PN-ABH-088
69887

HOST

A Model Development and Support System
for Implementing All Types of Models

User Guide

Version 2.00

Funding for the development of this program has been provided by the United States Agency for International Development through the Integrated Population and Development Planning (INPLAN) Project.

AUTHORS

Program System Design and Implementation

Mr. Gordon M. Cressman
Microcomputer Systems Analyst
Center for Development Policy
Research Triangle Institute

Dr. Beth G. Barnwell
Computer Systems Analyst
Center for Systems and Operations
Research Triangle Institute

Ms. Carole E. Sierpien
Program Systems Design Consultant
Chapel Hill, North Carolina

Documentation Design and Preparation

Mr. Gordon M. Cressman

Dr. Beth G. Barnwell

Ms. Carole E. Sierpien

Dr. Dorothy E. Stribling
Computer Systems Analyst
Center for Systems and Operations
Research Triangle Institute

Ms. Kirsten Olsen Crouch
Economist
Center for Development Policy
Research Triangle Institute

ACKNOWLEDGEMENTS

Funding for the development of this program has been provided by the United States Agency for International Development (USAID) through the Integrated Population and Development Planning (INPLAN) Project. The technical support of USAID professional staff in the Policy Development Division of the Office of Population is gratefully acknowledged.

We are indebted to the following people for spending a significant amount of time testing early versions of Host, and for making many valuable design suggestions: Dr. R. Scott Moreland, Development Economist; Dr. Luis A. Crouch, Development Economist; Dr. Dorothy E. Stribling, Computer System Analyst; and Ms. Hazel J. Ryon, Microcomputer Specialist, all of the Research Triangle Institute, and to Dr. James C. Knowles, President, The Knowles Corporation. We also owe thanks to Dr. James E. Kocher, Director of the INPLAN Project, for unflagging support and enthusiasm for this effort.

Trademarks

The following trademarks are mentioned in this document: Bernoulli Box is a trademark of IOMEGA Corporation; dBASE III is a trademark of Ashton-Tate, Inc.; Epson is a trademark of Epson America, Inc.; IBM PC-DOS is a trademark of International Business Machines Corporation; Lotus 1-2-3 is a trademark of Lotus Development Corporation; MS-DOS is a trademark of Microsoft, Inc.; Turbo Pascal and Turbo Toolbox are trademarks of Borland International, Inc.; and WordStar is a trademark of MicroPro International Corporation.

TABLE OF CONTENTS

Section *Page*

AUTHORS....i

ACKNOWLEDGEMENTS....ii

TABLE OF CONTENTS....iii

LIST OF FIGURES....viii

LIST OF TABLES....x

LIST OF SCREENS....xii

Chapter *Page*

1. INTRODUCTION....1-1
   - What Host is and What it Does....1-2
   - How this Document is Organized....1-3
   - Typographical Conventions....1-5
   - Reporting Problems and Getting Help....1-6
   - Registering Your Copy of Host....1-6

2. INSTALLATION....2-1
   - Required Operating System and Hardware....2-2
     - Memory Space....2-2
     - Disk Space....2-2
     - Numeric Coprocessor....2-4
   - Copying Host to a Hard Disk or Cartridge Disk....2-4
   - Creating or Modifying the CONFIG.SYS File....2-4
   - Using the DOS PATH Command....2-5

3. GETTING STARTED....3-1
   - Entering Host....3-2
     - Bad command or file name....3-3
     - Difficult to Read Screen Display....3-4
   - The Host Screen Display....3-4
   - Host Cursors....3-5
   - Using the Keyboard....3-5
   - Using the Menus....3-7
     - Cursor Selection....3-8
     - Key Selection....3-8
   - Naming Things in Host....3-8
   - User or Manager Access....3-10
     - User....3-10
     - Manager....3-10

Chapter *Page*

3. **GETTING STARTED** (cont.)
The Host File System....3-11
Configuration Files....3-11
Application Modules....3-12
Host Data Files....3-13
Library Files....3-14
Instruction Files....3-15
WARNING....3-16
Using Query Fields....3-18
Using Toggle Fields....3-20
Generating Reports....3-20
Using DOS From Within Host....3-23

4. **CONFIGURING HOST**....4-1
Default Files....4-3
Host Data File....4-3
Comparison File....4-4
Instruction File....4-4
Default Output Devices....4-4
Table Output Device....4-4
Graph Output Device....4-4
Default Directories....4-4
Data Directory....4-5
Module Directory....4-5
Library Directory....4-5
Instruction Directory....4-6
Printer Settings....4-6
Left Margin....4-6
Top Margin....4-6
Bottom Margin....4-6
Page Length....4-7
Line Length....4-7
Page Eject....4-8
Setup String....4-8
Monitor Setting....4-9
Saving Configuration Settings....4-9

5. **DEFINING DATA**....5-1
Defining Dimensions....5-4
Adding and Editing Dimensions....5-4
The Time Dimension PERIOD....5-8
The Next, Previous, and Find Commands....5-9
Saving Dimensions....5-10
Deleting Dimensions....5-10
Listing Dimensions....5-10
Reporting Dimensions....5-11
Defining Variables....5-11
Adding and Editing Variables....5-11
The Next, Previous, and Find Commands....5-21
Saving Variables....5-22

Chapter *Page*

5. **DEFINING DATA** (cont.)
Deleting Variables....5-22
Listing Variables....5-22
Reporting Variables....5-23

6. **DEFINING LABELS**....6-1
The Label Library....6-2
Label Formats....6-2
Creating and Editing Label Formats....6-3
The Label Format Description....6-4
Label Specifications....6-4
The Next, Previous, and Find Commands....6-7
Saving Label Formats....6-8
Using Label Formats....6-8
Expanding Label Formats....6-8
Deleting a Label Format....6-9
Listing Label Formats....6-9
Producing Label Reports....6-9

7. **DEFINING TABLES**....7-1
The Table Library....7-2
Table Definitions....7-2
Creating and Editing Table Definitions....7-4
The Table Title....7-5
The Variable List....7-6
The Table Dimensions....7-9
The Next, Previous, and Find Commands....7-17
Saving Table Definitions....7-17
Deleting Table Definitions....7-22
Listing Table Definitions....7-22
Producing Table Definition Reports....7-22
Generating Tables....7-22
Selecting a Reference Data File....7-22

8. **EDITING DATA**....8-1
Selecting a Data File....8-2
Selecting a Variable....8-2
If Data is not Found in the Data File....8-3
Saving Editor-supplied Default Values....8-3
The Data Editor Display....8-3
The Menu Space....8-3
The Work Space....8-4
Using Label Formats....8-7
Moving Through the Data....8-8
The Data Entry Cursor....8-8
Changing Row and Column Dimension Levels....8-8
Changing Other Dimension Levels....8-10
Changing the PERIOD Dimension Level....8-10
Entering Data....8-11

Chapter *Page*

8. EDITING DATA (cont.)
The Edit Menu....8-12
Saving Data....8-12
Selecting Another Variable....8-14
Selecting Another Data File....8-14

9. RUNNING MODULES....9-1
Selecting an Application Module....9-2
Selecting the Host Data File....9-2
Specifying the Starting Period....9-3
Specifying the Number of Periods....9-3
Running the Application Module....9-3

