEOS vendor, this file has both delete and write protection. In order to make any changes to this file, the write protection status must be removed so the file can be rewritten.

Let's examine another RENAME command entry:

```
RENAME USER1.EXEC S USER2.=<RETURN>
USER1.EXEC:S renamed USER2.EXEC:S
```

This example changes a file named USER1 EXEC to USER2 EXEC. The equal sign (wildcard character) means that the new file type is to be the same as the old file type...EXEC. Because the descriptions of the files were specific the default option in this case was NOQUERY. Remember the contents of the files were identical. RENAME has only changed the name of the file.

Suppose there are some data files on the system that are not to be changed or deleted. Type:

```
>RENAME *.DATA S (DELETE WRITE<RETURN>
FILE1.DATA:S Okay to change protection (Y/N)? Y<RETURN>
FILE1.DATA:S protection: DELETE, WRITE
FILE2.DATA:S Okay to change protection (Y/N)? N<RETURN>
FILE3.DATA:S Okay to change protection (Y/N)? Y<RETURN>
FILE3.DATA:S protection: DELETE, WRITE
>
```

Wildcards were used because it was not certain how many data files were involved. THEOS searched the S drive to locate all files with a file type of DATA. Because a wildcard was used as part of the file description, the QUERY option was automatically in effect. As can be seen, THEOS prompts for an indication whether or not protection status of the file is to be changed. In one instance a response of N was recorded. THEOS ignored that file and moved on to the next file matching the description. After all files matching the description were located, the user is returned to the THEOS prompt.

SUMMARY

The RENAME command provides two functions:

- Changing a file's protection status.
- Renaming an old file description.

Certain files that are part of the THEOS operating system like the SYSTEM.ERRMSG file are both write and delete protected. When such a file's contents must be changed, the RENAME command must be used to change its protection status.

Session 16: The ERASE Command

Now that we have had a chance to examine some of the commands that involve file and disk operations, it is appropriate that we begin our discussion of the THEOS command that permits the erasure of files. The ERASE command is capable of erasing any file stored on any disk provided that the file is not delete protected. If you need to erase a file that is delete protected, you must first change the protection status using the RENAME command. A privilege level of 1 is required to use ERASE.

The format of the ERASE command is:

```
ERASE FILENAME FILETYPE DRIVNAME (options)
```

FILENAME is the file name to be erased. FILETYPE is the file type to be erased. DRIVNAME is the drive where the file is located.

The use of wildcards with this command is convenient and is illustrated in the subsequent examples.

Option	Function
TYPE	FILENAME, FILETYPE and DRIVNAME to be erased are displayed on the screen before erasure. This is the default option.
QUERY	The system manager is asked on a file-by-file basis if it is ok to erase a file. This is the default option when wildcards are used.
NOQUERY	The system manager is not asked if it is ok to erase a file. This is the default option when no wildcards are used.

Figure 4.30. ERASE command options

USING THE ERASE COMMAND

Before executing the ERASE command it may be helpful to execute a FILELIST command. FILELIST will provide a listing of files on the currently logged account. These can be selected for possible erasure.

The ERASE command options are designed to protect against accidental file erasure. The following examples will illustrate this feature.

For example, suppose you were to type:

```
>ERASE USE COMMAND<RETURN>
USE.COMMAND:S, OK to erase (Y/N)?N<RETURN>
No files erased.
>
```

In this example FILENAME and FILETYPE are specified, but the drive where the file is located is not. THEOS automatically searches all attached disks. When it finds a file which matches the description typed, THEOS asks if it has found the correct file. Any response other than Y causes THEOS to respond that no files have been erased.

Let's examine another usage of ERASE. Suppose you were to type:

```
>ERASE USE COMMAND<RETURN>
USE.COMMAND:S, OK to erase (Y/N)?Y<RETURN>
USE.COMMAND:S erased
1 file(s) erased, 3 block(s) recovered.
>
```

This is identical to our previous example except that Y is entered in response to the prompt. After the <RETURN> key is pressed, THEOS erases the file and displays a message to that effect, along with the total number of files erased and the amount of disk space recovered which can be reused.

Let's see what happens when wildcard characters are indicated with ERASE. If you were to type:

```
>ERASE XYZ*.* S<RETURN>
No files erased
>
```

THEOS searches the S disk for a FILENAME of XYZ and a FILETYPE of *. Since no files on the S disk were found matching that description, no files were erased.

Suppose you entered ERASE with *.* specified as the FILENAME and FILETYPE as follows:

```
>ERASE *.* A<RETURN>
FILE1.COMMAND:A Okay to erase (Y/N)? Y<RETURN>
FILE1.COMMAND:A Protected
FILE2.DATA:A Okay to erase (Y/N)?Y<RETURN>
FILE2.DATA:A erased
FILE3.BASIC:A Okay to erase (Y/N)?Y<RETURN>
FILE3.BASIC:A erased
2 file(s) erased, 12 block(s) recovered.
>
```

Note that drive A was specified. The first file located was protected causing THEOS to ignore the request to erase it. A total of two files were erased, and twelve disk blocks were recovered for reuse.

Let's examine one more ERASE example:

```
>ERASE USE COMMAND S (NOQ<RETURN>
USE.COMMAND:S erased
1 file(s) erased, 3 block(s) recovered.
```

Note the use of the NOQUERY option. In order for this option to work, FILENAME, FILETYPE, and DRIVNAME must be specified.

The use of options with ERASE is far more limited than with the other THEOS commands. The reason should be clear...to avoid accidental erasure of files. Special caution should be exercised when the system manager is erasing files belonging to the system account. Files in the system account are public and as such every THEOS user has access to them.

WHEN TO USE THE ERASE COMMAND

The ERASE command is used when files become outdated or are no longer used. This applies mostly to application programs no longer

in use or data files created by those programs. Some application programs provide a menu option for erasing unwanted or unused files. Whenever those choices are available, use them rather than ERASE. When no provisions for erasing files have been furnished, as is the case with most word processing software, the ERASE command is the only means of erasing unused files.

SUMMARY

The ERASE command provides a means of erasing files belonging to the currently logged account. The command is designed such that the user is protected against the accidental erasure of files. Many software packages provide menu selections which provide for erasure of selected files. The ERASE command is recommended for use in those situations where unused files are occupying space which could be used for more current data.

Session 17: The EDIT and SCRIPT Commands

These commands provide THEOS users with limited word processing capability as well as the means to create and edit sequential files. The EDIT command, when invoked, creates sequential files containing user entered data. The SCRIPT command permits data entered via the EDIT command to be formatted on a printed page according to user specifications. This tutorial session will primarily be devoted to the EDIT command because it will be used in later sessions to create EXECs, and it is easier to master than SCRIPT. The EDIT command will be discussed first followed by a brief description of the SCRIPT command.

The EDIT Command

The EDIT command is the predominant method used to create THEOS sequential files. Our primary use for the EDIT command will be to create EXEC programs. (See chapter 6 on the EXEC language) Although this is our primary use for this command, the editor can be used to manipulate any type of sequential file.

The EDIT command provides THEOS users with a **file editor**. The file editor permits text to be changed, added, or deleted from a file. As a rule, EDIT commands are easily understood as they are English words whose definitions indicate the function they perform. For example, the INPUT command is used to put data into a file. Most EDIT commands can be abbreviated using their first letter. As usual we'll indicate any allowed abbreviation in bold.

Note: Although EDIT is considered a THEOS command, there are additional commands within EDIT. In this tutorial all commands relating to EDIT will be indicated in uppercase letters.

EDIT is a line orientated editor. This means that after a command is executed, such as INPUT, THEOS assumes that multiple lines of text will be entered and input is allowed on a line-by-line basis. When a character on a given line is incorrect and needs re-editing, the MODIFY command temporarily allows EDIT to function as a **character** editor.

Throughout this tutorial *editor* will refer to the editing of a sequential file after invoking the EDIT command

The format of the EDIT command is:

EDIT FILENAME FILETYPE DRIVNAME (option)

FILENAME is the name of the file to be edited. FILETYPE is the type of the file to be edited. DRIVNAME is the drive where the file is to be located. In order to enter the edit mode, all three parameters must be specified. Only two options are available for use with the EDIT command. They are:

BACKUP and NOBACKUP

BACKUP is the default option. The editor *automatically* generates a backup copy of the file being edited. The NOBACKUP option suppresses this feature. The only reason to suppress the auto-backup feature is if there is limited disk space. All examples will use the default of BACKUP.

If you were to type:

```
>EDIT NEWFILE EXEC S<RETURN>
NEW FILE:
EDIT:
*
```

THEOS would respond with a “NEW FILE” message because no file had previously been stored using that name. The asterisk character is the editor’s prompt character. The asterisk indicates that the editor is waiting for a command.

The Built-In Help Facility

There are many commands available within the EDIT command. To assist you in remembering the commands, an on-screen help facility has been provided. In order to activate the help facility type **HELP** at the edit prompt as shown below:

```
EDIT NEWFILE EXEC SRETURN
*HELP<RETURN>
```

A complete list of the available commands along with their correct syntax is displayed on the screen. Because the list of commands is longer than the number of lines available on most screens, the wait character [↑] is displayed at the bottom of the screen. Press any key to view the rest of the commands. The internal editor help commands are listed in figure 4.31.

AGAIN	
BOTTOM	
CASE	U L M A B
CHANGE	/str/str2[/linecnt [occurcnt [start]]]
COLUMN	
COMBINE	
CSI	command
DELETE	[/str/{count}]
DOWN	[count]/str/
DUP	count
ERROR	[count] (macro only)
FILE	[fn[.ft[:fd]]]
FIND	statement
GET	[file] [[/frstr/{frct} [/tostr/{toct}]]
HELP	
HTAB	[char [ON OFF]]
INPUT	statement
LINEMODE	[ON OFF]
LIST	[count]
LOCATE	[/str/]
MODIFY	[count]
NAME	[fn[.ft[:fd]]]
NEXT	[count]
NUMBER	[start incr]
PAGE	[number]
PROMPT	
PUT	[file] count/tostr/
PUTD	[file] count/tostr/
QUIT	[number]command
RENUMBER	[start [incr]]
REPLACE	statement
SAVE	[fn[.ft[:fd]]]
SKIP	[[-]nn] (macro only)
SPLIT	/str/
TABSET	[number] . . .]
TOP	
TRUNCATE	[number]
TYPE	[count]
UNNUMBER	
UP	[count]/str/
VERIFY	[[number] [ON OFF]]
WRAPMODE	[ON OFF]
X	[? statement]count]
Y	[? statement]count]
Z	[? statement]count]
Zone	[number]

Figure 4.31. Screen display of internal editor help messages

The Edit Commands

The commands we will use to add, modify, and store files on disk are:

- INPUT
- DELETE
- SAVE
- FILE
- QUIT
- **MODIFY** consisting of the **subcommands** insert (I), delete (D), replace (R), beginning of line (B), end of line (E), quit (Q), and end modify **<RETURN>**
- **<RETURN>**
- COLUMN
- BOTTOM
- TOP
- LIST
- LOCATE
- NUMBER
- UNNUMBER
- RENUMBER
- Arrow Keys

All commands are executed from the editor prompt (*) with the exception of the **MODIFY** subcommands.

Command	Function
INPUT	Permits new text to be added to a file.
DELETE	Allows lines of text to be deleted from a file.
SAVE	The file being edited is saved to disk but the file remains in memory for further modification.
FILE	The file being edited is saved to disk and a "file saved" message is displayed on the terminal. User is returned to CSI.
QUIT	The current edit session is terminated with no changes made to the file.
MODIFY	Permits modification of the current line of text. Its built-in subcommands provide easy editing of previously entered text.
<RETURN> key	Can be used to advance to the next line of text or to end the use of a MODIFY subcommand.
COLUMN	Displays column numbers across the terminal screen. Useful for positioning text on the screen.

BOTTOM	Positions the cursor at the end of the file being edited.
TOP	Positions the cursor at the top of the file being edited.
LIST	Permits the viewing of several lines of text on the screen for proof-reading or re-editing.
LOCATE	A particular string of characters anywhere in the text can be "found" using this command.
NUMBER	This command numbers each text line automatically in increments of 10.
UNNUMBER	Allows line numbers to be removed from a file.
RENUMBER	This command will renumber the text lines.
Arrow Keys	These keys are used for cursor movement within a file. When the down arrow is pressed, the message "DOWN 1" is displayed on the screen. When the up arrow is pressed, the message "UP 1" is displayed on the screen.

Figure 4.32. EDIT commands and their functions

There are a number of other editing commands available for use. Those described here should be sufficient to create and edit our files.

USING THE EDIT COMMANDS

Since we will be creating a new file (one that doesn't already exist on your system) it would be best if you sat down at your terminal and entered the following examples. Remember, the text to be entered via the keyboard is displayed in bold. Space bar means press the space bar.

In the first example we will illustrate the use of the **INPUT**, **RETURN**, **SAVE**, **FILE** and **QUIT** commands. To enter the editor from the THEOS prompt type:

```
EDIT NEWFILE EXEC S<RETURN>
NEW FILE:
EDIT:
*I<RETURN>
THIS IS A TEST OF THE EDITOR INPUT COMMAND. WHEN EDITING A FILE <RETURN>
TRY TO ANTICIPATE WHEN NEARING THE END OF A LINE BECAUSE THE EDITOR <RETURN>
WILL NOT MOVE AUTOMATICALLY TO THE NEXT LINE.<RETURN>
IF ON A NEW LINE YOU TYPE space bar
THEN PRESS THE RETURN KEY <RETURN>
THE EDITOR AUTOMATICALLY PRINTS A BLANK LINE.
IF YOU PRESS THE RETURN <RETURN> KEY TWICE
THIS RETURNS YOU TO THE EDIT PROMPT.<RETURN>
*
```

As you can see the edit prompt reappears. When inputting lines of text pressing return twice tells the editor you are finished. In each line of text above we pressed RETURN before the end of line.

If you would like to save your work to disk and continue editing you can type:

```
*SAVE<RETURN>
"NEWFILE.EXEC:S" saved.
```

Your work will be saved to disk and will also remain in memory for further editing.

If you would like to save your work and discontinue editing at this time, type:

```
*FILE<RETURN>
"NEWFILE.EXEC:S" filed.
```

Notice that the THEOS prompt appears. QUIT allows you to exit the editor without saving the file to disk.

```
*QUIT<RETURN>
>
```

In the preceding example, THEOS searches disk S for a file called NEWFILE.EXEC. When the file is not found, it automatically opens a new file with that name.

In the cases above where you entered "SAVE" or "FILE", the file was stored on disk S. Had you typed QUIT, there would have been no new file stored on disk. Also, note that the three commands SAVE, FILE, and QUIT must be typed in their entirety in order to be executed.

TOP, BOTTOM, AND COLUMN

Let's use another example to explore the use of the TOP, BOTTOM, and COLUMN commands. Using the file created in example 1 type:

```
EDIT NEWFILE EXEC<RETURN>
EDIT:
*
```

Here EDIT transferred the file from disk to memory so it is ready for editing. Since THEOS was able to find the file on disk, the "NEW FILE": message was not displayed. Now type:

```
*BOT<RETURN>
```

The EOF: message appears on the screen indicating that you have reached the **end of file**. If you press the up arrow, the message "UP 1" appears on the screen followed by the appropriate line of text. If you now press the down arrow, you will be moved down one line in the text and the message "DOWN 1" will appear on the screen. Once again you have been moved to the end of the file.

Note that when you entered:

***TOP<RETURN>**

the "TOF:" message appeared on the screen. This indicates you have reached the top of file.

Suppose you were to type:

```
*COL<RETURN>
 1          2          3          4          5          6          7
123456789012345678901234567890123456789012345678901234567890
TOF:
```

Note the series of numbers which appear on the screen. This command is useful when positioning text on the screen. The numbers provide a visual indication of screen columns one through eighty. If you would like to position some text beginning at position 20:

- Type I to begin inputting text after the EDIT prompt
- Use the space bar to move the cursor under column 20
- Begin typing

When inputting text, the left arrow can be used for deleting text (before a <RETURN> is entered); the space bar can be used for moving the cursor to the right without moving the cursor to the next line.

LOCATE

EDIT's LOCATE command allows you to locate a particular string of characters in text previously entered. In the following example we'll examine the usage of LOCATE. Assuming the editor is still active, first be certain you are at the top of the file by typing:

***TOP<RETURN>**
TOF:

Let's assume we want to find the word "automatically".

***L /AUTOMATICALLY<RETURN>**
*

The editor will then display the line containing the string of characters you wish found and display the EDIT prompt and wait for your next command. The correct syntax for the LOCATE command is:

LOCATE /desired string

Note that there must be a space between the command and the slash mark. Here is what actually appears on your screen:

30 WILL NOT MOVE AUTOMATICALLY TO THE NEXT LINE
*

MODIFY

Let's continue our discussion of the EDIT commands by examining the use of the MODIFY command and its subcommands. MODIFY offers an easy way of changing previously entered text. When the MODIFY command is executed, the line of text to be modified is displayed with the cursor positioned at the beginning of the line. Using the previous example let's assume that the screen appears as follows:

PRINTS A BLANK LINE. IF YOU PRESS <RETURN><RETURN> AT ANY TIME
*MOD<RETURN>
PRINTS A BLANK LINE. IF YOU PRESS <RETURN><RETURN> AT ANY TIME

The cursor is positioned at the beginning of the line with the modify mode active. If the changes you wish to make do not start at the beginning of the line, you can use the left and right arrow keys to position the cursor at the desired location.

Typing I (for insert) allows characters to be inserted starting with the position of the cursor. To terminate the insert mode press <RETURN>. Each time you type D (for delete) a character to the right of the cursor is deleted. To terminate the delete mode press <RETURN>. If you type R (for replace) the next character entered replaces the current character. When all the desired characters have been replaced, press <RETURN>. If E is entered the cursor will be repositioned at the end of the line. Typing B returns the cursor to the beginning of the line.

Practice using these commands with the line of text just displayed by executing **MODIFY**. Don't worry about making mistakes. The original file is still on disk and your changes will not be saved unless you type **SAVE** or **FILE**. If you don't want to save your text modifications, you can type **QUIT** <RETURN> so that any changes made are ignored and the modification mode is exited. Pressing <RETURN> while in the **MODIFY** mode tells the editor you are finished with the **MODIFY** subcommands. Pressing <RETURN> a second time exits the **MODIFY** mode and returns you to the **EDIT** prompt.

DELETE AND LIST

In this example we'll examine the **DELETE** and **LIST** commands. First load the file into memory by typing:

```
>EDIT NEWFILE EXEC S<RETURN>
EDIT:
* <RETURN>
10 THIS IS A TEST OF THE EDITOR INPUT COMMAND. WHEN EDITING A FILE
*DEL
20 TRY TO ANTICIPATE WHEN NEARING THE END OF LINE BECAUSE THE EDITOR
*DEL 2
40 IF ON A NEW LINE YOU TYPE [space bar]<RETURN> THE EDITOR AUTOMATICALLY
*TOP
TOF:
* <RETURN><RETURN><RETURN>
40 IF ON A NEW LINE YOU TYPE [space bar]<RETURN> THE EDITOR AUTOMATICALLY
*
50 PRINTS A BLANK LINE. IF YOU PRESS RETURN TWICE
*
EOF:
*TOP
*LIST5
TOF:
40 IF ON A NEW LINE YOU TYPE [space bar]<RETURN> THE EDITOR AUTOMATICALLY
50 PRINTS A BLANK LINE. IF YOU PRESS RETURN TWICE
EOF:
*
```

In this example, the <RETURN> key was pressed to display the first text line. Next, **DEL** was typed. **DELETE** by itself with no number after it will delete the current line of text. The cursor is then positioned at the beginning of the next line which is displayed. Next **DEL 2** was typed which instructed the editor to delete two lines of text. The cursor is positioned on the next available line of text and dis-

played. TOP was typed to go to the top of the file and the RETURN key was pressed three times to display the remaining two lines of text and the EOF message indicating the end of the file. To verify that only two lines of text remained, LIST 5 was typed (meaning “list 5 lines of text”). The fact that only two lines of text were displayed confirms that a total of three lines of text were deleted.

NUMBER, UNNUMBER, AND RENUMBER

In the following example, we will examine the NUMBER, UNNUMBER and RENUMBER commands. Suppose you were to type:

```
>EDIT NEWFILE EXEC<RETURN>  
EDIT:  
*NUMBER<RETURN>
```

If you now examine the contents of the file, you will find that each line has been assigned a number in increments of 10.

You can examine the lines one at a time by pressing <RETURN> or by pressing the down arrow. To retain the numbers as a permanent part of the file, type FILE at the EDIT prompt to store the line numbers to disk with the rest of the file. The line numbers could be later removed by re-editing the file, typing UNNUMBER in response to the EDIT prompt and re-saving the file.

The RENUMBER command is used to renumber lines of text in a file in a case where new lines of text have been inserted between two existing line numbers. The NUMBER command assigns line numbers starting with the number 10 with subsequent numbers in increments of 10. As we will see in chapter six on EXECS, a line must occasionally be inserted between two numbered lines. The RENUMBER command is useful in such situations.

THE SCRIPT COMMAND

The SCRIPT command is covered thoroughly in the THEOS documentation, and although it is powerful in its formatting capabilities, it is not designed for day to day word processing applications.

Some of the formatting features included in this command are: headers, footers, text justification, titling, variable line spacing, paging control, access to system time and date, input from terminal during processing, etc.

The SCRIPT command is used for the formatting of an existing sequential file. Although other means of creating sequential files are available in THEOS, the primary method is the EDIT command.

EMBEDDED COMMANDS

Unlike other THEOS commands, the SCRIPT commands are part of the file operated upon by that command. An embedded command is a command that is typed as part of the actual text. The SCRIPT commands must be preceded by a period so as to allow THEOS to interpret the lines as a formatting command. All embedded commands are entered at the time of text entry (EDIT). Let's examine an embedded SCRIPT command using the file we have been working on (NEWFILE EXEC).

```
>EDIT NEWFILE EXEC<RETURN>
EDIT:
*I[RETURN
.LM 15<RETURN>
<RETURN>
*FILE<RETURN>
>
```

The command we have embedded in our text is .LM (left margin); the number assigned to the left margin is 15. This means that when the file is printed, the left margin for all text in the file will be 15.

OTHER SCRIPT FEATURES

Multiple files can be linked together into one document using SCRIPT's LINK feature. *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 SCRIPT but may not be supported by the printer being used. Check the documentation provided by your printer manufacturer to see which features are supported and how they are implemented.

If you are doing extensive word processing, consider one of THEOS's word processing packages. They are easier to learn and use than are SCRIPT and EDIT.

If you would like to learn more about SCRIPT, consult your THEOS documentation for instruction on the use of the SCRIPT command.

SUMMARY

The EDIT and SCRIPT commands work together to provide a text processing system. The SCRIPT commands are formatting commands which are embedded in an edit file and control how the file is printed on paper. Only the most commonly used EDIT commands were described in this section. Refer to the THEOS documentation for details on those commands not covered here.

Session 18: The SET/START/STOP Commands

These THEOS commands are used together to define and set up system parameters such as time, date, memory allocation, multi-user operation, single-user operation, system history, etc. The SET command will be discussed first followed by START and then STOP.

THE SET COMMAND

This command allows such functions as time, date, and memory allocation to be changed as the needs of the system dictate. The format of the SET command is:

SET parameter status

Parameter is the function altered by the command. Status is the condition to which the parameter is changed.

More parameters are available in THEOS than those discussed here. We have limited our discussion to those parameters most useful in day-to-day system operation.

Certain privilege levels may be required to set some parameters.

Parameter	Function
DATE mm/dd/yy	Permits changing of the system date. Requires a privilege level of 5.
RDYMSG	Permits the on-screen display of the system ready message which consists of the return code, current system time and elapsed time of function executed.
TIME hh:mm:ss	Permits changing of the system clock. Requires a privilege of 5.
MEMORY	Allocates memory partitions for multi-user operation and memory size. Requires a privilege level of 5.

Figure 4.33. SET parameters and their functions

The following entry illustrates how SET can be used to change the system date:

>SET DATE 12/29/84<RETURN>

THEOS will reset the system date to December 29, 1984.

SET can be used as shown below to change the status of the system ready message:

>SET RDYMSG ON<RETURN>or SET RDYMSG OFF<RETURN>

The purpose of the system ready message is to display the following information:

- the system return code
- the current system time
- the time required to execute the last THEOS command typed

The return code is a number THEOS uses to determine whether or not commands have been executed without error. A return code of 0 indicates that a command operation was successfully completed. Other return codes will be discussed in chapter 6. The ready message information can be most useful when testing THEOS programs or EXECs that use the return code to display error messages.

To set a new system time type:

>SET TIME 15 24<RETURN>

Enter any char to set time to (15:24:00) **<RETURN>**

The current month and year are displayed along with the new time entry. Rather than automatically setting the time as soon as **<RETURN>** is pressed, the prompt provided by THEOS allows setting the exact time when a character is entered. Note that THEOS uses a twenty-four hour time clock.

THE SET MEMORY COMMAND

The SET MEMORY command is vitally important since it allocates memory partitions for multi-user operation. A memory partition is a portion of the total system memory assigned to each THEOS user. This partition allows the user enough memory to run application software and share the common microprocessor. Memory partitions

must be allocated in ascending order and deallocated in descending order. When the multi-user version of THEOS is booted, there is only one active memory partition.

Most multi-user THEOS systems are supplied with a minimum of 256K RAM which is sufficient for five users (each with 45K) and a printing spooler shared among all users. Before multi-user operation can begin, memory must be allocated to each user. Conversely, if in multi-user mode, memory must be deallocated before single-user operation can begin. To clarify, let's use an example.

In a system with 256K RAM, when the system is first booted one memory partition is automatically activated. This partition allocates approximately 45K for the user and 20K for the THEOS nucleus. To prepare for multi-user operation and allocate memory for user partitions two through five, type:

```
>SET MEM 2 1 1<RETURN>
>SET MEM 3 1 2<RETURN>
>SET MEM 4 1 3<RETURN>
>SET MEM 5 1 4<RETURN>
```

To prepare for single-user operation and deallocate the available memory type:

```
>SET MEM 5 1 4<RETURN>
>SET MEM 4 1 3<RETURN>
>SET MEM 3 1 2<RETURN>
>SET MEM 2 1 1<RETURN>
```

Enter the commands detailed in this section from the main terminal or from the terminal attached to port one.

The START and STOP Commands

These commands are used together primarily to activate and deactivate user partitions allocated by the SET MEMORY command.

The START command is used to activate the memory partitions. Using the example above, to activate the memory partitions for multi-user operation type:

```
>START 2<RETURN>  
>START 3<RETURN>  
>START 4<RETURN>  
>START 5<RETURN>
```

When the START command is executed, the user whose partition was most recently activated is prompted on the screen to "Logon please:". If that particular terminal has not yet been turned on, it may be necessary to enter a <RETURN> after start-up in order for the logon message to appear. Once the logon messages appear on the terminals, the system is ready for multi-user operation.

The STOP command is used to deactivate the memory partitions. In order to deactivate the multi-user system initialized in our examples, type:

```
>STOP 5<RETURN>  
>STOP 4<RETURN>  
>STOP 3<RETURN>  
>STOP 2<RETURN>
```

This command cannot be used unless a user has logged off. When a user on a particular terminal has logged off and the terminal is still active, the message: ****Partition has been stopped!**** appears on the screen. This informs any prospective users that the system is not available for multi-user operation.

The STOP command only stops the operation of a memory partition. It does not deallocate the memory. Deallocation is accomplished by the SET MEMORY command.

SUMMARY

The SET command is used to change certain parameters such as time, date, and memory allocation for multi-user operation. The START and STOP commands are necessary in order to activate and deactivate memory partitions to start and stop multi-user operation. These functions must be performed from the terminal attached to port one on the system.

Session 19: The SHOW Command

The SHOW command provides THEOS users with the ability to display the status of certain system parameters. The format of the SHOW command is:

SHOW parameter

Parameter	Function
BANK	Displays the available memory banks and their sizes.
DISK	Displays the status of all currently attached disks or, when specified, a particular disk.
MEMORY	Displays the current allocation and usage of memory (RAM).
PRIVLEV	Displays the privilege level of the currently logged user.
RDYMSG	Displays status (on or off) of the ready message switch.
SERIAL	Displays the serial number of the THEOS operating system in use.
TIME	Displays the current system time and date.
USERS	Displays information about any users currently logged onto the system.
WHO	Displays the account name of the currently logged account.

Figure 4.34. Show parameters

The following entry illustrates the use of the SHOW BANK command:

>SHOW BANK<RETURN>			
> Region	Low	High	Size
NUCLEUS	0000H	3FF4H	16363
Bank #00	3FF5H	FF7FH	49035
Bank #01	4000H	FFFFH	49152
Bank #02	4000H	FFFFH	49152
Bank #03	4000H	FFFFH	49152
Bank #04	4000H	9038H	20537

The data of most interest to us here is the bank number and the size of each bank. This command was executed on a THEOS system with 256K RAM. The information displayed is an indication of how the memory might be allocated in a THEOS system with five users and a print spooler.

The SHOW DISK command can be executed as follows:

```
>SH DI<RETURN>
```

```
Label: "HARDISK1",  
Archived to disk "TEST " on 05/19/85 10:44  
Capacity: 14,544K bytes (303-6-32),  
Available: 7,144K bytes (49%),  
Largest area: 7,109K bytes,  
830 files in use (out of 1,944)
```

Five lines of information about the disk are displayed.

- Directory label
- Disk capacity in K bytes, number of cylinders, number of heads, number of sectors per track
- Amount of storage available in K bytes and percentage
- Largest amount of contiguous disk space
- Number of files currently in the disk directory, number of entries allocated.

In addition the SHOW DISK command checks for proper disk space allocation and reports any misallocated space with an error message displayed on the line following the number of directory entries allocated. Misallocations are indicated by a + or - number in K bytes.

Negative misallocation indicates that more space has been allocated for file storage than is actually being used. Positive misallocation indicates that less space has been allocated than actually in use. These conditions are not normal and should be corrected before the system is used further. Negative misallocation is usually a result of improper closing of a file due to a *bug* in a software package or a power failure during system operation preventing proper exiting of a program. The positive misallocation is the more serious condition of two because it may mean that two files are sharing the same disk space and that one or both contain invalid data. See chapter 5 for instructions on how to correct misallocations.

The following example illustrates usage of the SHOW MEMORY command:

>SH MEM<RETURN>

```

Port Bank MEMLO MEMMX MEMHI SP--- PC--- Size Routine Username
      0000H 3FF4H          16373 NUCLEUS
1*    0 3FF5H FF7FH F809H F802H 4C01H 49035 SHOW    USER2
>
    
```

This command is used to display the current memory allocation. In this example there is only one user on the system. When this command is executed in a multi-user environment, information about each user is obtained. The information displayed is port, bank, size, routine, and user. Port references the port on the computer in which the terminal is plugged. Bank is the number of the memory bank containing the RAM for that user. Size is the amount of memory available to that user. Routine is the THEOS command currently being executed by that user. In this case the command is SHOW. Username is the account name of the currently logged user. The asterisk indicates the user executing the SHOW command.

As shown in the following example, the SHOW PRIVLE command indicates the privilege level of the currently logged user.

```
>SHOW PRIVLE<RETURN>
PRIVLEV =5
>
```

The on/off status of the ready message switch is displayed by the SHOW RDYMSG command as shown below:

```
>SHOW RDYMSG<RETURN>
RDYMSG =ON
```

The SHOW SERIAL command displays the serial number of the THEOS system in use as shown in the following example:

```
>SHOW SERIAL<RETURN>
SERIAL =00-00000 (whatever your serial number is)
```

As shown below the SHOW TIME command displays the current date and time:

```
>SHOW TIME<RETURN>
hh:mm:ss day of week, month/date, current year
>
```

The SHOW USERS command displays information about the currently logged users such as the port number of their terminal, their account name, and date and time of logon. In the following example, one user, USER3, is currently logged onto the system.

```
>SHOW USERS<RETURN>
Port Username Log-Date Log-Time
1* USER3 01/01/84 11:11:31
>
```

The SHOW WHO command displays the account name of the currently logged account.

```
>SHOW WHO<RETURN>
USER3 Logon 11:11:31 01/01/84
>
```

This command is useful because it is possible to forget the account name of the currently logged account. By typing this command you are instantly reminded of the account name under which you are logged on, as well as the time and date of the logon.

SUMMARY

The SHOW command is useful for displaying the status of certain THEOS parameters. It can be executed at any time from the THEOS prompt. The most commonly used parameters were examined in this tutorial.

Session 20: The SPOOLER Command

In a multi-user system there is a need to provide a means of controlling access to the printers attached to the computer. This is particularly important if only one printer is available. One of the benefits of a multi-user system over a single-user system is its ability to perform multiple jobs simultaneously. It would be most unfortunate if a printer created a time *bottleneck* because all users were waiting to use the printer. Fortunately, THEOS provides the spooling function which allows files specified for printing to be temporarily written to disk when the printer is *busy*.

When a *spooled* disk file is created, it is automatically assigned a filename by THEOS. The filenames are assigned in numerical order as they are created beginning with the number one. When the printer is available, each file is printed beginning with the first spooled file. In order to avoid unnecessarily occupying disk space, each file is erased after printing is completed.

CONTROL OF SPOOLER FUNCTIONS

Application software generally is written to automate operation of the spooler. For this reason our tutorial will be limited to discussing the commands related to activating and deactivating the spooler. For a discussion of all SPOOLER commands, refer to the THEOS documentation.

If THEOS commands are executed at the operating system level such as LIST or FILELIST and the spooler is active, the function of the spooler is automatic. THEOS writes the information to a disk file. If the printer is not busy, the file is immediately printed. If the printer is busy, the information remains in the file until the printer is available.

Use of THEOS's printer echo feature with the spooler is not recommended. Printer echo is the direct printing of a screen display by pressing ESC P RETURN. Use of this feature may result in misallocated disk space. If you want to use this feature, be certain that the spooler is not activated.

The format of the SPOOLER command is:

SPOOLER function

Function	Purpose
INIT	Initializes the spooler for use; must be performed before any of the spooler functions can be used. This command requires a privilege level of five.
QUIT	This command deactivates the spooler and requires a privilege level of five.
STOP	Stops printing after the current report. This command is used to terminate the use of the spooler. Termination of printing in the middle of a report should be handled automatically by application software.

Figure 4.35. SPOOLER functions

Using The SPOOLER Commands

In order to use the spooler functions the following rules must be observed:

- The device PRINTER1 must be attached.
- The file SYSTEM.SPOOLQUE must exist, preferably on a disk by itself. Early versions of THEOS (prior to version 5.6) caused spooled files to overwrite files on the disk where SYSTEM.SPOOLQUE was located.
- The SPOOLER INIT function must be performed.
- Execute all commands relating to the spooler from the main terminal attached to PORT1.

Note: The THEOS spooler only allows spooled files to be directed to the printing device attached as PRINTER1.

SINGLE-USER SPOOLING

It is possible to use the spooler in the single-user mode. Doing so *fre*s the terminal for another task as soon as the spooled file is written to the disk. Assuming the rules outlined above have been followed, to use the spooler while in single-user mode execute the following steps:

- Be certain PRINTER1 is attached
- Be certain the file SYSTEM.SPOOLQUE is located on an active drive
- Enter the following command:

```
>SPOOLER INIT<RETURN>
Spool parameters: Queue=A, Copies=1, Nohold.
Spooler has form "A" mounted.
Spooler is waiting for work.
>
```

This message informs you that the spooler is ready for operation.

Caution: Using the spooler in single-user mode means that approximately 5K of working memory is taken away from that user's working memory. Many THEOS programs require more than 40K bytes for proper operation. If that is the case it is recommended that you use the spooler in multi-user mode to maximize memory for each user partition.

MULTI-USER SPOOLING

Using the spooler in multi-user mode is a little more complex. To do so:

- Be certain PRINTER1 is attached
- Be certain the file SYSTEM.SPOOLQUE is located on an active drive
- The spooler must be loaded
- Use the SET MEMORY command to allocate memory for each user
- Allocate one additional memory partition for the spooler

To set up a two user system with spooler type:

```

>A PRINTER1 PORTn<RETURN>
>LOAD SPOOLER<RETURN>
>SET MEM 2 1 1; SECOND MEMORY PARTITION<RETURN>
>SET MEM 3 3500 1; SPOOLER PARTITION<RETURN>
>SPOOLER INIT<RETURN>
Spool parameters: Queue=A, Copies=1, Nohold.
Spooler has form "A" mounted.
Spooler, is waiting for work.
>START2<RETURN>
>

```

Line 1 ensures that a printer was attached to the port of your choice. Before attaching the printer, be certain that the desired port is active and has the baud rate set to match your printer's baud rate. Line 2 loads the spooler. Lines 3 and 4 allocate memory for the second user and spooler partitions. After the spooler is initialized, the second user partition is activated using the START command.

To set up a five user system with spooler type:

```

>ATTACH PRINTER1 PORTn<RETURN>
>LOAD SPOOLER<RETURN>
>SET MEM 2 1 1<RETURN>
>SET MEM 3 1 2<RETURN>
>SET MEM 4 1 3<RETURN>
>SET MEM 5 1 4<RETURN>
>SET MEM 6 3500 5<RETURN>
>SPOOLER INIT<RETURN>
Spool parameters: Queue=A, Copies=1, Nohold.
Spooler has form "A" mounted.
Spooler is waiting for work.
>START 2<RETURN>
>START 3<RETURN>
>START 4<RETURN>
>START 5<RETURN>

```

Line one attaches a printer to the port of your choice. Line two loads the spooler module. The following entries allocate memory for each user and spooler partition. The spooler is then initialized and each user partition started.

Stopping The Spooler In Single-User Mode

The following command terminates the spooling function while in the single-user mode:

```
>SPOOLER QUIT<RETURN>
>
```

This command can only be executed if the spooler is inactive. A privilege level of five is required to execute this command.

Note: In addition to terminating spooler operation, issuing a SPOOLER QUIT command causes the printer being used by the spooler to be detached from the system. If this printer is to be used after issuing the command, it must be reattached by typing:

```
>A PRINTER PORTn<RETURN>
>
```

Stopping The Spooler In Multi-User Mode

The spooler is generally deactivated in the multi-user mode when bringing the system down to single-user operation. It is mandatory that these commands be executed from the main terminal or the terminal attached to PORT1 because during the process of bringing the system down to single user mode, all user partitions except the main terminal attached to PORT1 will be deactivated.

Using our five user system example, we'll illustrate stopping the spooler and bringing the system down to single -user mode. From the main terminal type:

```
>SPOOLER STOP<RETURN>
>SPOOLER QUIT<RETURN>
>STOP 6<RETURN>
>SET MEM 6<RETURN>
>STOP 5<RETURN>
>SET MEM 5<RETURN>
>STOP 4<RETURN>
>SET MEM 4<RETURN>
>STOP 3<RETURN>
>SET MEM 3<RETURN>
>STOP 2<RETURN>
>SET MEM 2<RETURN>
>UNLOAD SPOOLER<RETURN>
```

The SPOOLER STOP command is used in case the spooler is printing a report at this time. This command tells the spooler to quit printing after the current report has finished printing.

The SPOOLER QUIT command is used to deactivate the spooler. Each user partition is then stopped and the memory partitions deactivated. The spooler module is then unloaded.

Note: If all users on the system are not logged off when the memory partitions are stopped, the error message "Partition is still active" is displayed. That memory partition remains active until the user has logged off and the STOP command re-executed.