10. GENERATING TABLES....10-1
Selecting a Table....10-2
Selecting a Host Data File....10-3
Selecting a Comparison File....10-3
Specifying an Output File....10-4
Specifying a Period List....10-4
Generating the Table....10-6
Problems Generating Tables....10-7
Incompatible Data File....10-7
Insufficient Memory....10-7
Table Too Wide....10-7
Importing Table Output into Lotus 1-2-3....10-8

11. INSTRUCTION FILES....11-1
Creating and Editing Instruction Files....11-4
The Interactive Prompting Facility....11-5
Saving Host Instruction Files....11-6
Host Instruction File Syntax....11-6
Backing-Up an Instruction File....11-9
Processing an Instruction File....11-10
Deleting an Instruction File....11-11

12. MANAGING FILES....12-1
Managing Host Data Files....12-3
Creating a Host Data File....12-3
Renaming a Host Data File....12-3
Backing Up a Host Data File....12-4
Deleting a Host Data File....12-4
Verifying a Host Data File....12-4
Packing a Host Data File....12-7
Managing Host Libraries....12-8
Creating a Library....12-8
Verifying a Library....12-8
Packing a Library....12-11
Using the Management Utilities....12-12
Read....12-12

Chapter *Page*

13. ADVANCED TOPICS....13-1
Executing Host Instructions from DOS....13-2
Using a Configuration File....13-2
The Error File HOSTERR.TXT....13-2
Host Instruction Syntax....13-3
Running Application Modules from DOS....13-4
Producing Tables from DOS....13-5
Executing the BASE Instruction from DOS....13-6
Executing the INCR Instruction from DOS....13-7
Executing the PERIOD Instruction from DOS....13-7
Executing the SUSPEND instruction from DOS....13-7
Processing Instruction Files from DOS....13-7
Using Host with Multiple Directories....13-9

Appendix *Page*

A. HOST SYSTEM FILES....A-2

B. USING HOST WITH FLOPPY DISKS AND RAM DISKS....B-2

C. KNOWN BUGS....C-2
Bugs Corrected in Host Version 2.00....C-2
Known Bugs in Version 2.00....C-3

D. PLANNED ENHANCEMENTS TO HOST....D-2

E. RUNNING HOST ON THE WANG PROFESSIONAL COMPUTER....E-2

F. VERIFY ERROR MESSAGES....F-2

GLOSSARY....G-2

INDEX....I-2

LIST OF FIGURES

Figure *Page*

1. The command structure of Host's Access Menu....3-10
2. A schematic of Host commands for obtaining reports....3-21
3. A sample page of a printed Label Report....3-22
4. Commands providing access to Host's default settings....4-3
5. Host printed page layout parameters....4-7
6. The basic components of a Host Data File....5-2
7. The command structure of Host's Dimension Definition facility....5-5
8. The command structure of Host's Variable Definition facility....5-12
9. A schematic of Host commands for manipulating Label Formats....6-3
10. A schematic of Host commands for defining and generating tables....7-3
11. A table produced using the variable list shown in Screen 19....7-7
12. An example Comparison Table with a percentage difference column....7-8
13. A table produced using default dimension Use settings....7-11
14. A table showing AGE used as a Row dimension....7-12
15. A table using the Label Format AGEBYFIV for the Row dimension AGE....7-13
16. A table using the Label Format WORKAGES for the Row dimension AGE....7-15
17. A table using Combine set to "No" for the Row dimension AGE....7-16
18. A table showing PERIOD used as a Column dimension....7-19
19. A table showing SEX used as a Nested dimension....7-20
20. A table showing the use of more than one Unnested dimension....7-21
21. The command structure of Host's Data Editor....8-2
22. Commands required to run a single Application Module....9-2
23. Commands required to generate a table from Host's Main Menu....10-2

Figure *Page*

24. A typical page of output from the Table Generator.....10-5

25. The command structure of Host's Instruction file facility.....11-2

26. A summary of paths through the Instruction File Processor.....11-10

27. The command structure of Host's File Management facility.....12-2

28. A summary of paths through Host's Data File verification facility.....12-5

29. A summary of paths through the Library verification facility.....12-9

30. An Instruction file modified to run as a "BAT" file from DOS.....13-8

31. An Instruction file using DOS replaceable parameters.....13-9

32. A possible multiple-directory configuration for use with Host.....13-10

33. A multiple-directory configuration with regional subdirectories.....13-12

LIST OF TABLES

Table *Page*

1. Microcomputers and operating systems recommended for use with Host.....2-3
2. Display adapters recommended for use with Host.....2-3
3. Host cursor movement and editing commands.....3-6
4. Host cursor movement commands for use in menus.....3-9
5. Examples of valid and invalid Host Identifiers.....3-9
6. A brief summary of Host Configuration Files.....3-12
7. A brief summary of Host Application Module files.....3-12
8. A brief summary of Host Data Files.....3-14
9. A brief summary of Label Library files.....3-15
10. A brief summary of Table Library files.....3-16
11. A brief summary of Instruction files.....3-16
12. Standard Host names for reports sent to disk files.....3-21
13. The size and range of Host variable Types.....5-14
14. Edit Mask characters available for numeric variables.....5-15
15. Edit Mask characters available for non-numeric variables.....5-17
16. Cursor movement and editing commands for dimension name lists.....5-21
17. Table Definition cursor movement and editing commands.....7-10
18. Data Editor Cursor movement and editing commands.....8-9
19. Data Editor function keys for changing the level of PERIOD.....8-11
20. Examples of valid and invalid period list entries.....10-6
21. Restrictions to DOS command use in Host Instruction files.....11-4
22. Function keys used in the Instruction File Editor.....11-6

Table *Page*

23. Default files and directories used by the RUN command.....13-5

24. Default files and directories used by the TABLE command.....13-6

LIST OF SCREENS

Screen *Page*

1. Host's title screen display....3-2
2. Host's Main Menu screen display....3-3
3. The six areas of the Host screen display....3-5
4. Host's Define Menu screen display....3-7
5. Host was interrupted while processing an Instruction file....3-17
6. The previous Host session was interrupted....3-18
7. A Host menu composed of Data File names....3-19
8. A variable name selection list....3-19
9. A dimension report screen display....3-23
10. A sample Host configuration facility screen display....4-2
11. A typical Dimension Definition screen display....5-6
12. A typical Variable Definition screen display....5-13
13. An example Label Format Editor display....6-4
14. A Label Format illustrating grouping label specifications....6-6
15. (Warning!) A Label Format with no grouping label specifications....6-6
16. A Label Format illustrating a computed label specification....6-7
17. An example screen showing an expanded Label Format....6-9
18. An example screen showing the creation of a Table Definition....7-4
19. An example variable specification screen of a Table Definition....7-6
20. An example dimension specification screen of a Table Definition....7-11
21. A Data Editor display of a periodic variable with no dimensions....8-5
22. A Data Editor display of a one-dimensional nonperiodic variable....8-5
23. A Data Editor display of a two-dimensional nonperiodic variable....8-6

Screen *Page*

24. A Data Editor display of a three-dimensional periodic variable.....8-7

25. A Data Editor display showing the Edit Menu.....8-13

26. A sample screen showing output of the Table Generator.....10-3

27. A typical Host Instruction File Editor display.....11-5

28. A syntax error was found while processing an Instruction file.....11-11

29. A run-time error occurred while processing an Instruction file.....11-12

30. An example showing a Host Data File being verified and rebuilt.....12-6

31. An example showing a Host Library being verified and rebuilt.....12-10

32. Configuration settings for multiple-directory system in figure 32.....13-11

ADDENDUM

UPGRADING FROM VERSION 1.02 TO VERSION 2.00 OF HOST

Differences Between Version 1.02 and Version 2.00

Several major and many minor improvements have been made to produce Version 2.00 of the Host system. The following is a list of the most significant changes.

- The Host User Guide has been completely rewritten and expanded. We have added a glossary, an index, and several new appendices. The Host Programmer Guide has been drafted and is currently being reviewed. We are also planning an abbreviated Host Introductory Guide for those who want to get started with a minimum of reading.

- Several changes in terminology have been introduced to generalize the system and improve its clarity. These are summarized in the following table.

Terminology Changes in Host Version 2.00

Old Term	New Term
Projection File (or Database)	Host Data File
Projection Module	Application Module
Project (execute Projection Module)	Run (an Application Module)
Run (an Instruction File)	Process (an Instruction File)
Nested by (dimension)	Nested
Unnested by (dimension)	Unnested
Describe Menu	Define Menu

- We have added a command to Host's Main Menu (*DOS) that allows you to execute any program or DOS command from within Host. This new feature is described on page 3-23, of the User Guide, **Using DOS From Within Host**.

- The Instruction File Processor now allows any command that you could issue at the DOS prompt to be included in a Host Instruction file. This new feature, which is described in Chapter 11 of the User Guide, allows you to issue DOS commands and execute other programs during the processing of a Host Instruction file. Thus, the special capabilities of other software packages can be smoothly and efficiently integrated with those of Host.

- We have added a File command to the Dimension, Variable, and Table menus. This allows you to change the current Host Data File without reconfiguring Host. Further, it is no longer necessary to declare a default Host Data File before choosing Dimension, Variable, or Table from Host's Define Menu. If there is no default Data File, Host asks you to enter the name of the Data File you want to work with

after you choose one of these Define Menu selections.

- You can now precede all Host Data File, Instruction File, and Library names with directory paths at Host prompts. Thus you can use files not in your default directories without reconfiguring Host.

- A default Instruction file directory has been added to the Configuration File, allowing you to keep all Instruction files in a directory other than the default Data Directory.

- Five bugs have been located and corrected (see **Appendix C**, KNOWN BUGS, for details).

Upgrading to Host Version 2.00

If this is your first version of Host, you may ignore the rest of this section. If you are upgrading to Host Version 2.00 from Host Version 1.01 or 1.02, then you should do the following.

1. Copy all of the Host System Files files from your new Host Version 2.00 diskettes to the Host directory of your hard disk or disk cartridge.

2. Execute Host's UPGRADE utility by typing "upgrade" at the DOS prompt as follows:

   C>**upgrade**[Enter]

   This example assumes Host Version 2.00 is on disk drive "C". If this is not the case, set your current drive to the disk drive that contains Host. The UPGRADE utility will upgrade all Configuration and Instruction files in ***all directories*** on the target disk drive. If you do not want some of these files upgraded, you should copy them to other disks before executing UPGRADE.

   You may specify the target disk drive when you execute UPGRADE by including a drive designation on the DOS command line. For example, if you wanted to upgrade all the files on drive "D" to be compatible with Host Version 2.00, then you would type the following at the DOS prompt:

   C>**upgrade d**[Enter]

   If you do not specify a target drive on the DOS command line, UPGRADE gives you a chance to type in a disk drive designation.

   If you have several cartridges or disks with Host files on them, you may repeat this step to upgrade the files on each cartridge or disk. Once all Configuration and Instruction files are upgraded you can erase the UPGRADE.COM file from your hard disk or disk cartridge.

3. Finally, important changes have been made to the Application Module include files HOST1.BOX and HOST2.BOX. You will need to recompile all Application Modules with these new include files before they will be compatible with Host Version 2.00. We know this is an inconvenience, but it is unavoidable. Host is still an evolving software system; we will continue to make it as easy as possible for you to transfer your work to future versions of Host.

CHAPTER 1
INTRODUCTION

INTRODUCTION

What Host is and What it Does

Host is a microcomputer software system designed to simplify the construction and use of all types of models. This system provides an integrated framework supporting all phases of model development, from program design through result presentation. Host's data manipulation facilities provide basic input and output routines for user-written program modules, as well as interactive data editing and reporting facilities for model users. Although originally designed for projection modeling applications, Host is general enough to be applicable to demographic, econometric, ecological, epidemiological and many other types of models.

Host includes several facilities common in most database management system (DBMS) software, including facilities for data definition, data labeling, full screen data editing, table generation and application program development. However, unlike most DBMS software, Host tailors these features to the types of data structures most useful for modeling. These structures range from simple scalar values to seventeen-dimensional matrices composed of real, integer, byte, character, or character string values. Host's multi-dimensional capabilities also include special features for working with periodic data (data that require a time dimension). Further, unlike most DBMS software, Host's application development programming language is Turbo Pascal™. Since Turbo Pascal is a widely used, general purpose language, this provides Host module developers access to an immense body of Turbo Pascal literature and enhancement products.

Usually, the first step in programming a large model for Host is to divide the model into separate sectors, or modules, each representing a logical subsystem of the whole. Each of these modules is then written as a Turbo Pascal program that includes Host-supplied support routines. The resulting Host Application Module can be run from within Host, or directly from the operating system. When running, most modules call Host support routines to read data from a Host Data File. A module can then use those data in calculations and, if desired, store results in the same Host Data File by calling another Host support routine.

Any task that can be expressed as a Pascal language computer program can be developed into a Host Application Module; Host does not restrict what an Application Module can do. Possibilities include large scale simulations, interactive training exercises, or computer graphics presentations. Modules can be highly interactive, with decisions made by the operator during execution, or can be run without manual intervention by accepting values stored using Host's Data Editor.

Host provides additional flexibility by allowing certain data characteristics to be modified interactively without requiring any changes in related program modules. When a variable is created within Host by a module developer, it is designed to be as large and disaggregated as possible; any user can then interactively change the structure of the variable within the limits established by the module developer. Models using Host can therefore be adapted to different situations with ease.

Host's advanced user interface has many features common in the best microcomputer software, including "Lotus®-style" menus and logical use of the keyboard. In addition, nearly all functions can be performed using either a Host command or the equivalent WordStar® command. Host has special features to help you respond to program prompts and consistent displays to help you quickly locate information on the screen.

Host's multi-dimensional, spreadsheet-like editor gives you convenient access to data in Host Data Files. Data displayed in the editor can be clearly labeled using Host's labeling facility. Investigation and comparison of alternative model scenarios can be done by executing a base run, then using the results as the base against which to compare other runs. Differences between runs can be identified readily using Host's table generation capabilities.