SUMMARY

The purpose of the THEOS spooler is to allow sharing of a common printer in a multi-user environment thereby eliminating a time *bottleneck* which may occur as each user accesses the printer. Examples illustrating the use of the spooler in single and multi-user mode have been provided. As we will see in chapters six and seven EXECs can be used in both single and multiuser modes to automatically activate the spooler.

Session 21: The SYSGEN Command

The SYSGEN command allows you to save the status of certain system parameters for future system start-ups. SYSGEN is used to save such system attachments as:

- disk drives
- tape drives
- printers
- modems

Other parameters such as time and date may be sysgened as well.

The format of the SYSGEN command is:

SYSGEN

In a new THEOS system, the only peripherals that have been sysgened are the main terminal (attached to PORT1) and the disk drive containing the system disk. Before issuing the SYSGEN command, you may want to examine the current value of the parameter using the ATTACH command.

HOW TO SYSGEN PERIPHERALS

In the session on the ATTACH command we examined how to tell THEOS that peripherals such as disk drives, printers and modems are to be attached. When the ATTACH command is executed these attachments are only in effect until the system is re-booted. In order to make the system easier to operate, the SYSGEN command permits all frequently used peripheral attachments to be saved. This means that when the system is re-booted all sysgened attachments remain in effect. Let's look at an example.

Step 1: Reboot your system.

Step 2: To see what is currently attached to the system type:

>A<RETURN>

THEOS will respond with a list of current attachments. Since a system boot was just executed and you have not attached any peripherals, this attachment table shows what peripherals have been sys-

generated by the hardware vendor. If you are using a hard disk system with multiple platters, it is possible that the platters other than the system platters have not been attached or sysgened. In order to attach and sysgen them type:

```
>A A DISK1<RETURN>
>A B DISK2<RETURN>
>SYSGEN<RETURN>
```

The first two command lines attached each platter. The SYSGEN command saved the attachments; when the system is rebooted platters A and B are still attached along with the system disk.

Let's examine the procedure for sysgening a printer. Type:

```
>A PRINTER1 PORTn (options<RETURN>
>SYSGEN<RETURN>
>
```

PORTn is the port number where the printer is to be attached. Options include such parameters as width, lines per page, baud rate, etc. These are included if you wish to specify values other than the defaults.

Let's examine one more SYSGEN example. Assume that it is the start of a new year and the system has been booted. The actual date is January 1, 1985. Your system has a built-in real time clock and THEOS shows that the current year is 1984 not 1985. To correct this problem type:

```
>SET DATE 1/1/85<RETURN>
>SYSGEN<RETURN>
>
```

If you type:

```
>SH DATE<RETURN>
January 1, 1985
>
```

The SH DATE command verifies that the new date has been correctly entered and sysgened.

Note: Using the SYSGEN command on floppy disk drives is not recommended. As long as the floppy disk is attached, there must be a disk in that drive at all times; otherwise a "disk not ready" message appears on the screen. Attach and detach floppy disk drives as they are needed.

Let's examine how SYSGEN is used with a modem. Type:

```
>A COMM PORTn<RETURN>  
>SYSGEN<RETURN>  
>
```

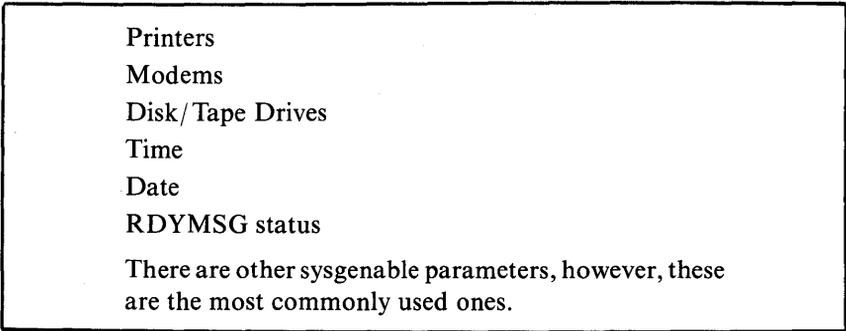


Figure 4.36. Sysgenable parameter/peripheral devices

WHAT IF YOU CHANGE YOUR MIND?

If the SYSGEN command has been executed and you want to restore the system to its original state, you can do so by executing either the ATTACH or SET command and then typing SYSGEN. The new values will be saved.

SUMMARY

The SYSGEN command provides a method of making system attachments and setting certain parameters *permanently* so they don't have to be reset when the system is rebooted.

Session 22: The SEND/RECEIVE and TERMINAL Commands

These commands provide THEOS users with the ability to transfer files from one computer to another either directly from port to port or over telephone lines via modem. The SEND/RECEIVE commands are logical complements of each other and are designed to send and receive any type of THEOS file, including programs from one THEOS system to another.

SEND/RECEIVE can be used when:

- transferring files port to port from one THEOS computer to another
- transferring files from one THEOS computer to another via modem over phone lines.

The TERMINAL command is a program used in conjunction with a modem to establish a connection with any computer. It has built-in provisions for sending or receiving sequential format files.

Before continuing with this tutorial, we recommend that you consult your hardware and modem documentation for proper cabling, switch settings, and connections.

THE RECEIVE COMMAND

The RECEIVE command is used to receive files from another THEOS computer. The format of the RECEIVE command is:

RECEIVE FILENAME FILETYPE DRIVNAME

Most often the files being received have the same FILENAME and FILETYPE as the files being sent. If this is the case it is not necessary to specify the FILENAME and FILETYPE. However the DRIVNAME where the file(s) are to be located must be indicated.

USING THE RECEIVE COMMAND

In order to use the RECEIVE command, a communications device (COMM) must be attached to an active port, the correct baud rate must be set and you must be logged onto the account which is to

own the files being received. If files are being sent directly from one computer to another, a modem is not necessary. In this case you can type:

```
>A COMM PORTn<RETURN>  
>
```

This command line instructs THEOS to establish PORTn as a communications link between two computers. Since a modem is not used, in this case, the default baud rate is 9600. The data is transferred quickly.

If a modem is being used, the command line as indicated above is entered. The baud rate might be specified as an option. However, in some computers baud rate is hardware controlled and may not be set from THEOS.

If it is possible to set the baud rate via THEOS, the following command line can be entered:

```
>A COMM PORTn (B1200<RETURN>  
>
```

This command line attaches a modem to the port of your choice with a baud rate of 1200.

As of this writing, 1200 baud is commonly used for sending data over standard telephone lines.

If communication between computers is by modem, it is first necessary to establish a connection. The examples below assume that the receiving computer has an auto-answer modem and the operator knows by the modem indicator lights that a connection has been successfully established.

Suppose you were to type:

```
>RECEIVE S<RETURN>  
Waiting for Sending Station  
Receiving FILENAME.FILETYPE:S  
Segment 1,2,3,etc.  
End of File Received  
End of Transmission
```

In this example, only the DRIVNAME was specified. This means that all files being received will have the same FILENAME and FILE-

TYPE as those being sent. When the two computers begin communicating, the message "Receiving FILENAME.FILETYPE:S" is printed and the segment number being transmitted is displayed. The number of segments will vary depending upon the size of the file being received. The larger the file the greater the number of segments to transmit.

Sometimes it may be useful to receive files onto a floppy disk for later transfer to a hard disk. To receive files onto a floppy disk type:

```
>A G FLOPPY1<RETURN>
>RECEIVE TESTFILE DATA G<RETURN>
>Waiting for Sending Station
Receiving TESTFILE.DATA:G
Segment 1,2,3,etc.
End of File Received
End of Transmission
>
```

In this example the floppy disk drive designated G was attached. The file being received was to be named TESTFILE.DATA. When the file has been successfully received, the "end of file" message is displayed along with the "end of transmission" message. If you are now finished with the G drive, be certain you detach it by typing:

```
>A G<RETURN>
```

SOME SPECIAL NOTES REGARDING THE RECEIVE COMMAND

- Files being received are automatically stored on the disk.
- Files being received are owned by the currently logged account, not the owner of the file as it is sent. Be certain you are logged on as the account who is to own the file. If the file is a program to be shared by all users, receive the file logged on as SYSTEM.
- If a file currently exists with the same name as the one being sent, the old file is automatically overwritten with the new one
- The file specifications, such as date and time, are identical to the file being sent.
- Sequential files are the easiest to send and therefore take the least amount of time. Program files and data files take considerably longer.

- If files are being sent over telephone lines errors in communication may occur. The SEND/RECEIVE programs have built-in error checking to prevent data transfer errors. If the telephone connection is particularly poor, command execution may be terminated and an error message such as "receiver disconnected" or "receiver timed out" is displayed.
- If receiving files by modem, be certain that the modem is "hung up" after all files have been received. The RECEIVE command has no provision for automatically hanging up the phone.

THE TERMINAL COMMAND

This command provides THEOS users with the ability to establish a communications link with another computer system via modem. Communication can be established with another THEOS computer, or a non-THEOS computer for SENDING or RECEIVING files. As figure 4.37 illustrates when the TERMINAL command is executed, the user is presented with a menu of choices. The examples provided will illustrate the use of the most frequently selected choices.

The format of the TERMINAL command is:

TERMINAL (options)

There are options that can be used with the TERMINAL command but since they can be changed from the terminal emulator menu they will not be discussed here. For more information refer to your THEOS documentation.

USING THE TERMINAL COMMAND: THE THEOS TERMINAL EMULATOR

The THEOS terminal emulator is the active part of the TERMINAL command. A menu is provided to make communicating with another computer by modem easy. To access this menu first be certain that COMM is attached to an active port and that the proper baud rate is set.

Next type:

>TERMINAL<RETURN>

The terminal emulator menu is then displayed. If you type TERMINAL before a COMM device is attached, the error message "COMM not attached" is displayed.

```
THEOS Terminal Emulator - version n.n
Modes: Duplex=F Autolf=N Ctl=N Null=N Master=N Class=0
Receive file = (ON= ,OFF= )
Send file = (ON= ,OFF= )

1 - Assign receive file
2 - Assign send file
3 - Define translate table
4 - Change modes
5 - Begin emulation
6 - Quit (return to THEOS)

Enter function: _
```

Figure 4.37. Screen display of the THEOS terminal emulator

Full duplex communication and no automatic line feed are two of the default values on the terminal emulator menu. These generally need not be changed. If necessary the mode values at the top of the screen can be changed by selecting option 4. When option four is selected the cursor is positioned on the mode line value to be changed. If no changes are to be made, the space bar can be pressed to move to the next value. Continue pressing the space bar until you reach the desired value or you return to the "Enter function:" prompt at the bottom of the screen.

The only values acceptable for the various modes are:

- Duplex=F or H
- AutoIf=Y or N
- Ctl=Y or N
- Master=Y or N
- Class=1 through 6
- space bar which moves from one value to another

Note: When using the terminal emulator to communicate with a non-THEOS system, only sequential format files can be sent or received.

Let's examine how to use the terminal emulator to dial a remote computer:

Step 1. Be certain the modem is attached to an active port and the correct baud rate is set.

Step 2. Type: **TERMINAL**<RETURN>

The terminal emulator menu is then displayed. Let's assume you have a Hayes Smartmodem or a modem that is Hayes compatible. In either case, the commands for dialing a telephone should be similar.

Step 3. From the menu type 5. This selection allows you to begin the dialing sequence. In the upper left-hand corner of your screen the following message is printed.

ESC C, to return to menu

On some terminals, such as the Televideo 950 and other 950 compatible models, pressing ESC ESC C is required to return to the menu.

Step 4. Turn on the modem. Type the dialing sequence required by the modem. To dial the THEOS User's Group type:

atdt18059655415<RETURN>

If at this time the characters just typed do not appear on the screen, it probably means that the cable connecting the modem to the computer is not wired correctly. You might try reversing pins 2 and 3 on one end of the cable.

Step 5. When the CONNECT message is displayed (under the dialing sequence), a successful connection with the remote computer has been established.

Step 6. Press <RETURN>. This tells the remote computer you are ready to communicate.

Step 7. The sign-on message from the THEOS User's Group should now be displayed on your screen.

Step 8. When you want to hang up the phone, you can follow the on-screen directions provided by the user's group communication software. This will log you off of their system, but will not hang up the phone. When you have terminated your conversation, you can either go to the modem command mode and issue the command required to hang up or simply turn off the modem.

Step 9. Press ESC C to return to the emulator menu.

Step 10. If you are finished with the emulator, press 6 to return to the THEOS prompt.

Note: If you are not using a Smartmodem or another autodial modem, when you see the ESC C message on the screen you must pick up the phone and begin dialing manually according to the directions supplied with your modem.

Step 11. The modem is still attached. If you want to detach it at this time type:

>A COMM<RETURN>

To send sequential files follow steps 3 through 5 described earlier. Press ESC C to return to the menu. Enter 2 from the emulator menu and type the FILENAME.FILETYPE and DRIVNAME to be sent. Next, enter 5 to begin emulation and the file will automatically be sent.

To receive sequential files type 1 from the emulator menu and enter the FILENAME.FILETYPE and DRIVNAME of the file to be received. Select option 5 from the emulator menu so you can see when a connection has been established. After establishing a successful connection, the file specified will be received and automatically stored on disk.

To hang up the phone, use the modem command mode or hang up the phone.

Note: Depending upon the quality of the telephone connection and how communication parameters are set, errors in transmission or reception of sequential files may occur, particularly at 1200 baud. Be certain to add the parity option in the ATTACH command if the remote computer requires it. The choices would be PE for Parity Even, PO for Parity Odd, or if no parity is specified, no parity will be used. If you typed,

> A COMM PORTn(PE<RETURN>

a COMM device with even parity would be attached.

THE SEND COMMAND

The SEND command is used to transmit files from one THEOS system to another. The format of the SEND command is:

SEND FILENAME FILETYPE DRIVNAME (options

FILENAME and FILETYPE is the file description to be sent. DRIVNAME is the drive where the file is located. The use of wildcards for the FILENAME and FILETYPE are permitted.

Option	Function
NOEOT	This option is useful when sending a group of files. It prevents the receiver from being disconnected after each file is sent.
QUERY	The sender is asked on a file-by-file basis if it is ok to send a file.
NOQUERY	The sender is not asked on a file-by-file basis if it is ok to send a file.
TALK	Allows the sender to send a message to the receiving computer after all files have been sent.

USING THE SEND COMMAND

In order to use the SEND command a COMM device must be attached to an active port and you must be logged onto the account which owns the files you wish to send. If you try to send files not owned by the currently logged account a "file not found" message is displayed. Since the SEND command is the logical complement of the RECEIVE command, the same instructions can be used when attaching a modem. The difference is that the **sender** is usually responsible for making the connection to the receiver. If a modem is not being used and files are being sent port to port, sending files is simple.

To send files port to port the two THEOS computers must be connected via a modem cable and the following command must be executed on both computers:

>A COMM PORTn<RETURN>

The receiver types:

>RECEIVE S (or some other drive)
Waiting for Sending Station

The sender then types:

>SEND TEST FILE S<RETURN>

If the receiving station is not ready the message:

Waiting for Receiving Station ****

is displayed.

If the receiving station is ready THEOS responds:

Segment 1,2,3,4, etc.
End of File

Had the sender typed:

>SEND TEST FILE S (TALK<RETURN>

the file would have been sent as indicated except that after the "end of file" message, THEOS would respond:

Enter TALK messages, terminated by an empty line.

The sender might type:

**That's it for the files I'm sending.
Turn off the modem and pick up the phone.
I want to talk to you.**
>

SOME SPECIAL NOTES ABOUT TELECOMMUNICATIONS

- When connecting a modem, the type of cable required is serial with pins one through eight and twenty connected. Some modem/computer interfaces require that pins two and three be reversed on one end of the cable.
- The same cable used with the modem can generally be used to connect computers when sending port to port.
- It is not possible to send or receive archived files.

How to Access and Operate a THEOS Computer by Modem

Many systems houses or system integrators sell their systems complete with modems so that should a problem arise with software or hardware, the computer can be operated from their office. The computer owner can also buy an additional terminal and modem and access the computer from home. To access and operate a THEOS computer by modem, follow these steps:

- Step 1. Set up the computer in multi-user mode.
- Step 2. Connect a modem to one of the ports where a terminal is usually connected. Connect the modem to the desired telephone line and leave the power on.
- Step 3. Set the baud rate for the modem port to 1200 either via a hardware switch (inside the computer) or by using the ATTACH command.
- Step 4. Using the terminal and modem you have at home, turn on the modem and terminal and initiate the dialing sequence as described in the section on the terminal emulator. This time, you will be dialing a number at your office where the computer modem is connected.
- Step 5. If a connection is successfully established, the "Logon please:" message is displayed on the screen after you press <RETURN>.
- Step 6. Logon as you would normally and proceed to operate the computer as if you were in the office.
- Step 7. When finished, logoff and turn off the terminal and modem.

Note: As long as the modem is attached to a port other than port one, it is not possible to bring the system down to single-user mode. Also, it is obvious that the ARCHIVE and RESTORE commands cannot be executed because an operator must be available to insert tapes or floppy disks as required.

The system should operate normally except that since the baud rate is 1200 as opposed to 9600, the time it takes to *paint* the screen display is relatively long. Therefore, word processing over the phone is generally not practical. Most accounting functions should work well. Also be aware that an occasional *garbage* character might appear on the screen. These characters are due to noise in the phone lines.

The computer views the modem attached to one of its ports as a terminal not a COMM device. If the computer is to be used in this manner on a regular basis, two modems are recommended, one as a terminal, the other as a file sending/receiving device.

SUMMARY

The SEND, RECEIVE, and TERMINAL commands provide THEOS users with the ability to:

- Send or receive any type of THEOS file port to port or via modem from another THEOS computer.
- Send or receive sequential format files from any computer via modem.
- Establish a communication link with any other computer via modem.

It is not necessary to buy additional communication software in most cases. Autodial/autoanswer modems are recommended for convenience.

The quality of telephone service varies from one part of the country to another as does long distance service. AT&T is recommended for 1200 baud communication to minimize errors.

If these commands are to be used regularly, practice with them until you feel comfortable. For the terminal emulator, try to find some 1200 baud 800 numbers you can dial or some local numbers of computer bulletin boards that provide 1200 baud service. If you are going to make a long distance call, contact the THEOS User's Group. You will find their bulletin board most friendly and helpful.

Session 23: How to Install Application Software

Unless a THEOS system was purchased from a systems house, it is not likely that application software is installed and ready to use. The purpose of this session is to provide some examples of how software might be installed on a THEOS system whose disks have been correctly formatted so they are ready to store files.

Application software may be provided on either floppy disks or tapes. Depending upon the number of files involved, the software may be in archive format or individual file format. Word processing and spreadsheet programs are the least complex in terms of the number of programs required to operate. Accounting software is the most complex. This session is not intended to replace documentation provided by software vendors but to demonstrate via examples the procedures for installing software.

Note: Most THEOS software is **serialized**. This means that when software is purchased it is matched to the serial number of your

operating system. It will only run on your system. If after installing and running the software an "Incorrect serial number" message is displayed, contact your software vendor immediately.

Let's suppose you are installing an accounting system whose files are supplied in archive format contained on one eight inch floppy disk, and owned by an account called AIM.

- Step 1. While logged onto the SYSTEM account, use the ACCOUNT command to set up user accounts and passwords for personnel who will be using the software. If the programs are to be used by everyone, the programs can be installed under the SYSTEM account. Data files can be created later under another account, perhaps an account called USER1 and its synonyms.
- Step 2. Assuming you are going to install the programs on the SYSTEM account, insert the diskette and type:

>A G FLOPPY1<RETURN>

The files can be restored to any currently attached drive; in this case we will use drive S. Type

>RESTORE G S (NEWACC AIM<RETURN>

THEOS will prompt with the RESTORE questions. When you are asked if it is "Ok to start restore (Y/N)?" type Y<RETURN>.

Each file description, as it is restored, will be displayed on the screen. The first command line issues an ATTACH command for floppy disk drive G. Next the RESTORE command is used specifying that files located on drive G will be restored to drive S. Using the NEWACC option and specifying AIM means that ownership of the files located on drive G that belong to the AIM account will be transferred to the currently logged account which is SYSTEM.

Note: If by accident you omit AIM and begin restoring files, all the files will be restored to some unknown account and cannot be accessed. If this happens, the easiest way to erase the files is to reformat the disk.

Step 3. Detach the floppy by typing:

>A G<RETURN>

Step 4. To display the main menu type:

>RUN BMENU<RETURN>

The main program selection menu will be displayed.

Since all of the programs have been installed, all that remains is to create the data files. If the data files are to be created under another account such as USER1, log onto that account and follow the directions supplied with the software for creating files. Most application software provides an EXEC program which will automatically create the files according to predetermined specifications. Then, once the files are created, a menu choice can be selected which will allow you to change the files to the size you require.

Accounting systems use indexed, direct, keyed, and sequential files to store their data. All but the sequential files must be created before the software can be used. The sequential files required are created automatically.

Installing a THEOS word processing package such as Spellbinder is a little less time consuming than the preceding example because only a few files are required to begin using the package and the files are supplied in their entirety. This means that the COPYFILE command can be used to transfer files from the floppy disk to the hard disk.

Step 1. Log onto the SYSTEM account. Type,

>A G FLOPPY1<RETURN>

to attach the floppy disk.

Step 2. Insert the floppy disk containing the programs and type:

>FILELIST G (OWN<RETURN>

This allows you to determine who owns the files. If the files are owned by the SYSTEM account, proceed to step three.

Step 3. Type:

>COPYFILE *.* G == S (NOQ<RETURN>

All files on the G drive will be copied to the S drive. As the Spellbinder files are copied their file descriptions are displayed.

Step 4. When all the files have been successfully copied type,

>A G<RETURN>

in order to detach the floppy.

Step 5. In order to configure the Spellbinder program for the type of terminal and printer being used, a special EXEC program called CONFIG is supplied. You can complete the configuration and begin using the program by typing,

>CONFIG<RETURN>

and answering the questions on the screen that pertain to the type of equipment being used. The Spellbinder documentation provides an explanation of how to answer the questions.

Step 6. After the CONFIG EXEC has been run, the program can be used by typing

>SPELL<RETURN>

You should still be logged onto the SYSTEM account. Since the Spellbinder programs are owned by SYSTEM, any user can access them. Should you log on to a different account, the program can still be run and in addition Spellbinder sequential files will be created on that account. Although anyone can use the programs, each user can create his own private files by virtue of the THEOS accounting feature.

SUMMARY

THEOS application software is supplied on floppy disks or tapes. In order to be used, it must be transferred to hard disk. Depending upon the number of programs the software may be supplied in either ARCHIVE format or as individual files. The documentation supplied with the software should detail all specific installation requirements.

5

THEOS System Files and Diagnostic Commands

Introduction

This chapter's purpose is to introduce the THEOS user to the types of files used to operate the system other than command files. These files include the following:

SYSTEM.DEVnn
SYSTEM.DEV NAMES
SYSTEM CLASSnn
SYSTEM ERRMSG
SYSTEM.EXECnn
SYSTEM.SPOOLQUE

We'll also discuss the REPAIR and KILL commands which are used to correct disk problems.

The SYSTEM.DEVnn Files

These files are supplied with every copy of THEOS. They are called **device driver** programs and are used to provide a series of instructions that tell THEOS how to access the peripherals such as disk drives, tape drives, and modems. The average THEOS user rarely manipulates these files other than copying them from one system disk to another.

The number nn indicates the number assigned to the file as part of its FILETYPE. A typical listing of SYSTEM.DEVnn files is illustrated in figure 5.1. When a peripheral device such as a disk drive or tape drive is ATTACHED, the instructions contained in these files are automatically stored in memory for use by THEOS.

SYSTEM.DEVNames CONTENTS

This file contains a list of names used by THEOS for attached devices. Figure 5.2 shows a sample listing of the contents of a SYSTEM.DEVNames file. This listing will vary depending upon the type of hardware being used. This list of names is used by THEOS every time an ATTACH command is executed. If an attempt is made to attach a device not on this list, the error message "Undefined physical device" will be displayed.

The list as supplied from the hardware vendor should be sufficient for most THEOS installations. Since the file is sequential, new device names can be added if necessary as long as certain guidelines are followed. It is beyond the scope of this book to discuss modification of this file. More on this subject can be found in the THEOS documentation.

The SYSTEM.CLASSnn Files

Each terminal has its own unique *codes* that are generated each time a key is pressed on the keyboard, as well as certain requirements for how the screen is *painted* with characters. These files provide

FILELIST SYSTEM.DEV##:*

01/16/85 17:25 Page 1

Frame--- Ftype-- Dr --Date-- Time- -Recs Blks Format- Label---

SYSTEM	DEV10	S	03/22/83	09:26	3	1	R	256	HARDISK1
SYSTEM	DEV11	S	03/22/83	09:26	3	1	R	256	HARDISK1
SYSTEM	DEV12	S	03/22/83	09:27	3	1	R	256	HARDISK1
SYSTEM	DEV13	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV14	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV15	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV16	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV17	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV18	S	03/22/83	09:28	3	1	R	256	HARDISK1
SYSTEM	DEV25	S	09/30/82	11:23	5	2	RW	256	HARDISK1
SYSTEM	DEV26	S	06/08/82	11:54	3	1	RW	256	HARDISK1
SYSTEM	DEV28	S	12/21/82	14:34	2	1	RW	256	HARDISK1

12 files. 13 blocks.

Figure 5.1. Sample listing of SYSTEM.DEVnn files

```
SYSTEM.DEVNAMES:HARDISK1  
  
FLOPPY1 1 D 1 2  
FLOPPY2 2 D 1 2  
DISK1 3 D 3 4 5 6 7 8  
DISK2 4 D 3 4 5 6 7 8  
DISK3 5 D 3 4 5 6 7 8  
DISK4 6 D 3 4 5 6 7 8  
DISK5 7 D 3 4 5 6 7 8  
DISK6 8 D 3 4 5 6 7 8  
CART 25 T  
REEL 26 T  
PORT1 18 CPSIO  
PORT2 17 CPSIO  
PORT3 16 CPSIO  
PORT4 15 CPSIO  
PORT5 14 CPSIO  
PORT6 13 CPSIO  
PORT7 12 CPSIO  
PORT8 11 CPSIO  
PORT9 10 CPSIO  
PORT10 9 CPSIO  
BISYNC 27 CSIO  
CENT 28 PO
```

Figure 5.2. Sample listing of contents of SYSTEM.DEVNAMES

THEOS with a set of instructions on how to interpret the keys pressed on your terminal keyboard and how to display characters. These files are also referred to as **class code drivers**. The number nn indicates the **code number**. Every copy of THEOS is supplied with a number of class code drivers so that the THEOS user is not restricted to a particular brand or model of terminal. The user can choose the terminal that best suits his requirements. These drivers allow for the usage of most popular terminals. Figure 5.3 contains a partial listing of some of the class code drivers supplied and the terminals the codes support.

Some application software packages may include class code drivers for terminals recommended for use with their software.

Code Number	Terminal	Model Number
1	Beehive	B100
2	ADDS	580, Envoy
3	Lear Siegler	ADM 3A
6	Hazeltine	1500
7	Televideo	912/920
7	Kimtron	ABM 85
10	Heath/Zenith	H19
15	ADDS	Regent 20
22	IBM	3101 (ESC ESC for system control)
25	Volker-Craig	VC404
31	Televideo	950 ESC ESC for system control
31	Kimtron	ABM 85H, KT7(ESC ESC for system control)
35	Lear Siegler	ADM 5
36	Televideo	910
37	Falco Data Products	TS-1
102	Televideo	Personal Terminal

Figure 5.3. Partial listing of THEOS-supported terminals

Terminals from different manufacturers may be able to use the same class code. Note in figure 5.3 that both Kimtron and Televideo terminals can use a class 7 class code driver.

To obtain a listing of the class code drivers on your system for the SYSTEM account, type:

```
>F SYSTEM.CLASS*<RETURN>
```

A listing of all class code drivers on your system will be displayed. The list should be similar to that in figure 5.4.

The SYSTEM.ERRMSG File and the ERRMSG Command

This file contains the **error**, **warning**, and **information messages** that are displayed in response an incorrect command line entry or some other error relating to improper system operation. Since this file is not a sequential file, it cannot be modified using the EDIT command. A program has been provided to add or modify error messages stored in this file.

The ERRMSG command allows modification of a message contained in this file. In order to use this command, you must be logged onto the SYSTEM account. Otherwise the message "Can only be run from SYSTEM account" is displayed.

It is possible to list the contents of the SYSTEM.ERRMSG file by typing:

```
>LIST SYSTEM.ERRMSG<RETURN>
```

THEOS responds by listing all of the messages contained in the file.

As figure 5.5 illustrates, the messages are numbered in sequence beginning with the number one. The file can contain a maximum of 255 messages. The message contains the message text as well as variable information determined at the time the message is displayed. Variable information is indicated by a number within braces such as { 1 }.

USING THE ERRMSG COMMAND

The format of the ERRMSG command is:

```
ERRMSG (number)<RETURN>
```

FILELIST SYSTEM.CLASS*:*

01/16/85 19:10 Page 1

Files owned by account: SYSTEM (0)

Fname---	Ftype--	Dr	--Date--	Time-	-Recs	Blks	Format-	Label---
----------	---------	----	----------	-------	-------	------	---------	----------

SYSTEM	CLASS227	S	02/09/83	11:26	3	1	RWD 256	HARDISK1
--------	----------	---	----------	-------	---	---	---------	----------

SYSTEM	CLASS31	S	02/09/83	11:26	3	1	RWD 256	HARDISK1
--------	---------	---	----------	-------	---	---	---------	----------

SYSTEM	CLASS7	S	07/30/81	14:33	3	1	RWD 256	HARDISK1
--------	--------	---	----------	-------	---	---	---------	----------

3 files, 3 blocks.

Figure 5.4. Sample listing of SYSTEM.CLASSnn files

```
1: [1] Syntax error.
2: [2] File name missing.
3: [3] File type missing.
4: [4] Drive code missing.
5: [5] Disk not attached.
6: [6] Disk not mounted.
7: [7] Invalid option.
8: [8] File not found.
9: [9] Disk full.
10: [10] Directory full.
11: [11] Invalid file name.
12: [12] Invalid drive code.
13: [13] File "{1}.{2}" not found.
14: [14] "{1}" is not recognizable.
15: [15] "{1}" is an invalid numeric.
16: [16] Expecting "{1}", found "{2}".
17: [17] Expecting "{1}", found end-of-line.
18: [18] Protected file.
19: [19] Insufficient privilege.
20: [20] Incorrect serial number.
21: [21] Expecting end-of-line, found "{1}".
22: [22] Required parameter missing.
23: [23] Must be "ON" or "OFF".
24: [24] Incorrect format, should be "{1}".
25: [25] Cannot detach the CONSOLE.
26: [26] Invalid record in "{1}.{2}".
27: [27] Cannot attach to an output only device.
28: [28] Cannot attach to an input only device.
29: [29] Can only attach to a random storage device.
30: [30] Cannot attach to a random storage device.
31: [31] Cannot detach the System disk.
32: [32] Undefined logical device.
33: [33] Undefined physical device.
34: [34] Must be sequential organization.
35: [35] File already exists.
36: [36] Organization code is missing.
37: [37] Missing option: "{1}".
38: [38] {1} is not attached.
39: [39] Device is not attached.
40: [40] File channel is not assigned.
41: [41] No files found.
42: [42] {1} file(s), {2} block(s).
43: [43] Must have at least two disks attached.
44: [44] Source and destination drives cannot be the same.
45: [45] No files erased.
46: [46] {1} file(s) erased, {2} block(s) recovered.
47: [47] Source and destination do not have the same capacity.
48: [48] Receiver disconnected.
49: [49] Receiver timed out.
50: [50] Must be private file.
51: [51] Load module contains unresolved external references.
52: [52] Cannot mix ABS with REL or COM modules.
53: [53] Cannot have multiple ABS modules.
54: [54] Symbol table overflow.
```

Figure 5.5. Partial listing of the SYSTEM.ERRMSG file

Number indicates the number of the message to be displayed. For example, suppose you entered the following:

```
>ERRMSG 42 <RETURN>
[42]P1 file(s), P2 block(s)
>
```

Error message 42 is used to report the number of files and disk space occupied by those files when a FILELIST command is executed. [42] is the number assigned to the error message. P1 is the variable indicating the number of files. P2 is the variable indicating the number of disk blocks.

In order to change the contents of the ERRMSG file, you must first be certain to change the write protect status of the file. As supplied, the SYSTEM.ERRMSG file is write protected. When a message is changed, the file must be re-written. In order to rewrite the file, its write protection must first be removed. To change the protection status of the file, use the RENAME command and type:

```
>RENAME SYSTEM.ERRMSG S (NOW[RETURN]
```

To enter the interactive maintenance mode type:

```
>ERRMSG <RETURN>
Message number: 99 <RETURN>
Old: THEOS Multi-user Operating System
New: THEOS Multi-user Operating System 5.6[RETURN]
Message number: <RETURN>
```

The message number to change is 99. THEOS prints the old message on the screen. Enter the new message after New: and press <RETURN>. If no more messages are to be changed, press <RETURN> to save the new message as part of the file. If more changes are to be made, enter a new message number to be changed.

Note: Making unnecessary changes in this file is not recommended particularly if the message contains a variable. Changes may limit the usefulness of this file in providing THEOS users with informative error messages. Study the THEOS documentation before making any changes in messages 99 through 102, which are sign-on mes-

sages displayed every time a successful logon is accomplished. Generally these are the only messages that warrant change.

The **SYSTEM.EXECnn** Files

These files are necessary to permit execution of EXEC programs. The purpose of these files is to save the status and variables of one or more EXEC programs while another EXEC is executing. In multi-user THEOS the number of SYSTEM.EXECn files is equal to the number of user partitions in the system. A system with five user partitions will include SYSTEM.EXEC1, SYSTEM.EXEC2, SYSTEM.EXEC3, SYSTEM.EXEC4, and SYSTEM.EXEC5.

These files are direct access files with a record length of 512. The number of records contained in the file determines how many EXEC programs may be used. As distributed the SYSTEM.EXECn files contain two records. This means that only two EXEC programs may be used.

CREATE can be used to create additional SYSTEM.EXECn files or to permit a greater number of EXECs to be used. In chapter 6 examples of creating and altering the size of SYSTEM.EXECn files will be provided.

Note: If a user partition does not have a SYSTEM.EXEC file, EXEC programs cannot be run. Any attempt to run an EXEC will return the user to the THEOS prompt.

The **SYSTEM.SPOOLQUE** File

SYSTEM.SPOOLQUE is a data file used by the print spooler. The spooler will not function if this file is not present on an attached disk. As indicated in the tutorial on the SPOOLER command, this file may reside on any attached disk. Preferably it should be put on a disk by itself because spooled files will be written to the disk containing this file. This will maximize the available disk space for spooled files and prevent the spooler from accidentally writing over existing files.

To verify the existence of this file from the SYSTEM account type:

```
>F SYSTEM.SPOOLQUE <RETURN>
```

If the file exists, THEOS will display information about the file in FILELIST command screen format.

To create this file, type:

```
>CREATE SYSTEM SPOOLQUE DRIVNAME (DIRECT RECLN 16 FILE 64<RETURN>
>
```

This command line will create the SYSTEM.SPOOLQUE file as a direct file with a record length of 16 and a file size of 64. In place of DRIVNAME specify the drive where you wish the file to be located. The FILE size specified determines the maximum number of spooled files that can be queued for printing. A file created according to the preceding specifications is more than adequate for most THEOS users.

System Diagnostics: Misallocations: The REPAIR and KILL Commands

The REPAIR and KILL commands are used to correct disk space misallocations. During the course of normal system operation, problems relating to storing files on disk should not occur. Misallocations are errors in disk space allocation which arise due to unforeseen problems such as power outages, program bugs, static electricity, and improper closure of files.

There are two kinds of misallocations:

- + misallocation occurs when 2 files attempt to occupy the same disk space.
- - misallocation occurs when there is an inconsistency between information in the disk directory and the file itself.

The more dangerous of the two conditions is the positive misallocation. This means that there are at least two files that probably contain inaccurate data. When this occurs it is sometimes possible to save one of the files.

Reporting Misallocations

Misallocations are reported using the SHOW DISK command. For example, if you type:

>SHOW DISK S<RETURN>

THEOS might respond:

```
S(1) Label: "SYSTEM ",
Backup onto "SYSTEM$B" on 01/15/85 23:30,
Capacity: 497K bytes, (77-1-26),
Available: 56K bytes (11%),
Largest area: 56K bytes,
95 files in use (out of 176)
+1K bytes mis-allocated
```

Under normal circumstances, there would be no misallocations reported. In this case, however, it is noted that 1K of disk space is misallocated. Were a negative misallocation to exist, the message “-1K bytes misallocated” would have been displayed.

The REPAIR Command

The REPAIR command is used to *view* disk allocation statistics and *optionally to correct* misallocations. When the REPAIR command is executed the disk specified is searched sequentially displaying:

```
File n FILENAME.FILETYPE \id
```

File n is the number in the directory beginning with 1 through the total number of files on the disk. The owning account id number is displayed after the backslash (\). If no number is displayed after the (\), the file is owned by the SYSTEM account.

During command execution the allocation of each file is checked against the previous file's allocation. File collisions are noted (positive misallocations). Any misallocations are also noted. The REPAIR command can be executed from any account but the system *must be* in single-user mode.

The REPAIR command has only one option — the FIX option. The FIX option is used to correct misallocations. The format of the REPAIR command is:

```
REPAIR DRIVNAME (option
```

DRIVNAME is the drive (S,A,B,C, etc.) to be repaired.

Note: Unless the FIX option is specified, any reported misallocations *will not* be corrected.

For example if you were to type,

```
>REPAIR A <RETURN>
File 1 DATA.FILE
File 2 TEST.DATE1
File 3 NEW.FILE2
File 4 OLD.FILE3
.
.
.
File 45 INDSIZES.SYSREF 4
Sector 252 is used but not allocated
Sector 256 is used but not allocated
Sector 428 is allocated but not used
Sector 440 is allocated but not used
```

The lines under File 45 are error messages describing the error conditions concerning this file. Since the files scroll by rapidly it may become necessary to pause execution of the REPAIR command to note the files that are causing the misallocation. To pause execution of the command, press ESC S, or on some terminals ESC ESC S. After the file has been noted, enter ESC S again and continue execution until all files have been examined.

The KILL Command

The KILL command is used to remove a misallocated file from the disk directory *without* de-allocating its disk space.

The format of the KILL command is:

```
KILL FILENAME FILETYPE DRIVNAME
```

The entire file description of the file must be specified. Using the ERASE command to remove the misallocated file will result in further misallocations. After determining which files are causing the misallocation via the REPAIR command, REPAIR can be used with the FIX option to correct the misallocation.

How To Correct Negative Misallocations

Negative misallocations indicate that the information in the disk directory and the file are not the same. To correct this type of misallocation follow these steps:

Step 1. Be certain that the system is in single-user mode. Execute a SHOW DISK command.

Note the drive with the misallocated file(s) and type:

>REPAIR DRIVNAME (FIX <RETURN>

Step 2. If error messages are noted underneath any file, this may mean the problem is more serious. Stop the scrolling by pressing ESC S and note the names of files with error messages. Press ESC S to continue the repair process. If no error messages are noted the repair process should be complete.

Step 3. To verify a successful repair operation type:

>SHOW DISK DRIVNAME<RETURN>

If the misallocation has been repaired normal system operation can resume. If not, try rebooting the system and execute another SHOW DISK command. If the problem persists contact your dealer.