Host allows automation of module execution and table generation using special Instruction files. Host Instruction files are similar to PC/MS-DOS batch ("BAT") files, allowing an operator to store a sequence of instructions for running modules and generating tables. This series of instructions can be processed at any time by specifying the Instruction file containing them. Executing interactive modules using an Instruction file can be effective in constructing many analytical and instructional models. In addition, Host Instruction files may consist entirely of instructions for generating tables and graphs. They can therefore be used to produce computerized "slide-shows" of previously calculated results.

How this Document is Organized

This Host User Guide provides the information necessary to use any model composed of Host Application Modules. This Guide explains how to do the following:

- install Host on a microcomputer;
- enter the Host System;
- understand and use Host's basic features;
- establish system-wide default settings;
- add and edit data definitions;
- define and attach explanatory text labels to data;
- define and generate tables;
- enter and edit data;
- run Application Modules;
- use Host Instruction files; and
- manage your file system.

This Guide does not explain how to write Host Application Modules using Turbo Pascal; this information is contained in the Host Programmer Guide.

The sections of this document are introduced in a sequence that should parallel your investigations of Host's features. Each chapter begins with a brief overview of the current topic followed by a discussion of related commands or operations in the order in which they appear on Host's menus. Examples are given illustrating concepts and options. We hope this organization is useful to those who use this Guide as a tutorial while operating Host. This document can also be used as a reference manual providing complete information about each Host feature and function.

The contents of each chapter of this Guide are as follows.

1. Chapter 1 briefly describes Host, then outlines the contents of this Guide and tells how to get help if you encounter problems.

2. Chapter 2 lists required hardware and gives instructions for installing Host on a microcomputer.

3. Chapter 3 tells how to enter Host and describes fundamental, system-wide concepts, including Host's use of the display screen and keyboard, the difference between a Host User and a Host Manager, and how to use Host's Query and Toggle fields.

4. Chapter 4 gives information on setting various system-wide default values to be used by Host whenever it is executed.

5. Chapter 5 explains how to define dimensions and variables in a Host Data File.

6. Chapter 6 describes the use of Host Label Formats to annotate data and to total data categories for presentation in tables.

7. Chapter 7 tells how to create and store Host Table Definitions describing the tabular form in which you wish data to be presented.

8. Chapter 8 explains how to enter and edit data in a Host Data File.

9. Chapter 9 describes how to run Host Application Modules.

10. Chapter 10 tells how to generate tables from Host Table Definitions.

11. Chapter 11 contains information about using Host Instruction files for automating Application Module execution, table generation, and other operations.

12. Chapter 12 explains how to access Host facilities for managing Host Data Files, as well as Host Label and Table Libraries. This chapter also tells how to use Host Utilities.

13. Chapter 13 covers topics for advanced Host Users, including running Application Modules, generating Tables and running Host Instruction files directly from the operating system, and organizing your Host files in separate subdirectories.

14. Appendix A contains a list and brief description of all Host System Files on the distribution disks.

15. Appendix B gives recommendations for running Host using any combination of floppy disks or RAM disks.

16. Appendix C lists known errors in Host.

17. Appendix D describes additions and improvements planned for future versions of Host.

18. Appendix E tells how to run Host on the Wang Professional Computer.

19. Appendix F explains error messages that may be produced by Host's file verification facility as well as how to correct these errors if they occur.

Host is a complex program with many powerful features. Although its user interface is consistent and often intuitive, some of the more advanced concepts may require more time to understand. As you encounter unfamiliar terms or concepts, refer to the Table of Contents for related topics and check the Glossary for a brief definition of new terms.

One final note before you proceed. This manual assumes you are familiar with the basic commands of the PC/MS-DOS operating system such as COPY and PATH. You should understand fundamental operating system terms such as disk file, current drive, current directory and path. You should know the rules for directory paths, file names and file name extensions. Familiarity with tree-structured directories, and the PC/MS-DOS commands dealing with them, would also be helpful. If you are not familiar with these terms and concepts, you should review your operating system manual.

Typographical Conventions

When a special term or phrase is used for the first time in this document, it is *italicized* and defined. These definitions are collected in the Glossary at the end of the Guide.

In examples of interactive dialogs with Host or the DOS operating system, information entered by the microcomputer user appears in **boldface** type; program or operating system prompts and responses appear in ordinary type.

Keys on the alphanumeric keyboard are represented by placing the key mnemonic between square brackets*. For example, [Esc] represents the escape key, and [F4] represents programmable function key 4. When two or more keys must be used in combination, this is shown by separating their symbols with a plus sign. For example, holding down the [Alt] key while pressing the [I] is represented by [Alt]+[I]. This is the Host command meaning "insert a blank line above the current line."

For most Host cursor movement and editing commands, Host also recognizes the equivalent WordStar command. When this alternative is available, the WordStar equivalent is given in standard WordStar notation enclosed in parentheses. For example, in WordStar notation, holding down the [Ctrl] key while pressing the [N] key is represented as ^N. Therefore, in a section discussing the Host Insert Line Command, this command appears as [Alt]+[I] (^N).

---

* The key mnemonics used are those appearing on the IBM AT microcomputer keyboard. The IBM PC keyboard uses special arrow symbols to label the [Shift] and [Enter] keys; other compatible microcomputers may have other differences in key mnemonics. Appendix E includes a partial table of Wang Professional Computer keyboard key equivalences.

Reporting Problems and Getting Help

Despite significant effort expended in designing and testing this program system, we know it can never be free of errors. If you are having problems using operating system commands, consult the documentation provided with your computer and operating system. If you are having problems with Host, call or write:

Host User Support
Center for Development Policy
Research Triangle Institute
P.O. Box 12194
Research Triangle Park, North Carolina 27709 USA

Phone: (919) 541-7218
Telex: 802509 RTI RTPK
Cable: RESTRINS, Raleigh, NC

If you think you have discovered an error in Host, try to recreate the circumstances causing it to see if the problem is repeatable. You must be able to describe in detail the data and sequence of events that induce an error before the Host support staff can attempt to diagnose a problem.

A printed copy of any error messages issued by Host, the Turbo Pascal Compiler, or your operating system will help the Host support staff reply promptly to a written request for help. Printouts of relevant display screens, produced using [Shift]+[PrtSc], can be extremely helpful, since they provide an accurate record of the context in which an error message occurred. In some situations, a diskette containing copies of the data files involved may also help in locating the problem.

Registering Your Copy of Host

In the front pocket of this Guide, you will find a Software Registration Form with a return envelope. The Host support staff keep a file of registration information allowing us to inform Host users of corrections and improvements to Host. If you want your name added to this file, you should register your copy of Host by filling out and returning the enclosed form. If you have received Host in a workshop or seminar conducted by the Research Triangle Institute, your copy of Host should be registered for you automatically.

CHAPTER 2
INSTALLATION

INSTALLATION

Installing Host on your microcomputer requires that you do the following:

- verify that you have the required operating system and hardware;
- copy Host to a hard disk or cartridge disk;
- create or modify the CONFIG.SYS file; and
- use the DOS PATH command.

Each of these steps is described in detail in the following sections.

Required Operating System and Hardware

Host is designed to run on IBM PC or compatible microcomputers running version 2.0 or later of the PC-DOS or MS-DOS operating system. Although these two operating systems are not identical, we are not aware of any differences affecting Host. Both operating systems are referred to throughout the remainder of this document as PC/MS-DOS or simply DOS. Host has been verified to function correctly with the microcomputers and operating systems listed in Table 1. Host has also been tested, and functions correctly, with the display adapters listed in Table 2. If your microcomputer or display adapter is not listed in these tables, this does not mean Host will not run on your equipment; this means we have not tested that particular combination. If your hardware is compatible with that listed in Tables 1 and 2, Host is not likely to have problems running on it. Host can be configured for an IBM Monochrome Display, IBM Color Display or a compatible display monitor. If you are using a composite monochrome display monitor with a color display adapter, you can configure Host to use monochrome display attributes only.

Memory Space

Host itself can operate in 256K of Random Access Memory with all the machines and operating systems listed above. The memory demands of Application Modules running within Host, however, depend on their construction. Documentation provided with Application Modules should list their specific memory requirements. Memory demanding utilities (print spoolers, RAM disks, etc.) and memory resident programs (spelling checkers, keyboard macro utilities, etc.) decrease the amount of memory available to Host. In practice, we recommend at least 384K, and ideally 512K or more of Random Access Memory.

Disk Space

It is possible, but not practical, to run Host using only two 360K floppy disk drives (see **Appendix B**, USING HOST WITH FLOPPY DISKS AND RAM DISKS). However, Host has

TABLE 1.

Microcomputers and operating systems recommended for use with Host.

Microcomputer	Operating System	Version Number
IBM Personal Computer	IBM PC-DOS	2.0 - 3.1
IBM Personal Computer XT	IBM PC-DOS	2.0 - 3.1
IBM Personal Computer AT	IBM PC-DOS	2.0 - 3.1
Compaq Portable/Portable II	MS-DOS	2.11
	IBM PC-DOS	2.0 - 3.1
Compaq Deskpro	MS-DOS	2.11
	IBM PC-DOS	2.0 - 3.1
Wyse PC	MS-DOS	2.11
Kaypro 2000	MS-DOS	2.11
Kaypro 16	MS-DOS	2.11
Sperry PC/IT	MS-DOS	3.1
Wang Professional Computer*	PC-DOS	3.1

* Requires use of the Wang IBM PC Monochrome Emulation Card.

TABLE 2.

Display adapters recommended for use with Host.

Display Adapter
IBM Enhanced Graphics Display Adapter
IBM Monochrome Display Adapter
IBM Color Display Adapter
Hercules Graphics Card
Genoa Spectrum*
Paradise Modular Graphics Card

* The Genoa Spectrum will not display different cursor sizes for insert and over-type entry modes when operating in IBM Color Display Adapter emulation mode.

been designed to be operated from a hard disk, RAM disk, or high-speed cartridge disk system. The files comprising the complete Host Model Development and Support System occupy almost 1.5 megabytes of disk storage space. If you do not intend to develop new Host program modules or execute certain Host functions directly from DOS, the necessary disk space is just under 1.2 megabytes; and if you intend to access only a subset of Host's facilities, even less space is required. **Appendix A**, HOST SYSTEM FILES, includes the information necessary to decide which files on the distribution disks you really need to copy. In practice, you probably need at least 2 to 3 megabytes of disk space to use Host, since associated module and data files may require much additional space.

Numeric Coprocessor

This version of Host requires a numeric coprocessor chip (Intel 8087 or 80287) and can not function without it. If you do not have this chip installed in your microcomputer, Host "locks-up" soon after you execute it and you must reboot your system to regain control. Host uses the numeric coprocessor to improve the speed and accuracy of its calculations. Nearly all "desk-top" microcomputers are designed to accommodate this chip and it is increasingly common for it to be installed. Many portable "lap-top" computers, however, do not provide a socket for a numeric coprocessor. A version of Host that runs on these machines will be available soon.

Copying Host to a Hard Disk or Cartridge Disk

Host is ordinarily distributed on 5¼-inch, 360K double-sided diskettes (on request, Host is also available on a ten megabyte IOMEGA® Bernoulli Box™ cartridge). You should first copy the files on these disks to a hard disk or disk cartridge, then put the distribution disks away for safe keeping. Before copying the Host files from the distribution disks, verify that enough disk space is available on the hard disk. About two megabytes are required to store all of the files on the distribution disks.

To avoid conflicts with other files already on your system, it is best to place all Host system files together in a new directory or subdirectory. The directory containing the Host system files is referred to throughout the remainder of this Guide as Host's *Home Directory*. In the following examples, the hard disk is assumed to be drive "C". If this is not the case, simply replace the "C" in the following examples with the appropriate drive letter. Begin by creating a new directory as follows:

```
C>md host[Enter]
```

The files from each Host distribution diskette can now be copied into the new directory by first placing the diskette in drive A, then issuing the following command:

```
C>copy a:*.* host[Enter]
```

Creating or Modifying the CONFIG.SYS File

Host requires that certain operating system characteristics be set through the use of the CONFIG.SYS file. The use of the operating system configuration file CONFIG.SYS is explained in detail in your PC-DOS or MS-DOS manual. The CONFIG.SYS file may contain certain commands that modify the way DOS functions or that make it recognize certain foreign hardware devices. DOS searches the root directory of the boot disk for this file during boot-up. If found, its contents are interpreted by the operating system, otherwise DOS operates using default configuration values. You can make changes to CONFIG.SYS using any text editing software. However, if you use WordStar you must be sure to edit CONFIG.SYS as a Non-document file.

Host requires that the maximum number of concurrently open files permitted by DOS be at least 16. If a CONFIG.SYS file is already present on your DOS boot disk, you need to add the following statement to the file before executing Host:

```
files = 16
```

Using a number larger than 16 does not affect Host, but the memory required by the resident portion of DOS increases by 48 bytes for each additional file above the default value of 8. If the value of this setting is too low, you will receive a message similar to the following when you attempt to use certain Host functions:

```
Too many files open (file \HOST\LIBRARY\LABEL.IDX) in DIMTASKS.CHN.
Closing all data files...
Host program system can not continue.
```

The performance of Host, and many other programs as well, will improve significantly if you add the following statement to CONFIG.SYS:

```
buffers=24
```

This sets the number of sector buffers to 24, thereby providing some additional temporary storage space to DOS, and reducing the number of disk accesses required for many operations. You may request DOS to use between 1 and 99 buffers, but each buffer above the default value of 2 (3 for the IBM Personal Computer AT) requires 528 additional bytes.

If the file CONFIG.SYS is not on your DOS boot disk, you can easily create it by instructing the DOS COPY command to accept input from the console. The following example illustrates this technique, assuming your DOS boot disk is Drive "C".

```
C>copy con config.sys[Enter]
files=16[Enter]
buffers=24[Enter]
^Z[Enter]

    1 File(s) copied
```

The symbol "^Z" is the DOS screen representation of the key combination [Ctrl]+[Z].

It is important to remember that changes to CONFIG.SYS do not affect DOS unless you reboot your microcomputer system after making the changes.

Using the DOS PATH Command

In order to use the example model provided with Host, or to execute Host from a directory other than the one which contains it, you must tell DOS the path to the file HOST.COM. The simplest method is to enter the following command directly from the DOS prompt as shown below:

```
C>path c:\host[Enter]
```

This works for the current session, but will not be in effect the next time you boot your microcomputer. It is better to include this line in your AUTOEXEC.BAT file.