Correcting Positive Misallocations

Positive misallocations are the most hazardous as they indicate that two files are attempting to occupy the same disk space. To correct this condition follow these steps:

Step 1. Be certain the system is in single-user mode. Then type:

>REPAIR S (or drive to be repaired) <RETURN>

Step 2. As the files are listed to the screen, press ESC S (or on some terminals ESC ESC S) to pause command execution to note the file(s) with the error message under it.

Step 3. Log on to the account who owns the file and execute the KILL command. Type:

KILL FILENAME.FILETYPE DRIVNAME<RETURN>

Step 4. Type:

REPAIR DRIVNAME (FIX)<RETURN>

Step 5. After all files have been displayed type:

>SHOW DISK DRIVNAME<RETURN>

The misallocation should have been corrected. Sometimes it may be necessary to reboot the system in order to eliminate the misallocation message. If the misallocation has not been corrected, it is advisable to contact the company the system was purchased from for further instructions.

Summary

In addition to the THEOS command files, a number of other files are required by THEOS. These files are supplied as standard equipment with every copy of THEOS. As is the case with the SYSTEM.ERRMSG and SYSTEM.EXECnn files, it is possible to modify the contents of the files to better suit the needs of the user.

The diagnostic commands, REPAIR and KILL, have been provided to correct disk problems caused by misallocated files. Disk misallocations occur as a result of a system power failure during a disk operation, or a *bug* in a program that does not close a THEOS file properly.

6

The THEOS EXEC Language

Introduction

One of THEOS's most powerful features is the EXEC language. Unlike the THEOS commands, the EXEC language is a process control language implemented with user-written EXEC programs to perform repetitive functions of the operating system. It allows the user to write programs that facilitate day to day system operation. We must master the EXEC language to write these programs. The EXEC language is similar to the JCL (Job Control Language) used on mini and mainframe computers. It has the added feature of being able to interact with the operand and allows for conditional command execution.

The language can be used to write EXEC programs which can:

- effectively isolate the user from the operating system via a series of menus
- minimize operator error
- minimize the knowledge required to use THEOS commands

For example, an EXEC program might be written to perform daily disk backups. The operator would simply type the name of the EXEC program and the program would provide instruction on the loading of disks, their location, what to do with them later and would perform the backup.

This chapter consists of a tutorial on EXEC language usage. By the end of this chapter you will be able to begin writing your own EXECs which will make your system “menu driven.”

The EXEC language consists of two parts. The first is the EXEC file or program containing instructions, and the second is the EXEC language processor which interprets and executes the instructions. The language processor executes an EXEC program as if the operator were entering instructions for the computer directly from the keyboard. As far as we are concerned, the language processor is transparent to the user and will not be dealt with here.

An EXEC program is a sequential format file which is created and edited using the EDIT command. This file may contain:

- * CSI commands
- * EXEC instructions
- * data for a user program
- * data for executing another EXEC

It is the job of the language processor to read and interpret the instructions in the EXEC file and execute them. These instructions may be in the form of THEOS command lines or EXEC instructions.

Beginning with the section on the “BEGSTACK instruction,” whenever the user is asked to “type:” a command, we are assuming that the user has first executed the EDIT command and has entered the subcommand of INPUT. It may be helpful to review the tutorial session on the EDIT command before continuing.

Special Situations: System Cancel, IPL.EXEC, and User Logon File

SYSTEM CANCEL

When THEOS is operating under control of an EXEC, it may sometimes be necessary to terminate the operation of that EXEC. To do so enter the sequence:

ESC Q (on some terminals ESC ESC Q)
Cancel EXEC (Y/N)?

If a “Y” is entered, control of the system will return to CSI. If a response of “N” is entered, the EXEC continues to control the system.

Note: It is recommended that only a response of “Y” be entered because entering “N” may cause unpredictable results. In versions of THEOS 6.0 or greater the user is automatically returned to CSI without the cancel EXEC message.

IPL.EXEC AND USER LOGON FILES

One of THEOS’s more “user friendly” features is the execution of an EXEC program when the system is first turned on or when a user logs onto a new account.

After the system is “booted” and the date and time entered, THEOS will automatically execute an EXEC program named IPL.EXEC. This EXEC usually belongs to the IPL account and its synonym accounts or the SYSTEM account. An IPL account is required for proper operation.

The file IPL.EXEC should contain the instructions or commands to be executed following booting of the system. These commands might include setting up the system in multi-user mode, attaching printers, logging off the system, and subsequent display of the "Logon please:" message.

Note: When no IPL account exists, the user is requested to log on to an account via the "Logon please:" message.

After a successful logon is accomplished, THEOS searches that account's disk directory for an EXEC file equal to the account name. Let's assume we have logged onto an account called USER1. If a file called USER1.EXEC exists, it will be executed automatically like IPL.EXEC. The USER1.EXEC may in turn contain a series of instructions that execute another EXEC program which displays a series of menu choices of programs to be used that day.

USER1.EXEC is an example of a *user logon file*. IPL.EXEC is a special example of a user logon file. Let's examine the use of a user logon file.

When requested to,

Logon please: **USER1** <RETURN>

the operator logs on as USER1 and THEOS searches the disk for an EXEC file named USER1. If USER1 is found the following line might be displayed:

PRESS RETURN KEY TO GO TO MENU

After the <RETURN> key is pressed, the operator may be presented with a series of choices as shown below.

```
Ashley Communications, Inc.
22:33:51 Friday, January 25, 1985
MASTER MENU
1. ACCOUNTING/ORDER ENTRY/INVENTORY CONTROL
2. FINANCIAL MODELING
3. TERMINAL PROGRAM
4. Spellbinder WORD PROCESSING
5. SYSTEM UTILITIES
6. SEND MESSAGES TO ANOTHER USER
7. RETRIEVE MESSAGES (EXECUTE MAILBOX)
8. LOGOFF SYSTEM/LOGON TO NEW ACCOUNT
9. multiWrite WORD PROCESSING
10. QUIT FOR THE DAY
ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN:
```

Figure 6.1. Menu

In this example there may have been as many as three EXECs involved: IPL.EXEC, USER1.EXEC, and MASTER.EXEC. MASTER.EXEC is responsible for displaying a series of menu choices on the screen.

The EXECn Files

As discussed in chapter 5 in order for EXEC programs to be run, a file named SYSTEM.EXECn must exist for each user partition. The purpose of this file is to store program variables used by the EXECs during execution. You can check the number of SYSTEM.EXECn files included on your system by logging onto the SYSTEM account and typing:

```
>F SYSTEM.EXEC# <RETURN>
```

THEOS responds with a list similar to figure 6.2.

```

FILELIST SYSTEM.EXEC#:*                               03/07/86 18:06 Page 1
Fname--- Ftype-- Dr --Date-- Time-- --Recs Blks Format- Label---
SYSTEM EXEC1 S 03/07/86 18:04 4 2 D 512 HARDISK1
SYSTEM EXEC2 S 03/07/86 18:04 4 2 D 512 HARDISK1
SYSTEM EXEC3 S 03/07/86 18:04 4 2 D 512 HARDISK1
3 files, 6 blocks.

```

Figure 6.2 List of SYSTEM.EXECn files

Figure 6.2 identifies three SYSTEM.EXEC files each containing four records. The number of records in each file determines the maximum number of EXECs that can be used. The fact that there are three files means that only three user partitions can use EXEC programs. If the system has enough memory available for five user partitions, then two more SYSTEM.EXECn files must be created in order for partitions four and five to run EXEC programs. The files are also delete and write protected.

Let's examine how to increase the number of records in a SYSTEM.EXECn file as well as how to create additional SYSTEM.EXECn files for the other user partitions.

CHANGING AN EXISTING SYSTEM.EXECn FILE

Step 1. Be certain you are logged onto the SYSTEM account. Remove the delete and write protection from the file by typing:

```
>RENAME SYSTEM.EXEC1 S (NOW NOD<RETURN>
```

Step 2. Before a new file can be created the old one must first be erased. Type:

```
>ERASE SYSTEM EXEC1 S <RETURN>
```

Step 3. Type:

```
>CREATE SYSTEM.EXEC1 S (DIRECT FILE 12 RECL 512<RETURN>
```

Step 4. To verify that the file was successfully created type:

>F SYSTEM.EXEC1 S|<RETURN>

If other SYSTEM.EXECn files must be changed, follow steps two through four above for the other files.

Note: If the files to be changed are not first erased, the error message “File already exists” will be displayed and the new file will not be created.

CREATING A NEW SYSTEM.EXECn FILE

Step 1. Be certain you are logged onto the SYSTEM account.

Step 2. To create a new file with the capability of using twelve EXEC programs type:

>CREATE SYSTEM.EXEC4 S (DIRECT FILE 12 RECL 512,<RETURN>

Step 3. To verify that the file was successfully created type:

>F SYSTEM.EXEC4 S<RETURN>

This is the SYSTEM.EXECn file for the fourth user partition. To create another file for the fifth user partition, follow steps one through three above.

The SELECTED.EXEC File

In tutorial session nine of chapter 4 on the FILELIST command, one of the options discussed was EXEC. The purpose of this option was to write to disk a list of files obtained by the FILELIST command. When the EXEC option is used with FILELIST, a special file called SELECTED.EXEC is written to disk. This file can be used with many THEOS commands and options to minimize keystrokes typed and automate certain functions.

The format for using SELECTED.EXEC is:

EXEC SELECTED COMMAND (COMMAND OPTIONS

COMMAND is the THEOS command to be executed. **OPTIONS** are the options to be associated with that command.

Let's see how we could have used SELECTED.EXEC to help with the creation of SYSTEM.EXECn files.

Step 1. Be certain you are logged onto the SYSTEM account.

Step 2. Type:

>F SYSTEM.EXEC# S (EXEC <RETURN>

A file called SELECTED.EXEC is now present on the S disk.

Step 3. Type:

```
>EXEC SELECTED RENAME (NOD NOW <RETURN>  
>RENAME SYSTEM EXEC1 S (NOD NOW)  
SYSTEM.EXEC1:S protection: NODELETE, NOWRITE  
RC=0, 13:49:02, ET=0.00 minutes.  
>RENAME SYSTEM EXEC2 S (NOD NOW)  
SYSTEM.EXEC2:S protection: NODELETE, NOWRITE  
RC=0, 13:49:04, ET=0.01 minutes.  
>RENAME SYSTEM EXEC3 S (NOD NOW)  
SYSTEM.EXEC3:S protection: NODELETE, NOWRITE  
ET=0.02 minutes.
```

If you were to type:

>FILELIST SYSTEM.EXEC# <RETURN>

You will see that the protection status of the existing SYSTEM.EXECn files has been changed.

Step 4. Type:

```
>EXEC SELECTED ERASE <RETURN>  
>ERASE SYSTEM EXEC1 S  
SYSTEM.EXEC1:S erased  
1 file(s) erased, 6 block(s) recovered.  
RC=0, 14:32:42, ET=0.01 minutes.  
>ERASE SYSTEM EXEC2 S  
SYSTEM.EXEC2:S erased  
1 file(s) erased, 2 block(s) recovered  
RC=0, 14:34:44, ET=0.01 minutes.  
>ERASE SYSTEM EXEC3 S  
SYSTEM.EXEC3:S erased  
1 file(s) erased, 2 block(s) recovered.
```

All of the SYSTEM.EXECn files have now been erased.

Step 5. Type:

```
>EXEC SELECTED CREATE (DIRECT FILE 12 RECL 512)<RETURN>  
>CREATE SYSTEM EXEC1 S (DIRECT RECL 512 FILE 12)  
>CREATE SYSTEM EXEC2 S (DIRECT RECL 512 FILE 12)  
>CREATE SYSTEM EXEC3 S (DIRECT RECL 512 FILE 12)  
>
```

Steps one and two are most important because the SYSTEM.EXECn files must be owned by the system account and the list of files in the SELECTED EXEC generated by the FILELIST command must be complete in order to create all three files in the later steps. Using SELECTED.EXEC has allowed us to save time by entering three command lines rather than nine.

Components of the EXEC Programming Language

There are many components of the EXEC language. These components consist of:

- constants and variables
- labels
- operators
- comments
- keywords some of which are part of EXEC instructions
- EXEC instructions

EXEC CONSTANTS AND VARIABLES

A constant in an EXEC program consists of a string of characters, either letters or numbers, not preceded by an ampersand (&) character. A constant answers a request by THEOS for operator input.

We have seen the use of constants in our tutorial sessions when asked to type a response to a THEOS prompt. For example in the INITDISK command if you were to initialize an attached disk you might type:

```
>INITDISK C (FORMAT SIZE 8 <RETURN>
INITDISK will erase all files on drive C(4)
Do you wish to continue (Y/N)? Y <RETURN>
Enter disk label: TEST <RETURN>
Number of directory entries = 32
Increment between adjacent sectors = 1
Offset between adjacent tracks = 0
Number of surfaces = 1
Number of tracks per surface = 77
Number of sectors per track = 13
Is above information correct (Y/N)? Y <RETURN>
```

In this example the entries Y, TEST, and Y are constants entered by the user. These entries could be placed in an EXEC as constants and entered by the EXEC program to help automate the process of initializing disks.

Variables are symbols which may represent any user entered input. In the example above, the responses to the THEOS questions were known in advance. If the answers to THEOS prompts are not known in advance, variables can be used so that whatever input is entered becomes a constant. Using the previous example, let's presume it was not known in advance which drive we wanted to initialize. We could choose a variable, according to the guidelines described below, called &DRIVE to represent the drive to be initialized. We could then place a line in the EXEC program that would prompt the system user to enter the name of the drive to be initialized. After the drive is entered, the variable &DRIVE accepts that value. The most frequently used forms of variables in THEOS EXECs are true variables and reserved variables. These are described in the following sections.

TRUE VARIABLES

True variables are indicated by an ampersand character followed by an alphabetic character optionally followed by numeric characters. Although the length of variables is technically unlimited, in most cases variables no greater than eight characters are required.

The guidelines for true variable usage are as follows:

- the first eight characters must be unique
- variable names *may not* duplicate any of the EXEC language key words including the ampersand
- the value of variables is determined by a READ instruction
- only sixteen true variables may be used in any one EXEC program

Examples of true variables include:

- &DRIVE
- &SEL
- &PICKONE
- &STOPNOW

RESERVED VARIABLES

Reserved variables are indicated by reserved variable names, such as &NULL or &RETCODE. The value of a reserved variable is determined by its function. For example, the &NULL variable indicates an empty value (0).

This variable can be used in an EXEC when input from the operator is required. If the operator keeps pressing the [RETURN] key, THEOS interprets this as a zero value. As long as the value of &NULL remains 0, the operator is not permitted to move past this point in the program.

The &RETCODE variable is a numeric value that indicates the system *return code*. Each THEOS command or program executed has a return code value associated with it. These return code values can be used in EXEC programs to determine whether or not the next step in the EXEC program will be executed.

Return Code Value	Meaning
0	Normal program execution
4	Warning errors encountered
8	Syntax errors detected in command line
12	Severe error detected (file not found, etc.)
16	Potentially disastrous error detected (disk/system integrity may be suspect)
253	Fatal run-time error during execution of RUN command perhaps while running an application program
254	Program cancelled by operator using System Cancel-key

Figure 6.2. System Return Codes

These return code values are assigned by THEOS and cannot be changed. More examples of how and where to use reserved variables will be provided later.

LABELS

An EXEC program can include labels to be used as comments or as a reference point for branching to another area of the program. A label is indicated by a minus sign (-) followed by an alphabetic

character and optionally by other alphanumeric characters. The length of a label is limited to eight characters including the negative sign. Labels must start in column one of the line. If line numbering in the EDIT command is being used, no spaces other than the one following the line number are permitted. Some examples of labels are:

- -BEGIN
- -START
- -STOP
- -AGAIN
- -OPTION1
- -OPTION2

If a label -STARTINPUT is used, an error will result because the label contains more than eight characters. It is recommended that no keywords be used as labels to avoid confusion.

OPERATORS

Operators permit arithmetic operations to be performed with variables in an EXEC program.

Comparison operators are used between two variables or constants to indicate a relationship. Comparison operators are allowed only with the &IF, &WHILE, and &UNTIL instructions. There are six comparison operators.

Operator	Symbol	Function
EQ	or =	Indicates equality
NE	or <>	Indicates not equal
LT	or <	Indicates less than
GT	or >	Indicates greater than
LE	or <=	Indicates less than or equal to
GT	or =>	Indicates greater than or equal to

Figure 6.3. Comparison operators

Examples demonstrating comparison operator usage will be provided in the section on the EXEC &IF instruction.

COMMENTS

When writing an EXEC program, it may be helpful to insert comment lines within the program as documentation. Comments can be inserted in the program as long as they are preceded by a semi-colon (;). For example as you are starting an EXEC, you might want to have the first few lines read as follows:

```
10: ;Master exec started on 1/19/85
20: ;copyright 1985 by Walter Stagner
30: ;line 40 is where the EXEC begins
40: -START
```

Lines ten through thirty are comment lines.

KEYWORDS

Keywords are the part of the EXEC instruction that tell the EXEC processor what to do. The keywords are predetermined by THEOS and cannot be changed. To write EXEC programs you must learn what these keywords do. Keywords are covered in conjunction with the EXEC instructions in the next section. Figure 6.4 lists all of the THEOS EXEC keywords.

```
&BEGSTACK
&BEGTYPE
&CAT
&CONTROL
&CRT
&END
&ERROR
&ESC
&GOTO
&IF
&INDEX
&LEN
&LINE
&LIT
&NULL
&PAGE
&QUIT
&READ
&REPEAT
```

```
&RETCODE
&SKIP
&SPACE
&STACK
&SUB
&TYP
&TYPE
&UNTIL
&WAIT
&WHILE
&*
&$
```

Figure 6.4. EXEC keywords

THE EXEC INSTRUCTIONS

The EXEC instructions consist of keywords that issue instructions to the EXEC processor for executing tasks. In addition virtually any THEOS command can be executed from within an EXEC program. For further flexibility, EXEC instructions may be combined or used together. Examples of combined EXEC instructions will be provided later in this chapter.

Approximately 25 keyword instructions exist in the THEOS EXEC language. We will discuss those most often used in alphabetical order. For information on those instructions not covered, consult the THEOS documentation.

THE &BEGSTACK INSTRUCTION

The &BEGSTACK instruction is used when data is placed in an EXEC program that would normally be entered by the operator. The format of this instruction is:

&BEGSTACK

The instruction should be placed on a line by itself to avoid confusion. The lines of data should be entered on individual lines. All &BEGSTACK data should be placed *before* the THEOS command that is to use the data. For example when using the INITDISK command, THEOS asks three questions before the initialization process begins:

Do you wish to continue (Y/N)?
Enter disk label:
Is the above information correct?

To use the **&BEGSTACK** instruction to automatically enter this data, type:

```
&BEGSTACK  
Y  
TEST  
Y  
&END  
INITDISK G (FORMAT H 2
```

The list of data elements is always terminated by the **&END** instruction. The THEOS command to be executed follows the **&END** instruction.

THE **&BEGTYPE** INSTRUCTION

The **&BEGTYPE** instruction permits the display of information on the screen. The format of the instruction is:

```
&BEGTYPE  
line(s) of text
```

This instruction is used when information is to be displayed during EXEC execution for help purposes. The information is not analyzed in any way. Each line of text entered following **&BEGTYPE** is displayed on a separate line. In order to tell the EXEC processor you are through entering text, the **&END** instruction must be used on a line following the last line of text. For example:

```
&BEGTYPE  
You can start typing some text here  
and continue on this line. If you are through type &END on next line.  
&END
```

&CONTROL

This instruction allows you to control the display of any THEOS commands executed from within the EXEC program. The format of the instruction is:

&CONTROL ON| OFF| TRACE

&CONTROL ON means that all THEOS commands executed from within the EXEC program will be displayed. &CONTROL OFF means that all THEOS commands will be executed without being displayed. &CONTROL TRACE means that all THEOS commands and EXEC instructions along with their line numbers are displayed. This is most useful during the writing of EXEC programs to trace the execution of the program and to locate "bugs."

&CRT

This instruction serves two purposes. It can be used to position the cursor to any position on the terminal, and it can be used to utilize screen attributes. The format of &CRT is:

&CRT <column# variable> <line# variable>

Column# is the value of the column number where you wish the cursor positioned. This value must be numeric and is generally a number between one and seventy-nine.

Line# is the value of the line number where you wish the cursor positioned. This value must be numeric and is generally a number between one and twenty-four.

As figure 6.5 illustrates <variable> indicates the screen control function to be performed.

Variable	Function
HOME	Moves cursor to upper left hand corner of screen
CLEAR	Clears the screen
EOS	Erase to end of screen
EOL	Erase to end of line
BELL	Sounds the terminal bell/buzzer/beeper
BON	Turns blink attribute on
BOFF	Turns blink attribute off
RVON	Turns reverse video on
RVOFF	Turns reverse video off
ULON	Turns underline on
ULOFF	Turns underline off

Figure 6.5. Screen control functions

Before using any of the screen control functions shown in figure 6.5, particularly the reverse video and underline attributes, be certain they are supported by your terminal.

Examples of the &CRT command are listed below:

&CRT 12,13	positions cursor to line 13, column 12
&CRT 7,6	positions cursor to line 6, column 7
&CRT BELL	sounds the terminal bell/buzzer/beeper
&CRT CLR	clears the screen

The CSI Instruction

All THEOS commands and application software packages can be executed from within an EXEC program by using the CSI instruction. The format of the CSI instruction is:

<command-text>

Command-text can specify:

- the command to be executed as well as the command option
- the name of the program to be run
- the name of another EXEC to be run

A CSI instruction is entered as shown in figure 6.6. When executed from within an EXEC program, it is as if the command line were typed at the THEOS prompt.

Instruction	Result
LIST IPL.EXEC S (PRINT	Lists the file IPL.EXEC to the printer
RUN OM0000	Runs a program called OM0000.BASICCOM
SINGLE EXEC	Executes another EXEC called SINGLE EXEC

Figure 6.6. CSI instruction examples

Note: When CSI instructions are issued via THEOS commands it is recommended that the entire command name be spelled out. This will speed execution time of the command.

The &END Instruction

The &END instruction is used to terminate the lines of text entered by the &BEGTYPE instruction or the list of data elements entered via the &BEGSTACK instruction. This instruction must appear on a line by itself. The format of the instruction is:

&END

If the &END instruction is omitted, an error message is displayed on the screen, and the EXEC program is terminated.

The &ERROR Instruction

The &ERROR instruction allows you to specify an instruction to be executed in the event that a system error occurs during execution of a THEOS command or EXEC instruction. The format of the instruction is:

&ERROR <instruction>

The instruction can be a valid EXEC instruction. The &ERROR instruction is most often used with the &RETCODE, &GOTO, and &TYPE EXEC instructions. For example if the following line is part of an EXEC,

&ERROR &GOTO -ERROR

when an error in execution occurs, the EXEC program will proceed to the line of the program that contains the label -ERROR. Here another series of EXEC instructions determines what happens next.

Consider the following example:

&ERROR &TYPE Error &RETCODE detected — Program aborted!

If an error occurs the &TYPE instruction will display:

Error nnn detected — Program aborted!

nnn is the system return code value. This should help the operator determine the cause of the error. EXEC execution would be terminated and control of the system would return to THEOS.

Note: The line containing the `&ERROR` instruction should be placed near the beginning of the EXEC. If an error occurs *before* the `&ERROR` instruction is encountered, the `&ERROR` instruction will have no effect. It is possible to use multiple `&ERROR` instructions in other parts of the EXEC.

The `&GOTO` Instruction

The `&GOTO` instruction allows you to *branch* to another area of the EXEC program. The format of the `&GOTO` instruction is:

```
&GOTO label
```

Label is any valid EXEC label. `&GOTO` examples follow:

```
&ERROR &GOTO -ERROR  
&GOTO -AGAIN  
&GOTO -START  
-LOOP1 &GOTO -RESTART
```

As can be seen, `&GOTO` need not be on a line by itself. It is often used with other EXEC instructions and labels. `&GOTO` is a useful instruction as it allows a branch to another area of the EXEC for re-execution without re-entering the lines of the EXEC being branched to.

The `&IF` Instruction

This instruction permits variables to be tested; depending upon the results of the test, an instruction is executed. The format of `&IF` is:

```
&IF <relation> <instruction>
```

Relation is an operator such as EQ, NE, GT, etc.

Instruction is an EXEC instruction. The `&IF` instruction evaluates the relation and, if the relation is true, executes the instruction specified. If the relation is false the instruction following the relation (operator) is ignored, and EXEC execution continues with the instruction on the next line.

Consider these lines in an EXEC:

```
50: &IF &SEL GT 2 &GOTO -START
60: &IF &SEL LT 1 &GOTO -START
70: &IF &SEL EQ 1 &QUIT
80: &IF &SEL EQ 2 &GOTO -CONT
```

Line 50 could be interpreted as, “If the variable called &SEL is greater than two go to the EXEC label called -START.” Line 60 could be interpreted as, “If the variable called &SEL is less than one go to the EXEC label called -START.” Line 70 could be interpreted as “If the variable called &SEL is equal to one then execute the &QUIT EXEC instruction.” Line 80 could be interpreted as “If the variable called &SEL is equal to two go to the EXEC label called -CONT.”

These lines are part of an EXEC we will see again later. The value of the variable &SEL is determined by operator prompted input. These lines are designed to ensure that the operator is only allowed to enter a value between one and two. Any other value is unacceptable.

The &QUIT Instruction

The &QUIT instruction allows you to terminate EXEC execution. The format of this instruction is:

```
&QUIT
```

When the &QUIT instruction is encountered, the execution of the EXEC is terminated at that point. If the EXEC was invoked from the keyboard (by typing in the name of the EXEC), control will return to THEOS and the prompt will be displayed. If the EXEC was invoked from another EXEC program, control returns to the EXEC instruction *following* the one invoking this EXEC.

The &READ Instruction

When the &READ instruction is encountered, execution of the EXEC is paused and keyboard input from the operator is requested. The EXEC prompt character (:) is displayed, and before execution is resumed the operator must make an entry. For our purposes the &READ instruction will be used to read operator input which will then be assigned to a variable. The format of the &READ instruction is:

&READ <true variable>

True variable is any user assigned variable preceded by the & character. Consider these EXEC lines:

```
30: &TYPE What Is Your Choice?
40: &READ &SEL
50: &IF &SEL GT 2 &GOTO -START
60: &IF &SEL LT 1 &GOTO -START
70: &IF &SEL EQ 1 &QUIT
80: &IF &SEL EQ 2 &GOTO -CONT
```

Line 30 causes,

What Is Your Choice?:

to be displayed on the screen. Line 40 can be interpreted as “Read the operator-entered value which will be assigned to the variable &SEL.” After the value of &SEL is “read” EXEC execution resumes.

The &SPACE Instruction

This instruction allows blank lines to be placed on the screen. It is used with the &TYPE instruction when it becomes necessary to put spaces between typed lines to make them more readable. The format of this instruction is:

&SPACE <count>

Count is the number of blank lines to be displayed. If no number is specified for <count>, one blank line is displayed.

Examples:

&SPACE	displays one blank line
&SPACE 4	displays four blank lines
&SPACE X	displays X blank lines

It is a good practice to place this instruction on a line by itself.

The &TYPE Instruction

&TYPE is used to display characters on the screen for user prompts. The format of this instruction is:

&TYPE data

Data can be any string not exceeding eight characters. Any words exceeding eight characters will be **truncated** to eight by the EXEC processor. The &TYPE instruction was mentioned in the preceding example with &READ because these instructions are most often used together. Consider the lines:

```
30: &TYPE What Is Your Choice?  
40: &READ &SEL
```

As indicated above, this line is used to prompt the operator for input. The &READ instruction is used in the next line to assign the value entered to the variable &SEL.

When using the &TYPE instruction to display user prompts, it is recommended that words be eight characters or less to avoid truncation. If it becomes necessary to use words that are longer or multiple lines of prompting text, the &BEGETYPE instruction is recommended.

The &WAIT Instruction

The &WAIT instruction causes the EXEC program to pause until the operator types a key. Its purpose is to allow the operator time to view a screen of data. The format of this instruction is:

```
&WAIT
```

This instruction should be placed on a line by itself.

Upon execution &WAIT invokes the same process as THEOS system commands when displaying information on the screen. When this instruction is encountered by the EXEC processor an up arrow character (↑) is displayed at the bottom left corner of the screen. EXEC execution continues when the operator enters either <RETURN> or presses the <space bar>. Next, the screen may be cleared, more data may be displayed, or if the &QUIT instruction is encountered, the EXEC is terminated and the user is returned to CSI (>).

Consider the following EXEC lines:

```
10: &TYPE ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU  
20: &WAIT
```

The `&TYPE` instruction displays the message advising the operator that the archive process has been completed. The `&WAIT` instruction will pause EXEC execution until the operator enters `<RETURN>`.

Using The EXEC Language

We will now consider a series of EXECs which can be used as models for writing other EXECs.

In some examples you may be asked to type in the examples and try executing them, while in others the EXEC operation and function will be discussed. Feel free to use any of these EXECs on your system or to modify them as your needs require. Complete listings of all EXECs can be found in the appendix or on disk available directly from Weber Systems, Inc.

We will start with some simple examples. Since it is desirable for all users to have access to EXECs, be certain you are logged onto the SYSTEM account when editing these files.

Example 1

This EXEC is a simple six line affair which prints six blank lines on the screen, displays a message, waits for operator input, and then executes another EXEC. The purpose of the six blank lines is to provide a space for messages to be displayed which may have been sent to the user while logged off. Let's take a look at the EXEC.

```
0005 ;USER1 EXEC
0010 &CONTROL OFF
0020 -START
0030 &SPACE 6
0040 &BEGTYPE
0050         PRESS RETURN TO CONTINUE
0060 &END
0070 &WAIT
0080 MASTER EXEC
```

The purpose of each line is as follows:

- Line 5 is a comment line.
- Line 10 turns off display of THEOS command lines.
- Line 20 is a -START label providing a reference point.
- Line 30 prints six blank lines on the screen.
- Line 40 issues the &BEGTYPE instruction.
- Line 50 is the text to be displayed by &BEGTYPE. An attempt was made to center the text by using spaces. Remember you can use the COLUMN command from within the editor to display a ruler line. This makes the job of centering text easier.
- Line 60 issues the &END instruction to terminate execution of the &BEGTYPE instruction.
- Line 70 pauses program execution until the operator enters a key.
- Line 80 is a THEOS command line.

Example 2

The purpose of this EXEC is to illustrate how variables can be used to accept operator input from the keyboard and use this input to execute THEOS commands. A method of preventing premature exiting of an EXEC program is also shown.

From the THEOS prompt type:

```
EDIT ASSIST.EXEC<RETURN>
NEW FILE:
EDIT:
```

after the EDIT prompt type:

```
*LINE ON <RETURN>
*INPUT<RETURN>
```

Begin entering each of the following lines as indicated. Press <RETURN> at the end of each line.

```
0010 &CONTROL OFF
0020 -START
0030 &CRT CLR
0040 &TYPE PLEASE TYPE THE THEOS COMMAND
      YOU WISH HELP ON OR [RET] TO LIST CMD.
0050 &READ &SEL
0060 &CRT CLR
0070 HELP &SEL
```

```
0080 &WAIT
0090 &CRT CLR
0100 &CRT 30,14
0110 &TYPE MORE HELP? Y/N
0120 &READ &SEL
0130 &IF &SEL = Y &GOTO -START
0140 &IF &SEL = N &CRT CLR
0150 &QUIT
*FILE<RETURN>
THEOS responds with:
ASSIST.EXEC:S saved.
```

After the edit mode has been entered, `LINE ON` was entered to turn on the automatic line numbering feature of the editor. On some THEOS systems, when editing an EXEC, the default condition of the `EDIT` command is `LINE ON`.

Let's examine the purpose of each line.

- Line 10 turns off display of any THEOS commands executed.
- Line 20 uses the `-START` label to signify the start of the program.
- Line 30 clears the screen.
- Line 40 displays a line of text on the screen and asks for operator input.
- Line 50 reads the `&SEL` variable which contains the user entered response.
- Line 60 clears the screen.
- Line 70 executes the THEOS `HELP` command using the user input stored in the `&SEL` variable from line 50.
- Line 80 ensures that the help messages remain on the screen until the operator presses `[RETURN]` or `<space bar>`.
- Line 90 clears the screen.
- Line 100 positions the cursor at column 30, line 14 in order to center line 110.
- Line 110 asks the operator if more help is required and asks for a response.
- Line 120 reads the response and stores it in the variable `&SEL`.
- Line 130 uses the `&IF` instruction and the `=` operator to determine what happens next. If the variable equals `Y` then the `&GOTO` instruction is used to go to the `-START` label where execution of the EXEC resumes.

- Line 140 uses the same instruction as line 130 except this time the N response is considered. If the response is N, the screen is cleared (&CRT CLR) and the &QUIT instruction is executed.

Note that in some cases, multiple instructions can be entered on the same line as in line 140.

To try this EXEC after it has been “filled” type:

ASSIST <RETURN>

If you have successfully entered it with no errors, it will perform as indicated above.

One of the goals of an EXEC may be to isolate the user from the operating system. In order to achieve this goal, it is undesirable for an EXEC to be prematurely exited. Using the example above, lines 130 and 140 provide instructions on what to do as long as the response is Y or N. What if the response is any other keyboard character including <RETURN>?

In the EXEC, as it is presently written, pressing any key other than Y or N will exit to THEOS. Let's see how we can correct this problem. Another line can be added by entering the editor:

```
>EDIT ASSIST.EXEC<RETURN>
EDIT:
*
```

Use any of the edit commands to access line 140. If you've forgotten these commands, type HELP at the edit prompt to display them. After line 140 is displayed type:

INPUT <RETURN>

The editor will automatically display a new line with a new line number. Enter the following:

```
145: &IF &SEL NE Y &IF &SEL NE Y &GOTO -START<RETURN>
```

At the edit prompt type:

FILE <RETURN>

After the EXEC has been filed try using it. If any response other than N is entered, the &GOTO instruction causes execution to continue beginning at the -START label. Line 145 also illustrates the use of multiple EXEC instructions on the same line.

Using The IPL.EXEC

Assuming a user account called IPL exists, the IPL.EXEC can be utilized. On most THEOS systems, the IPL.EXEC is responsible for readying the system for operation. As you already know, the system can be readied for use by entering keyboard commands. The primary purpose of this EXEC is to *completely automate* system start-up.

The IPL.EXEC we will use as an example occupies over two pages. As EXECs increase in their complexity, it is recommended that they be written in small sections each beginning with a line label for easy identification. It is best to outline *on paper* what functions you wish the EXEC to perform. Each function should begin with a label. The following example will illustrate the use of labels.

The IPL.EXEC automatically executes the following tasks:

- provide the user a choice of single or multi-user operation
- attach the main printer
- set the system up in a single-user mode and execute a LOGOFF command
- set the system up in multi-user mode, displaying a series of messages describing the process and execute a LOGOFF command
- execute a SHOW DISK command to warn the operator of any disk misallocations
- request a logon

To conclude our discussion of EXEC, our example will focus on incorporating CSI instructions into EXECs.

Example 3

```
0005 -START
0010 &CONTROL OFF
0020 &CRT CLR
0030 &TYPE (S)INGLE (M)ULTI USER MODE OR MULTI MODE WITH (N)O SPOOLER?
0040 &READ &MODE
0050 &IF &MODE=N &GOTO -SMUNOSP
0060 &IF &MODE=M &GOTO -MULTI
0070 ATTACH PRINTER PORT6
0080 &CRT CLR
0090 &BEGTYPE
0100     Now displaying disk space...
0110
0120 &END
0130 SHOW DISK
0140 &BEGTYPE
0150
0160     System Ready...
0170 &END
0180 LOGOFF
0190 -MULTI
0200 &CRT CLR
0210 &BEGTYPE
0220     Starting multi user mode for 5 terminals...
0230 &END
0240 &CRT 1 03
0250 &BEGTYPE
0260 Attaching main printer...
0270 &END
0280 ;ATTACH PRINTER
0290 ATTACH PRINTER1 PORT6
0300 &BEGTYPE
0310 Loading spooler...
0320 &END
0330 LOAD SPOOLER
0340 &BEGTYPE
0350 Setting memory for terminal 2...
0360 &END
0370 SET MEM 2 1 1
0380 &BEGTYPE
0390 Setting memory for terminal 3...
0400 &END
0410 SET MEM 3 1 2
0420 &BEGTYPE
0430 Setting memory for terminal 4...
0440 &END
0450 SET MEM 4 1 3
0460 &BEGTYPE
0470 Setting memory for terminal 5...
0480 &END
0490 SET MEM 5 1 4
0500 SET MEM 6 3500 5
0510 &CRT CLR
0520 &BEGTYPE
0530 Memory is set for the system...
```

```
0540 &END
0550 SPOOLER INIT
0560 &BEGTYPE
0570 Starting terminal 2...
0580 &END
0590 START 2 (PORT2 L79 P23 C7
0600 &BEGTYPE
0610 Starting terminal 3...
0620 &END
0630 START 3 (PORT3 L79 P23 C7
0640 &BEGTYPE
0650 Starting terminal 4...
0660 &END
0670 START 4 (PORT4 L79 P23 C7
0680 &BEGTYPE
0690 Starting terminal 5...
0700 &END
0710 START 5 (PORT5 L79 P23 C7
0720 SPOOLER START
0730 &CRT CLR
0740 &BEGTYPE
0750     Now displaying disk space...
0760
0770 &END
0780 SH DI
0790 &BEGTYPE
0800
0810     System Ready...
0820 &END
0830 LOGOFF
0840 -SMUNOSP
0850 &CRT CLR
0860 &CRT 20 13
0870 &BEGTYPE
0880 PROCESSING MULTI-USER WITHOUT SPOOLER!!!
0890 &END
0900 &CRT CLR
0910 &BEGTYPE
0920 Attaching main printer...
0930 &END
0940 ;ATTACH PRINTER FOR MULTI-USER WITHOUT SPOOLER
0950 ATTACH PRINTER1 PORT6
0960 &TYPE Setting memory for terminal 2...
0970 SET MEM 2 1 1
0980 &TYPE Setting memory for terminal 3...
0990 SET MEM 3 1 2
1000 &TYPE Setting memory for terminal 4...
1010 SET MEM 4 1 3
1020 &TYPE Setting memory for terminal 5...
1030 SET MEM 5 1 4
1040 &CRT CLR
1050 &TYPE Starting terminal 2...
1060 START 2 (PORT C7
1070 &TYPE Starting terminal 3...
1080 START 3 (PORT3 C7
1090 &TYPE Starting terminal 4...
1100 START 4 (PORT4 C7
```

```
1110 &TYPE Starting terminal 5...
1120 START 5 (PORT5 C7
1130 &CRT CLR
1140 &BEGTYPE
1150     Now displaying disk space...
1160
1170 &END
1180 SHOW DISK
1190 &BEGTYPE
1200
1210     System Ready...
1220 &END
1230 LOGOFF
```

As can be seen, the line labels and their functions used in this EXEC are:

- -START indicates the starting point for the EXEC and contains the instruction for attaching the main printer.
- -MULTI starts the section which invokes multi-user mode.
- -SMUNOSP starts the section which invokes multi-user mode with no spooler.

The most important feature of this EXEC is the extensive use of THEOS CSI commands.

In an EXEC, CSI commands are entered on individual lines just as if you were typing a command line. The only difference is that EXEC line entry is accomplished using the EDIT command.

The -START section includes three THEOS commands: ATTACH, SHOW DISK, and LOGOFF.

The -MULTI section includes the ATTACH, LOAD SPOOLER, SET MEMORY, SPOOLER INIT, START, SHOW DISK, and LOGOFF commands.