The purpose and use of the AUTOEXEC.BAT file are explained in detail in your PC/MS-DOS manual. The AUTOEXEC.BAT file may contain any standard DOS commands that you want executed when DOS is first loaded into memory. DOS searches the root directory of the boot disk for this file during boot-up. If it is found, the commands in the file

are executed in sequence.

If an AUTOEXEC.BAT file is already present on your boot disk, you must add the following line to the file:

```
path c:\host
```

If a path command is already in the file, the path to the HOST.COM file may be added to the paths already listed. The resulting path command might appear as follows:

```
path c:\dos;c:\util\norton;c:\host
```

You can make changes to AUTOEXEC.BAT using any text editing software. However, if you use WordStar you must be sure to edit AUTOEXEC.BAT as a Non-document file. If the file AUTOEXEC.BAT is not already on your boot disk, you may easily create it by instructing the DOS COPY command to accept input from the console. This is the same technique described above for creating the CONFIG.SYS file. In the following example, your DOS boot disk is assumed to be in the "C" disk drive.

```
C>copy con autoexec.bat[Enter]
path c:\host[Enter]
^Z[Enter]

    1 File(s) copied
```

As with the CONFIG.SYS file, changes to AUTOEXEC.BAT do not affect your system unless you reboot your microcomputer after making the changes.

CHAPTER 3
GETTING STARTED

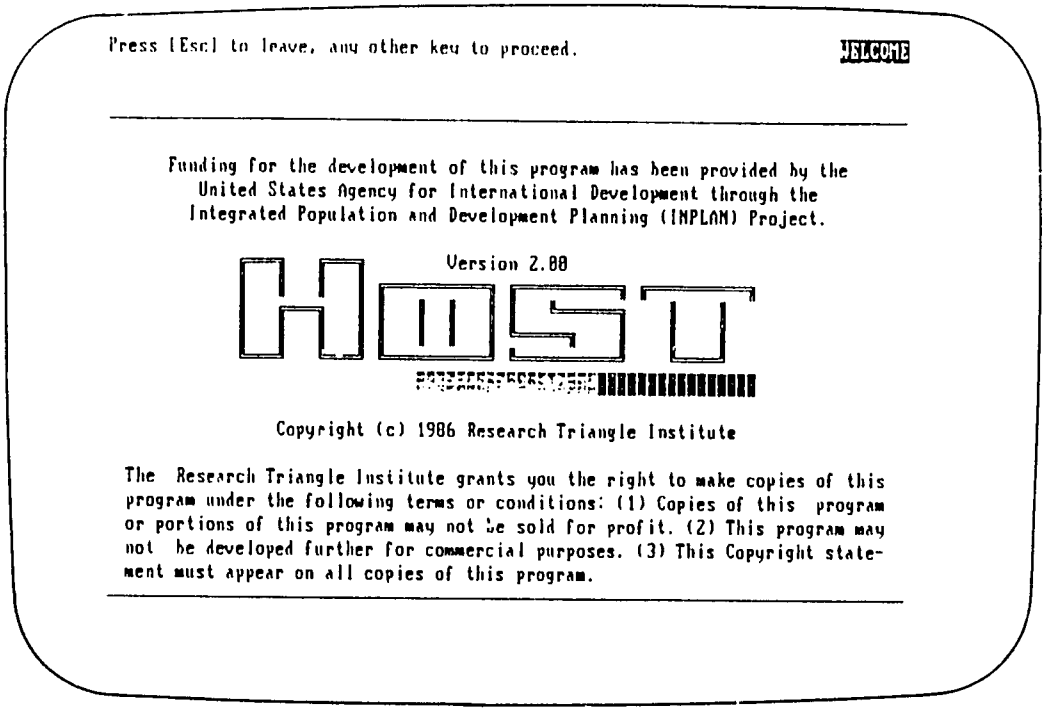
GETTING STARTED

Entering Host

If you have followed the installation procedures described in Chapter 2, you can enter Host by typing "host" at the operating system prompt as follows:

C>**host**[Enter]

The result should be a display similar to that shown in Screen 1. Do not be concerned if there are minor differences between your display and Screen 1. Variations may occur because of differences between display hardware or Host versions. If your screen is difficult to read or you have received an error message, refer to the sections at the end of this chapter.



Screen 1. Host's title screen display.

If you have succeeded in displaying Host's title screen, first press any key to proceed, then select an operator access level from Host's Access Menu. Usually you should choose User access by pressing [Enter]. Manager access is intended for those defining new data structures or developing Application Modules; these operations are explained in the Host Programmer Guide. The difference between User and Manager access is explained in more detail in following sections of this Guide.

After you have selected an operator access level, Host displays its Main Menu. Screen 2 shows Host's Main Menu screen display. The following chapters of this Guide

explain in detail each selection on this menu. The remainder of this chapter explains general concepts used throughout Host.

```
                                                          MAIN MENU
Configure Define Edit Run Tabulate Graph Instruct Manage Access *DOS Quit
Specify file names, directories, printer and monitor settings.
____________________________________________________________________________


  * Use the arrow keys to highlight your menu choice then press
    [Enter], or press the key corresponding to the first letter
    of your menu choice.  Pressing the [Esc] key is the same as
    selecting the choice "Quit".
____________________________________________________________________________
```

Screen 2. Host's Main Menu screen display.

Bad command or file name

If, when trying to enter Host, you receive the following message:

Bad command or file name

then DOS can not find the file HOST.COM because either

- HOST.COM is not in the current directory and the DOS PATH command has not been properly issued (as described in Chapter 2, **INSTALLATION**), or
- HOST.COM is neither in the current directory nor the directory named in the DOS PATH command.

If you are trying to enter Host from its Home Directory, then either the file HOST.COM is not in that directory or, for some reason, the directory can not be read correctly by DOS.

If you are trying to enter Host from some other directory, first verify that the DOS PATH command has been correctly issued by using the DOS SET (Set Environment) command. If the PATH displayed is not correct, issue the correct PATH command at the DOS prompt, or correct the PATH command in your AUTOEXEC.BAT file and reboot your microcomputer. If the PATH displayed by the SET command is correct, then either the file HOST.COM is not in Host's Home Directory or, for some reason, the directory can not be read correctly by DOS.

If you suspect that you disk's directory structure is damaged, use the DOS CHKDSK (Check Disk) command or the Norton Utilities "DT" (Disk Test) program to detect and correct any problems. When you are sure that the directory structure is correct, recopy HOST.COM from the Host distribution disks.

Difficult to Read Screen Display

If Host's title screen is on your display but it is difficult to read, then Host is probably not configured for your screen display. To correct this problem, enter the following sequence of key strokes:

[Space], [Enter], [Enter], [M], [Space], [/], [S], and [Q].

You probably can not see what is happening on the screen, so be careful when pressing these keys. After you have entered the above sequence, Host should be correctly configured for your display and you should see Host's Main Menu on the screen. Changing Host's configuration settings is described completely in Chapter 4, **CONFIGURING HOST**.

The Host Screen Display

Screen 3 shows how Host divides the screen display into six separate areas.

1. The upper left corner of the screen is the *Message Space*. It is reserved for messages telling you what the program is doing, or has done.