The -SMUNOSP section includes the same commands as the -MULTI section minus the commands for spooler start-up.

Use the EDITOR to enter the lines for this EXEC and begin using it. Before making any modifications, try using the EXEC as it is. If necessary, you can customize it later.

Appendix E is devoted to describing a turnkey THEOS system which uses a series of EXECs to isolate the user from the operating system. Figure 6.7 lists these EXECs and their functions.

EXEC Name	EXEC Function
IPL.EXEC	System start-up
MASTER.EXEC	Allows user to select programs from a menu
UTIL.EXEC	Performs system maintenance functions
CPY.EXEC	Performs disk copies as requested from UTIL.EXEC
SAFECOPY.EXEC	Performs archives as requested from UTIL.EXEC
RES.EXEC	Performs restore as requested from UTIL.EXEC
SINGLE.EXEC	Brings system down to single-user mode
USER1.EXEC	All USER EXECs direct users to the MASTER.EXEC
USER2.EXEC	
USER3.EXEC	
USER4.EXEC	
USER5.EXEC	

Once the system manager has installed these EXECs, it should *never* be necessary for the user to see the THEOS prompt and enter a command line. Complete line by line listings of all of these EXECs are provided in the appendix.

Depending upon its complexity, an EXEC can become quite large. Once an EXEC begins to occupy more than 10K of disk space, it becomes inconvenient to edit and difficult to trace its operation. For this reason, it is recommended that smaller EXECs be used that are called into action by a menu EXEC. That is precisely what happens in the UTIL.EXEC. Although the UTIL.EXEC itself is very large, it calls smaller EXECs into action when certain functions need to be performed such as archiving, restoring, and copying files. Study the MASTER.EXEC and the UTIL.EXEC to see how this is done.

If you plan to use these EXECs be certain that they are owned by the SYSTEM account. If you type them in manually check your work periodically for errors. Generic installation instructions are provided when the EXECs are purchased on diskette.

Summary

The purpose of this chapter was to familiarize the system manager with the EXEC language so that EXECs or EXEC programs could be written to automate operating system functions or to isolate users from the operating system. Sample EXECs have been provided to illustrate the use of the language and may be used in their present form on your system. It is possible for you to take what you have learned here and apply your knowledge by writing your own EXECs or by modifying those presented in this book.

7

Using Menus Created By The EXEC Language

This chapter and those that follow are designed for the non-technically oriented THEOS user. You can read these chapters without having read the preceding, although chapter 4 would be helpful.

This chapter is designed to illustrate the use of EXEC programs in a turnkey THEOS 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 work the computer.

The EXECs which are used here are designed to eliminate the need to enter THEOS command lines. The discussion that follows assumes that all of the EXEC's listed in appendix E have been installed by the system manager without modification. We also assume that:

- The THEOS operating system has been installed.
- Application software has been installed.
- User accounts have been set-up, possibly including passwords.

- A SYSTEMn.EXEC has been set up for each user partition with a file size of twelve, allowing all EXECs discussed here to be utilized.
- Certain lines of the MASTER.EXEC have been modified by the system manager to accommodate the program names of your particular application software.

The THEOS system used by the author had virtually no EXEC programs of any kind supplied with it. The EXECs discussed here were written to facilitate system use, particularly for novice users. Implementing these EXEC's on your system may also make it easier to use.

The use of three EXECs will be examined in detail. These EXECs are:

- IPL.EXEC
- MASTER.EXEC
- UTIL.EXEC

Starting The System: The IPL.EXEC

Using an EXEC to start an THEOS system is as simple as the following steps indicate:

- Step 1. Turn on the computer, terminal(s), and printer.
- Step 2. Press the reset button on the computer if required.
- Step 3. Press the key marked ESC.
- Step 4. Type:

BS

On some systems it may be necessary to input the time and date. If your system requires that information, enter it.

After the time and date have been entered THEOS responds:

```
Logon please:IPL
Sign-on message 1
Sign-on message 2
THEOS Multi-user version 5.6A
Logon at 19:43:53, on 01/28/85
```

Sign-on messages one and two are replaced by information about your particular type of computer hardware. The screen is quickly cleared and replaced by the following message:

{S}ingle or {M}ulti user mode or multi user with {N}o spooler?:

Step 5. Select the mode of operation you desire. If S <RETURN> is entered, the system is activated for one user and a printer. If M <RETURN> is entered, the system is activated for multi-user operation for five terminals, the print spooler, and a printer.

```
Starting multi user mode for 5 terminals. . .
Attaching main printer. . .
Loading spooler. . .
Setting memory for terminal 2. . .
Setting memory for terminal 3. . .
Setting memory for terminal 4. . .
Setting memory for terminal 5. . .
Memory is set for the system. . .
Starting terminal 2. . .
Starting terminal 3. . .
Starting terminal 4. . .
Starting terminal 5. . .
        Now displaying disk space. . .

S(1) Label: "SYSTEM ",
        Backup onto "SYSTEM$B" on 01/15/85 23:30,
        Capacity: 497 bytes, (77-1-26),
        Available: 56K bytes (11%),
        Largest area: 56K bytes,
        95 files in use (out of 176)

        System Ready. . .
```

Figure 7.1. Screen display of multi-user with spooler start-up

If N <RETURN> is entered, the system is activated for multi-user operation without the print spooler.

```
PROCESSING MULTI—USER WITHOUT SPOOLER!!!!
Attaching main printer. . .
Setting memory for terminal 2. . .
Setting memory for terminal 3. . .
Setting memory for terminal 4. . .
Setting memory for terminal 5. . .
Starting terminal 2. . .
Starting terminal 3. . .
Starting terminal 4. . .
Starting terminal 5. . .

          Now displaying disk space. . .

S(1) Label: "System ",
      Backup onto "SYSTEM$B" on 01/15/85 23:30,
      Capacity: 497 bytes, (77-1-26),
      Available: 56K bytes (11%),
      Largest area: 56K bytes,
      95 files in use (out of 176)

          System Ready. . .
```

Figure 7.2. Screen display of multi-user with no spooler start-up

After system set-up in the desired mode is completed, the user is requested to Logon please.:

- Step 6. At this time, type the name of the account to which you wish to logon. The list of user account names acceptable by the system is determined by the list of account names entered by the system manager. The account name is usually the name of the user, making it easy to remember.

As you can see, it is not necessary to enter any command line or know anything about THEOS itself. All that is necessary is to know which mode of operation you desire for the system and your user account name.

Once you have successfully logged onto an account, you are asked to "PRESS RETURN TO CONTINUE."

Running Application Software From A Menu: The MASTER.EXEC

Presumably the software to be used on the system has already been selected and installed. In order to run an application package, it is necessary to type RUN followed by the name of the program and <RETURN>. The purpose of the MASTER.EXEC is to isolate the user from the operating system and to provide the user with menu choices. All that is necessary to run a program is to type the number of the program and press <RETURN>; the EXEC does the rest.

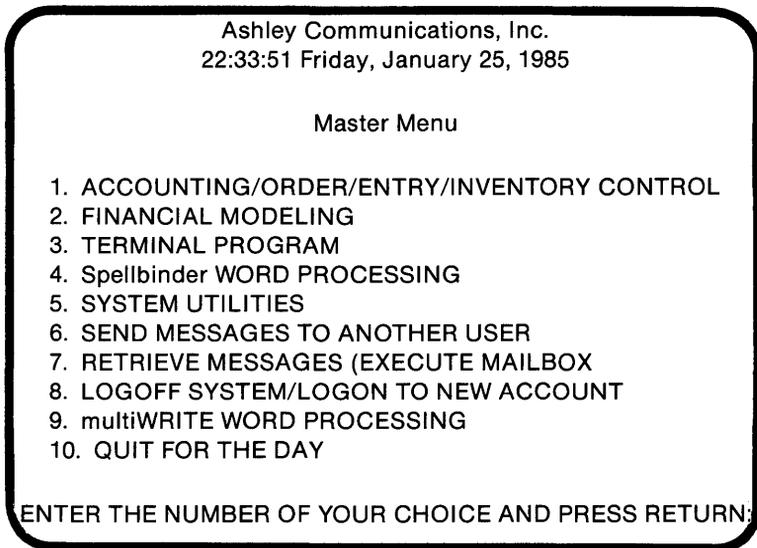


Figure 7.4. The Master menu

Let's examine the function of each option

OPTION 1: ACCOUNTING PROGRAM

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. There are several fine spreadsheet programs available for THEOS.

OPTION 3: TERMINAL PROGRAM

The terminal program is the THEOS Terminal Emulator. When this option is selected, the user is taken to the Terminal Emulator menu where communication with a remote computer can be established and data sent, received, or stored for future use.

OPTION 4: SPELLBINDER WORD PROCESSOR

The Spellbinder word processing package is selected with this option. Spellbinder is a very flexible and popular word processing program. It has been integrated with many other software packages where word processing is required.

OPTION 5: SYSTEM UTILITIES

When this option is selected, the user is taken to a utility menu where system maintenance functions such as account maintenance, archiving, restoring, and erasing files can be performed. This option is provided as a convenience for the system manager and may be password protected. In addition, most of the functions on this utility menu require that the user be logged onto the SYSTEM account. The use of the utility menu is described in detail below.

OPTION 6: SENDING MESSAGES TO ANOTHER USER

Sending messages to another user is a built-in feature of THEOS. When terminals are dispersed throughout a building, it may be convenient to send messages to others currently using the computer.

To send messages to another user type:

6<RETURN>.

The following screen is displayed:

This option allows you to send brief messages to other users on the system. In order to use this function, you must know the account name of the person(s) you wish to send the message to. You may even request that the person you are sending a message to send YOU a reply after receiving your message.

1. SEND SAME MESSAGE TO ALL USERS CURRENT
2. SEND MESSAGE TO SPECIFIC USER
3. RETURN TO MAIN MENU

ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN:

If choice one is selected, the screen is cleared and the following,

Enter message text, terminate input by empty line.

?

is displayed. You may type any message you like. When finished press the <RETURN> key twice and you are returned to the master menu.

If choice two is selected, the screen is cleared and you are asked:

Who would you like to send a message to?:

Enter the account name of the person to receive the message.

If that person is not currently logged on, THEOS will display:

User is not logged on.

Do you wish to deposit in mailbox?

If you type in Y<RETURN> you are then asked to:

Enter message text, terminate input by empty line.

After entering the message press the <RETURN> key twice to return to the master menu. If you type in N <RETURN> in response to the mailbox question, you are immediately returned to the master menu.

Choice three on this menu is provided in case you change your mind about entering messages at this time. When choice three is selected, you are returned to the master menu without sending any messages.

When messages are sent to a user that is currently logged on, the message is displayed on that user's screen immediately without affecting the work currently being undertaken at that terminal.

When messages are deposited in a user's mailbox, that user is notified of this at the time of logon. In order to retrieve messages deposited in the mailbox, option seven from the master menu must be selected.

OPTION 7: RETRIEVE MESSAGES

This option executes the THEOS MAILBOX command. Assuming you have just logged onto the system and a user has deposited a message in your mailbox, the message will appear on your terminal after the logon time and date.

"You have messages waiting. To retrieve, execute MAILBOX".

To retrieve the messages, select option seven from the master menu. When option seven is selected the screen is cleared and the following is displayed:

From ACCOUNTNAME, on date, at time;
Message text.

ACCOUNTNAME is the account sending the message. Date is the date the message was sent. Time is the time the message was sent. Message text is the message itself.

At the bottom of the screen the following text is displayed:

If screen is blank then, sorry, you have no messages waiting.

PRESS RETURN KEY TO CONTINUE

The purpose of this text is to provide information to the user in the event option seven is selected and there are no messages in that user's mailbox. Pressing the <RETURN> key returns you to the master menu.

OPTION 8: LOGOFF SYSTEM/LOGON TO NEW ACCOUNT

On most systems this option is selected at the end of the day to terminate use of the system. When this option is selected, the screen is cleared and the following message is displayed:

WOULD YOU LIKE TO MAKE BACKUP COPIES OF TODAY'S WORK? Y/N:

If a response of N<RETURN> (or any response other than Y) is entered, the user is automatically logged off the system. If a response of Y<RETURN> is entered, the utility menu appears for the purpose of making backup copies of data files. Making backup copies is usually the responsibility of the system manager, however in some companies, each user may be responsible for making his own copies. If the system manager has modified the MASTER.EXEC such that a password is required to access to the utility menu (and you don't know the password) then making backups is not your responsibility.

OPTION 9: multiWRITE WORD PROCESSING

Selecting option nine presumes that the multiWRITE word processing package has been installed on your system. Why two word processing packages on the same system? Each package has certain strengths the other lacks. See chapter eight for more details on this option.

OPTION 10: QUIT FOR THE DAY

This option may not be required for all computer systems. The computer hardware being used during the writing of this book is an IBC Middi-Cadet with a 5¼ inch 20 Megabyte hard disk. Before system shut-down, the computer requires that a special THEOS command called HOME be executed to position the heads which read the disk safely in the *landing zone*. If this command is not executed, it is possible that damage to the drive may occur and data may be lost. When this command is executed, no message is displayed on the screen. You can hear the heads being re-positioned if you are close to the computer.

A Word About Master Menu Error Messages

Due to the simplicity of the MASTER.EXEC there are few error messages likely to be displayed. Even if an error message is displayed, it will be brief and the user will immediately be returned to the master menu. The most likely messages to be displayed are "XYZ.COMMAND not found," or "XYZ.PROGRAM not found." These messages mean that the program names being called for by the EXEC are not present on the system. If either message is displayed, consult your system manager.

Performing System Maintenance Functions: The UTIL.EXEC

When option five is selected from the master menu, the user is taken to the system utilities menu. If UTIL.EXEC has been modified by the system manager to require a password for access, the average user will not be permitted to access the utilities menu. Because this menu is important in the turnkey operation of a THEOS system, we will examine the function of each option on the utilities menu in detail. If you encounter any errors when attempting to execute the UTIL.EXEC, report the error to the system manager. The size of the EXEC program may have to be reduced to accommodate your hardware.

Ashley Communications, Inc.

19:49:32 Tuesday, January 29, 1985

SYSTEM UTILITIES MENU

- | | |
|------------------------------------|----------------------|
| 1. COPY FILES | 7. SINGLE USER MODE |
| 2. ARCHIVE TO FLOPPIES/TAPE | 8. MULTI USER MODE |
| 3. RESTORE/VERIFY FLOPPIES or TAPE | 9. MULTI USER MODE |
| 4. ACCOUNT MAINTENANCE | 10. ERASE FILES |
| 5. REPORT MIS-ALLOCATIONS STATUS | 11. LIST DISK FILES |
| 6. FORMAT FLOPPIES/TAPE | 12. RETURN TO MASTER |
| 13. LOGOFF SYSTEM | |

TYPE [C] TO CONTINUE COPY [S] TO SELECT NEW FILES [Q] TO QUIT:

Figure 7.4. The System Utilities menu

Most options included help information with their first screen display. When on-screen help is not available, the user is prompted for specific information and help should not be needed. At times, when prompted for input, it is not possible to continue past that point unless an entry is made. If at any time you are unsure about what to do next, keep making entries and/or press <RETURN> until an option to quit or return to a menu is presented. After an option has been successfully executed to completion, you are returned to the utilities menu. Remember that <RETURN> must be pressed to complete each entry.

Because some utility functions require that the system be in single-user mode, it is recommended that the main terminal be located close to the computer for convenient access to disk and tape drives.

OPTION 1: COPY FILES

This option allows the THEOS COPYFILE command to be executed. The COPYFILE command permits files to be copied in their entirety on a file-by-file basis or in groups using wildcard characters. Only files belonging to the presently logged account may be copied. When option 1 is selected the following screen is displayed:

This option will allow you to make backup copies of your files either on a file by file basis, or all files. In order to make these copies, you will be prompted by a series of questions on the screen to make it easy and convenient for you. But first, some special notes:

- 1 Your floppy disk drive is designated as G.
- 2 Your main system disk is labelled as S. Subsequent platters are labelled as [A,B,C, etc].
3. If you just want to copy all of your files, just type * when prompted to do so. Otherwise, copying is on a file-by-file basis. You may copy either from the hard disk to floppy or from one hard disk platter to another. However, if you are copying from hard disk to floppy disks you must first INSERT AN INITIALIZED DISK INTO THE FLOPPY DRIVE.

PRESS [C] TO CONTINUE OR [F] IF YOU NEED TO FORMAT FLOPPIES

If F is entered, you will be returned to the utility menu so that you may select the option to format floppy disks or tapes. If C is entered, execution continues to the next prompt. The screen is cleared and you are asked:

What is the file name you wish to copy? [* for all files]:
Enter the file name of the file to be copied.
What is the filetype to go with the above name? [* for all files]:
Enter the file type of the file to be copied.
Which disk are you copying from? [S-A-G/ ETC.]:
Enter the disk letter being copied from.
Which disk are you copying to? [S-A-G / ETC.]:
Enter the disk being copied to.

After <RETURN> is pressed, the terminal will beep and display the following:

FILENAME FILETYPE WILL BE COPIED FROM DRIVE X TO
DRIVE Y

[* * MEANS THAT ALL FILES WILL BE COPIED]

The following message,

TYPE [C] TO CONTINUE [S] TO SELECT NEW FILES [Q] TO QUIT:

is displayed at the bottom of the screen. If Q is selected, you are returned to the utility menu. If S is selected, you are given the opportunity to enter new file descriptions. If C is selected, the COPYFILE command is executed according to the specifications entered. The screen is cleared and the following is displayed:

Please wait ...
FILENAME.FILETYPE:X copied to FILENAME.FILETYPE:Y
.
.
.
.

All files specified for copying are displayed during the copy process. When all files have been copied, (^) will be displayed in the lower left-hand corner of the screen. Press <RETURN> at this time. The screen is cleared, and the following prompt appears:

DO YOU WISH TO MAKE MORE COPIES? Y/N:

If N is entered, you are returned to the utility menu. If Y is entered, you will be asked to enter new file descriptions, and the copying process will be repeated. Should any errors be encountered during execution, the following message is displayed:

THE SYSTEM HAS DETECTED AN ERROR!!
If you wish to continue making copies you must start over and retry the copy operation. Type YES for a Retry, anything else to return to the System Utilities Menu.

If you wish to try again “YES” must be entered. Any other response returns you to the utility menu.

OPTION 2: ARCHIVING TO FLOPPY DISKS OR TAPE

The archive option is frequently selected. This option is used to make backup copies of all files, particularly those too large to fit on one disk in their entirety. When this option is selected the following screen is displayed:

In order to make an ARCHIVE, you must be in SINGLE-USER mode. In addition, you must have INITIALIZED FLOPPIES OR TAPES ready to insert into the disk (or tape) drive when prompted to do so. If you need to INITIALIZE FLOPPIES OR TAPES please choose the option to return to the System Utilities Menu. In addition if you need to bring the system down to SINGLE-USER mode, please choose the option to return to the System Utilities Menu.

1. RETURN TO SYSTEM UTILITIES MENU
2. CONTINUE WITH ARCHIVE PROCEDURES

What is your Choice?:

If one is entered, you are returned to the utility menu. When two is entered, the next archive screen is displayed:

The ARCHIVE is useful because it allows you to make copies of files that would normally be too large to fit on just one floppy disk. Once you have made your initial choices, the system prompts you on what you are to do next.

******* SPECIAL NOTES *******

1. THE FLOPPIES/TAPES YOU USE FOR THE ARCHIVE SHOULD CONTAIN ANY PREVIOUS FILES YOU ARE USING AS THEY WILL BE ERASED DURING THE ARCHIVE.
2. DEPENDING UPON THE SIZE OF YOUR FILES, YOU MAY HAVE TO USE MORE THAN 1 FLOPPY DISC (IF YOU ARE ARCHIVING TO FLOPPY).

ARCHIVE TO [T]APE OR [F]LOPPY?:

If T is entered, the archiving procedure for tapes is begun. If F is entered, the archiving procedure for floppy disks is begun. Because the procedure for archiving floppy disks is similar to the procedure for tapes, a discussion of the floppy disk procedure should suffice for both. After F is entered, the next screen is displayed:

You will now be able to specify HOW you wish to archive. The most economical way of making an archive is to select only those files that have been modified ON OR AFTER THE DATE YOU ENTER. However, ONCE A MONTH we recommend that you make a complete archive. If you are making a COMPLETE archive, have plenty of initialized floppies ready.

1. COMPLETE ARCHIVE
2. ARCHIVE YOUR ACCOUNT BY DATE
3. ARCHIVE ONLY YOUR ACCOUNT

What Is Your Choice?:

When it is necessary to backup everything on the system, choice one is selected. The screen is then cleared, and the following is displayed:

Which drive do you wish to ARCHIVE from [S/A/B ETC.]:

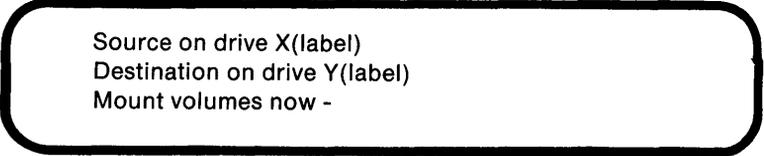
After the drive to ARCHIVE from is specified, the following message is displayed:

ALL FILES ON DRIVE X WILL BE ARCHIVED TO DRIVE G

X will be indicated as the drive you entered. This EXEC automatically assumes the floppy disk drive is designated as G. The following message:

PRESS [C] TO CONTINUE ARCHIVE [S] TO SELECT NEW DRIVE OR [Q] TO QUIT:

is then displayed providing a new series of choices. If S is chosen, a new drive can be selected. If Q is chosen, you are returned to the utility menu. If C is chosen, the screen is cleared and the following screen displayed:

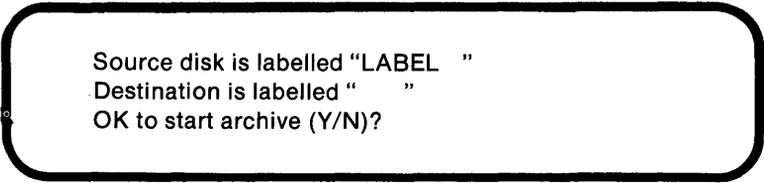


```
Source on drive X(label)
Destination on drive Y(label)
Mount volumes now -
```

This is an information message showing the assigned name of the platter to archive from X(label) and the assigned name of the disk to receive the archive Y(label).

“Mount volumes now -” is a request by THEOS for the user to insert media into the drive and input a response when ready to continue. Enter either Y or <RETURN> to continue. If N is entered, the error message described below is displayed.

When continuing, the next screen displayed is:



```
Source disk is labelled "LABEL "
Destination is labelled " "
OK to start archive (Y/N)?
```

This is an information message confirming that drives specified for archiving are correct. THEOS then asks if it is OK to continue. To continue, enter “Y.” If “N” is entered the error message described below is displayed. If it is OK to continue, THEOS displays:

```
Selecting and sorting files.
X files selected.
```

X is the number of files THEOS has selected for archiving. If all goes as planned the archive process continues displaying each file description archived. Most of the archive process is automatic but you

must be ready to insert the next volume of media when prompted to do so. Most THEOS computers use double sided double density floppy disks which can hold approximately one megabyte of data. If you have five megabytes of data files to backup, have five diskettes at hand. *Do not leave the system unattended at this time!*

Upon successful completion of the archive, the screen is cleared and the message is displayed.

ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU

Remember that each hard disk platter must be archived individually. If your system includes more than one hard disk platter, this procedure must be repeated for each one.

Choice two is the one most frequently used. This option allows backup copies to be made of those files changed or updated since the last time a backup was made. Since program files are rarely changed or updated, when this option is selected mostly data files will be archived.

Note: It is recommended that at least three sets of archives be kept at hand for the purpose of archiving a minimum of three times per week. Archives could be made on Monday, Wednesday, and Saturday. Rotate the archives, if you like, to conserve the use of floppy disks or tapes. For example, if you have three archives from the previous week's work, when you archive on Monday of the new week, use the diskettes from the previous Monday's archive.

The screen displays and choices in option two are similar to option one except instead of being asked which drive to archive from, you are asked to:

Please type the date of your last ARCHIVE in the format MM/DD/YY:

The input is read by the EXEC and automatically determines which files are to be included for archiving. Only files dated on or after the date entered are archived.

Choice three is provided for those users who are required to backup their own data files. When this option is selected, only those files belonging to the currently logged account will be selected for archiving. Remember that archives must be made in single-user mode.

This means that in order for a user to archive his own files, the procedure must be executed from the main terminal.

Errors During The Archiving Procedure

When errors occur during the archive process the following message is displayed:

THE SYSTEM HAS DETECTED AN ERROR!!
If you wish to continue the ARCHIVE procedure, you must start over and retry the operation. Type YES FOR A RETRY, anything else to return to the System Utilities Menu.

In order to re-try the archive procedure, you must type YES. Anything else automatically returns you to the utility menu. Before a second archive is attempted, try using a different set of floppy disks. The first set might be defective.

OPTION 3: RESTORE/VERIFY FLOPPIES OR TAPE

In THEOS the RESTORE command is the mirror image of the ARCHIVE command. The purpose of this option is to restore files to the disk that have been previously archived and to verify the integrity of files just archived. Restoring files may be necessary to replace data files that have become corrupted or damaged. When this option is selected the following screen is displayed:

In order to RESTORE, you must be in SINGLE-USER mode. In addition, you must have your ARCHIVED VOLUMES (EITHER TAPES OR FLOPPIES) ready to insert into the tape (or disk drive) when prompted to do so. If you need to bring the system down to SINGLE-USER mode, please choose the option to return to the System Utilities Menu.

1. RETURN TO SYSTEM UTILITIES MENU
2. CONTINUE WITH RESTORE PROCEDURES

What is your Choice?:

If one is selected, you are returned to the utility menu. If two is selected the following is displayed:

RESTORE FROM [T]APE OR [F]LOPPY?:

When F is selected the next screen is displayed:

You will now be able to specify HOW you wish to restore. The options should be self-explanatory. The RESTORE/VERIFY ARCHIVE option is used to VERIFY the integrity of the ARCHIVE you just made. If you have a multi-volume archive, this procedure will take a few minutes.

1. COMPLETE RESTORE
2. RESTORE SELECTED FILES
3. RESTORE ONLY YOUR ACCOUNT
4. RESTORE/VERIFY ARCHIVE

What Is Your Choice?:

The menus and choices of this option are essentially the same as for the archive option. For this reason, choices one through three on the restore menu should be self explanatory.

Choice four permits the integrity of files just archived to be verified. This choice is selected immediately after an archive is made and should provide you some peace of mind. If this choice is executed successfully, the files just archived are intact and are not damaged or corrupted in any way.

Note: The archive and restore menu selections should not be executed if disk space misallocations exist. If there are misallocations, they must be corrected first. Report them to the system manager immediately.

OPTION 4: ACCOUNT MAINTENANCE

When this option is selected the user must be logged onto the SYSTEM account and have a THEOS privilege level of five. If the user is not logged onto the SYSTEM account the message: "Must be run from SYSTEM account" is displayed. The purpose of this option is to allow the adding, changing, and deleting of user accounts. When selected, the following screen is displayed:

This option allows you to perform the following functions:

Add An Account.

Change An Existing Account (Passwords, Privilege Level, etc.)

Delete An Account. ***WARNING*** Please do not choose this option without contacting the SYSTEM MANAGER. Once accounts are deleted, you may NEVER again access the files belonging to that account.

USER ACCOUNT MAINTENANCE OPTIONS:

1. LIST CURRENT ACCOUNTS (PASSWORDS, etc.)
2. ADD/CHANGE/DELETE ACCOUNTS
3. RETURN TO UTILITY MENU

ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN:

When choice one is selected the contents of the SYSTEM.ACCOUNT file are displayed so that the user can view the current accounting structure with regard to account synonyms, passwords, and privilege levels.

Choice two enters the user in the interactive mode of account maintenance where accounts can be added, changed, or deleted. THEOS displays:

Mode (Add/Change/Delete/End)?

After all changes to the accounting file have been made, the user selects E for end, and all changes become part of the SYSTEM.ACCOUNT file. Next, the screen is cleared and the user is asked:

CREATE EXEC TO ALLOW USER TO ACCESS MENUS? Y/N

If a response of N is entered, the user is returned to the account menu. If a response of Y is entered, the following message is displayed:

TYPE THE NAME OF THE ACCOUNT JUST ENTERED:

Enter the name of the account. Upon entry an EXEC file is created for the new user account so that all menus can be accessed. In the event that more EXECs need to be created, the following message is displayed:

ANY MORE NEW ACCOUNTS? Y/N:

If no more new accounts are to be added, the user is returned to the account menu. If more EXECs need to be created, the user is returned to the "CREATE EXEC..." prompt.

Choice three returns the user to the system utilities menu.

OPTION 5: REPORT MISALLOCATION STATUS

Misallocations are errors in allocation of disk space caused by improper file closure, static electricity, power failures, and other causes unknown. Whatever the cause, the misallocations must be corrected immediately. The function of this option is not to correct the misallocation but rather report the file(s) causing the problem. The procedure for correcting misallocated disk space is described in detail in chapter six.

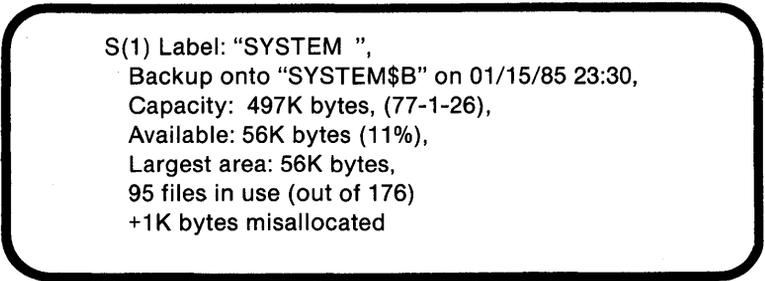
When selected this option's functions are:

- execute a SHOW DISK command to report any misallocations
- allow selection of the drive to be repaired
- report the existence of the file(s) causing the misallocation

Upon execution the message:

Now displaying disk space...

is displayed and a SHOW DISK command executed.



```
S(1) Label: "SYSTEM ",  
Backup onto "SYSTEM$B" on 01/15/85 23:30,  
Capacity: 497K bytes, (77-1-26),  
Available: 56K bytes (11%),  
Largest area: 56K bytes,  
95 files in use (out of 176)  
+1K bytes misallocated
```

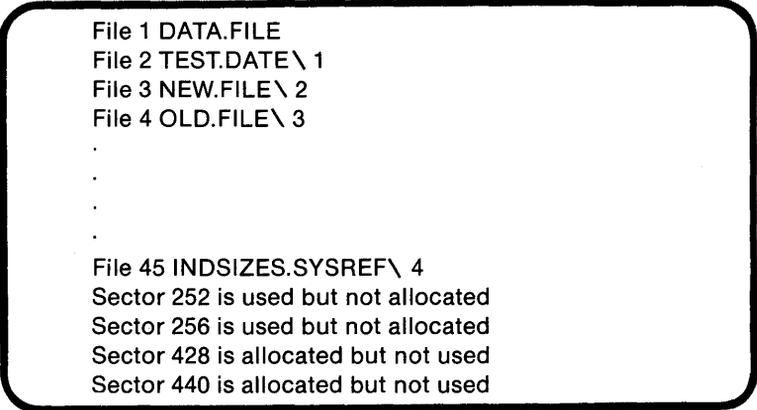
Figure 7.5. SHOW DISK display

This display will show the status of all currently attached disks. Note the disk letter with the misallocation. In this example disk S has a misallocation.

The up-arrow (↑) in the lower left-hand corner of the screen indicates more information to follow so press <RETURN>. You are asked:

```
DO YOU SHOW ANY MISALLOCATIONS ?  
TYPE Y OR N (YES OR NO) AND PRESS RETURN:
```

If you respond with N you are returned to the utility menu. If you respond with Y an information message is displayed and the name of the drive to be repaired is requested. The program asks several times if it is OK to continue. If OK, then the repair process begins.



```
File 1 DATA.FILE  
File 2 TEST.DATE \ 1  
File 3 NEW.FILE \ 2  
File 4 OLD.FILE \ 3  
.  
.  
.  
.  
File 45 INDSIZES.SYSREF \ 4  
Sector 252 is used but not allocated  
Sector 256 is used but not allocated  
Sector 428 is allocated but not used  
Sector 440 is allocated but not used
```

Figure 7.6. Screen display of the disk repair process

Each file in the directory beginning with 1 through the total number of files on the disk is displayed.

After the backslash character (\) the owning account id number is displayed. As the repair process progresses it may become necessary to temporarily stop execution so that the file descriptions of damaged files may be noted. Press ESC S (on some terminals ESC ESC S) to stop. Press ESC S again to re-start. It is recommended that the entire

disk be examined. On disks containing a large number of files, this process may require five to ten minutes during which time the screen must be carefully observed. When finished the following message is displayed:

If this program has reported to you that there is space or files on the disk that are unaccounted for, the recommended method of fixing the problem is to copy (not backup) the files from the disk with the error to an empty disk (such as an initialized floppy). Then, go back and use the KILL command to remove the bad file from the disk. You will now be given an opportunity to go back to look again at your disk space. If you are STILL showing misallocations, it is best to follow the procedure above (ie return to utility menu and copy files).

After this message is displayed, a SHOW DISK command is re-executed and you are returned to the "DO YOU SHOW ANY MISALLOCATIONS?" prompt. Since the misallocation has not yet been corrected, answer N and return to the utility menu. Follow the procedures outlined in chapter six to correct the misallocation.

OPTION 6: FORMAT FLOPPIES/TAPE

Before data can be written on floppy disk or tapes, these must first be initialized or formatted. When this option is selected the INITDISK and INITTAPE commands are automatically executed. In order for these commands to be executed, a floppy disk drive or tape drive must be attached. This EXEC performs that function automatically. Floppy disk G and TAPE1 are the designated drives. Figure 7.7 illustrates the floppy disk formatting process. Be certain to insert a floppy disk before option one from the format disks menu is executed.

This method of formatting does not allow the use of any type of floppy disk other than a double-sided double density disk. In addition

This program will initialize floppy disks/tapes for making backup copies of your data. Please insert a blank diskette into the disk drive with the label side up. Make sure that the write-protect tab is over the hole at the bottom. Otherwise, you will have to start over. If you are using tape please insert the tape into the tape drive.

YOU MAY NOW CHOSE YOUR OPTIONS:

1. INITIALIZE FLOPPIES
2. INITIALIZE TAPES
3. QUIT

ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN:1

INITDISK will erase all files on drive G (1)

Do you wish to continue? (Y/N)? Y

Enter disk label: TEST

Number of directory entries=176

Increment between adjacent sectors=2

Offset between adjacent tracks=0

Number of surfaces=2

Number of tracks per surface=77

Number of sectors per track=26

Double Density, Double Side

Is the above information correct (Y/N)? Y

Figure 7.7. Display of floppy disk formatting procedure

no flexibility is permitted as far as entering a disk label. Every floppy disk formatted will have the same label. To provide this flexibility, the system manager will have to modify this EXEC.

Note that the user is not asked for any input until the following question is displayed:

DO YOU WISH TO FORMAT MORE FLOPPIES? Y/N:

When a response of N is entered in response to the "DO YOU WISH TO FORMAT MORE FLOPPIES" question, you are returned to the utility menu.

OPTION 7: SINGLE USER MODE

Upon selection this option automatically converts the system to single user operation. The following screen is displayed upon execution:

This option will bring the system down to single-user operation for purposes of making ARCHIVE copies and any other system functions where it is NECESSARY to be in single-user mode. In order to bring the system down to single-user mode, all users MUST BE LOGGED OFF. If this is not the case, you will see "partition still active" messages on the screen as the system attempts to come down to single-user mode. If you have not succeeded in bringing the system down to single-user you will have to see to it that EVERYONE has logged off, and re-select the SINGLE USER option from the utilities menu.

Next, the currently logged users are displayed with the following message:

If you show anyone on the system OTHER THAN System or BG #6 (print spooler), then you must choose N below.

HAVE ALL USERS BEEN NOTIFIED FOR SINGLE USER MODE? (Y or N)

If Y is entered then an attempt is made to start single-user operation. If N is entered, the message:

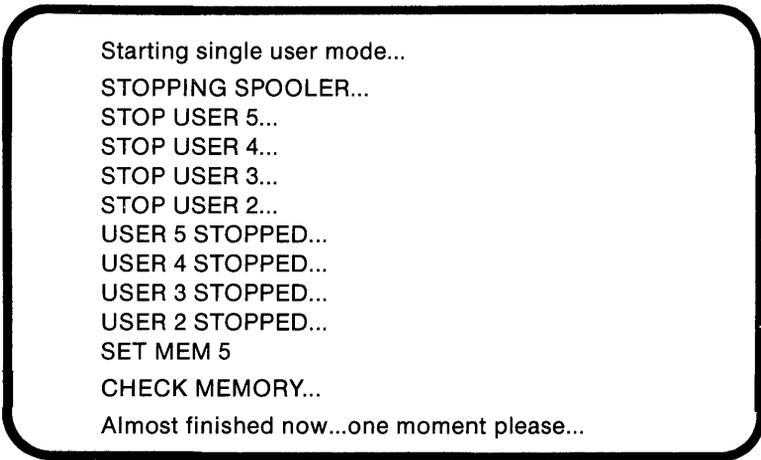
* WARNING * - SINGLE USER MODE WITHIN 5 MINUTES!!!!!!!!!!!!

is displayed on the screen to warn the user to finish the current task and logoff the system.

When single-user mode processing begins the message:

```
PROCESSING SINGLE USER MODE!!!!!!!!!!
```

is displayed on the screen followed by a series of messages as shown in figure 7.8.



```
Starting single user mode...
STOPPING SPOOLER...
STOP USER 5...
STOP USER 4...
STOP USER 3...
STOP USER 2...
USER 5 STOPPED...
USER 4 STOPPED...
USER 3 STOPPED...
USER 2 STOPPED...
SET MEM 5
CHECK MEMORY...
Almost finished now...one moment please...
```

Figure 7.8. Single-user mode screen messages

Other messages may appear between those shown in figure 7.8 such as “Spooler not initialized” or “User is not logged on.” These messages are provided by THEOS to inform you of the system status.

After successful execution one more question remains to be answered:

```
CONTINUE [S]INGLE USER MODE OR [M]ULTI USER WITHOUT SPOOLER?:
```

If S is entered, the system remains in single-user mode and you are returned to the utility menu. If M is entered, the system is set up in multi-user mode without the use of the spooler.

OPTION 8: MULTI-USER MODE

This option is selected when single-user mode is no longer required, such as after a successful archive or restore operation. When

selected, the IPL.EXEC is re-executed and the screen display is the same as for the IPL.EXEC described earlier. After execution, the "Logon please:" message is displayed.

OPTION 9: MULTI-USER MODE WITHOUT SPOOLER

When this option is selected the following screen is displayed:

This function is useful when the System Manager needs to print out long reports, such as monthly statements, that may tie up the Spooler for long periods of time. However, before the system can be operated WITHOUT the spooler, the system must be brought down to Single User Mode and then brought up in Multi User WITHOUT the spooler.

The up-arrow (↑) is displayed in the lower left-hand corner of the screen. When RETURN is pressed, the user is taken to the single-user screen. After the system has been brought down to single-user mode, select M when presented with the prompt:

CONTINUE [S]INGLE USER MODE OR [M]ULTI USER WITHOUT SPOOLER?:

After M is selected the following message is displayed:

PROCESSING MULTI-USER WITHOUT SPOOLER!!!!!!

Screen messages are displayed in a manner similar to the IPL.EXEC except for the fact that the spooler is not activated. When this option has been successfully executed, the user is returned to the utility menu.