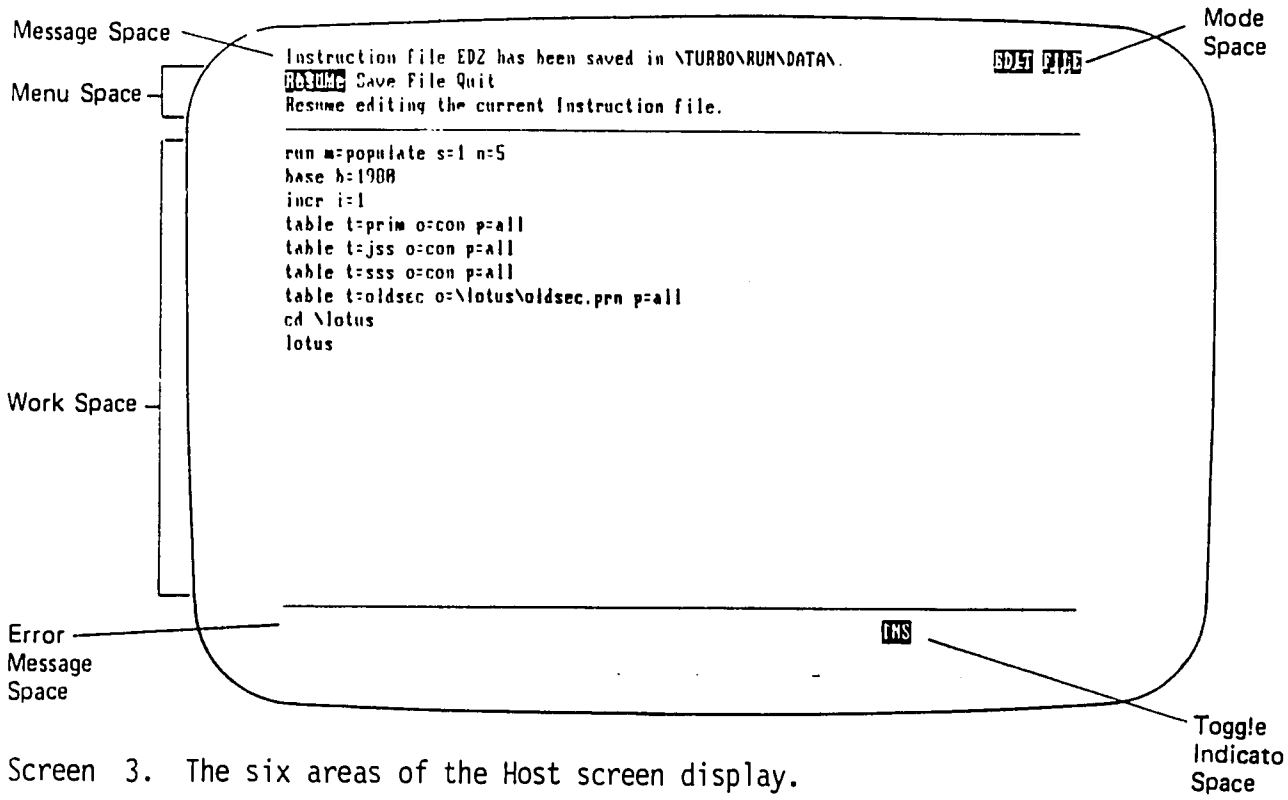
2. The upper right corner of the screen is the *Mode Space* and always contains a *Mode Indicator*. The Mode Indicator shows you where you are within the Host program system. It is useful for orienting yourself and reporting the location of errors to the Host support staff.

3. The second two lines of the screen are the *Menu Space*. When not used for displaying other information, this area is used for displaying "Lotus-style" horizontal menus. Use of this type of menu is explained in a following section of this Guide.

4. The center of the display, bounded at top and bottom by horizontal lines, is the *Work Space*. Usually, it is used for data entry and display. When not needed for this purpose, however, it may be used to display other helpful information.

5. The lower left corner of the display is the *Error Message Space* and is reserved for short error messages. When a potentially serious error has occurred, the short message displayed in this space may be supplemented by information in the Work Space.

6. The lower right corner of the screen is the *Toggle Indicator Space*. This area is used for displaying the status of the toggle keys [Ins], [Caps Lock], [Num Lock] and [Scroll Lock]. In the Data Editor, it is also used to display the status of the [End] key.



Screen 3. The six areas of the Host screen display.

Host Cursors

A *cursor* is a special symbol, color, or shape used to show position on the display screen. Host provides facilities for working with several types of information and, according to which facility you are using, uses several types of cursors.

When you are to select an item from a horizontal menu or vertical list, Host displays the current selection within a highlighted block. This type of cursor is also used to show position when editing data values in the Host Data Editor. In the Instruction File Editor, however, Host displays the current instruction line using bright characters, and other instruction lines using dim characters. Finally, when you are entering data one character at a time, Host marks the current character with a blinking rectangle or square. A blinking rectangular cursor shows that Host is in over-type mode, while a blinking square cursor shows that Host is in insert mode. Host occasionally combines the above cursor types to show both line and character position on the screen.

Using the Keyboard

No attempt is made in this Guide to familiarize you with the standard IBM microcomputer keyboard arrangement. This information is included in the Guide to Operations manual provided with all IBM microcomputers and most IBM compatible microcomputers.

Host has been designed to make effective use of the standard IBM microcomputer keyboard. Most special cursor movement and editing keys have their standard meanings. In addition, Host recognizes many WordStar cursor movement and editing commands. Table 3 lists all Host cursor movement and editing keys and WordStar equivalents.

TABLE 3.

Host cursor movement and editing commands.

IBM AT Command Key	Function in Host	WordStar Command*
[Esc]	Same as "Quit" on menus. Same as [/] in fields, but restores original field contents before displaying menu. Interrupts printing (similar, not identical, to WordStar ^U).	^U
[Home]	Beginning of line or field. First row of first column of data (Data Editor).	^QS
[End]	End of line or field. End Mode (Data Editor).	^QD
[Ctrl]+[Home]	Top of data or information on screen.	^QE
[Ctrl]+[End]	Bottom of data or information on screen.	^QX
[↑]	Up one line.	^E
[↓]	Down one line.	^X
[←]	Left one character or data entry.	^S
[→]	Right one character or data entry.	^D
[Ctrl]+[←]	Left one word. Window left (Data Editor).	^A
[Ctrl]+[→]	Right one word. Window right (Data Editor).	^F
[Ins]	Insert/overtype toggle.	^V
[Del]	Delete character under cursor.	^G
[←—]	Delete character to left of cursor.	[←—]
[PgUp]	Up one window full of information.	^C
[PgDn]	Down one window full of information.	^R
[Ctrl]+[PgUp]	Top of data or information.	^QR
[Ctrl]+[PgDn]	Bottom of data or information.	^QC
[Tab Left]	Left one window full of data (Data Editor).	None
[Tab Right]	Right one window full of data (Data Editor).	None
[Alt]+[D]	Delete line or row.	^Y
[Alt]+[E]	Erase current entry.	^QY
[Alt]+[I]	Insert line or row above current line or row.	^N
[Alt]+[S]	Search for name or partial name (List Facility).	^QF
[Alt]+[U]	Restore previous data entry ("undo").	None