OPTION 10: ERASE FILES

As more and more sequential data files are created with word processing and spreadsheet programs, it may become necessary to purge the system of old files no longer being used. The erase files option provides a quick and easy means of erasing files. Upon selection the following screen is displayed:

This program will allow you to erase files from your hard disk that are no longer useful to you...such as old data files. Be certain that the files no longer have any use to you because once you ERASE them, they CANNOT be recovered. If you are unsure about erasing a particular file, don't erase it. In order to erase a file, you must SPECIFY BOTH the

FILENAME
AND
FILETYPE

and specify the DRIVE which the file resides on.

After the RETURN key is pressed, the following message is displayed:

WOULD YOU LIKE TO LIST YOUR FILES? Y/N :

If Y is entered the user is prompted to first enter the filename, file type, and drive where the file is located. If the location of the file is uncertain, RETURN can be pressed and all attached disks are searched. All files belonging to the currently logged account will be listed. Files to be erased can be noted. Before continuing the user is asked if more files are to be listed.

An entry of "N" indicates that the user already knows the file names and file types of the files to be erased. Even if the exact names are not known, wildcard characters (*, , #, etc.) can be used. Each file matching description entered is displayed and THEOS asks if it is OK to erase particular file. Remember that each sequential file created has that corresponding file with a file type of BACKUP. To clean up the disk, backup files should be erased as well.

When a response of Y is entered, the screen is cleared and the following messages displayed:

FILENAME TO ERASE:
FILETYPE TO ERASE:
DRIVE WHICH FILE RESIDES ON:

Responses to prompts are required; otherwise you will not be permitted to continue. When entries have been completed the message,

FILENAME FILETYPE WILL BE ERASED
FROM DRIVE DRIVNAME

is displayed in the center of the screen. The following message:

[E] TO ERASE FILES [S] TO
SELECT NEW FILES OR [Q] TO QUIT:

is displayed at the bottom of the screen to permit new entries to be made or to allow you to exit the operation.

Selecting E allows the file erasure procedure to continue. Selecting S allows you to select different files to be erased. Selecting Q allows the file erasure procedure to be aborted returning you to the utility menu. If E is selected the system displays,

Please wait...

FILENAME FILETYPE DRIVNAME Ok to Erase (Y/N)?
FILENAME FILETYPE DRIVNAME Ok to Erase (Y/N)?

until all files matching the description entered have been displayed. You must respond to each prompt with Y or N to continue. When N is entered, THEOS ignores that file and moves on to the next one matching the specified description. No files will be erased unless a response of Y has been entered. When all files matching the specified description have been erased (or ignored) the following message is displayed:

MORE FILES TO ERASE? Y/N:

If no more files are to be erased, you are returned to the menu. If more files are to be erased, the EXEC allows new files to be selected.

OPTION 11: LISTING FILES

This option allows you to list the files belonging to the currently logged account. This list can be used to examine the status of the number of files belonging to the currently logged account. If it is noted that a number of files are no longer being used, a hard copy of the list may be printed and option ten may be selected to erase files no longer needed. To obtain a list of files select option eleven; the following is displayed:

FILENAME TO LIST:
FILETYPE TO LIST:
DRIVE WHICH FILE RESIDES ON? RETURN IF NOT SURE:
[H]ARD COPY OR [S]CREEN?:

Entries must be made before continuing. Wildcards can be used in place of specific file names. If the drive where the file is located is unknown, the system allows the RETURN key to be pressed. All attached drives are then searched. The list of files may be printed on the main printer (hard copy) or on the terminal [S]creen.

After all files matching the description entered have been displayed, the following message is displayed:

DO YOU HAVE MORE FILES TO LIST? Y/N:

If N is entered, you are returned to the system utilities menu. If Y is entered, you are prompted to enter new file descriptions.

OPTION 12: RETURN TO MASTER MENU

When you've finished with the system utilities menu, option twelve is selected to return to the master menu.

OPTION 13: LOGOFF SYSTEM

This option is selected to quit for the day or to log on to a new account. In either case when chosen the following messages are displayed:

Have you remembered to make backups of today's work?
Type M to return to MENU or L to LOGOFF the system:

Entering M returns you to the utilities menu where the daily backups may be performed. Entering L logs you off the system and displays the "Logon please." message. At this point either the system can be shut down, or you may log on to a new account.

The System Cancel Key

On rare occasions it may be necessary to use the system cancel key.

One place where its use may be suggested is in the event of an error message displayed on an unfamiliar screen. When the sequence ESC Q (on some terminals ESC ESC Q) is entered, THEOS displays:

CANCEL EXEC (Y/N)

If a response of Y is entered, the user is returned to the THEOS prompt and EXEC execution terminates. If a response of N is entered, EXEC execution should continue. However, the result of an N response can be unpredictable and entering N is not recommended in most cases.

Summary

This chapter has described the use of EXEC programs in a turn-key THEOS installation. Many “computer experts” take a dim view of isolating the user from the operating system. They feel that the user should have an “intimate” relationship with the computer and its operating system. The fact of the matter is that most business users care little about how the operating system or computer works. They are interested in executing business applications without days and weeks of intensive training. THEOS business application software should include EXECs, such as these discussed here, that can simplify their installation and operation.

8

A User's Introduction To THEOS 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, mail 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 small multi-user computer may suit your needs. You may 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 included about the strengths and weaknesses of specific software packages.

As with other microcomputer operating systems a wide variety of THEOS application software exists for almost any application imaginable. Unfortunately, the quality ranges from amateurish to professional. As this book is not primarily intended to provide evaluations of software, discussions of each software category will be brief. Where possible comments on software will be based on actual hands-on use.

Why You Probably Haven't Heard of These Software Packages

The THEOS operating system was designed to be a multi-user operating system. Most software packages that are heavily advertised and promoted, such as dBase and Lotus 1-2-3™, were written for use with Apple DOS™, CP/M™, or MS-DOS™, all of which are single-user operating systems. Generally it would be impractical to take existing single-user software and make it THEOS compatible. To take advantage of THEOS, it is 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 THEOS.

Some software vendors make demonstration packages available in order to allow a user to examine the capability of the software. These demos range in price from \$10.00 to \$100.00. The demos perform identically to a fully functioning package except that the file size is limited to only a few records. Most THEOS software is *serialized* to each copy of the THEOS operating system. This means that when purchasing software, the vendor requires the serial number of your operating system. This practice helps to discourage software piracy. You can make as many backup copies of the software as you like but the software will only work on the THEOS operating system with your serial number. If an attempt is made to run the software on another THEOS machine, the message "Incorrect serial number" is displayed.

THREE TYPES OF SOFTWARE EVERYONE CAN USE

Word Processors

Word processing software allows you to type mailing lists, letters, 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. There are two excellent word processing packages for THEOS: Spellbinder and multiWRITE. Although the purpose of the two programs is identical, the operation and features of each differ. Depending upon your needs, you may find it necessary to utilize both programs.

Note: Word processing software may sometimes require lengthy disk access time 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 or three users perform word processing tasks simultaneously.

SPELLBINDER

This is one of the few THEOS programs which is also available for other operating systems. The programmers have done a commendable job of adapting it to the THEOS environment. Some of the features of Spellbinder™ include:

- built-in mail merge facility
- ability to print mailing labels, including multiple copies of each label
- forms fill-in capability

- smaller documents can be combined into larger ones
- row and column calculations can be performed within a document
- documents can be printed to the screen as they appear on paper
- documents wider than eighty characters can be edited
- documents larger than memory capacity can be edited
- utilization of the THEOS spooler
- backup files are automatically stored on disk for each file edited
- documents can be edited and printed without being saved on disk

Perhaps the most important feature of Spellbinder is that it is easy to learn. A tutorial manual is supplied with the program in addition to the regular documentation. Working through the tutorial and practicing with the examples is recommended. Bear in mind, however, that all Spellbinder™ documentation is written with the CP/M user in mind. Rather than write new documentation specifically for THEOS the marketers of Spellbinder chose to use their existing CP/M documentation, supplementing it with three or four pages noting the differences between CP/M and THEOS terminology and conventions. When working through each example, remember to use THEOS formats when entering program commands instead of the CP/M formats provided. You might consider going through the documentation with a pencil in hand, substituting THEOS conventions for the CP/M conventions provided.

It is the system manager's responsibility to configure Spellbinder to work with the THEOS system. If improperly configured, lines of text will not be displayed on the screen properly. As supplied, the program is designed to work with the most popular terminals such as Televideo. If you are not using Televideo terminals, the configuration must be changed using the supplied program called "CONFIG."

As supplied, it is not possible to use the function keys on most terminals to execute Spellbinder commands. Instead *control key* sequences are used. For example to toggle between EDIT and COMMAND mode, the sequence Ctrl Q is used. To generate Ctrl Q, press the letter Q while holding down the key labelled Ctrl.

The editing portion of the program is separate from that portion which executes commands. While editing, it is possible to insert, use tabs, backspace and correct errors, perform deletions, move across, up or down through the document. When commands are executed, such as printing or storing a file on disk, the command mode must be entered.

Although it is possible to work with documents larger than memory, Spellbinder's method of "swapping" text from memory to the disk is somewhat less than convenient, leaving much room for operator error. When an edited file is larger than memory, a command must be issued to save that portion of the file to disk, clearing the memory for more text. Unfortunately, once the text is written to disk, it cannot be brought back into memory without issuing a command to close the file and start over by re-loading the edited file into memory.

The more powerful features of the program are executed using *macros*. Macros are special programs supplied with and executed within the Spellbinder program which permit mail list merging, forms fill-in, on-screen calculations, sorting lists, etc. Mastering the use of the macros is time consuming and requires considerable practice, primarily due to the limited amount of documentation supplied for them

COMMENTS

In summary, Spellbinder is a very flexible word processing program that can transform a THEOS computer into a powerful word processing tool. An inexperienced user need spend only a few hours with the program to master the basics. The main weakness of the program are its documentation and method of handling large files. In addition, some other word processing programs now offer the convenience of built-in spelling checkers.

Program: Spellbinder

Manufacturer: Lexisoft, Inc. Box 1378 Davis, CA 916/758-3630

User support available: by telephone at user's expense.

Price: \$550

Distribution medium: eight inch floppy disk

Special requirements: minimum of 44K of memory is required to run the program. This leaves 32K for the program itself and 12K for the document being edited.

multiWRITE™

This program was written specifically for the THEOS operating system. It features:

- menu-driven operation
- edit/command mode similar to Spellbinder
- mail list merging
- spelling checker with 80,000 word Random House Dictionary
- 30,000 word Random House thesaurus
- documents larger than memory can be edited easily
- special configuration files for a variety of terminals and printers
- on-line help facility available at the touch of a key covers many program features
- a built-in program to convert Spellbinder formatted files to multiWRITE files
- automatic control of THEOS spooler functions

Unlike Spellbinder, if you happen to own a brand of terminal that is supported by multiWRITE, it is possible to configure the function keys on that terminal so that they may be utilized to execute certain commands such as insert, delete a word, indent, screen up, screen down, etc. Depending upon the location of the function keys on your keyboard and whether or not you have become used to using Ctrl key sequences as most Spellbinder users have, you may or may not find this a useful feature. If you have become accustomed to the Ctrl key sequences of Spellbinder, it is possible to configure multiWRITE to use those same sequences to execute commands.

Upon entering the program the user is presented with a first choices menu. The menu choices are:

- select an existing document
- create a new document
- use directory functions
- how to use multiWRITE
- return to THEOS

A sample letter is provided with the program which is useful for examining how lines of text can be typed and formatted. As with Spellbinder, formatting commands can be embedded in the text to supplement the global formatting commands selected just before printing.

When a document is created you are asked for the its name, the drive where the document is to be located, and a short description. This information is used for the multiWRITE file directory. MultiWrite maintains its own directory of document files which can be viewed later. Spellbinder requires some knowledge of THEOS commands because it does not maintain its own directory of word processing files. MultiWrite requires virtually no knowledge of THEOS commands.

In multiWRITE, as an edited document grows in size, the program automatically writes it to disk when working memory is exceeded. The size of a document is limited only by disk capacity. Because of the way multiWRITE handles files larger than memory, it is possible to edit any file from beginning to end without re-loading.

Since the THEOS print spooler is important in a multi-user environment, it is equally important that the software use the spooler correctly. MultiWrite does just that. No THEOS commands are required to use the spooler. The program can handle printing multiple copies of a document and can also terminate all printing with a few keystrokes. During printing the number of the copy being printed as well as the page number is displayed. The program always tries to keep the user well informed.

COMMENTS

MultiWRITE is both easy to use and powerful. It is the only THEOS word processor that is supplied with a spelling checker. It is the only microcomputer word processor with an on-line thesaurus. The documentation is nicely packaged in a binder and is typeset. It is primarily a reference guide. Multiwrite is now supplied with a task-oriented tutorial.

A number of features are documented but not thoroughly explained. These include:

- Tabs must be set before use (no built-in defaults).
- To indent you must specify how much to indent (no built-in defaults).
- Depending on your hardware configuration you may not be able to use the thesaurus in multi-user mode with the spooler.
- A minimum of 44K of memory is required.
- When the dictionary is compiled a backup dictionary file is automatically created which occupies 1/2 megabyte of disk space.

MultiWRITE lacks the macros included in Spellbinder. If you require sorting, forms fill-in, and extensive mail list management capabilities, you may find you need both Spellbinder and multiWRITE.

Program: multiWRITE

Manufacturer: Logic Extensions Resources
9651 Business Center Drive
Rancho Cucamonga, CA 91730
714/980-0046

User support available: by telephone at user's expense

Price: \$695 complete with spelling checker and thesaurus

Distribution medium: eight inch floppy disk; 5¼ inch floppy disk

Special requirements: minimum of 44K working memory

Note: Working with documents larger than ten pages is cumbersome with either word processing program. To improve speed and ease of editing, try working with smaller document files and combining them later if necessary. This maximizes productivity by minimizing slowdown of the computer.

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 THEOS 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 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 THEOS spreadsheets: Scratchpad 3.1 and MultiCALC 2.0.

SCRATCHPAD 3.1 AND multiCALC 2.0™

These programs will be discussed together as they are quite similar. Both are fine programs which accomplish the same goals, but they differ slightly in features, screen displays, and operation. The programs are designed along the lines of popular spreadsheet programs such as Supercalc or Multiplan. Scratchpad seems to resemble Supercalc while multiCALC resembles Multiplan. They both use the / (slash) key plus the first letter of the command to execute a command. On-screen help is available for both programs. Scratchpad provides general help messages about the function of various commands while multiCALC provides specific help messages about the commands being executed whenever the ? key is pressed. Both program's basic features are quite similar. These are listed below:

- Worksheet size is memory independent.
- Horizontal and vertical "windows" with full synchronization are available.
- Formulas over 100 characters in length are possible.
- Individually defined column widths are possible.
- In-cell editor allows changes without re-typing.
- Spreadsheets can be merged.
- 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, LOOPUP, MAX, MIN, PI, SUM, SQR, SIN, COS, TAN, ARCTAN, LOG, EXP, FAC, NPN, +, -, *, /, ^ (exponentiation), <, <=, >, =, >, <>.
- Both programs can use European number formats.
- Control of printing functions is available with automatic pagination, column re-titling, etc.
- Formatted 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 Scratchpad is the built-in templates for:

- Internal rate of return
- Amortization
- Future value
- Regular deposits
- Regular payments

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 Scratchpad keeps video attributes such as reverse video, half intensity, etc. to a minimum, that program requires no special terminal configuration program. Because multiCALC does make extensive use of video attributes, it is supplied with various terminal configuration utilities. In addition multiCALC is able to support some of the newer color display terminals.

MultiCalc+ is a special version of multiCALC containing an additional set of commands for operation with a hard copy graph/chart plotter. User prompts are provided for controlling graph type (pie, stacked bar, comparison bar, line drawing, user label); graph size; pen hatching; and pen colors.

Comments

Both spreadsheet programs perform adequately, and it would be difficult to recommend one program over the other. The documentation provided with both programs is adequate but more material could be provided for the hands-on tutorial approach. The templates provided with Scratchpad are certainly a step in that direction.

Both companies provide user support. Business Information Technologies has diversified into other operating systems. Logic Extension Resources is 100% committed to THEOS.

Program: Scratchpad

Manufacturer: Business Information Technologies, Inc.
5 Burlington Square P.O. Box 426
Burlington, VT 05402
802/658-5600 contact Chris Pratt

User support available: by telephone at user's expense

Price: \$395.00

Distribution medium: 8 inch floppy disk,
5¼ inch floppy disk

Special requirements: any THEOS based system

Program: multiCALC

Manufacturer: Logic Extension Resources
9651 Business Center Drive
Rancho Cucamonga, CA 91730
714/980-0046 contact Rhett Tindall

User support available: by telephone at user's expense

Price: \$495 mutliCALC+ \$695

Distribution medium: 8 inch floppy disk
5¼ inch floppy disk

Special requirements: any THEOS based system

Database Managers

Database management software for THEOS permits users to design a customized data entry program tailored to their 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. Comprehensive documentation is an important ingredient of a good database management program. The best database managers use tutorials provided with pre-constructed 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, records, and fields. First the file is defined. Next the number of records the file is to contain is defined. Think of a file drawer...the number of records is the number of file folders you wish it to hold. Finally the fields within each record are defined. The fields are the individual data items comprising each record.

For example, if you were creating a database for mailing labels you might create a file called "mail." You might define that file to store one hundred records. The fields making up each record could be:

Name	Address	City	State	Zip
------	---------	------	-------	-----

After data is entered, it is possible to sort the file alphabetically or numerically. When sorting is completed, a report could be designed called "mail label." The report format might resemble the following:

Name
Address
City State Zip Code

Because we are using a hard disk based system, there are few limitations in constructing a database. The maximum number of records that can be stored in a database is 30,000 to 65,000, depending upon the software. We will examine two database programs: UserBase and Control-8.

UserBase

Userbase is a database program designed specifically for THEOS which is ideally suited for use in a multi-user environment. No special programming experience is required to use the program. Help is available at each step by entering /HELP. It is also possible to *back up* to a previous step or exit the program altogether by pressing the ESC key twice.

As an instructional tool, a special program called INTRO can be run which aids the first-time user in understanding the UserBase approach to information management. The INTRO program describes how to use each UserBase module as well as certain keyboard conventions.

UserBase Menu Aid

Copyright (c) 1983 UserWare International

BUILDING A DATABASE

- 1 Allocating disk space (ALLOCATE)
- 2 Defining your database (DEFINE)

ENTERING DATA (single database)

- 3 Adding data (MAINTAIN, ADD)
- 4 Modifying data (MAINTAIN, MOD *)
- 5 Locating data (MAINTAIN, LOC *)
- 6 Inquiring data (MAINTAIN, INQ *)

SORTING AND SELECTING

- 7 Sort and select data (SORT)
- 8 Flagging records (POINTER)

DEVELOPING REPORTS

- 9 Writing a report (REPORT **)
- 10 Listing a database (DUMP **)

ALTERING DATA

- 11 Transferring data (TRANSFER**)

SYSTEM EXIT

- 98 Exit to THEOS
- 99 Logoff

* indicates SORT may be used first

** indicates SORT must be used first

Enter option number:

(Enter /HELP for assistance)

Figure 8.1. The UserBase Menu

The most frequently used program modules can be selected from the UserBase menu, shown in figure 8.1. This menu lists some of the UserBase modules along with a brief description of their function. Each module is numbered in the order used when a new database is created.

When creating a new database with UserBase, follow the steps listed below:

- ALLOCATE disk space.
- DEFINE the data fields.
- MAINTAIN the database.
- SORT the data file.
- Design and print REPORTs according to your specifications.

After a database is created and the MAINTAIN program invoked, the screen does not display fields as they might appear on a form. You can use the FORM module to create a screen which mimics a form you are presently using. The FORM module allows you to construct screens. If your form is too large to be accommodated by a single screen, multiple screens are possible. In this case, instead of using the MAINTAIN module to add or inquire about data, the SCREEN module is invoked.

The FORM and SCREEN modules are preferable to MAINTAIN if data security is important. These modules interact with THEOS privilege levels. Depending upon his privilege level, a user may be permitted to add, modify, delete, or view records in the file. The user with the lowest privilege level will only be able to view records. These modules also permit on-screen calculations within fields.

Most data files created by UserBase will be THEOS indexed files resulting in almost instantaneous record access.

Since data is entered in a random fashion, the ability to sort a data file is critical. Through a simple question and answer dialog, complex sorting criteria can be specified.

REPORT is the UserBase module that constructs customized formatted reports. When invoked the user is prompted through a series of questions which will construct the desired report format. For convenience, the report format can be stored on disk and recalled at will, saving considerable time. The REPORT module makes excellent use of the THEOS spooler. It displays job status statistics during execution, which notify the user that work is progressing as well as the approximate time required to complete the job.

TRANSFER can be used to construct a new database from an old one. TRANSFER is used to move data from one database to another or to effect changes within the same database, such as adding records or fields.

COMMENTS

UserBase's numerous features and ease of use compare favorably with *any* other personal computer database system. UserBase was originally implemented (and is still being sold) for the Digital Equipment Corp. PDP-11 minicomputer under the name USER-11.

A demonstration program is available for \$100.00. The \$100.00 can be credited towards the purchase of the complete package within thirty days of purchase. The demo includes complete documentation and a tutorial that takes you step-by-step through the basics of using the package. UserBase can also be used as an application software generator. Contact UserWare for a list of application packages written with UserBase.

Programs: UserBase

Manufacturer: UseWare International
2235 Meyers Avenue
Escondido, CA 92025-1070
619/741-8825 contact Rich Baldwin

User support available: by telephone at user's expense
UserBase training is optionally available
at UserWare facilities in Escondido, CA
CA for extra charge.

Price: \$795

Distribution medium: 8 inch floppy disk
5¼ inch floppy disk or
Onyx cartridge

Special requirements: Any THEOS based system. Terminals should have cursor addressing, reverse video, erase-to-end-of screen, erase-to-end-of-line, and bell capabilities. UserBase consists of over 100 programs and help text files requiring approximately 1 MB of disk space.

Control-8

Control-8 is another THEOS database management system. Control-8 is composed of a series of individual programs plus the ACCESS, SORT, and FORMS modules.

Control-8 allows the user to design the database. After the specifications of the database have been determined, data can be entered. Reports can be constructed using the ACCESS inquiry language, data can be sorted using the SORT module and custom formatted reports can be created using FORMS.

DATABASE DESIGN AND DATA ENTRY USING CONTROL-8

Control-8 is a screen-oriented database, which means that data fields can be positioned anywhere on the screen. The user must define the screen layout. It is usually helpful to determine the screen's appearance before using the program. Some important features of the screen generator are:

- The user can be prompted with on-screen help messages during data entry.
- Video attributes such as reverse video, underline, blinking, etc. can be used to further assist in data input if supported by your terminal.
- Data fields can be edited without re-typing the entire field saving you time.
- Up to nine screen pages of data are possible.
- A database can include as many as 65,535 individual records.
- Data files are indexed resulting in quick access to records.
- Up to 80 characters per field are allowed.

Consider the screen design in figure 8.2.

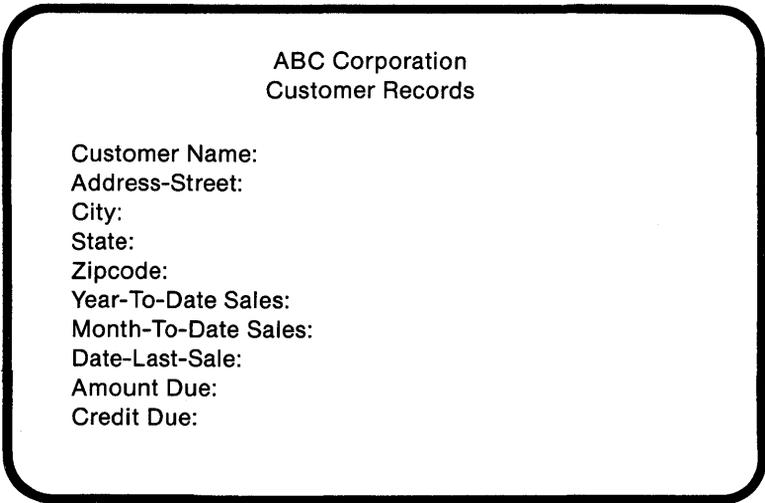


Figure 8.2. Control-8 screen design

This is an example of a single screen database. The purpose of each data field in this example should be self explanatory. You may already be using a form such as this one to keep track of sales in your company.

REPORT GENERATION

Control-8's ACCESS language allows information to be obtained from the database by entering an English-like sentence. These sentences can be used to list selected information from multiple files or from any field of data within a file. ACCESS can be considered the **report writer** feature of the program. Frequently used report formats can be stored on disk and later recalled so that the report can be re-generated without re-entering the ACCESS sentence.

The ACCESS language permits a columnar format report to be output to either the terminal or printer. The report can be printed with page headings, subtotal footers, and column headings. If you require special report formats such as mail labels, specialized forms, or other non-columnar formats, you may need to use the FORMS program. FORMS allows special report formats to be designed easily and works with the ACCESS language to produce customized reports.

Consider the following ACCESS sentence:

```
LIST ALL CUSTOMER NAMES WITH YEAR-TO-DATE SALES GREATER THAN $10,000.
```

The report generated might appear as follows:

Customer Name	Year-To-Date Sales
Ashley	\$12,541
Barnaby	\$34,456
Audio Plus	\$87,567
YSL Records	\$56,098

COMMENTS

The Control-8 package is very powerful and flexible in terms of features, performance, and value for the money. The documentation provided is quite adequate and follows a format similar to the documentation for the THEOS operating system itself. For this reason it is

recommended that the system manager be the one to construct the databases and report formats. Most users will not want to read the THEOS documentation in order to learn Control-8. Once the database is created it is a simple matter to teach someone how to perform data entry. Unlike UserBase, Control-8 does not include menus or a built-in on-screen help facility. It is possible for EXEC programs to be written to permit users to access the Control-8 programs via menus. At the very least when database management is to be performed, it should be available as a choice from the MASTER MENU EXEC.

Whether you choose UserBase or Control, some hands-on training is recommended preferably by the manufacturer or by one of his authorized dealers.

Program: Control-8 Relational Database Management System

Manufacturer: Phase One Systems, Inc.

7700 Edgewater Drive Suite 830

Oakland, CA 94621

415/562-8085 Hildy Langewis

User support available: by telephone at user's expense

Control-8 training is optionally available at Phase One Systems in Oakland, CA. for extra charge.

Price: \$595 for all modules including FORMS. FORMS is available separately for users who already have other modules.

Distribution medium: eight inch floppy disk

Special requirements: any THEOS-based system

General Business Accounting Software

Two options are available to those considering computerizing their business's accounting functions:

- construct your own accounting package using one of the database programs described earlier
- use one of the readily available 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 THEOS 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 THEOS 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.

We will examine two general accounting systems. The first is the Open Systems Accounting System; the second is the Ashley Accounting System.

OPEN SYSTEMS ACCOUNTING SYSTEM

The Open Systems accounting system is written in COBOL as opposed to THEOS BASIC. This means that in order to use the software, the COBOL language must be installed on your system. The software was written to utilize the record locking feature of the THEOS operating system. Record locking allows multiple users to access the same data file but prevents access to records in the file currently in use by another user. Password protection is also provided to prevent unauthorized access to certain reports and portions of the program.

Perhaps the most striking feature of the software is the sheer bulk of documentation provided. The software is available for purchase in modules. Each accounting module is individually boxed and comes with a beautifully typeset and bound installation guide as well as a user's guide. The installation guide is provided to instruct the system manager in the proper installation of the software. Configuration sheets are provided for each module to aid the system manager in data file set-up.

The user's guide provides a separate card which lists the data entry commands. For consistency's sake each user's guide is divided into the following sections:

- General information
- How to use the software module
- File maintenance
- Sample reports
- Accounting summary
- Data security
- System messages
- Index specifying page numbers which explain execution of specific tasks

Each user's guide includes numerous illustrations and screen displays, making it easy to follow the examples provided. Upon entering the Open Systems Accounting System, the menu as shown in figure 8.3 is displayed.

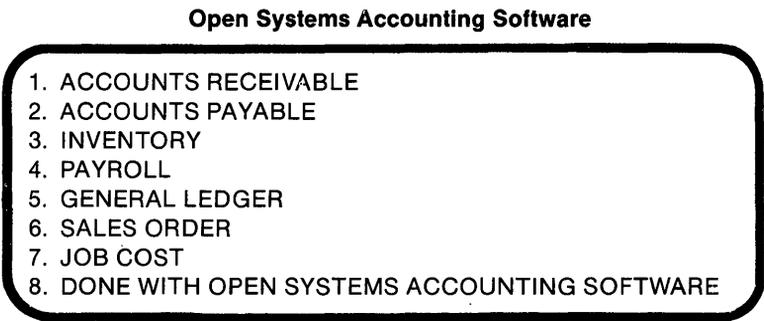


Figure 8.3. Open Systems Accounting main menu

If option three was selected, the main inventory menu would be displayed as shown in figure 8.4.

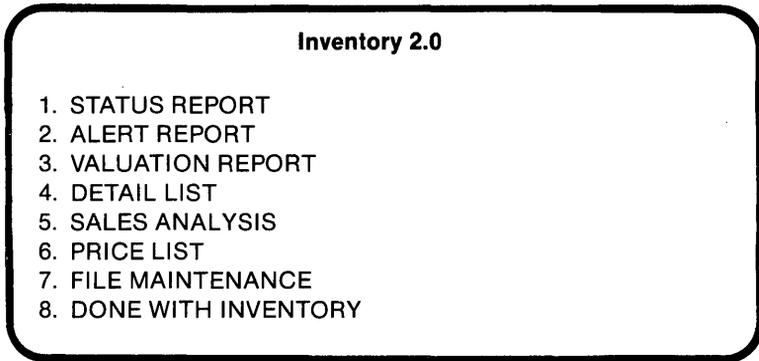


Figure 8.4. Open Systems inventory menu

Selections one through six in this menu deal with the available inventory reports. Selection seven allows you to maintain the inventory file including receipt of items and price changes. After any inventory related function is selected you are returned to this menu. Option eight can be selected to return to the main menu.

The primary benefit of any fully integrated accounting system, such as the Open System package, is efficiency. When an order is entered, the product ordered is subtracted from the inventory file, an accounts receivable transaction is generated and posted which in turn flows through to general ledger. All of this is done automatically with only a single entry.

COMMENTS

Accounting software is by nature far more complex than a word processor or spreadsheet. The programs alone can require over one million characters of disk storage. The Open Systems Software is one of the least complex accounting systems available for THEOS. This could be a blessing or a curse depending upon your needs. The menus are short and simple but so are the data entry screens. If you can adjust the way you do business to accommodate the software, it may be ideal for you. The available report options are fairly limited. It is unfortu-

nate that the Report Writer module is not available for the THEOS version of this software, as it would allow users to construct reports not present as menu selections in each module.

Program: Open System Accounting Software

Manufacturer: Open Systems
430 Oak Grove
Minneapolis, MN 55404
612/870-3515

User support available: Client Support Service with 800 number for annual fee based on the number of software modules purchased. Credit cards are accepted for this service.

Price: \$695 for each module

Distribution medium: eight inch floppy disk

Special requirements: THEOS RM/COBOL language. 64K memory partition for each user. Check with hardware manufacturer.

Ashley Accounting System

The Ashley Accounting System consists of the following modules:

- Accounts receivable
- Accounts payable
- Order entry
- Inventory control
- Point of sale
- Purchase order tracking
- General ledger

The software is written in THEOS BASIC and requires no additional language. The software is composed of over four hundred individual programs requiring four million characters of disk space. For this reason, it is recommended that hardware with a minimum of 20 MB of disk space be used to operate the system. This would allow space for the accounting software, data, and other software such as a word processor or spreadsheet. Unlike the Open Systems software, the documentation for this software is incomplete, though adequate to make use of the software. The Ashley system's appeal lies in its flexibility and ease of operation.

The main menu appears as illustrated in figure 8.5.

The Ashley Accounting System
Control Module
Your Company Name

1. FILE MAINTENANCE	File maintenance module
2. ORDER ENTRY	Order entry module
3. ACCOUNTS RECEIVABLE	Accounts receivable module
4. ACCOUNTS PAYABLE	Accounts payable module
5. INVENTORY CONTROL	Inventory control module
6. SALES ANALYSIS	Sales analysis module
7. GENERAL LEDGER	General ledger module
8. PERIODIC MAINTENANCE	Periodic maintenance module
S. SETUP	General setup information
R. RETURN	Return to system

Please select function _

Figure 8.5. Ashley Accounting System main menu

Only the modules purchased become part of the main menu. The menu shown in figure 8.5 assumes that order entry, receivables, payables, sales analysis, and general ledger were purchased. In order to select a function, only the number of the desired module need to be pressed. It is not necessary to press <RETURN> except during data entry and report selection and generation. In most cases, it is possible to *back up* one step to a previous menu simply by pressing the ESC key twice (three times on certain terminals), or if a menu is present, by pressing <RETURN>.

Each menu choice within each module is numbered. For example if option three, ACCOUNTS RECEIVABLE, were selected, the following menu would be displayed:

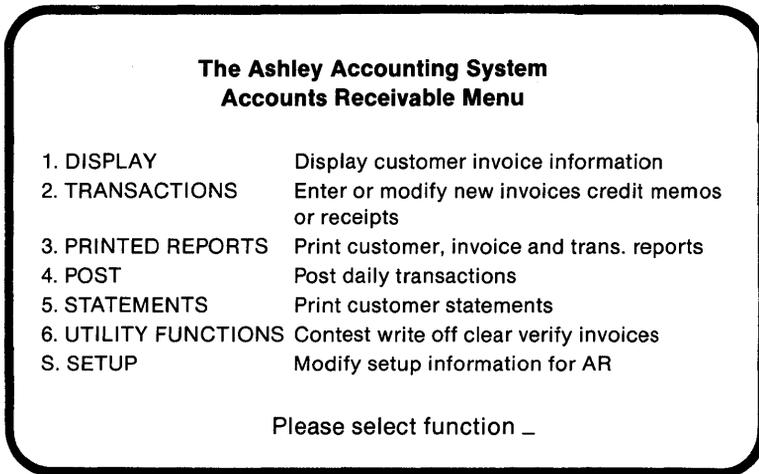


Figure 8.6. Ashley Accounting Accounts Receivable menu

As software gains in complexity, additional menus are required to facilitate its use. It is also true that when software is used daily, the user becomes so familiar with the software that the menus become an obstacle and slow down data entry. Not so with the Ashley system. Once a user knows in advance which menu choices are desired, a series of numbers corresponding to the menu choices can be entered transporting the user to the desired location for immediate data entry. For example, using the menus in figure 8.5 and 8.6, by choosing 3 and 2, you are taken to the accounts receivable transaction menu. The menu illustrated in figure 8.6 is bypassed.

If a number is entered incorrectly when selecting a function, the ESC key can be used to *backup* or the [RETURN] key can be pressed.

On-screen help is available in certain areas of the program, particularly in order entry. Help is available by pressing CTRL A.

Besides ease of use, how good is the software? This software has been used successfully by a variety of retail and wholesale businesses, although it is probably better suited for use in a wholesale business with warehousing facilities. The size of data files is limited only by the disk space and the limitations imposed by THEOS. Five screen pages of information are maintained on each inventory item. The information stored for each inventory item takes up roughly 1K of disk space. A wide variety of pricing schemes are possible as well.

The modules which have been used extensively and most successfully are accounts receivable, accounts payable, order entry, and inventory control. Accounts receivable and payable are essentially mirror images of each other in terms of operation. These modules should be more than adequate for many businesses. While the general ledger is generally sufficient for most businesses, it may not provide enough breakdown of individual expenses or accounts for some. Certain reports, such as the commission report, are not as flexible as they might be because it is not possible to pay commission based only on invoices paid or to pay variable rates of commission depending upon the product category. Passwords can be employed in certain areas of the program to deny unauthorized users access.

COMMENTS

In many ways this software is the opposite of the Open Systems software. It lacks professional documentation yet it can almost be used without it. This software has many built-in features that have not been well documented. Once file sizes have been determined, they can easily be expanded or contracted without losing valuable data. A feature yet to be implemented is a report writer. A report writer would allow users to generate reports not present on any menu.

Program: The Ashley Accounting System

Manufacturer: Ashley Computer Systems
19431 Business Center Drive #27
Northridge, CA 91324
818/885-6800

User support available: Telephone support is available at user's expense. Local software support provided by regional dealer located near end-user.

Price: \$595 for each module.

Distribution medium: eight inch floppy disk.

Special requirements: Minimum 20 MB hard disk system.

Vertical Market Packages

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.

THEOS offers many vertical market packages. Two of these include the Computec Client Accounting System and the Eagle Company Video Management System. We will not discuss vertical packages here as the diversity of these prohibits such an examination.

WHERE TO FIND THEOS SOFTWARE

Software Directories

The multi-user computer market is quite different from the personal computer market. The cost of the least sophisticated THEOS system with software is well over \$7000.00. Not only are most computer stores unable to deal with such high price points, their in-store 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.

THEOS APPLICATION SOFTWARE DIRECTORY

This publication is not strictly a software directory. It is a collection of advertisements for hardware and software, news and commentary about THEOS, questions and answers about THEOS as well as an alphabetical listing of over 1000 vendors of THEOS software worldwide. 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 time.

Some software packages are sold separately while others must be purchased as complete turn-key packages. Unfortunately, the directory doesn't indicate this fact.

The publication is available directly from:

Phase One Systems, Inc.
7700 Edgewater Drive Suite 830
Oakland, CA 94621
415/562-8085

The cover price is \$10.00

THEOS TIMES

THEOS Times is a new publication now available from THEOS Software Corp., the new marketers of the THEOS operating system. The publication is to be issued quarterly. It will provide up-to-the minute information about THEOS8 and THEOS86, the 16-bit version of THEOS as well as additional listings of available THEOS software.

The publication will be available directly from:

THEOS Software Corp.
201 Lafayette Circle Suite 100
Lafayette, CA 94549
415/283-4290 Contact: Callie Webb

Again many of the packages listed in THEOS Software must be purchased with hardware.

HARDWARE DISTRIBUTORS

National hardware distributors such as OmniSource are a good place to start looking for THEOS software systems. Companies such as this distribute a number of computer lines and peripherals. In addition, they selectively distribute software such as word processors, spreadsheets and general accounting systems. They are not inclined to sell to end users directly. They will refer you to a local dealer who will be able to provide service and support. For more information contact:

OmniSource
455 Bearcat Drive
Salt Lake City, UT 84115
801/485-6664

Summary

THEOS was conceived as a business-oriented operating system. Therefore it is no surprise that over 1000 varied business application programs are available. No piece of software is perfect so don't expect to find software that will exactly match the way you currently conduct your business.

9

THEOS86™

Introduction

With the vast array of books available on the UNIX operating system, one would think UNIX to be the only 16-bit operating system available for microcomputers. This is *not* the case. The 16-bit version of THEOS was first introduced as OASIS-16. THEOS Software Corp., the new marketers of THEOS, have changed its name to THEOS86 as it was developed to utilize the Intel 8086 microprocessor. Although THEOS8 was first developed for 8-bit Z80 microprocessors, THEOS86 has been designed to accommodate the more powerful features of the 16-bit microprocessor. In this chapter we will discuss the differences between an 8-bit and a 16-bit operating system and the advantages of using an operating system such as THEOS86. In particular we will discuss some of the advantages of THEOS86 versus THEOS8 and we'll compare THEOS with the highly touted UNIX operating system. We will conclude with a discussion of the availability of hardware and software for THEOS86.

This chapter uses the term “THEOS8” when referring to THEOS and “THEOS86” when referring to the 16-bit version of THEOS. Both operating systems are now marketed and licensed by THEOS Software Corp. and Timothy Williams, the copyright holder of the operating system. THEOS Software Corp. is discussed in more detail in chapter ten.

Speed...8-bit vs. 16-bit Microprocessors

An 8-bit microprocessor is capable of processing data one character at a time. A 16-bit microprocessor can produce data two characters at a time. Therefore, the theoretical advantage of a 16-bit processor over an 8-bit processor is speed; the 16-bit processor should be twice as fast.

Another factor which helps to determine speed of data processing is the **clock speed** of the microprocessor. Clock speed is an indication of how fast the microprocessor can execute its instructions. Clock speed is measured in Megahertz (million cycles per second abbreviated MHz). The faster the clock speed, the faster the processor. Some of the early 8-bit microprocessors had clock speeds of only two MHz. Now there are 8-bit microprocessors on the market which have clock speeds of 8 MHz. The processing speed of this latest generation of 8-bit microprocessors is comparable to the speed of some 16-bit processors!

Another factor which helps to determine speed of data processing is the **clock speed** of the microprocessor. Clock speed is an indication of how fast the microprocessor can execute its instructions. Clock speed is measured in Megahertz (million cycles per second abbreviated MHz). The faster the clock speed, the faster the processor. Some of the early 8-bit microprocessors had clock speeds of only two MHz. Now there are 8-bit microprocessors on the market which have clock speeds of 8 MHz. The processing speed of this latest generation of 8-bit microprocessors is comparable to the speed of some 16-bit processors!

OTHER ADVANTAGES OF 16-BIT COMPUTERS

Other advantages of 16-bit computers include:

- Processing speed is 20% faster than 8-bit systems.
- More users are possible.
- More memory is available for running programs.
- Large amounts of disk space are available for storing data files.

Note: Much of the speed advantage of 16-bit processors is lost because there are more instructions for a 16-bit operating system to execute.

Over the past few years, the cost of 16-bit hardware has dropped considerably. The introduction of the IBM PC and the PC/XT has substantially increased the size of the 16-bit market. The sheer number of PC/XTs sold has created a market large enough for the marketers of THEOS to begin promoting THEOS86, the 16-bit version of THEOS.

An Overview of THEOS86

Like THEOS8, THEOS86 is a multi-user operating system designed for the business market. It is compatible with THEOS8 software (THEOS8 software can be easily translated to run under THEOS86) but takes advantage of the capabilities of 16-bit processors such as the 8086, 8088, and the 68000. The operating system is written in the C language so that it is easy for hardware vendors to create implementations of THEOS86 for their machines. There are a number

of hardware vendors supporting THEOS86. They include Seiko, Altos, and IBM for the PC/XT, and others.

THEOS86 uses a similar command structure to THEOS8 but with expanded capabilities for each command. For example in THEOS8 it is easy for the system manager to set up user accounts. The only parameters are:

- Account name
- Synonym name
- Password
- Privilege

THEOS86 has seventeen *additional* parameters that may be determined for each account. The complexity of the THEOS86 accounting structure permits more flexibility but requires considerably more thought before set-up.

Figure 9.1 compares the features of THEOS8 and THEOS86. We'll discuss each of these features in the following sections.

	THEOS8	THEOS86
Maximum number of users	9	32
Maximum amount of disk storage	167MB	4 Gbytes
Number of print spoolers	1	4
Maximum memory for each user	64K	256K
Graphics capability	NONE	YES

Figure 9.1. A comparison of features of THEOS8 and THEOS86

MAXIMUM NUMBER OF USERS

An 8-bit THEOS system is ideal for use in businesses whose sales volume is \$250,000 to ten million dollars annually. A typical THEOS8 installation has no more than three users and a printer on the system simultaneously. Plenty of room for later expansion is possible.

Any multi-user system slows down as users are added. An 8-bit THEOS system with nine users accessing the system simultaneously is not practical. Realistically, if you expect more than 5 users to be accessing the system simultaneously, you should consider implementing a THEOS86 system.

MAXIMUM AMOUNT OF DISK STORAGE

Do you realize how much disk space 167MB is? Think back to our discussion of software in chapter eight. The accounting package which required 1000 characters for each inventory item would only require 30MB for 30,000 inventory items! Even, with 60,000 inventory items, only 60MB of disk space is occupied. There is plenty of room left for customer files, accounts receivable transactions, and so on.

MULTIPLE PRINT SPOOLERS

The impatience of the users plays a role in the importance of this feature. Storing data to be output to the printer in spooler files allows the user to move on to other tasks. In THEOS8, the purpose of the spooler is to eliminate a *time bottleneck* which occurs during printing. When a print request is made, the data to be printed is temporarily stored on disk in a spooler file. When the printer is ready, THEOS sends the spooled file to the printer while automatically erasing it from the disk. The number of spooled files possible is limited only by the available disk space and the record length specified by the system manager when the SYSTEM.SPOOLQUE file was created. This file helps control spooler functions.

Some software allows supplementary printers to be accessed from a menu. These printers would have no spooling capability. When a print command is issued to a printer without spooling capability, the user would have to utilize another terminal to continue working on the system until printing was completed.

THEOS86's capacity of four spooled files per user (versus one allowed in THEOS8) offers a real advantage in systems where the printer is used extensively. This permits printing of different files on the same or different forms and the use of multiple printers by application software.

MAXIMUM MEMORY FOR EACH USER

The maximum amount of memory available for each user in a THEOS8 system is 64K. With the portion of RAM required for THEOS subtracted from the available memory, each user ends up with

about 44-45K. Programmers have complained about the hardships involved in writing programs for this small amount of memory. They claim with more memory they could write better programs. They are right, to some extent.

THEOS8 programmers have done an excellent job with the tools given them. To some extent they have side stepped memory constraints by writing smaller programs which they linked together. When word processors and spreadsheets required additional memory for text, temporary work files were created on disk, and data was swapped in and out of memory as needed. The primary drawback with this approach is slower execution. Disk access is time consuming. Storing data in memory and editing it there is much faster.

THEOS8 programmers now writing for THEOS86 can write better and more compact programs because of the limitations imposed upon them by THEOS8. In many cases 64K of memory is adequate. Application software requiring more than 64K of memory must be implemented on THEOS86.

GRAPHICS

Pie charts, bar charts and other computer-generated graphics can be helpful. How useful are these charts and how much are you willing to pay for this capability? Much of the attraction of software for personal computers has been the attractive screen displays that certain programs create. THEOS8 has no graphics capability. If this feature is important, consider THEOS86.

Hardware

The following manufacturers have implemented THEOS86 for their machines.

- Seiko
- Altos
- IBM for the PC/XT
- Onyx
- Advanced Digital

The minimum hardware requirements for THEOS86 are:

- 16-bit microprocessor
- 348K bytes RAM
- Terminal
- Minimum of 5MB of hard disk storage.

Depending upon the hardware vendor, these requirements may vary. A brief discussion of THEOS86 hardware vendors follows.

SEIKO

Although Seiko is known for their watches, they also make an excellent 16-bit computer system and have been working with 16-bit THEOS since 1982. They have developed their own implementation which includes a CP/M Adapter™ utility. This allows their system to run *off the shelf* programs like SuperCalc, Wordstar, and dBase II. They have written their own installation guide for THEOS86 which makes installing a THEOS86 system easy for anyone familiar with THEOS8.

Application software for dentists, architects, travel agents, florists, realtors, and many other vertical markets is available through Seiko. They even have 24 hour support.

A major advantage to anyone considering the purchase of a Seiko system is the financial stability of the company. Many smaller companies lacking the money to stay in business will come and go.

ALTOS

Altos is a recognized leader in the multi-user hardware community, though their involvement with THEOS has been limited. Presently, two of their 16-bit machines support THEOS86.

IBM

The name IBM is synonymous with computer. For this reason alone, the number of potential THEOS86 users is enormous. Many businesses which have purchased single-user systems, such as the PC/XT, are now finding they need a multi-user system. The PC/XT

can be transformed into a multi-user system by installing THEOS86 and some peripherals. The only drawback is that software currently running under PC-DOS or MS-DOS won't work under THEOS86.

Under THEOS86 up to three users can share a single PC/XT. A terminal is all that is required to add users, saving you money. Each terminal shares the XT's memory, disk drives, printer, and other attached peripherals.

The minimum hardware requirements for PC/XT THEOS86 operation are:

- IBM PC/XT
- 348K RAM
- Monochrome display card
- Monochrome monitor
- One RS232C adapter for each additional user (up to 2)

The system can be expanded up to 640K of RAM. THEOS86 supports any popular brand of terminal such as Televideo or Kimtron, parallel or serial printers, and communications devices such as modems.

ONYX

The Onyx 186D series microcomputer is a multi-user system which supports THEOS86. Onyx has been involved with THEOS since its inception.

Many THEOS distributors and dealers implement the system on Onyx hardware. They offer over-the-phone consulting, a software subscription service providing up-to-the-minute information on a variety of THEOS software, classroom training, and their own documentation.

ADVANCED DIGITAL

The Super System II, introduced in November, 1984, features an 8MHz 16-bit microprocessor, up to 1MB of RAM, a choice of disk configurations between 20 and 330 MB, and either an eight inch or 5¼ inch floppy disk drive for loading software and backup. Contact the company for pricing and a list of its dealers.

Software

The THEOS Application Software Directory and THEOS Times are the best places to begin looking for software for THEOS86. Many software developers advertise that their software is available for both THEOS8 and THEOS86. Figure 9.2 includes a partial listing of software vendors and their 16-bit products.

Company	Product(s)
Phase One Systems	Control-16
Logic Extension Resources	multiCALC, multiWRITE
Groundstar	accounting software
UserWare International	Userbase Database Management
Computer Printed Words	Spellvue Spelling Checker
Seiko	CP/M Adapter™, vertical market software

Figure 9.2 THEOS software vendors.

THEOS vs. UNIX

Anyone considering the purchase of a 16-bit computer must consider the purchase of the operating system too. The tremendous financial resources of AT&T, the developers of UNIX, have given UNIX a real push into the market. Hardware vendors have taken advantage of the advertising dollars being spent and daily introduce new products which are UNIX compatible. UNIX advocates claim that UNIX is a superb environment for writing programs and that it is the only universal operating system. It may be a great operating system for writing programs. After all it was written by computer programmers working for AT&T. As far as being universal, does this mean it will run on any computer? Not hardly. What can be said about UNIX is that it has become the buzzword of the mid 1980's.

Are you a computer programmer? Probably not. Your main concern is most likely running application software to help you operate your business easily and profitably. To better appreciate the differences between the two operating systems from the user's point of view, let's review their design goals.

Anyone considering the purchase of a 16-bit computer must consider the purchase of the operating system too. The tremendous financial resources of AT&T, the developers of UNIX, have given UNIX a real push into the market. Hardware vendors have taken advantage of the advertising dollars being spent and daily introduce new products which are UNIX compatible. UNIX advocates claim that UNIX is a superb environment for writing programs and that it is the only universal operating system. It may be a great operating system for writing programs. After all it was written by computer programmers working for AT&T. As far as being universal, does this mean it will run on any computer? Not hardly. What can be said about UNIX is that it has become the buzzword of the mid 1980's.

Are you a computer programmer? Probably not. Your main concern is most likely running application software which help you operate your business easily and profitably. To better appreciate the differences between the two operating systems from the user's point of view, let's review their design goals.

Design Goals Of THEOS and UNIX

Development work on UNIX begin in 1969 with Ken Thompson, a Bell Labs employee, on the DEC PDP-7 minicomputer. His goal was to develop an operating system ideally suited to programmers working with this computer. In its first incarnation UNIX was a single-user operating system. Development continued until the operating sytem could run on a variety of minicomputers and could support several hundred thousand bytes of memory, large disk storage devices and many users. Little thought was given to making UNIX a commerically viable product.

THEOS development began in the mid-1970's on a Z80-based microcomputer. The goal was to bring the sophisticated features found only on minicomputers and mainframes, such as multi-user capability, to small business computers. The first versions of THEOS ran on machines with a maximum of 64K per user (THEOS) and floppy disk drives. THEOS has been refined to accommodate up to 640K of memory and over 150MB of disk storage.

Although no operating system yet written can be totally user friendly, THEOS is one of the easiest to use. It is possible to isolate

users from the operating system be writing EXEC programs to automate system maintenance and run application software. If it becomes necessary to use the operating system directly, on-line help is available for all operating system commands by typing,

>HELP COMMAND NAME

at the operating system prompt. Command names are descriptive of their functions.

Using UNIX requires either a resident expert or a series of shell programs to assist the user. Shell programs are similar to THEOS EXECs and are supplied with a system by a turnkey system vendor. These shell programs must be written by a UNIX programmer. If shell programs are not provided, an on site UNIX expert is required. UNIX commands are often not descriptive of their function and on-line help is generally not available. A novice user of a UNIX system could unwittingly cause damage to the system because no questions are asked before a command is executed. A UNIX system always assumes the operator is knowledgeable, which is not always an accurate assumption.

DATA FILE MANAGEMENT

One of the most important tasks of an operating system is data file management. One of the original design goals of UNIX was to make its file system simple. All UNIX files have the same structure. There is no built-in indexed or keyed file access. Indexed or keyed access must be provided by the application software. The means by which data files are accessed determines the speed of data retrieval.

THEOS was designed for business use. It provides built-in capability for a variety of file types (such as indexed and keyed files) and data can be retrieved quickly.

FILE AND RECORD LOCKING

Multi-user system programmers must always allow for occurrences where one user wishes to access another's files or records. For example an accounting program that does extensive writing to a data

file should protect the entire file from another's use. This feature is called **file locking**.

If a file is not locked in its entirety other users may access the file but not individual records within the file. This feature is called **record locking**. Without locking data can be entered by system users with no controls. This could result in errors in data file.

THEOS86 provides built-in data security because record locking is an integral part of the operating system. File and record locking in UNIX must be undertaken by the application software. THEOS86 allows the programmer to concentrate on writing functional software without worrying about file access.

COST vs. PERFORMANCE

THEOS8 and THEOS86 were designed to be low in cost. Both are capable of running on computers costing as little as \$4000. The design of the UNIX operating system requires extensive use of the microprocessor particularly during multi-user operation. For this reason it is recommended for use on high performance computers with ample capacity disk drives. Such machines are generally in the \$7000 to \$15,000 price range.

DOCUMENTATION

One of the reasons for the large number of books of UNIX is the poor documentation provided by AT&T. Most UNIX vendors supplement this documentation with their own material. The documentation for THEOS, as originally written, was not written for the first time user but it is informative. Even individuals with limited computer experience can use it. THEOS Software Corp. is writing new documentation for both THEOS8 and THEOS86. Figure 9.3 compares THEOS86 and UNIX features.

	THEOS86	UNIX
Application	Business	Research & Development
Software dev. capab.	Good	Extensive
Languages supported	4	10+
Disk space required for operating system	3MB	3-5MB
File locking	Yes	No
Record locking	Yes	No
Filing system	Comprehensive	Application dependent
Hardware required for efficient operation	\$4000 min.	\$8000 min.

Figure 9.3. THEOS86 and UNIX feature comparison

Summary

Most of the advantages of THEOS86 over THEOS8 center around speed, greater number of users, and greater disk storage capacity. Here are some valid reasons for considering THEOS86:

- You will require a system with more than five terminals
- You wish to implement THEOS on IBM hardware. The IBM PC/XT will only run THEOS86.
- You require graphics capabilities which THEOS8 lacks.

10

Further Considerations

Introduction

THEOS was originally marketed by Phase One Systems, Inc. That firm no longer holds the marketing rights to THEOS whose marketing is now being handled by THEOS Software Corp. In this chapter we will discuss THEOS Software Corp., the latest version of THEOS8 and the THEOS User's Group.

THEOS Software Corp.: The New Marketers of THEOS

In September of 1984 THEOS Software Corp. acquired the exclusive world-wide marketing rights to the THEOS8 and THEOS86 operating systems. Previously, both operating systems were marketed by Phase One Systems, Inc. THEOS8 was marketed under the generic term of OASIS while THEOS86 was marketed as OASIS-16. This change of venue is viewed by the THEOS community as a major step in achieving greater visibility in the marketplace for THEOS. Timothy Williams, the author of THEOS8, THEOS86 and related products, is also the founder and chairman of the board of THEOS Software Corp. With the developer of THEOS taking a personal interest in the company marketing it, THEOS should gain more market visibility.

This new marketing arrangement has no direct effect on users. Users have no reason to contact THEOS Software Corp. directly. The operating systems are licensed to hardware vendors who sell them with their hardware. However, better communication between THEOS Software Corp. and the hardware vendors will mean better support for THEOS users through their dealers.

THEOS VERSION 6.1

No operating system is perfect and THEOS is no exception. The majority of this book was written with word processing software running under THEOS 5.6, a version released in 1983. This author only recently received a copy of 6.1 for evaluation. As of this writing many THEOS hardware vendors have yet to release their versions of 6.1

The refinement of THEOS8 has resulted in improvements in performance and ease of operation. Many of the refinements benefit programmers more than users. We will discuss those refinements that directly benefit end users.

Note: Each hardware vendor uses most of the THEOS programs as they exist. However, each vendor may add other commands and utility programs which make THEOS more compatible with his hardware.

THEOS8 Version 6.1 Refinements

THE SYSTEM CANCEL KEY

In an earlier chapter we mentioned that using the system cancel key (ESC Q) to cancel an EXEC operation causes the message:

Cancel EXEC (Y/N)?

to be displayed. If “Y” is entered the EXEC will be cancelled and you will be returned to the THEOS prompt. If “N” is entered, the result can be unpredictable. Sometimes the system might hang while other times EXEC execution will continue.

Another problem occurred when a key other than “Y” or “N” was pressed. Again the results were unpredictable. Due to these unpredictable results, the use of the system cancel key in early THEOS8 versions was discouraged.

In version 6.1 as soon as the sequence ESC Q is entered, EXEC execution is terminated immediately. The purpose of the system cancel key is to cancel the command or routine being executed. No operator intervention is required.

ARCHIVING IN MULTI-USER MODE

The UTIL.EXEC program described in this book provides a menu choice for creating an archive. The help screen includes a message which warns the operator that single-user mode is required for archiving. Single-user operation is no longer required to make archives. If this EXEC is used with any version of THEOS prior to 6.1 and the archive choice is selected while in multi-user mode, the error message “Must be single-user mode or private disk volume” is displayed, and you are returned to the menu. If you are using 6.1 and specify the (PUBLIC option you can archive in multi-user mode.

The ability to archive in multi-user mode is a time-saving convenience. Most archives are undertaken at the end of a business day after everyone has finished using the computer. The ability to archive in multi-user mode means the system manager need not bring the system down to single-user mode before making an archive.

Note: The RESTORE command requires single-user mode operation.

NEW HELP MESSAGES

Additional help screens are available with version 6.1. By typing,

>HELP EXEC<RETURN>

help for writing EXEC programs will appear on the screen. Although EXEC isn't a command it is a feature important enough to warrant on-screen help. Earlier problems with the formatting of help messages on the screen have been corrected.

NEW EXEC FEATURES

Some application software eliminates the requirement of pressing <RETURN> when making choices from menus. For consistency's sake it would be nice if this capability were possible with EXEC programs. Now it is. The EXEC programs in the appendix can be modified by the system manager to provide this convenience. Information on the routines to add to the EXEC's can be obtained by typing:

>HELP EXEC<RETURN>

ADDITIONAL WORKING MEMORY

At least 2K bytes of additional memory are now available because one of the files required for spooler operation is an integral part of the SYSTEM.NUCLEUS file. Remember in chapter eight on software, we mentioned that the multiWRITE word processing program could not utilize the thesaurus on some systems in multi-user mode. With the additional memory available the thesaurus can be used in multi-user mode. The additional memory should benefit most systems, permitting them to be more versatile.

THE THEOS USER'S GROUP

The OASIS User's Group (OUG) since its establishment in February, 1981, has grown to over nine hundred members, most of whom are computer professionals. The group has compiled an extensive library of software contributed by its members. The software consists of over seven hundred multifile programs consisting of the following types and percentages of software:

- 20% utility programs
- 20% business and finance
- 20% engineering and math
- 25% games
- 10% education and miscellaneous categories

Membership is \$35.00 per year. The membership fee includes one disk volume containing samples of user-contributed software as well as a newsletter subscription. Members can order new library additions at a cost of \$35.00 for two diskette volumes. Twelve volumes are offered annually. A service fee of \$35.00 is charged for old volumes in the library.

Belonging to the user's group can provide you with up-to-the-minute information about THEOS enhancements and/or problems which will help your system run smoothly. In addition, you may find some of the user-contributed software useful. Since most of the people using THEOS are in business, you may be able to share experiences with other users which are mutually beneficial.

If the THEOS User's Group is of interest to you contact:

Fred Bellomy
THEOS User's Group
P.O. Box 2400
Santa Barbara, CA 93120
805/965-0265



Appendix A. Glossary

<i>Boot</i>	The term for starting-up the computer.
<i>Byte</i>	The computer equivalent of one character.
<i>Class Code</i>	A THEOS program which optimizes the performance of a terminal (or other peripheral such as a printer or modem) with a computer. Each brand and model of terminal requires a class code for correct screen displays.
<i>Command Line</i>	A series of characters entered after the THEOS prompt (>) which tells THEOS what function to perform. These characters consist of a THEOS command separated by a space and an open parenthesis (followed by command options.
<i>Commands</i>	Directives for an action by THEOS. These directives consist of names such as COPYFILE, ACCOUNT, ARCHIVE, RESTORE, and so on, and are indicative of the functions which they perform.
<i>Control Key</i>	This key must be depressed <i>together</i> with another key to change the value of another key. The key marked "Ctrl" is held down with another key to execute a command.
<i>Cursor</i>	A marker on a terminal screen indicating that characters may be entered. The type of cursor is selectable according to the options permitted by various terminals. These options include: a small filled rectangle, a slow or rapidly flashing rectangle, a flashing or non-flashing underscore.
<i>Database Manager</i>	A piece of application software analogous to an electronic filing cabinet. Many pieces of information can be stored or sorted. Customized reports may be generated.
<i>Database</i>	A file created by a database manager which can be maintained, sorted and reported upon.
<i>Default Value</i>	A value automatically assumed (or in some cases automatically entered) by the software, saving the operator time and providing protection against mistakes. Default values apply to operating system commands and options as well as to data entry during execution of application software.
<i>DRIVNAME</i>	A letter (S, or A through G) which specify disk drives in a THEOS System.

<i>ESC(ape)</i>	This key is pressed <i>before</i> another key to execute a specific function or command. Some terminals may require that this key be pressed twice.
<i>EXEC</i>	A special program in THEOS which can perform operating system functions automatically as well isolate a user from the operating system.
<i>EXEC Language</i>	A programming language which permits EXEC programs to be written. It is similar to the Job Control Language found on many larger more expensive computers.
<i>Execute</i>	To carry out a command or run a program.
<i>Field</i>	An item within a file such as a name or address.
<i>File</i>	Analogous to a file folder. A file can contain the records of a database.
<i>File Description</i>	Component parts of a THEOS command line consisting of filename, filetype, and drivname.
<i>Filename</i>	The first component of a THEOS file description. The file name can be no longer than eight characters and they must start with an alphabetic character.
<i>Filetype</i>	The second component of a THEOS file description. The filetype categorizes the file as to its function. File types can be no longer than eight characters and must start with an alphabetic character. Some file types have particular meaning to THEOS. Examples of file types are COMMAND, EXEC, DATA, and so on.
<i>Hardware</i>	Any piece of a computer system which is tangible such as disks, disk drives, modems, microprocessors, printers, and so on.
<i>K or Kbytes</i>	One kilobyte (1024 bytes). The equivalent of 1024 characters. Used in reference to disk storage or RAM.
<i>Logon</i>	Message displayed in a THEOS system which requires that a user type an account name. If the account name exists in the SYSTEM.ACCOUNT file, the user may be asked to type in a password. After successful log on is accomplished the user may then access all programs and data files which are owned by that account.
<i>Media</i>	Any materials such as hard disks, floppy disks or tapes used to store computer data. Floppy disks tapes are generally classified as removable media
<i>MB or Megabyte</i>	One million characters of storage. Commonly used when referring to disk drive storage capacity.

<i>Menu</i>	A series of options or choices is displayed on the screen. When an option is selected a program is run or an operating system command is executed. A menu facilitates use of a computer system.
<i>MHz</i>	A measurement of frequency: million cycles per second.
<i>Modem</i>	A device which permits exchange between two computers over telephone lines.
<i>Operating System</i>	The software which allows the user to communicate with the computer in a language which is mutually agreeable.
<i>Option</i>	A stipulation imposed on an operating system command. In the following example: <p style="text-align: center;">ARCHIVE G S (ACCOUNT</p> the ACCOUNT option stipulates that only those files belonging to the currently logged account be archived. THEOS command options were listed as each command was discussed.
<i>Peripheral</i>	Any device such as a terminal, printer, or modem which is separate from the computer itself.
<i>Prompt</i>	A symbol displayed on the screen which indicates that the software is ready to accept input. The THEOS prompt is indicated by >. A prompt may also be a series of on-screen messages provided to suggest possible input.
<i>RAM</i>	An acronym for Random Access Memory. Used in conjunction with a number (such as 64K) to indicate the amount of memory available.
<i>Record</i>	Similar to an index card in a file of cards where each card constitutes a name, address, city, state, and zip code of an individual. The card containing this information is a record in the file.
<i>Record Locking</i>	A software feature which permits only one person at a time to access or change a specific record in a common data file. Record locking permits use of the file by other users and is preferable to file locking which prevents use of the entire file when accessed by a user.
<i>Software</i>	A series of instructions which tells the computer what to do. These instructions may be in the form of an operating system or application software.
<i>Spreadsheet</i>	Application software which permits calculations to be made according to user entered specifications in a row and column format.

<i>Syntax</i>	The specific sequence in which words are placed in a command line or sentence.
<i>Template</i>	A pre-configured pattern of data entry which can be re-used. An example of a template in a spreadsheet is a loan amortization calculator. An example of a template in a database is the design of a database which may include name, address, city, state, and zip code.
<i>Toggle</i>	To turn on or off; usually accomplished by issuing a command within software.
<i>Turnkey</i>	" <i>Turn the key</i> " or turn on the computer and have it do something meaningful with little training. Used in reference to " <i>menu driven</i> " systems. Menus provide choices for users to make and are executed automatically.
<i>UNIX</i>	A 16-bit operating system developed by AT&T.
<i>Vertical Market Software</i>	Application software developed for specific businesses.
<i>Z80</i>	Generic term for the 8-bit microprocessor used in THEOS systems.

Appendix B.

THEOS Command Summary

Command	Function
ACCOUNT	Maintenance for user accounts
ARCHIVE/RESTORE	Back-up copies of hard disks
ATTACH	Associate peripherals with computer
CHANGE	Change the privilege level of a THEOS command
COPYFILE	Make copies of individual THEOS files
CREATE	Pre-allocate disk space for any non-sequential THEOS file
ERASE	Erase unused files and recover their disk space
FILELIST	Provide a listing of disk files
INITDISK/TAPE	Format disks or tape for storing data
LIST	List the contents of data files
LOGOFF/LOGON	The only means of accessing user accounts
MAILBOX/MSG	Internal message system provided by THEOS
RENAME	Change the names of files or their protection status
SCRIPT/EDIT	THEOS line editor and formatting programs
SET	Provides the only means of regulating THEOS switches and allocating user memory.
START	Activates user memory partitions
STOP	De-activates user memory partitions
SHOW	Used to display the status of THEOS switches
SPOOLER	Controls THEOS printing spooler functions
SYSGEN	Makes the status of THEOS switches or ATTACHments "permanent" helping to automate the system
SEND/RECEIVE/ TERMINAL	Provide the means of sending, receiving and storing files to another computer

Appendix C.

Common THEOS Warning and Error Messages

Basic Rules

Errors occur during THEOS command execution or from within application software packages for the following reasons:

- Operator error
- Damaged or defective media (disks, diskettes, or tapes)
- Hardware problem
- A bug in operating system or application software

The reasons are listed in order of their frequency. For those occasions when errors are not caused by operator error, we included the following discussion.

Disk Errors

Disk error messages are displayed in the following format:

Disk <disk label> <#> <message> <cyl> <head> <sect>

<disk label> is the logical disk label S,A,B, etc. <#> is the physical disk number attached to the logical disk label. <message> is a specific error message as indicated below. The <cyl> <head> <sect> provide information about the location on the disk where the error(s) occurred.

The specific disk error messages which appear in the <message> area are:

- Disk not ready
- Data transfer error
- Not initialized
- Track not found
- Sector not found
- Write protected
- Disk changed

Responses To Disk Error Messages

- Message:** disk not ready
- Probable Cause:** Floppy disk not loaded in the drive, or disk not attached.
- Remedy:** Insert an initialized floppy or floppy with data you wish to read in the attached drive. Press <RETURN>. If a floppy is not required, use the ATTACH command to detach the floppy disk drive. If the disk to be accessed is not attached, execute the ATTACH command.
-
- Message:** data transfer error
- Probable Cause:** static electricity, defective floppy disk, defective floppy disk drive, hard disk problem.
- Remedy:** Press <RETURN> to retry the operation. If you suspect a defective floppy disk press Q to quit the operation and return to CSI. Insert a newly initialized and verified floppy disk and reissue the original command. This message usually occurs during execution of a COPYFILE, ARCHIVE or RESTORE command.
-
- Message:** not initialized
- Probable Cause:** data cannot be written to the disk because it has not been initialized.
- Remedy:** Press Q to quit the operation, exit to CSI and use the INITDISK command to format the disk.
-
- Message:** track not found
- Probable Cause:** defective disk or disk drive.
- Remedy:** Press <RETURN> to retry the operation. Usually occurs during ARCHIVE, COPYFILE, RESTORE or VERIFY commands. If the retry doesn't work and the "Unrecoverable" message appears, press Q to quit the operation and try another disk. If you are restoring files use a backup copy.
-
- Message:** sector not found
- Probable Cause:** defective disk or disk drive.
- Remedy:** Press [RETURN] to retry the operation. If retry doesn't work and the "Unrecoverable" message appears press Q to quit the operation and try another disk or backup copy.
-
- Message:** write protected
- Probable Cause:** the disk has a write protect tab in place or has been protected using the RENAME command.

- Remedy:** Remove write protect tab or use the command:
`RENAME FILENAME FILETYPE DRIVE NAME
(NOD NOW[RETURN])`
to change protection status of the file.
- Message:** disk changed
- Probable Cause:** a floppy disk has been inserted with a different label than the previous disk. The label was assigned during execution of the INITDISK command.
- Remedy:** Press M to mount the disk. This informs THEOS that you want to use the disk with a different label.

Notes

- When R or [RETURN] is pressed for a retry of an operation, THEOS will automatically retry the operation a predetermined number of times (usually eight). At this point it is best to type Q to return to CSI and re-execute the command.
- If disk errors persist and are not a result of defective floppy disks, contact your vendor for help for possible repair of the disk drive.

Errors During The Booting Process

The following files are required to qualify as a system disk:

- SYSTEM.NUCLEUS
- SYSTEM.CSI
- SYSTEM.EXECLANG
- SYSTEM.EXEC1
- SYSTEM.DEV NAMES
- SYSTEM.ERRMSG ✓
- SYSTEM.DEVnn
- SYSTEM.CLASSnn
- SYSTEM.ACCOUNT

If any one of these files is not present and the disk is used to boot the system, either the system will not boot at all and will hang or the following errors may be displayed:

- Disk error <message>
- Not Found: "SYSTEM.NUCLEUS"

- Not Found: "SYSTEM.EXEC1"
- Not Found: "SYSTEM.CSI"
- Not Found: "SYSTEM.ERRMSG"

If any of these messages are displayed, either the disk is defective or one of the above files is missing.

Information Messages

The messages in the SYSTEM.ERRMSG file are provided to keep the user informed during interaction with THEOS. To view or print the messages in this file type:

> LIST SYSTEM ERRMSG[RETURN]

The contents of the file will be listed on the terminal. The following entry,

> LIST SYSTEM.ERRMSG (PRINTER1[RETURN]

lists the contents of the file to the printer.

Many of the messages are self-explanatory. Some are not. Here are some examples:

- | | |
|-----------------|---|
| Message: | Syntax error |
| Meaning: | A typing mistake occurred. |
| Message: | Protected file |
| Meaning: | File has delete or write protection. In order to modify the contents of a file using the EDIT command, you must be logged onto the account which owns the file and change its protection status. |
| Message: | Backup file protected |
| Meaning: | You are trying to edit a file which doesn't belong to your account. Log onto the account which owns the file and edit the file there. |
| Message: | Incorrect serial number |
| Meaning: | The serial number of your operating system does not match the serial number of your software. In order to operate the software, it must either be unserialized or serialized to your copy of THEOS. |
| Message: | Must be single-user mode or private disk volume. |
| Meaning: | The command being executed requires single-user mode in order to execute. This message usually occurs during ARCHIVE or RESTORE. |
| Message: | Maximum nesting exceeded. Nesting error. |

Meaning:

The SYSTEM.EXECn file has a record length smaller than the number of EXECs you wish to execute. You must use the CREATE command to increase the number of records in each SYSTEM.EXECn file according to the maximum number of EXEC programs you wish to execute. If you will be using fifteen EXEC programs, the record length of the SYSTEM.EXECn file must be fifteen or greater.

Appendix D. Using MCI Mail To Transmit Files

A new service provided by MCI permits letters to be sent electronically to any subscriber or sent and delivered by U.S. mail. It is also possible to send large documents of unlimited size to another subscriber for transfer to the subscriber's computer. Any THEOS sequential file may be sent in this manner using the THEOS Terminal Emulator program.

Word Processors and Communication Software

Most word processing programs use a variety of control characters and other commands which confuse communication software. Before files can be sent, these characters must be stripped from the document. Fortunately the multiWRITE word processing software provides a special program which will prepare the file automatically. Assuming you have finished editing a multiWRITE file and you wish to prepare it for electronic transmission, at the THEOS prompt type:

```
>MWPRINT FILENAME (THEOS <RETURN>
```

You are presented with the multiWRITE printing screen. Press ESC ESC and the message "Printing to FILENAME.MWPRINT" will be displayed. You will be returned to the THEOS prompt upon completion.

Sending Files

The MASTER.EXEC as supplied will automatically transport you to the THEOS terminal emulator and perform the ATTACH command. If you are using this EXEC, you may skip steps two and three.

To send files follow the steps outlined below.

- Step 1. Physically connect a modem to an unused port on the computer and set the proper baud rate (if it is hardware controlled) and turn the "clear to send" switch (inside the computer) off; turn on the modem.

Step 2. Enter the following ATTACH command:

>ATTACH COMM PORTn (W8[RETURN]

n is the port number where the modem is connected. W8 is an option that specified a transmitted word length of eight bits. If the baud rate can be software controlled, you can use the Baud option to specify B300 or B1200.

Step 3. At the THEOS prompt type:

>TERMINAL[RETURN]

The THEOS terminal emulator menu should appear on the screen.

Step 4. Select option 5 and follow the directions for dialing the MCI access number and the directions for dialing the telephone with your modem.

Step 5. Once you have successfully logged onto the MCI system and you have selected the option to create an MCI file, follow the instructions in your MCI guide. When you are ready to transmit, type ESC C to return to the terminal emulator menu.

Step 6. Select option 2 and type the file name and filetype to be sent.

Step 7. Select option 5 again. The file will automatically be sent regardless of its size. If file transmission was successful a ^ Z (control Z) character is displayed in the lower left corner of the screen. This is the end-of-file marker.

Step 8. Follow the MCI directions for sending the file and logging off their system. When logoff has been accomplished hang up the phone and press ESC to return to the terminal emulator menu. Type 6 to return to THEOS.

How To Know If Your File Was Not Received Properly

As the file is being sent its contents are displayed on the screen. Occasionally it may appear as if a character or two has been lost. Don't worry about it. These missing characters are usually errors which occur as characters are echoed back to your computer. The time to worry is when your terminal starts beeping wildly. This indicates a failure to communicate between your computer and the MCI computer. In this case a substantial number of characters will be lost. Wait a few minutes until sending resumes and the ^ Z character is displayed. Then try sending the file again.

If you still have problems wait until after 11pm on weeknights or anytime Saturday and Sunday when fewer users are accessing on the MCI network.

Appendix E. THEOS EXEC Programs

```

0005 ; THIS USER1 EXEC IS EXECUTED AFTER LOG ON AND TAKES
0007 ; THE USER TO THE MASTER MENU
0010 &CONTROL OFF
0020 -START
0030 &SPACE 6
0040 &BEGTYPE
0050
                                PRESS RETURN TO CONTINUE
0060 &END
0070 &WAIT
0080 MASTER EXEC ; EXECUTE MASTER EXEC

0010 ; MASTER MENU EXEC RUNS PROGRAMS BASED ON USER SELECTIONS
0025 ; COPYRIGHT 1985 BY WALTER STAGNER
0030 -AGAIN
0040 &CONTROL OFF
0050 -START
0060 &CRT CLEAR
0070 &CRT 25 ;
0080 &BEGTYPE ; PAINT SCREEN
0090 Ashley Communications, Inc.
0100 &END
0105 ; LINE 90 CAN BE REPLACED BY YOUR COMPANY NAME
0110 &SPACE 1
0120 &CRT 21,3
0130 SHOW TIME ; THIS LINE DISPLAYS THE CURRENT TIME AND DATE
0135 ; IF YOU DON'T LIKE REVERSE VIDEO DELETE THE LINES
0137 ; WITH THOSE COMMANDS...THE EXEC WILL FUNCTION OK
0140 &CRT RVON ; TURN ON REVERSE VIDEO
0150 &BEGTYPE
0160
                                MASTER MENU
0170 &END
0180 &CRT RVOFF ; TURN OFF REVERSE VIDEO
0190 &BEGTYPE
0200
0210
                                1. ACCOUNTING/ORDER ENTRY/INVENTORY CONTROL
0220
                                2. FINANCIAL MODELING
0230
                                3. TERMINAL PROGRAM
0240
                                4. Spellbinder WORD PROCESSING
0250
                                5. SYSTEM UTILITIES
0260
                                6. SEND MESSAGES TO ANOTHER USER
0270
                                7. RETRIEVE MESSAGES (EXECUTE MAILBOX)
0280
                                8. LOGOFF SYSTEM/LOGON TO NEW ACCOUNT
0290
                                9. multiWrite WORD PROCESSING
0300
                                10. QUIT FOR THE DAY
0310 &END
0315 ; LINES 200 THROUGH 300 CAN BE MODIFIED TO MATCH YOUR SOFTWARE
0320 -SEL
0330 &CRT 15 17
0340 &CRT RVON
0350 &TYPE ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN \
0360 &CRT RVOFF
0370 &READ &SEL ; READ USER INPUT AND ASSIGN VALUE TO &SEL VARIABLE
0380 &IF &SEL LT 1 &GOTO -SEL
0390 &IF &SEL GT 10 &GOTO -SEL
0395 ; LINES 380 AND 390 WILL NOT ACCEPT INPUT OTHER THAN A
0397 ; NUMBER BETWEEN 1 AND 10

```

314 THEOS User's Handbook

```
0400 &IF &SEL = 1 OM8000 EXEC
0410 &IF &SEL = 2 &GOTO -FINMOD ; RUN A SPREADSHEET PROGRAM IF PRESENT
0420 &IF &SEL = 3 &GOTO -TERM ; OASIS TERMINAL EMULATOR
0430 &IF &SEL = 4 &RUN SPELL ; RUNS SPELLBINDER WORD PROCESSING
0440 &IF &SEL = 5 UTIL EXEC ; SYSTEM UTILITIES MENU
0450 &IF &SEL = 6 &GOTO -MESS ; INTERNAL MESSAGE SYSTEM
0460 &IF &SEL = 7 &GOTO -MAIL ; RETRIEVE MESSAGES
0470 &IF &SEL = 8 &GOTO -LOGOFF ; LOGOFF/LOGON TO NEW ACCOUNT
0480 &IF &SEL = 9 WP EXEC ; WORKS IN CONJUNCTION WITH MULTIWRITE SOFTWARE
0490 &IF &SEL = 10 HOME ; THIS LINE EXECUTES AN OASIS COMMAND "HOME"
0492 ; LINE 490 IS REQUIRED BY THE IBC MIDDY CADET TO SHUTDOWN
0493 ; IF YOUR SYSTEM DOES NOT REQUIRE THIS COMMAND MODIFY LINE 490
0495 ; LINES 400, 410, 430, 480, AND 490 CAN BE MODIFIED TO
0497 ; MATCH THE REQUIREMENTS OF YOUR HARDWARE AND SOFTWARE
0500 &GOTO -AGAIN ; THE PURPOSE OF THIS LINE IS TO CAUSE THIS
0505 ; EXEC TO BE RE-EXECUTED UPON EXIT FROM A PROGRAM OR EXEC FUNCTION
0510 -LOGOFF ; LOGOFF SYSTEM OR LOGON TO NEW ACCOUNT
0520 &CRT CLR
0530 &TYPE WOULD YOU LIKE TO MAKE BACKUP COPIES OF TODAY'S WORK? Y/N \
0540 &READ &SEL ; READ USER INPUT AND ASSIGN VALUE TO &SEL VARIABLE
0550 &IF &SEL <> Y &IF &SEL <> N &GOTO -LOGOFF
0560 &IF &SEL = Y UTIL EXEC ; GO TO UTILITIES MENU FOR BACKUPS
0570 &CRT CLR
0580 LOGOFF
0590 -FINMOD ; NO SPREADSHEET PROGRAM CURRENTLY INSTALLED
0600 &CRT CLEAR
0610 &BEGTYPE
0620 This function is not available on your system. Contact your dealer
0630 if you would like more information on this option.
0640 &END
0650 &CRT 25 13
0660 &BEGTYPE
0670 PRESS RETURN TO CONTINUE
0680 &END
0690 &WAIT
0700 &GOTO -START
0710 -LIST ; LIST DISK FILES OF CURRENTLY LOGGED ACCOUNT
0720 &CRT CLR
0730 &BEGTYPE
0740 This option will list all files on your currently attached
0750 disks. When you see the "up arrow" or "~" in the lower left hand
0760 corner of the screen, this means you are to press the RETURN key to
0770 to continue listing your files.
0780
0790
0800 Press RETURN KEY To Begin Listing Your Files
0810 &END
0820 &WAIT
0830 -RELIST
0840 &CRT CLR
0850 &BEGTYPE
0860 Please wait...I'm looking for your files!
0870 &END
0880 FILELIST (DATE ; EXECUTES FILELIST COMMAND WITH DATE OPTION
0890 &WAIT
0900 -ASK
0910 &CRT CLR
0920 &CRT 20 12
0930 &TYPE DO YOU WISH TO LIST MORE FILES? Y/N \
```

```

0940 &READ &SEL
0950 &IF &SEL = N &GOTO -AGAIN
0960 &IF &SEL = Y &GOTO -RELIST
0970 &IF &SEL NE Y &IF &SEL NE N &GOTO -ASK
0980 -UTIL ; TO ENABLE THIS FUNCTION MODIFY LINE 440
0985 ; CHANGE UTIL EXEC TO &GOTO -UTIL
0990 &CRT CLR
1000 &BEGTYPE
1010 In order to enter the sytem utilities menu, you must be logged on to the
1020 SYSTEM ACCOUNT. In order to enter the system account, you will be
1030 asked to type the correct password. If you do not type the correct
1040 password, you will be returned to the Main Menu.
1050
1060 PRESS RETURN TO CONTINUE
1070 &END
1080 &WAIT
1090 &CRT CLR
1100 L SYSTEM
1110 &GOTO -START
1120 &QUIT
1130 -MESS ; INTERNAL MESSAGE SYSTEM
1140 &CRT CLR
1150 &BEGTYPE ; PAINT SCREEN
1160 This option allows you to send brief messages to other users
1170 on the system. In order to use this function, you must know
1180 the account name of the person(s) you wish to send the message
1190 to. You may even request that the person you are sending a
1200 message to send YOU a reply after receiving your message.
1210
1220 1. SEND SAME MESSAGE TO ALL USERS CURRENTLY LOGGED ON
1230 2. SEND MESSAGE TO SPECIFIC USER
1240 3. RETURN TO MAIN MENU
1250 &END
1260 &CRT 15 20
1270 &CRT BELL
1280 &TYPE ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN \
1290 &READ &SEL
1300 &IF &SEL GT 3 &GOTO -MESS
1310 &IF &SEL LT 1 &GOTO -MESS
1320 &IF &SEL = 1 &GOTO -SNDALL
1330 &IF &SEL = 2 &GOTO -SNDSP
1340 &IF &SEL = 3 &GOTO -START
1350 -SNDALL ; SEND MESSAGE TO ALL USERS
1360 &CRT CLR
1370 MSG * ; EXECUTE OASIS MESSAGE COMMAND
1380 &GOTO -START
1390 -SNDSP ; SEND MESSAGES TO SPECIFIC USERS
1400 &CRT CLR
1410 &TYPE Who would you like to send a message to? \
1420 &READ &USERNAME
1430 MSG &USERNAME ; EXECUTE OASIS MESSAGE COMMAND
1440 &GOTO -START
1450 -MAIL ; RETRIEVE INTERNAL MESSAGES
1460 &CRT CLR
1470 MAILBOX ; EXECUTE OASIS MAILBOX COMMAND
1480 &CRT 5 20
1490 &BEGTYPE
1500 If screen is blank then, sorry, you have no messages waiting.
1510 &END

```

316 THEOS User's Handbook

```
1520 &CRT 25 22
1530 &BEGTYPE
1540 PRESS RETURN KEY TO CONTINUE
1550 &END
1560 &WAIT
1570 &GOTO -START
1620 -TERM ; LINE 1630 IS SET UP FOR MCI MAIL/MODEM ON PORT9
1630 ATTACH COMM1 PORT9 (W8 ; MODIFY ACCORDING TO YOUR NEEDS
1640 TERMINAL ; EXECUTE OASIS TERMINAL COMMAND
1650 ATTACH COMM1 ; DE-TACH COMM DEVICE
1660 &GOTO -START

0002 ; THIS EXEC IS USED TO SET UP THE SYSTEM IN SINGLE OR MULTI USER MODE
0003 ; COPYRIGHT 1985 BY WALTER STAGNER
0005 -START
0010 &CONTROL OFF
0020 &CRT CLR
0030 &TYPE {S}ingle or {M}ulti user mode or multi user with {N}o spooler?
0040 &READ &MODE ; IF INPUT IS ANY CHARACTER OTHER THAN N OR M
0045 ; THE SYSTEM IS AUTOMATICALLY SET UP FOR SINGLE USER MODE
0047 ; AND THE USER IS LOGGED OFF
0050 &IF &MODE = N &GOTO -SMUNOSP ; MULTI USER MODE NO SPOOLER
0060 &IF &MODE = M &GOTO -MULTI ; MULTI USER MODE
0070 ATTACH PRINTER1 PORT6 ; PRINTER MUST BE PHYSICALLY ATTACHED
0075 ; TO PORT 6 IN ORDER TO WORK OR MODIFY LINE 70
0080 &CRT CLR
0090 &BEGTYPE
0100     Now displaying disk space...
0110
0120 &END
0130 SHOW DISK ; SHOW DISK COMMAND CHECKS FOR MIS-ALLOCATIONS
0140 &BEGTYPE
0150
0160     System Ready...
0170 &END
0180 LOGOFF
0190 -MULTI ; COMMANDS FOR MULTI USER SET-UP
0200 &CRT CLR
0210 &BEGTYPE
0220     Starting multi user mode for 5 terminals...
0230 &END
0240 &CRT 1 03
0250 &BEGTYPE
0260 Attaching main printer...
0270 &END
0280 ; ATTACH PRINTER FOR MULTI-USER SPOOLER
0290 ATTACH PRINTER1 PORT6
0300 &BEGTYPE
0310 Loading spooler...
0315 ; IF VERSION OF OASIS PRIOR TO 6.1 IS USED REPLACE THIS LINE
0317 ; WITH A LOAD SPOOLER COMMAND
0320 &END
0340 &BEGTYPE
0350 Setting memory for terminal 2...
0360 &END
0370 SET MEM 2 1 1
0380 &BEGTYPE
0390 Setting memory for terminal 3...
```

```
0400 &END
0410 SET MEM 3 1 2
0420 &BEGTYPE
0430 Setting memory for terminal 4...
0440 &END
0450 SET MEM 4 1 3
0460 &BEGTYPE
0470 Setting memory for terminal 5...
0480 &END
0490 SET MEM 5 1 4
0500 SET MEM 6 3500 5 ; ALLOCATES A MEMORY PARTITION FOR THE SPOOLER
0510 &CRT CLR
0520 &BEGTYPE
0530 Memory is set for the system...
0540 &END
0550 SPOOLER INIT ; ACTIVATES THE SPOOLER
0560 &BEGTYPE
0570 Starting terminal 2...
0580 &END
0585 ; MODIFY ALL START LINES ACCORDING TO CLASS CODE REQUIRED
0587 ; FOR YOUR TERMINAL...WE ARE ASSUMING CLASS 7
0590 START 2 (PORT2 L79 P23 C7
0600 &BEGTYPE
0610 Starting terminal 3...
0620 &END
0630 START 3 (PORT3 L79 P23 C7
0640 &BEGTYPE
0650 Starting terminal 4...
0660 &END
0670 START 4 (PORT4 L79 P23 C7
0680 &BEGTYPE
0690 Starting terminal 5...
0700 &END
0710 START 5 (PORT5 L79 P23 C7
0720 SPOOLER START
0730 &CRT CLR
0740 &BEGTYPE
0750     Now displaying disk space...
0760
0770 &END
0780 SHOW DISK
0790 &BEGTYPE
0800
0810     System Ready...
0820 &END
0830 LOGOFF
0840 -SMUNOSP ; MULTI USER SET-UP WITH NO SPOOLER
0850 &CRT CLR
0860 &CRT 20 13
0870 &BEGTYPE
0880 PROCESSING MULTI-USER WITHOUT SPOOLER !!!!!
0890 &END
0900 &CRT CLR
0910 &BEGTYPE
0920 Attaching main printer...
0930 &END
0940 ;ATTACH PRINTER MULTI-USER NO SPOOLER
0950 ATTACH PRINTER1 PORT6
0960 &TYPE Setting memory for terminal 2...
```

```
0970 SET MEM 2 1 1
0980 &TYPE Setting memory for terminal 3...
0990 SET MEM 3 1 2
1000 &TYPE Setting memory for terminal 4...
1010 SET MEM 4 1 3
1020 &TYPE Setting memory for terminal 5...
1030 SET MEM 5 1 4
1040 &CRT CLR
1050 &TYPE Starting terminal 2...
1060 START 2 (PORT2 C7
1070 &TYPE Starting terminal 3...
1080 START 3 (PORT3 C7
1090 &TYPE Starting terminal 4...
1100 START 4 (PORT4 C7
1110 &TYPE Starting terminal 5...
1120 START 5 (PORT5 C7
1130 &CRT CLR
1140 &BEGTYPE
1150     Now displaying disk space...
1160
1170 &END
1180 SHOW DISK
1190 &BEGTYPE
1200
1210     System Ready...
1220 &END
1230 LOGOFF
```

```
0010 ; THIS EXEC PERFORMS SYSTEM MAINTENANCE FUNCTIONS AND EXECUTES
0020 ; OASIS OPERATING SYSTEM COMMANDS BY MAKING SELECTIONS FROM MENU
0030 ; COPYRIGHT 1985 BY WALTER STAGNER
0040 -AGAIN
0050 &CONTROL OFF
0060 -START ; PAINT SCREEN
0070 &CRT CLR
0080 &BEGTYPE
0090
0100     Ashley Communications, Inc.
0110 &END
0115 ; LINE 90 CAN BE MODIFIED WITH YOUR COMPANY NAME
0120 &SPACE 1
0130 &CRT 20,3
0140 SHOW TIME ; DISPLAYS THE CURRENT TIME AND DATE
0150 &CRT RVON
0160 &BEGTYPE
0170
0180     SYSTEM UTILITIES MENU
0180 &END
0190 &CRT RVOFF
0200
0210 &CRT 1,6
0220 &BEGTYPE
0230 1. COPY FILES
0240 &END
0250 &CRT 1,8
0260 &BEGTYPE
0270 2. ARCHIVE TO FLOPPIES/TAPE
0280 &END
0290 &CRT 1,10
0300 &BEGTYPE
```

```
0310 3. RESTORE/VERIFY FLOPPIES or TAPE
0320 &END
0330 &CRT 1,12
0340 &BEGTYPE
0350 4. ACCOUNT MAINTENANCE
0360 &END
0370 &CRT 1,14
0380 &BEGTYPE
0390 5. REPORT MIS-ALLOCATION STATUS
0400 &END
0410 &CRT 1,16
0420 &BEGTYPE
0430 6. FORMAT FLOPPIES/TAPES
0440 &END
0450 &CRT 42,6
0460 &BEGTYPE
0470 7. SINGLE USER MODE
0480 &END
0490 &CRT 42,8
0500 &BEGTYPE
0510 8. MULTI USER MODE
0520 &END
0530 &CRT 42,10
0540 &BEGTYPE
0550 9. MULTI USER MODE W/O SPOOLER
0560 &END
0570 &CRT 42,12
0580 &BEGTYPE
0590 10. ERASE FILES
0600 &END
0610 &CRT 42,14
0620 &BEGTYPE
0630 11. LIST DISK FILES
0640 &END
0650 &CRT 42,16
0660 &BEGTYPE
0670 12. RETURN TO MASTER MENU
0680 &END
0690 &CRT 24,18
0700 &BEGTYPE
0710 13. LOGOFF SYSTEM
0720 &END
0730 -SEL
0740 &CRT 15 21
0750 &CRT RVON
0760 &TYPE ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN \
0770 &CRT RVOFF
0780 &READ &SEL ; THIS LINE READS USER INPUT
0785 ; &SEL IS THE VARIABLE ASSIGNED TO THE USER INPUT
0790 ; ONLY NUMBERS BETWEEN 1 AND 13 ARE ALLOWED
0800 &IF &SEL LT 1 &GOTO -START
0810 &IF &SEL GT 13 &GOTO -SEL
0820 &IF &SEL = 1 CPY EXEC ; THIS EXEC COPIES FILES
0830 &IF &SEL = 2 SAFECPY EXEC ; THIS EXEC MAKES ARCHIVES
0840 &IF &SEL = 3 RES EXEC ; THIS EXEC RESTORES FILES
0850 &IF &SEL = 4 &GOTO -AMAIN ; USER ACCOUNT MAINTENANCE
0860 &IF &SEL = 5 &GOTO -FIX ; REPORT MIS-ALLOCATION STATUS
0870 &IF &SEL = 6 &GOTO -INIT ; INITIALIZE DISKS AND TAPES
0880 &IF &SEL = 7 &GOTO -SINGLE ; SINGLE USER MODE
```

```
0890 &IF &SEL = 8 IPL EXEC ; MULTI USER MODE
0900 &IF &SEL = 9 &GOTO -MULTI ; MULTI USER MODE NO SPOOLER
0910 &IF &SEL = 10 PURGE EXEC ; THIS EXEC ERASES FILES
0920 &IF &SEL = 11 &GOTO -FILE ; LIST DISK FILES
0930 &IF &SEL = 12 MASTER EXEC ; RETURN TO MASTER MENU
0940 &IF &SEL = 13 &GOTO -LOGOFF ; LOGOFF SYSTEM OR LOGON TO NEW ACCOUNT
0950 &GOTO -AGAIN
0960 -INIT ; INITIALIZE DISKS AND TAPES
0970 &ERROR &GOTO -ERROR ; GO TO AN ERROR HANDLING ROUTINE
0980 &CRT CLEAR
0990 &BEGTYPE ; PAINT SCREEN
1000 This program will initialize floppy disks/tapes for making backup
1010 copies of your data. Please insert a blank diskette into
1020 the disk drive with the label side up. Make sure that the
1030 write-protect tab is over the hole at the bottom. Otherwise,
1040 you will have to start over. If you are using tape please
1050 insert the tape into the tape drive.
1060 &END
1070 &SPACE 4
1080 &BEGTYPE
1090
1100
1110
1120
1130
1140 &END
1150 &CRT 16 20
1160 &TYPE ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN \
1170 &READ &SEL ; READ USER INPUT
1175 ; ONLY NUMBERS 1 THROUGH 3 ARE ALLOWED
1180 &IF &SEL GT 3 &GOTO -INIT
1190 &IF &SEL LT 1 &GOTO -INIT
1200 &IF &SEL = 1 &GOTO -ID
1210 &IF &SEL = 2 &GOTO -IT
1220 &IF &SEL = 3 &GOTO -START
1230 -ID ; BEGIN INITIALIZING PROCESS
1240 A G FLOPPY1 ; ASSUMES THE FLOPPY DRIVE IS DESIGNATED G
1250 &CRT CLR
1260 &BEGSTACK ; AUTO DATA FILL-IN FOR QUESTIONS ASKED BY OASIS
1270 Y
1280 TEST
1290 Y
1300 &END
1310 INITDISK G (FORMAT H 2 ; EXECUTE INITDISK COMMAND WITH
1315 ; THE FORMAT OPTION AND DESIGNATE DOUBLE SIDED DISK
1320 &CRT CLR
1330 &BEGTYPE
1340 Please wait while I verify your diskette....
1350 &END
1360 VERIFY G ; DISKETTE VERIFICATION
1370 &BEGTYPE
1380 Diskette verification completed!!!
1390 &END
1400 &CRT 18 12
1410 &TYPE DO YOU WISH TO FORMAT MORE FLOPPIES? Y/N \
1420 &READ &SEL
1430 &IF &SEL = N
1440 &IF &SEL = Y &GOTO -ID
1450 A G ; DE-TACH FLOPPY DISK DRIVE
```

```
1460 &GOTO -AGAIN
1470 -SINGLE ; SINGLE USER MODE
1480 &CRT CLEAR
1490 &BEGTYPE ; PAINT SCREEN
1500     This option will bring the system down to single-user operation
1510     for purposes of making ARCHIVE copies and any other system
1520     function where it is NECESSARY to be in single-user mode. In
1530     order to bring the system down to single-user mode, all users
1540     MUST BE LOGGED OFF. If this is not the case, you will see
1550     "partition still active" messages on the screen as the system
1560     ATTEMPTS to come down to single-user mode.
1570
1580
1590     If you have not succeeded in bringing the system down
1600     to single-user you will have to see to it that EVERYONE
1610     has logged off, and re-select the SINGLE USER option
1620     from the utilities menu.
1630 &END
1640 &WAIT
1650 &CRT BELL
1660 &CRT CLR
1670 SHOW USERS ; SHOW WHO IS CURRENTLY LOGGED ON
1680 &BEGTYPE
1690
1700 If you show anyone on the system OTHER THAN System or BG #6 (print
1710     spooler), then you must choose N below.
1720 &END
1730 &SPACE 3
1740 &CRT 5 16
1750 &BEGTYPE
1760 HAVE ALL USERS BEEN NOTIFIED FOR SINGLE USER MODE? (Y or N)
1770 &END
1780 &CRT 65 16
1790 &READ &A
1800 &IF &A NE Y &GOTO -NOMSG
1810 &CRT 20 13
1820 &BEGTYPE
1830 PROCESSING SINGLE USER MODE !!!!!!!!!!!!!
1840 &END
1850 SINGLE EXEC ; EXECUTE SINGLE EXEC
1860 &CRT CLR
1870 -ASK4
1880 &CRT 5 20
1890 &TYPE CONTINUE [S]INGLE USER MODE OR [M]ULTI USER WITHOUT SPOOLER? \
1895 ; LINE 1890 GIVES OPERATOR A CHOICE OF MODE OF OPERATION
1900 &READ &SEL ; READ USER INPUT
1910 &IF &SEL <> S &IF &SEL <> M &GOTO -ASK4
1920 &IF &SEL = M &GOTO -SMUNOSP
1930 &IF &SEL = S &GOTO -EXIT
1940 &GOTO -EXIT
1950 -NOMSG
1960 &CRT 5 13
1970 &BEGTYPE
1980 NOTIFYING ALL USERS - SINGLE USER MODE WITHIN 5 MINUTES !!!!!!!!!!!!!
1990 &END
2000 MSG * WARNING - SINGLE USER MODE WITHIN 5 MINUTES !!!!!!!!!!!!!
2005 ; LINE 2000 SENDS MESSAGE TO ALL CURRENTLY LOGGED USERS
2010 -EXIT ; STAY IN SINGLE USER MODE AND RETURN TO MENU
2020 &GOTO -START
```

322 THEOS User's Handbook

```
2030 -FIX ; REPORT DISK FILE ALLOCATION STATUS
2040 &CRT CLR
2050 &BEGTYPE
2060     Now displaying disk space...
2070 &END
2080 SHOW DISK ; CHECK DISK STATUS
2090 &WAIT
2100 &CRT CLR
2110 &BEGTYPE
2120 DO YOU SHOW ANY MISALLOCATIONS ?
2130 TYPE Y OR N (YES OR NO) AND PRESS RETURN
2140 &END
2150 -RSPND
2160 &CRT 41 2
2170 &READ &A
2180 &IF &A NE Y &IF &A NE N &GOTO -RSPND
2190 &IF &A NE Y &GOTO -START
2200 &GOTO -REP
2210 &SPACE 1
2220 -REP
2230 &CRT CLR
2240 &BEGTYPE
2250     You are now ready to repair your misallocated disk space. You
2260     may only repair one drive at a time AND you must be in SINGLE-USER
2270     MODE to perform the repair function.
2280 &END
2290 &SPACE 3
2300 -ENTER
2310 &CRT 10 12
2320 &TYPE Please enter the drive name to be repaired S/A/ ETC. \
2330 &READ &DRIVNAME
2340 &CRT 14 15
2350 &TYPE Drive &DRIVNAME Is The One You Wish To Repair ? \
2360 &READ &A
2370 &IF &A = N &GOTO -ENTER
2380 -CHECK
2390 &CRT 25 20
2400 &TYPE OK TO CONTINUE? Y/N \
2410 &READ &SEL
2420 &IF &SEL = N &GOTO -AGAIN
2430 &IF &SEL NE Y &IF &SEL NE N &GOTO -CHECK
2440 &CRT CLR
2450 &ERROR &GOTO -AGAIN
2460 REPAIR &DRIVNAME ; EXECUTE OASIS REPAIR COMMAND
2470 &WAIT
2480 &CRT CLR
2490 &BEGTYPE
2500     If this program has reported to you that there is space or files
2510     on the disk that are unaccounted for, the recommended method of
2520     fixing the problem is to copy (not backup) the files from the
2530     disk with the error to an empty disk (such as an initialized floppy).
2540     Then, go back and use the KILL command to remove the bad file from
2550     the disk.
2560     If you are STILL showing misallocations,
2570     it is best to follow the procedure above (ie return to utility
2580     menu and erase and copy files or contact your vendor).
2590 &END
2600 &WAIT
2610 &GOTO -FIX
```

```
2620 -COPY ; THIS FUNCTION NOT IMPLEMENTED, ENDS AT LINE 2790
2625 ; USE SELECTION ONE ON UTILITIES MENU TO COPY FILES
2630 &CRT CLR
2640 &BEGTYPE
2650 Choosing this option allows you to make copies on a file-by
2660 basis to your floppy disks for selective backup purposes.
2670 &END
2680 &SPACE 12
2690 &BEGTYPE
2700 Press Any Key To Continue
2710 &END
2720 &WAIT
2730 -MAIN
2740 &CRT CLR
2750 &TYPE Please type the account you wish to Logon to \
2760 &READ &LOG
2770 &IF &LOG = &NULL &GOTO -MAIN
2780 LOGON &LOG
2790 &GOTO -AGAIN
2800 -AMAIN ; INTERACTIVE USER ACCOUNT MAINTENANCE
2810 &CRT CLR
2820 &BEGTYPE ; PAINT SCREEN
2830 This option allows you to perform the following functions:
2840
2850 Add An Account.
2860 Change An Existing Account (Passwords,Privilege Level, etc.)
2870 Delete An Account. ***WARNING*** Please do not choose this option
2880 without contacting the SYSTEM MANAGER. Once accounts are
2890 deleted, you may NEVER again access the files belonging to that
2900 account.
2910
2920 USER ACCOUNT MAINTENANCE OPTIONS:
2930
2940 1. LIST CURRENT ACCOUNTS (PASSWORDS, etc.)
2950 2. ADD/CHANGE/DELETE ACCOUNTS
2960 3. RETURN TO UTILITY MENU
2970 &END
2980 &CRT 15 20
2990 &TYPE ENTER THE NUMBER OF YOUR CHOICE AND PRESS RETURN \
3000 &READ &SEL ; READ USER INPUT
3005 ; ONLY NUMBERS BETWEEN 1 AND 3 ARE ALLOWED
3010 &IF &SEL LT 1 &GOTO -AMAIN
3020 &IF &SEL GT 3 &GOTO -AMAIN
3030 &IF &SEL = 1 &GOTO -LIST
3040 &IF &SEL = 2 &GOTO -MOD
3050 &IF &SEL = 3 &GOTO -AGAIN
3060 -LIST ; LIST USER ACCOUNTS ON SCREEN
3070 &CRT CLR
3080 ACC (T ; EXECUTE OASIS ACCOUNT COMMAND WITH TYPE OPTION
3090 &WAIT
3100 &GOTO -AMAIN
3110 -MOD ; ENTER INTERACTIVE ACCOUNT MAINTENANCE MODE
3120 &CRT CLR
3130 ACC ; EXECUTE OASIS ACCOUNT COMMAND
3140 -CREACCT ; AUTOMATIC CREATION OF EXEC TO ACCESS MENUS
3145 ; USING THE USER1 EXEC FILE AND COPYFILE COMMAND
3150 &CRT CLR
3160 &TYPE CREATE EXEC TO ALLOW USER ACCESS TO MENUS? Y/N \
3170 &READ &SEL
```

324 THEOS User's Handbook

```
3180 &IF &SEL = N &GOTO -AMAIN
3190 &IF &SEL = Y &GOTO -CREACC1
3200 -CREACC1 ; GO AHEAD AND CREATE THE NEW ACCOUNT EXEC.
3210 &TYPE TYPE THE NAME OF THE ACCOUNT JUST ADDED \
3220 &READ &ACCOUNT
3230 COPYFILE USER1 EXEC S &ACCOUNT EXEC S ; PUT THE EXEC ON S DRIVE
3240 &TYPE ANY MORE NEW ACCOUNTS? Y/N \
3250 &READ &SEL
3260 &IF &SEL = N &GOTO -AMAIN
3270 &IF &SEL = Y &GOTO -CREACCT
3280 &GOTO -AMAIN
3290 -ERROR ; ERROR HANDLING ROUTINE FOR INITDISK COMMAND
3300 &CRT CLR
3310 &BEGTYPE
3320 -----
3330                                     THE SYSTEM HAS DETECTED AN ERROR!!
3340     If you wish to continue formatting diskettes/tapes, you must start over
3350     and reselect that option from the menu.
3360 -----
3370 &END
3380 &SPACE 10
3390 &BEGTYPE
3400                                     PRESS RETURN TO GO BACK TO MENU
3410 &END
3420 &WAIT
3430 A G ; DE-TACH FLOPPY DISK DRIVE
3440 A TAPE1 ; DE-TACH TAPE DRIVE
3450 &GOTO -START
3460 -IT ; INITIALIZE TAPES
3470 &ERROR &GOTO -ERROR ; GO TO INITTAPE ERROR ROUTINE
3480 A TAPE1 CART ; EXECUTE OASIS ATTACH COMMAND FOR TAPE DRIVE
3490 &CRT CLR
3500 &BEGTYPE
3510 Please wait .... TENSIONING TAPE
3520 &END
3530 INITTAPE (TENSION ; TENSION TAPE
3540 &BEGTYPE
3550     INITIALIZING TAPE
3560 &END
3570 INITTAPE ; EXECUTE OASIS INITTAPE COMMAND
3580 &CRT CLR
3590 &CRT 20 12
3600 &TYPE DO YOU WISH TO FORMAT MORE TAPES? Y/N \
3610 &READ &SEL
3620 &IF &SEL = N
3630 &IF &SEL = Y &GOTO -IT
3640 A TAPE1 ; DE-TACH TAPE DRIVE
3650 &GOTO -AGAIN
3660 -MULTI ; MULTI USER MODE NO SPOOLER
3670 &CRT CLR
3680 &BEGTYPE
3690     This function is useful when the System Manager needs to print
3700     out long reports, such as monthly statements, that may tie
3710     up the Spooler for long periods of time. However, before the
3720     system can be operated WITHOUT the spooler, the system must be
3730     brought down to Single User Mode and then brought up in Multi
3740     User WITHOUT the spooler.
3750 &END
3760 &WAIT
```

```
3770 &GOTO -SINGLE
3780 -SMUNOSP ; MULTI USER MODE NO SPOOLER
3790 &CRT CLR
3800 &CRT 20 13
3810 &BEGTYPE
3820 PROCESSING MULTI-USER WITHOUT SPOOLER !!!!!
3830 &END
3840 &CRT CLR
3850 &BEGTYPE
3860 Attaching main printer...
3870 &END
3880 ATTACH PRINTER1 PORT6 (L80 ENAB1
3890 &TYPE Setting memory for terminal 2...
3900 SET MEM 2 1 1
3910 &TYPE Setting memory for terminal 3...
3920 SET MEM 3 1 2
3930 &TYPE Setting memory for terminal 4...
3940 SET MEM 4 1 3
3950 &TYPE Setting memory for terminal 5...
3960 SET MEM 5 1 4
3970 &CRT CLR
3980 &TYPE Starting terminal 2...
3985 ; MODIFY THE START LINES ACCORDING TO THE CLASS CODE DRIVER
3987 ; YOUR TERMINALS USE WE ASSUME CLASS 7 IN THIS CASE
3990 START 2 (PORT2 C7
4000 &TYPE Starting terminal 3...
4010 START 3 (PORT3 C7
4020 &TYPE Starting terminal 4...
4030 START 4 (PORT4 C7
4040 &TYPE Starting terminal 5...
4050 START 5 (PORT5 C7
4060 &GOTO -START
4070 -FILE ; LIST YOUR DISK FILES
4080 A G FLOPPY1 ; ATTACH FLOPPY DISK AS DRIVE G
4090 &CRT CLR
4100 &TYPE FILENAME TO LIST \
4110 &READ &FILENAME
4120 &IF &FILENAME = &NULL &GOTO -FILE
4130 &TYPE FILETYPE TO LIST \
4140 &READ &FILETYPE
4150 &IF &FILETYPE = &NULL &GOTO -FILE
4160 &TYPE DRIVE WHICH FILE RESIDES ON? PRESS RETURN IF NOT SURE \
4170 &READ &D
4180 &SPACE
4190 -INQ ; GIVE USER A CHOICE OF WHERE TO SEND FILELIST
4200 &TYPE [H]ARD COPY [S]CREEN OR [Q]UIT? \
4210 &READ &LIST
4220 &IF &LIST <> H &IF &LIST <> S &GOTO -INQ
4230 &IF &LIST = H FILELIST &FILENAME &FILETYPE &D (PRINTER1
4240 &CRT CLR
4250 &TYPE Please wait ....
4260 FILELIST &FILENAME &FILETYPE &D
4270 &WAIT
4280 &IF &LIST = Q
4290 A G ; DE-TACH FLOPPY DISK DRIVE
4300 &GOTO -AGAIN
4310 &CRT CLR
4320 &CRT 20 14
4330 &TYPE DO YOU HAVE MORE FILES TO LIST? Y/N \
```

326 THEOS User's Handbook

```
4340 &READ &SEL
4350 &IF &SEL = Y &GOTO -FILE
4360 &IF &SEL = N
4370 ;DETACH FLOPPY
4380 A G
4390 &GOTO -AGAIN
4400 -LOGOFF ; LOGOFF SYSTEM OR LOGON TO NEW ACCOUNT
4410 &CRT CLR
4420 &SPACE4
4430 &CRT RVON
4440 &CRT BELL
4450 &BEGTYPE
4460          Have you remembered to make backups of today's work?
4470 &END
4480 &CRT RVOFF
4490 &CRT 15,13
4500 &TYPE Type M to return to MENU or L to LOGOFF the system \
4510 &READ &SEL
4520 &IF &SEL <> M &IF &SEL <> L &GOTO -LOGOFF
4530 &IF &SEL = M &GOTO -AGAIN
4540 &CRT CLR
4550 LOGOFF
```

```
0005 ; THIS EXEC BRINGS THE SYSTEM DOWN TO SINGLE USER MODE
0007 ; COPYRIGHT 1985 BY WALTER STAGNER
0010 &CONTROL OFF
0020 &CRT CLR
0030 &BEGTYPE
0040          Starting single user mode...
0050 &END
0060 &CRT 1 03
0070 &BEGTYPE
0080 STOPPING SPOOLER...
0090 &END
0100 SPOOLER STOP
0110 SPOOLER QUIT
0120 STOP 6
0130 SET MEM 6
0140 &CRT CLR
0150 &TYPE STOP USER 5...
0160
0165 ; THE LINES STARTING WITH FORCE WILL CAUSE ANY CURRENTLY
0167 ; LOGGED USER TO BE LOGGED OFF AUTOMATICALLY
0170 FORCE 5 LOGOFF
0180 &TYPE STOP USER 4...
0190
0200 FORCE 4 LOGOFF
0210 &TYPE STOP USER 3...
0220
0230 FORCE 3 LOGOFF
0240 &TYPE STOP USER 2
0250
0260 FORCE 2 LOGOFF
0270 STOP 5
0280 &CRT CLR
0290 &TYPE USER 5 STOPPED...
0300 STOP 4
0310 &TYPE USER 4 STOPPED...
```

```
0320 STOP 3
0330 &TYPE USER 3 STOPPED...
0340 STOP 2
0350 &TYPE USER 2 STOPPED...
0360 SET MEM 5
0370 &CRT CLR
0380 &TYPE CHECK MEMORY...
0390
0400 SET MEM 4
0410 SET MEM 3
0420 &CRT CLR
0430 &TYPE Almost finished now..one moment please...
0440
0450 SET MEM 2
0460 UNLOAD SPOOLER ; UNLOADS THE SPOOLER MODULE IN VERSIONS
0465 ; OF OASIS PRIOR TO 6.1
0470 A PRINTER1 PORT6 (L80,P56, ENAB1
0475 ; MODIFY LINE 470 IF ATTACHING PRINTER TO DIFFERENT PORT
0490 UTIL EXEC ; RETURN USER TO SYSTEM UTILITY MENU

0010 ; THIS EXEC IS USED TO RESTORE PREVIOUSLY ARCHIVED FILES.
0020 ; COPYRIGHT 1985 BY WALTER STAGNER
0030 &CONTROL OFF
0040 -START
0050 &ERROR &IF &RETCODE LT 16 &GOTO -ERROR ; ERROR HANDLING ROUTINE
0060 &ERROR &IF &RETCODE GT 16 &GOTO -ERROR ; ERROR HANDLING ROUTINE
0070 &ERROR &IF &RETCODE = 16 &GOTO -NEXT ; RESTORED FILES OK?
0080 &CRT CLR
0090 &BEGTYPE ; PAINT SCREEN
0100 In order to RESTORE, you must be in SINGLE-USER mode.
0110 In addition, you must have your ARCHIVED VOLUMES (EITHER TAPES OR
0120 FLOPPIES) ready to insert into the tape (or disk drive) when
0130 prompted to do so. If you need to bring the system down to SINGLE-
0140 USER mode, please choose the option to return to the System
0150 Utilities Menu.
0160
0170 1. RETURN TO SYSTEM UTILITIES MENU
0180
0190 2. CONTINUE WITH RESTORE PROCEDURES
0200 &END
0210 &CRT 26,16
0220 &TYPE What is your Choice? \
0230 &READ &SEL ; READ USER INPUT
0240 &IF &SEL GT 2 &GOTO -START
0250 &IF &SEL LT 1 &GOTO -START
0260 &IF &SEL = 1 UTIL EXEC
0270 &IF &SEL = 2 &GOTO -CONT
0280 -ERROR ; ERROR HANDLING ROUTINE
0290 &CRT CLR
0300 &CRT BELL
0310 &BEGTYPE
0320 -----
0330
0340 THE SYSTEM HAS DETECTED AN ERROR!!
0350 If you wish to continue the RESTORE procedure, you must start
0360 over and retry the operation. Type YES for a RETRY, anything
0370 else to return to the System Utilities Menu.
0380
```

```

0390 -----
0400 &END
0410 &SPACE
0420 &READ &YN
0430 &IF &YN NE YES
0440 A G ; DE-TACH FLOPPY DISK DRIVE
0450 A TAPE1 ; DE-TACH TAPE DRIVE
0460 &QUIT
0470 &IF &YN = YES &GOTO -START
0480 -CONT
0490 &CRT CLR
0500 &CRT 20 12
0510 &TYPE RESTORE FROM [T]APE OR [F]LOPPY? \
0520 &READ &SEL ; CHOOSE BETWEEN TAPE OR FLOPPY DISK
0530 &IF &SEL <> T &IF &SEL <> F &GOTO -CONT
0540 &IF &SEL = T &GOTO -TAPE
0550 &IF &SEL = F &GOTO -FLOP
0560 -FLOP ; RESTORE FROM FLOPPY DISK
0570 &CRT CLR
0580 &BEGTYPE
0590     You will now be able to specify HOW you wish to restore.
0600     The options should be self-explanatory. The RESTORE/VERIFY
0610     ARCHIVE option is used to VERIFY the integrity of the
0620     ARCHIVE just made. If you have a multi-volume archive
0625     this procedure will take a few minutes and you must be
0627     present to change disks when prompted to do so.
0630
0640
0650
0660
0670
0680
0690
0700
0710
0720 &END
0730 &CRT 26,17
0740 &TYPE What Is Your Choice? \
0750 &READ &SEL ; READ USER INPUT
0760 &IF &SEL LT 1 &GOTO -FLOP
0770 &IF &SEL GT 4 &GOTO -FLOP
0780 &IF &SEL = 1 &GOTO -COMFLOP
0790 &IF &SEL = 2 &GOTO -SELECT
0800 &IF &SEL = 3 &GOTO -ACC
0810 &IF &SEL = 4 &GOTO -RESVER
0820 -COMFLOP ; RESTORE ALL FILES OPTION
0830 &CRT CLR
0840 &TYPE Which drive is to RECEIVE the restored files [S/A/B ETC.] \
0850 &READ &D
0860 &IF &D = &NULL &GOTO -COMFLOP
0870 &CRT BELL
0880 &CRT 15,12
0890 &TYPE ALL FILES ON YOUR FLOPPY WILL BE RESTORED TO DRIVE &D
0900 -ASK2 ; GIVE USER A CHANCE TO MAKE NEW SELECTIONS
0910 &CRT BELL
0920 &CRT 10,23
0930 &TYPE [C] TO CONTINUE [S] TO SELECT NEW DRIVE OR [Q] TO QUIT \
0940 &READ &IC
0950 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASK2

```

```
0960 &IF &IC = Q &QUIT
0970 &IF &IC = S &GOTO -COMFLOP
0980 A G FLOPPY1 ; ATTACH FLOPPY DISK AS DRIVE G
0990 &CRT CLR
1000 &TYPE Please wait ....
1010 RESTORE *.* G &D (REPLACE ; RESTORE ALL FILES ON G TO NEW DRIVE
1020 &WAIT
1030 &CRT CLR
1040 &CRT BELL
1050 &CRT 20,14
1060 &TYPE RESTORE IS COMPLETE PRESS RETURN TO GO TO MENU \
1070 &WAIT
1080 A G ; DE-TACH FLOPPY DISK
1090 &QUIT
1100 -TAPE ; RESTORE ALL FILES FROM TAPE
1110 &CRT CLR
1120 &TYPE Which drive is to RECEIVE the restored files [S/A/B ETC.] \
1130 &READ &D
1140 &IF &D = &NULL &GOTO -TAPE
1150 &CRT BELL
1160 &CRT 12,12
1170 &TYPE ALL FILES ON TAPE1 WILL BE RESTORED TO DRIVE &D
1180 -ASK3 ; GIVE USER A CHANCE TO MAKE NEW SELECTIONS
1190 &CRT BELL
1200 &CRT 10,23
1210 &TYPE [C] TO CONTINUE [S] TO SELECT NEW DRIVE OR [Q] TO QUIT \
1220 &READ &IC
1230 &IF &IC <> C &IF &IC <> S &IF &IC <> Q &GOTO -ASK3
1240 &IF &IC = Q &QUIT
1250 &IF &IC = S &GOTO -TAPE
1260 A TAPE1 CART ; ATTACH TAPE DRIVE
1270 &CRT CLR
1280 RESTORE *.* TAPE1 &D (REPLACE ; RESTORE ALL FILES FROM TAPE1
1290 &WAIT
1300 &CRT CLR
1310 &CRT BELL
1320 &CRT 20,14
1330 &TYPE RESTORE IS COMPLETE PRESS RETURN TO GO TO MENU \
1340 &WAIT
1350 A TAPE1 ; DE-TACH TAPE DRIVE
1360 &QUIT
1370 -RESVER ; RESTORE/VERIFY ARCHIVE
1375 ; THIS OPTION VERIFIES THE INTEGRITY OF AN ARCHIVE BY
1377 ; TRYING TO RESTORE FILES BUT BECAUSE THE (NEWFILE
1378 ; OPTION IS USED NO FILES ARE ACTUALLY RESTORED
1380 &CRT CLR
1390 &TYPE Which drive do you wish to RESTORE from [A/B/G ETC.] \
1400 &READ &D
1410 &IF &D = &NULL &GOTO -RESVER
1420 &CRT BELL
1430 &CRT 15,12
1440 &TYPE ALL FILES ON DRIVE &D WILL NOW BE VERIFIED
1450 -ASK ; GIVE USER A CHANCE TO SELECT NEW OPTIONS
1460 &CRT BELL
1470 &CRT 5,23
1480 &TYPE [C] TO CONTINUE [S] TO SELECT NEW DRIVE OR [Q] TO QUIT \
1490 &READ &IC
1500 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASK
1510 &IF &IC = Q &QUIT
```

```
1520 &IF &IC = S &GOTO -RESVER
1530 ATTACH G FLOPPY1 ; ATTACH FLOPPY DRIVE AS G
1540 &CRT CLR
1550 &TYPE Please wait while I verify your files...
1560 &SPACE
1570 RESTORE &D S (NEWFILE NOQUERY ; RESTORE FILES WITH (NEWFILE OPTION
1580 -NEXT ; THE FOLLOWING STEPS ARE EXECUTED PROVIDED THAT
1585 ; ARCHIVE VERIFICATIONS WAS SUCCESSFUL
1590 &CRT CLR
1600 &CRT BELL
1610 &CRT 20 14
1620 &TYPE VERIFY IS COMPLETE PRESS RETURN TO GO TO MENU
1630 &WAIT
1640 A G ; DE-TACH FLOPPY DISK DRIVE
1650 &QUIT
1660 -SELECT ; THIS FEATURE NOT IMPLEMENTED
1670 &CRT CLEAR
1680 &CRT 25 15
1682 &BEGTYPE
1683 THIS FEATURE NOT YET IMPLEMENTED
1684 &END
1685 &WAIT
1690 &GOTO -START
1700 -ACC
1710 &CRT CLEAR
1715 &CRT 25 15
1717 &BEGTYPE
1718 THIS FEATURE NOT YET IMPLEMENTED
1728 &END
1730 &WAIT
1740 &GOTO -START
1750 ; SINCE OPTIONS 3 AND 4 HAVE YET TO BE IMPLEMENTED
1760 ; USE THE SAFECY EXEC AS A MODEL AND TRY TO IMPLEMENT
1770 ; THESE FUNCTIONS YOURSELF
1780 ; THE SAFECY EXEC CAN BE USED AS A MODEL SINCE THE
1790 ; ARCHIVE AND RESTORE OPERATIONS ARE MIRROR IMAGES
1800 ; OF EACH OTHER
1810 ; THESE OPTIONS HAVE BEEN IMPLEMENTED IN THE SAFECY EXEC

0010 ;THIS EXEC IS USED TO MAKE ARCHIVES
0020 ;COPYRIGHT 1985 BY WALTER STAGNER
0030 &CONTROL OFF
0040 -START
0050 &ERROR &GOTO -ERROR
0060 &CRT CLR
0070 &BEGTYPE ; PAINT SCREEN
0080 In order to make an ARCHIVE, you must be in SINGLE-USER mode.
0090 In addition, you must have INITIALIZED FLOPPIES OR TAPES ready to
0100 insert into the disk (or tape) drive when prompted to do so. If you
0110 need to INITIALIZE FLOPPIES OR TAPES please choose the option to return
0120 to the System Utilities Menu. In addition if you need to bring the
0130 system down to SINGLE-USER mode, please choose the option to
0140 return to the System Utilities Menu.
0150
0160 1. RETURN TO SYSTEM UTILITIES MENU
0170
0180 2. CONTINUE WITH ARCHIVE PROCEDURES
0190 &END
```

```

0200 &CRT 26,16
0210 &TYPE What is your Choice?
0220 &READ &SEL
0230 &IF &SEL GT 2 &GOTO -START
0240 &IF &SEL LT 1 &GOTO -START
0250 &IF &SEL = 1 UTIL EXEC
0260 &IF &SEL = 2 &GOTO -CONT
0270 -ERROR ; ERROR HANDLING ROUTINE
0280 &CRT CLR
0290 &CRT BELL
0300 &BEGTYPE
0310 -----
0320                               THE SYSTEM HAS DETECTED AN ERROR!!
0330                               If you wish to continue the ARCHIVE procedure, you must start
0340                               over and retry the operation. Type YES FOR A RETRY, anything
0350                               else to return to the System Utilities Menu.
0360 -----
0370 &END
0380 &SPACE
0390 &READ &YN
0400 &IF &YN NE YES
0410 A G
0420 A TAPE1
0430 &QUIT
0440 &IF &YN = YES &GOTO -START
0450 -CONT ; CONTINUE WITH ARCHIVE PROCEDURE
0460 &CRT CLR
0470 &BEGTYPE
0480     The ARCHIVE is useful because it allows you to make copies of
0490     files that would normally be too large to fit on just one
0500     floppy disk. Once you have made your initial choices, the
0510     system prompts you on what you are to do next.
0520
0530             ***** SPECIAL NOTES *****
0540
0550     1. THE FLOPPIES/TAPES YOU USE FOR THE ARCHIVE SHOULD NOT CONTAIN
0560     ANY PREVIOUS FILES YOU ARE USING AS THEY WILL BE ERASED DURING
0570     THE ARCHIVE.
0580     2. DEPENDING UPON THE SIZE OF YOUR FILES, YOU MAY HAVE TO USE
0590     MORE THAN 1 FLOPPY DISC (IF YOU ARE ARCHIVING TO FLOPPY).
0600 &END
0610 &CRT 20 15
0620 &TYPE ARCHIVE TO [T]APE OR [F]LOPPY? \
0630 &READ &SEL ; READ USER INPUT
0640 &IF &SEL <> T &IF &SEL <> F &GOTO -CONT
0650 &IF &SEL = T &GOTO -TAPE
0660 &IF &SEL = F &GOTO -FLOP
0670 -FLOP ; ARCHIVE TO FLOPPY DISK
0680 &CRT CLR
0690 &BEGTYPE
0700     You will now be able to specify HOW you wish to archive. The most
0710     economical way of making an archive is to select only those files
0720     that have been modified ON OR AFTER THE DATE YOU ENTER. However,
0730     ONCE A MONTH we recommend that you make a complete archive. If you
0740     are making a COMPLETE archive, have plenty of initialized floppies
0750     ready.
0760
0770             1. COMPLETE ARCHIVE
0780

```

332 THEOS User's Handbook

```
0790                 2.  ARCHIVE YOUR ACCOUNT BY DATE
0800
0810                 3.  ARCHIVE ONLY YOUR ACCOUNT
0820 &END
0830 &CRT 26,16
0840 &TYPE What Is Your Choice? \
0850 &READ &SEL
0860 &IF &SEL LT 1 &GOTO -FLOP
0870 &IF &SEL GT 3 &GOTO -FLOP
0880 &IF &SEL = 1 &GOTO -COMFLOP ; ARCHIVE EVERYTHING
0890 &IF &SEL = 3 &GOTO -ACC ; ARCHIVE ONLY CURRENTLY LOGGED ACCOUNT
0900 -BEGIN ; ARCHIVE FILES SINCE THE LAST ARCHIVE
0910 &CRT CLR
0920 &TYPE Please type the date of your last archive in the format MM/DD/YY \
0930 &READ &ARDATE
0940 &IF &ARDATE = &NULL &GOTO -BEGIN
0950 &OK = &LEN &ARDATE
0960 &IF &OK GT 8 &GOTO -BEGIN
0970 -BEGIN1
0980 &TYPE Which drive do you wish to ARCHIVE from [S/A/B ETC.] \
0990 &READ &D
1000 &IF &D = &NULL &GOTO -BEGIN1
1010 &CRT BELL
1020 &CRT 5,12
1030 &TYPE ALL FILES DATED ON OR AFTER &ARDATE WILL BE ARCHIVED FROM &D TO G
1040 -ASK ; PERMIT USER TO MAKE NEW SELECTIONS
1050 &CRT BELL
1060 &CRT 5,23
1070 &TYPE [C] TO CONTINUE ARCHIVE [S] TO SELECT NEW DATE OR DRIVE [Q] TO QUIT \
1080 &READ &IC
1090 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASK
1100 &IF &IC = Q &QUIT
1110 &IF &IC = S &GOTO -BEGIN
1120 A G FLOPPY1 ; ATTACH FLOPPY DISK AS DRIVE G
1130 &CRT CLR
1140 &TYPE Please wait ....
1150 ARCHIVE &D G (&ARDATE ACCOUNT VERIFY NOQUERY
1155 ; LINE 1150 EXECUTES OASIS ARCHIVE COMMAND
1160 &CRT CLR
1170 &CRT BELL
1180 &CRT 20 14
1190 &TYPE ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU
1200 &WAIT
1210 A G ; DE-TACH FLOPPY DISK DRIVE G
1220 &QUIT
1230 -COMFLOP ; ARCHIVE EVERYTHING
1240 &CRT CLR
1250 &TYPE Which drive do you wish to ARCHIVE from [S/A/B ETC.] \
1260 &READ &D
1270 &IF &D = &NULL &GOTO -COMFLOP
1280 &CRT BELL
1290 &CRT 12,12
1300 &TYPE ALL FILES ON DRIVE &D WILL BE ARCHIVED TO DRIVE G
1310 -ASK2 ; ALLOW USER TO MAKE NEW SELECTIONS
1320 &CRT BELL
1330 &CRT 5,23
1340 &TYPE [C] TO CONTINUE ARCHIVE [S] TO SELECT NEW DRIVE [Q] TO QUIT \
1350 &READ &IC
1360 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASK2
```

```

1370 &IF &IC = Q &QUIT
1380 &IF &IC = S &GOTO -COMFLOP
1390 A G FLOPPY1 ; ATTACH FLOPPY AS DRIVE G
1400 &CRT CLR
1410 &TYPE Please wait ....
1420 ARCHIVE &D G (NOQ VERIFY
1425 ; LINE 1420 EXECUTES OASIS ARCHIVE COMMAND
1430 &CRT CLR
1440 &CRT BELL
1450 &CRT 20 14
1460 &TYPE ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU
1470 &WAIT
1480 A G ; DE-TACH FLOPPY DISK DRIVE G
1490 &QUIT
1500 -TAPE ; ARCHIVE TO TAPE
1510 &CRT CLR
1520 &BEGTYPE
1530     You will now be able to specify HOW you wish to archive. The most
1540     economical way of making an archive is to select only those files
1550     that have been modified ON OR AFTER THE DATE YOU ENTER. However,
1560     once a month we recommend that you make a complete archive. If you
1570     are making a complete archive, have an extra initialized tape
1580     cartridge handy.
1590
1600                                     1. COMPLETE ARCHIVE
1610
1620                                     2. ARCHIVE BY DATE
1630 &END
1640 &CRT 26,16
1650 &TYPE What Is Your Choice? \
1660 &READ &SEL
1670 &IF &SEL LT 1 &GOTO -TAPE
1680 &IF &SEL GT 2 &GOTO -TAPE
1690 &IF &SEL = 1 &GOTO -COMTAPE
1700 -BEGTAPE ; ARCHIVE TO TAPE USING DATE OPTION
1710 &CRT CLR
1720 &TYPE Please type the date of your last archive in the format MM/DD/YY \
1730 &READ &ARDATE
1740 &IF &OK GT 8 &GOTO -BEGTAPE
1750 &IF &ARDATE = &NULL &GOTO -BEGTAPE
1760 -ASKTAPE1
1770 &TYPE Which drive do you wish to archive from [S/A/B ETC.] \
1780 &READ &D
1790 &IF &D = &NULL &GOTO -ASKTAPE1
1800 &CRT BELL
1810 &CRT 5,12
1820 &TYPE ALL FILES DATED ON OR AFTER &ARDATE WILL BE ARCHIVED FROM &D TO TAPE
1830 -ASKTAPE ; ALLOW USER TO MAKE NEW SELECTIONS
1840 &CRT BELL
1850 &CRT 5,23
1860 &TYPE [C] TO CONTINUE ARCHIVE [S] TO SELECT NEW DATE OR DRIVE [Q] TO QUIT
1870 &READ &IC
1880 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASKTAPE
1890 &IF &IC = Q &QUIT
1900 &IF &IC = S &GOTO -BEGTAPE
1910 A TAPE1 CART ; ATTACH TAPE DRIVE
1920 &CRT CLR
1930 &TYPE Please wait ....
1940 ARCHIVE &D CART (&ARDATE VERIFY NOQUERY

```

```
1945 LINE 1940 EXECUTES OASIS ARCHIVE TO TAPE COMMAND
1950 &CRT BELL
1960 &CRT 20 14
1970 &TYPE ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU
1980 &WAIT
1990 A TAPE1 ; DE-TACH TAPE DRIVE
2000 &QUIT
2010 -COMTAPE ; ARCHIVE EVERYTHING TO TAPE
2020 &CRT CLR
2030 &TYPE Which drive do you wish to ARCHIVE from [S/A/B ETC.] \
2040 &READ &D
2050 &IF &D = &NULL &GOTO -COMTAPE
2060 &CRT BELL
2070 &CRT 12,12
2080 &TYPE ALL FILES ON DRIVE &D WILL BE ARCHIVED TO TAPE
2090 -ASK3
2100 &CRT BELL
2110 &CRT 8,23
2120 &TYPE [C] TO CONTINUE ARCHIVE [S] TO SELECT NEW DRIVE [Q] TO QUIT \
2130 &READ &IC
2140 &IF &IC <> C &IF &IC <> Q &IF &IC <> S &GOTO -ASK3
2150 &IF &IC = Q &QUIT
2160 &IF &IC = S &GOTO -COMTAPE
2170 A TAPE1 CART
2180 &CRT CLR
2190 &TYPE Please wait ....
2200 ARCHIVE &D TAPE1 (NOQ VERIFY)
2210 &CRT CLR
2220 &CRT BELL
2230 &CRT 20 14
2240 &TYPE ARCHIVE IS COMPLETE, PRESS RETURN TO GO TO MENU
2250 &WAIT
2260 A TAPE1
2270 &QUIT
2280 -ACC ; ARCHIVE TO FLOPPY FILES BELONGING TO CURRENTLY LOGGED ACCOUNT
2290 &CRT CLR
2300 &TYPE Which drive do you wish to ARCHIVE from [S/A/B/ ETC.] \
2310 &READ &D
2320 &CRT BELL
2330 &SPACE
2340 &TYPE ALL FILES THAT BELONG TO YOUR ACCOUNT ON DRIVE &D WILL BE ARCHIVED TO
2350 -ASK5
2360 &CRT BELL
2370 &CRT 10 23
2380 &TYPE PRESS [C] TO CONTINUE ARCHIVE OR [Q] TO QUIT \
2390 &READ &IC
2400 &IF &IC <> C &IF &IC <> Q &GOTO -ASK5
2410 &IF &IC = Q &QUIT
2420 A G 1
2430 &CRT CLR
2440 ARCHIVE *.* &D G (ACCOUNT NOQ VERIFY)
2445 ; LINE 2440 EXECUTES OASIS ARCHIVE COMMAND
2450 &CRT CLR
2460 &CRT BELL
2470 &CRT 20 14
2480 &TYPE ARCHIVE IS COMPLETE PRESS RETURN TO GO TO MENU
2490 &WAIT
2500 A G
2510 &QUIT
```

```

0005 ; THIS EXEC IS USED WITH MULTIWRITE WORD PROCESSING
0007 ; TO SPEED SELECTION OF FILES TO EDIT
0008 ; COPYRIGHT 1985 BY WALTER STAGNER
0010 -START
0020 &CONTROL OFF
0030 &CRT CLR
0040 &TYPE MWRITE FILE TO EDIT or RETURN TO GO TO MWRITE MENU [Q] TO QUIT \
0045 ; LINE 40 REQUESTS THE USER TO TYPE THE NAME OF FILE TO EDIT
0050 &READ &SEL ; READ USER INPUT AND ASSIGN VALUE TO &SEL VARIABLE
0060 &IF &SEL = Q &QUIT ; RETURN TO OASIS OR MASTER MENU
0070 MWRITE &SEL ; RUN MULTIWRITE PROGRAM AND EDIT USER ENTERED FILE
0080 &GOTO -START

0010 ; THIS EXEC IS USED TO COPY FILES IN THEIR ENTIRITY
0020 ; COPYRIGHT 1985 BY WALTER STAGNER
0030 &CONTROL OFF
0040 -START
0050 &ERROR &GOTO -ERROR ; GO TO ERROR HANDLING ROUTINE
0060 &CRT CLEAR
0070 &BEGTYPE ; PAINT SCREEN
0080      This option will allow you to make backup copies of your files.
0090      either on a file by file basis, or all files. In order to make
0100      these copies, you will be prompted by a series of questions
0110      on the screen to make it easy and convenient for you. But first,
0120      some special notes:
0130      1. Your floppy disk drive is designated as G.
0140      2. Your main system disk is labelled as S. Subsequent platters
0150      are labelled as [A,B,C, etc.].
0160      3. If you just want to copy all of your files, just type * when
0170      prompted to do so. Otherwise, copying is on a file-by-file basis.
0180
0190 You may copy either from the hard disk to floppy or from one hard disk
0200 platter to another. However, if you are copying from hard disk to floppy
0210 disks you must first INSERT AN INITIALIZED DISK INTO THE FLOPPY DRIVE.
0220 &END
0230 &CRT 8 24
0240 &CRT BELL
0250 &TYPE PRESS [C] TO CONTINUE OR [F] IF YOU NEED TO FORMAT FLOPPIES \
0260 &READ &IC
0270 &IF &IC <> F &IF &IC <> C &GOTO -START
0280 &IF &IC = F &QUIT
0290 -BEGIN
0300 &CRT CLR
0310 &TYPE What is the file name you wish to copy? [* for all files] \
0320 &READ &FILENAME ; READ USER INPUT
0330 &OK = &LEN &FILENAME ; CHECK FOR PROPER VARIABLE LENGTH
0340 &IF &OK GT 8 &GOTO -BEGIN
0350 &TYPE What is the filetype to go with the above name? [* for all files] \
0360 &READ &FILETYPE
0370 &OK = &LEN &FILETYPE
0380 &IF &OK GT 8 &GOTO -BEGIN
0390 &TYPE Which disk are you copying from? [S-A-G/ ETC.] \
0400 &READ &D
0410 &TYPE Which disk are you copying to? [S-A-G / ETC.] \
0420 &READ &DE
0430 &CRT BELL

```

```

0440 &CRT 12 14
0450 &TYPE &FILENAME &FILETYPE WILL BE COPIED FROM DRIVE &D TO DRIVE &DE
0460 &BEGTYPE
0470
0480             [* * MEANS THAT ALL FILES WILL BE COPIED]
0490 &END
0500 &CRT 5 24
0510 &TYPE TYPE [C] TO CONTINUE COPY [S] TO SELECT NEW FILES [Q] TO QUIT \
0515 ; LINE 510 ALLOWS USER TO MAKE NEW ENTRIES
0520 &READ &IC
0530 &IF &IC = S &GOTO -BEGIN
0540 &IF &IC = Q &GOTO -QUIT
0550 &IF &IC = C &GOTO -STCPY
0560 -QUIT
0570 A G
0580 &QUIT
0590 -STCPY ; START THE COPYING PROCESS
0600 A G FLOPPY1 ; ATTACH FLOPPY DISK AS DRIVE G
0610 &CRT CLR
0620 &TYPE Please wait ...
0630 COPYFILE &FILENAME &FILETYPE &D = = &DE (REPLACE VERIFY NOQ
0635 ; LINE 630 EXECUTES OASIS COPYFILE COMMAND WITH OPTIONS
0640 &WAIT
0650 &CRT CLR
0660 &CRT 20 14
0670 &TYPE DO YOU WISH TO MAKE MORE COPIES? Y/N \
0680 &READ &SEL
0690 &IF &SEL = Y &GOTO -BEGIN
0700 &IF &SEL = N
0710 ;DETACH FLOPPY1
0720 A G
0730 UTIL EXEC
0740 -ERROR ; ERROR HANDLING ROUTINE
0750 &CRT CLR
0760 &CRT BELL
0770 &BEGTYPE
0780 -----
0790             THE SYSTEM HAS DETECTED AN ERROR!!
0800             If you wish to continue making copies you must start over and
0810             retry the copy operation. Type YES for a Retry, anything else
0820             to return to the System Utilities Menu.
0830 -----
0840 &END
0850 &SPACE
0860 &READ &YN
0870 &IF &YN NE YES UTIL EXEC
0880 &IF &YN = YES &GOTO -START

0010 ; THIS EXEC IS USED TO ERASE FILES
0020 ; COPYRIGHT 1985 BY WALTER STAGNER
0030 -AGAIN
0040 &CONTROL OFF
0050 -START
0060 &CRT CLR
0070 &BEGTYPE ; PAINT SCREEN
0080             This program will allow you to erase files from your hard disk
0090             that are no longer useful to you...such as old data files.
0100             Be certain that the files no longer

```

```
0110         have any use to you because once you ERASE them, they CANNOT
0120         be recovered. It is suggested that if you are unsure about
0130         erasing a particular file, don't erase it!
0140
0150         In order to erase a file, you must SPECIFY BOTH the
0160
0170                 FILENAME
0180
0190                 AND
0200
0210                 FILETYPE
0220
0230         and specify the DRIVE which the file resides on.
0240
0250         You will always be allowed to change your mind before
0260         erasing a file.
0265
0270         Entries must be made in response to the prompts.
0280         Otherwise you will not be permitted to continue.
0290 &END
0300 &WAIT
0310 -BEGIN
0320 &CRT CLR
0330 &TYPE WOULD YOU LIKE TO LIST YOUR FILES? Y/N \
0340 &READ &SEL ; READ USER INPUT
0350 &IF &SEL = Y &GOTO -FILE
0360 &IF &SEL = N
0370 &IF &SEL NE Y &IF &SEL NE N &GOTO -BEGIN
0380 -ERASE ; BEGIN ERASE PROCEDURE
0390 &CRT CLR
0400 &TYPE FILENAME TO ERASE \
0410 &READ &FILENAME ; ASSIGN FILENAME INPUT TO VARIABLE &FILENAME
0420 &IF &FILENAME = &NULL &GOTO -BEGIN
0430 &TYPE FILETYPE TO ERASE \
0440 &READ &FILETYPE ; ASSIGN FILETYPE INPUT TO VARIABLE &FILETYPE
0450 &IF &FILETYPE = &NULL &GOTO -BEGIN
0460 ; SOME ENTRY OTHER THAN [RETURN] MUST BE MADE
0470 &TYPE DRIVE WHICH FILE RESIDES ON \
0480 &READ &DRIVNAME ; ASSIGN DRIVE INPUT TO VARIABLE &DRIVNAME
0490 &CRT BELL
0500 &CRT 15 12
0510 &CRT RVON
0520 &TYPE &FILENAME &FILETYPE WILL BE ERASED FROM DRIVE &DRIVNAME
0525 ; LINE 520 DISPLAYS THE FILENAME FILETYPE AND DRIVNAME TO ERASE
0530 &CRT 58 12
0540 &CRT RVOFF
0550 -ASK ; ALLOW USER TO MAKE NEW ENTRY
0560 &CRT 10 23
0570 &TYPE [E] TO ERASE FILES [S] TO SELECT NEW FILES OR [Q] TO QUIT \
0580 &READ &IC
0590 &IF &IC <> E &IF &IC <> S &IF &IC <> Q &GOTO -ASK
0600 &IF &IC = S &GOTO -ERASE
0610 &IF &IC = Q &QUIT
0620 &CRT CLR
0630 &TYPE Please wait ....
0640 ERASE &FILENAME &FILETYPE &DRIVNAME (QUERY
0650 ; LINE 590 EXECUTES OASIS ERASE COMMAND WITH QUERY OPTION
0660 -ASK2
0670 &CRT CLR
```

```
0680 &CRT 25 14
0690 &TYPE MORE FILES TO ERASE? Y/N \
0700 &READ &SEL
0710 &IF &SEL <> Y &IF &SEL <> N &GOTO -ASK2
0720 &IF &SEL = Y &GOTO -ERASE
0730 &IF &SEL = N &QUIT
0740 -FILE
0750 &CRT CLR
0760 A G FLOPPY1
0770 &TYPE FILENAME TO LIST \
0780 &READ &FILENAME
0790 &IF &FILENAME = &NULL &GOTO -FILE
0800 &TYPE FILETYPE TO LIST \
0810 &READ &FILETYPE
0820 &IF &FILETYPE = &NULL &GOTO -FILE
0830 &TYPE DRIVE WHICH FILE RESIDES ON? RETURN IF NOT SURE \
0840 &READ &D
0850 &SPACE
0860 &TYPE [H]ARD COPY OR [S]CREEN? \
0870 &READ &LIST
0880 &IF &LIST NE H &IF &LIST NE S &GOTO -ASK
0890 &IF &LIST = H FILELIST &FILENAME &FILETYPE &D (PRINTER)
0900 &IF &LIST = S FILELIST &FILENAME &FILETYPE &D
0910 &WAIT
0920 &CRT CLR
0930 &CRT 20 14
0940 &TYPE DO YOU HAVE MORE FILES TO LIST? Y/N \
0950 &READ &SEL ; READ USER INPUT
0960 &IF &SEL = Y &GOTO -FILE
0970 &IF &SEL = N
0980 ;DETACH FLOPPY
0990 A G
1000 &GOTO -ERASE
```

Appendix F. THEOS8 Terminals

The list below, provided by THEOS Software Corp., indicates those terminals which THEOS8 currently supports. Some software companies may supply class codes for other terminals not listed here. These files are supplied as standard equipment with THEOS8 and are stored on the system disk with a file description of:

SYSTEM.CLASSnn

where SYSTEM is the filename, CLASSnn is the filetype and nn is the code number.

Code	Terminal Manufacturer and Model
1	Beehive B100
2	ADDS 580 Envoy
3	Lear Siegler ADM 3A
4	Soroq IQ 120
5	Special (VDM)
6	Hazeltine 1500
7	Televideo 912/920 Kimtron ABS85
8	Perkin Elmer FOX
9	Perkin Elmer BANTOM
10	Heath H19
11	Infoton 100
12	Vector Graphic Flashwriter (VDM)
13	Tandy Radio Shack TRS80 II (VDM)
14	Elbitt VDU 1920
15	ADDS Regent 20
16	Qume Sprint 5
17	Diablo 16 (printer)
18	DTC 302
19	NEC Spinwriter 1510 (printer)
20	Informer 304
21	Data Media
22	IBM 3101 (ESC ESC for system control)

Code	Terminal Manufacturer and Model
23	S D Systems VDB 8024 (VDM)
24	Intertube
25	Volker Craig VC404
26	Teleray 1000 series
27	Tektronix 40
28	Visual 200
29	Heath H19 (Heath mode) Uses ESC ESC for system control
30	Wyse 50, Zentec Zepher
31	Televideo 950, Kimtron ABM85H, Kimtron KT7 Use ESC ESC for system control
32	Hazeltine Modular One
33	Kontron FS 180 (VDM)
34	Visual 200 (SD Systems)
35	Lear Siegler ADM 5
36	Televideo 910
37	Falco Data Products TS-1
38	Wyse 100 (Altos 1)
39	DEC VT-52 (Uses ESC ESC for system control)
40	ADDS Viewpoint
42	Ampex Dialogue 80
43	Altos II
44	Televideo 924
45	Vucom 1
48	Seiko 8620
49	Ampex D81/D125/D150/D175
52	Geveke Visa 30
81	Hazeltine Esprite
82	Qume QVT 102
85	Liberty Freedom 100

Figure F.1. Listing in class code order of THEOS8 class codes for terminals.

Notes

- Some printers require class codes for proper operation. Printer class codes are indicated by (printer).
- Some class codes not written specifically for a terminal may work acceptably such as class 7 and class 31. For both of these class codes we have indicated other terminals which can use these codes.
- If you plan to use a terminal not on this list, check with your vendor to be certain it will work with your system. If not a class code can be written for you by a THEOS programmer.
- Some computer hardware vendors may add other class codes to their implementation of THEOS.
- Whether or not a class code is available for a terminal is a good indication of its popularity in the marketplace.

Index

A

Abbreviations 46
Account add 65
ACCOUNT command 61-67
Account deletion 66
Account history 32
Account maintenance 64-67
Account quit 66
ARCHIVE command 73-78
ASSIST.EXEC 215-215
ATTACH command 67-73

B

Backup 73-78
Bootstrap 17

C

CHANGE command 83-84
Class code 71, 177
Clock speed 33, 34, 280
Command line 52-54

Command operands 53
Command summary 60-61
Contiguous 41
Control and Sort 41
COPYFILE command 84-89
CREATE command 89-92
Csi 24, 59

D

Data 13
Deadlocking 37
Default options 53, 61, 93
Delimiters 46
Device drivers 23-24, 70, 174
Documentation conventions 46

E

EDIT command 126-138
Editor command summary 130-131
Editor 127
 Bottom 133
 Col 134

Delete/list 136-137
File 132
Locate 134
Modify 135
Number/unnumber/renumber 137-138
Quit 132
Save 132
Top 134
Equal sign 86-88
ERASE command 123-126
EXEC instructions 202-211
 &BEGSTACK 202-203
 &BEGTYPE 203
 Constants and variables 197-199
 &CONTROL 203-204
 &CRT 204-205
 CSI 205
 &END 206
 &ERROR 206
 &GOTO 207
 &IF 207-208
 Keywords 201-202
 Labels 199-200
 Operators 200
 &QUIT 208
 &READ 208-209
 &SPACE 209
 &TYPE 209-210
 &WAIT 210
EXEC language 189-219
EXEC SELECTED 101
EXEC 47

F

File description 48-49
File locking 33, 289-290
FILELIST command 92-101
Files 12, 40-41
Flaw map 105
Format 103

G

Graphics 284

H

Hang System 117
Help 37-38, 289

I

Id numbers 62-63
INITDISK command 102-107
INITTAPE command 107-109
Inputfile 85
IPL.EXEC 191-193, 215-219

K

Keyboard 13-14
KILL command 185

L

LIST command 109-114
Logdev 68
LOGOFF/LOGON 114-117

M

MAILBOX/MSG commands 117-120
Misallocation 145, 146, 183-187

N

Nucleus 23

O

Operating system 16-20
Outputfile 85

P

Passwords 30
Phydev 68
Private files 28
Privilege level 30, 64
Programs 13
Prompting characters 47
Protecting status 92, 93, 120
Public files 28

Q

QCOPY 42-43

R

R/M Cobol 42
RECEIVE command 157-160
Record locking 33, 289-290
Records 13
Remote operation 166-167
RENAME command 120-122
REORG 42-43
REPAIR command 184-185
 REPAIR fix option 184-185
RESTORE command 78-83
Return code 199

S

SCRIPT command 138-139
 Boilerplating 139
 Embedded commands 138-139
 Textprocessing 139
Search order 47
Security 27
SELECTED.EXEC file 195-197
SEND command 164-166
Serialized 38, 168-169
SET command 140-142
 Date 141
 Memory allocate/deallocate 141-142
 Memory partition 141-142
 RDYMSG 141
 Time 141
Shared files 28
SHOW command 144-148
 Bank 144
 Disk 145
 Memory 146-147
 Priv 147
 Rdymsg 147
 Serial 147
 Time 147
 Users 148
 Who 148
Shutdown 58-59
Software installation 168-171
 NEWACC option 169
Software 12, 251-278
SPOOLER command 148-153
 Spooled files 148-149
 Spooler control 149

 Spooler deactivation 152-153
 Spooler setup 150-152
 SYSTEM.SPOOLQUE 150, 182-183
SQUEEZE 42-43
START/STOP commands 142-143
Startup 56-58
Synonym account 30, 62, 64
SYSGEN command 154-156
SYSTEM account 28-29, 61, 106-107, 115
System cancel (ESC Q) 52, 108, 191
System control keys 52
System disk 47, 103, 105
SYSTEM.CLASSn 175
SYSTEM.DEVNAMEs 68, 69, 174
SYSTEM.ERRMSG 115, 122, 123, 178-182
SYSTEM.EXECnn 182, 193-195
SYSTEM.LOGON 115
SYSTEM.NUCLEUS 72

T

Telecommunications 157-168
TERMINAL command 160-164
THEOS 6.1 294-296
THEOS Software Corp. 294
THEOS user's group 297
THEOS version 6.1 294-296
THEOS86 279-291
Time sharing 32

U

UNIX 279-287-291
User accounting 28
USER.LOGON file 191-193

V

Volume id 108

W

Wild cards 50-52, 88

Z

Z80 33-34

More Computer Books from Weber Systems Inc.

All About 1-2-3	\$5.95	IBM PCAT Repair Guide	\$19.95
1-2-3 Worksheets	\$7.95	Artif. Int. for PCAT	\$19.95
How to Use Supercalc	\$5.95	HP Laserjet US HB	\$19.95
How to Build a Program	\$9.95	Atari XE User's HB	\$15.95
The Programmer's Toolbox	\$9.95	Atari ST User's HB	\$ 9.95
World of PC-DOS	\$7.95	Atari XL User's HB	\$15.95
MS-DOS	\$7.95	Sanyo 550/555 US HB	\$17.95
Beyond Beginning Basic	\$7.95	Sanyo 775/885 US HB	\$17.95
Microsoft COBOL	\$9.95	Okidata Printer US HB	\$15.95
LISP: An Introduction	\$19.95	Appleworks User's HB	\$ 9.95
Atari Assembly Language	\$14.95	Applewriter Handbook	\$19.50
Multimate Handbook	\$15.95	Inside the Model 100	\$19.95
Turbo Pascal	\$15.95	Symphony Programming	\$17.95
Used Computer Price Guide	\$12.95	Bus. Gr. with 1-2-3	\$19.95
PC-Blue Library	\$14.95	Plan. and Bud. 1-2-3	\$17.50
PICK User's Handbook	\$17.95	Bus. Pr. Solv. 1-2-3	\$19.95
IBM PCAT User's Handbook	\$17.95	Bus. Dec. Making 1-2-3	\$19.50
Wordstar Handbook	\$16.50	Anal. Fin. State. Sym.	\$19.95
XENIX User's Handbook	\$17.95	Introduction to ADA	\$19.95
Micro. Sys Princ.-6502	\$25.95	Introduction to Modula 2	\$19.95
Architect. of the 8048	\$19.95	Introduction to BASIC	\$15.95
Computer Arch. & Org.	\$25.95	PASCAL	\$15.95
Lotus 1-2-3 Models	\$15.95	Beginning Forth	\$16.95
Model 100 Prog. Book	\$12.95	Beginning Fortran	\$15.95
Atari XE & XL Prog. BK	\$12.95	Introduction to Fortran	\$19.95
IBM PC Bus. Software	\$17.95	6502, 65C02, 65816 HB	\$14.95
Sanyo MBC Bus. Soft.	\$17.95	THEOS/OASIS Handbook	\$17.95
Computers for Everybody	\$ 4.95	Fund. of Micro. Arch.	\$39.95
Microsoft COBOL	\$ 9.95	Anal. and Des. of Dig. Cir.	\$39.95
Bits, Bytes, & Buzzwords	\$ 3.95	Micro. Systems Engineering	\$39.95

Title	Price	Quantity	Total
Catalogue			NC

Book Subtotal _____

Shipping & Handling (\$2.50 per item) _____

Ohio Residents Add 6½% Sales Tax _____

GRAND TOTAL _____

Method of Payment:

Check **Charge my:** VISA Master Card American Express

Card Number _____ Exp. Date _____

Cardholder Name _____

Ship To: _____

Name: _____

Address _____

City _____ State _____ Zip _____

Send to: Weber Systems Inc. 8437 Mayfield Road, Chesterland, OH 44026

Toll-free order number 800-851-6018



\$17.95

THEOS User's Handbook

THEOS User's Handbook is a clear, practical guide to the THEOS microcomputer operating system. This book benefits the system manager and the novice user. First it is a "hands-on" guide to the use of the commands required to maintain a THEOS system for the system manager. Second it describes everything the novice needs to know in order to put THEOS to work. Many examples and tutorials are included to clarify discussions.

Two chapters are devoted exclusively to the THEOS EXEC language. This is a powerful, yet easy to use language which allows you to construct menus making any THEOS system virtually menu-driven!

The following topics are covered in **THEOS User's Handbook**:

- * System security
- * Filing system
- * System maintenance
- * THEOS command tutorials
- * EXEC language
- * Examples of EXECs
- * Creation of a "turn-key" system
- * Software availability
- * Reviews of software
- * Future enhancements

This book is required reading for anyone using THEOS, or anyone considering a multi-user microprocessor operating system for business applications.

ISBN: 0-938862-31-6

LC 85-11619



WEBER SYSTEMS, INC.

8437 Mayfield Road, Chesterland, Ohio 44026