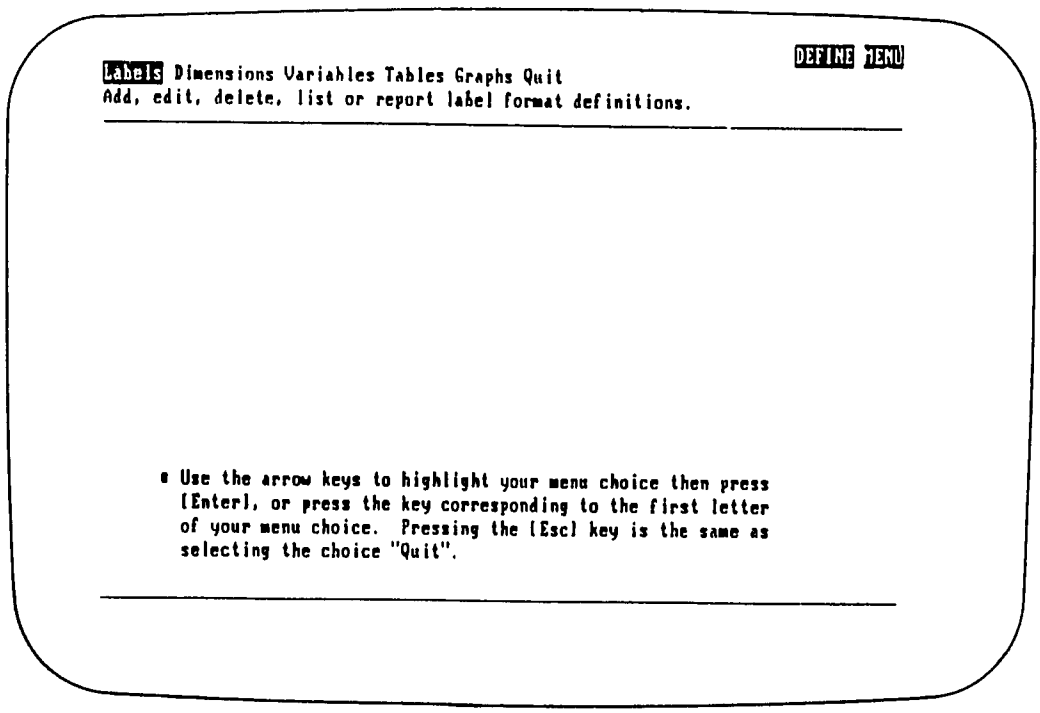
* WordStar commands are represented using normal WordStar notation. In WordStar notation, holding down the [Ctrl] key while pressing [U] is represented by ^U.

Because Host must provide facilities for entering and editing many types of information, certain editing commands have different functions in different Host facilities. When such differences exist, they are noted in Table 3. Tables of key commands specific to different Host facilities are included in other chapters of this Guide.

One key that is important throughout Host is the [Esc] key. Whenever you are selecting from a menu, responding to a question, entering data, or printing a report or table, pressing [Esc] tells Host to return to the previous menu or question. If this can not be done without losing some of your work, Host gives you a chance to save that work before continuing. In general, whenever you want to stop doing anything in Host, Press [Esc].

Using the Menus

Most functions in Host are selected from "Lotus-style" horizontal menus. These menus are displayed in the Menu Space at the top of the screen. Whenever the Menu Space is occupied by something other than a menu, pressing [/] or [Esc] displays a menu of available commands. Screen 4 is an example showing Host's Define Menu. The top line of each menu displays a list of available items or commands. One of these commands is within a highlighted block called the *menu cursor*, which indicates the current selection. The bottom line of each menu displays a short explanation of the current selection. Often this explanation includes a list of other commands that become accessible if that selection is entered. There are two methods of entering commands from these menus: *cursor selection* and *key selection.*



Screen 4. Host's Define Menu screen display.

Cursor Selection

To select a command from the menu using the cursor, use the keys [←], [→], [Home], and [End] to move the cursor to the desired selection. As the cursor moves, the explanation below the command list changes to correspond with the highlighted command. When the menu cursor is on the desired command, press [Enter].

When there are more items or commands in a menu than can be displayed across the Menu Space, an arrowhead symbol is displayed at one or both ends of the menu. These arrowhead symbols show that additional selections are available in the indicated direction. These additional selections can be displayed by moving the cursor beyond the last item or command currently displayed at the end of the Menu Space. In addition to using [←], [→], [Home], and [End], you can use [Ctrl]+[→] to display the next line of menu selections, and [Ctrl]+[←] to display the previous line of menu selections. Screen 7 on page 3-19 shows an example of a menu extending beyond available screen display space. Table 4 lists the cursor movement commands available in menus.

Key Selection

Usually, each command or item on a menu begins with a unique letter. Rather than moving the menu cursor to your selection then pressing [Enter], you can press the key corresponding to its first letter. This technique is faster than cursor selection and is designed to improve the efficiency of users familiar with the program. Some menus, however, are composed of file names over which the program has no control. These menus often contain several file names beginning with the same letter. Since key selection scans the items from left to right, selecting the first item beginning with the letter entered, it might not locate the intended item; here, cursor selection is the safer method.

Naming Things in Host

Each Application Module, Host Data File, Instruction file, or table output file has a name assigned to it by the Host operator who creates it. There are many places in Host where you are asked to name of one of these files. These names must conform to the DOS rules for file names as documented in your PC/MS-DOS manual. Whenever a file name is entered into Host, each character is converted to upper case and any character not meeting the DOS rules is rejected. It is therefore impossible to enter an invalid file name into Host.

Each Host Label Format, Dimension, Variable, Table Definition, or Graph Definition also has a name assigned to it by the Host operator who creates it. There are numerous places in Host where you are asked for the name of one of these items. When it is necessary to distinguish these item names from file names, this Guide calls them *Identifiers*. Host Identifiers must conform to the following three rules:

1. they must begin with a letter (A-Z) or underline character (_);
2. the first letter may be followed by any combination of letters (A-Z), digits (0-9), or underline characters (_); and
3. the name may be from one to eight characters long.

22

TABLE 4.

Host cursor movement commands for use in menus.

IBM AT Command Key	Function in Host	WordStar Command*
[Esc]	Same as the menu selection "Quit" if that selection appears on the menu. (Similar, not identical, to WordStar ^U.)	^U
[Home]	First section on menu.	^QS
[End]	Last selection on menu.	^QD
[←]	Left one menu selection. Last selection on menu if already on first selection.	^S
[→]	Right one menu selection. First selection on menu if already on last selection.	^D
[Ctrl]+[←]	Previous page of menu selections if there are more selections than will fit on the screen.	^A
[Ctrl]+[→]	Next page of menu selections if there are more selections than will fit on the screen.	^F

* WordStar commands are represented using normal WordStar notation. In WordStar notation, holding down the [Ctrl] key while pressing [U] is represented by ^U.

Those familiar with Turbo Pascal should recognize that these are the same rules used for Turbo Pascal identifiers except for the length limit.

Like Turbo Pascal, Host does not distinguish between upper and lower case characters in Identifiers. When an Identifier is entered into Host, each character is converted to upper case and any character not meeting the above rules is rejected. Therefore, it is impossible to enter an invalid Identifier into Host. Table 5 shows some examples of valid and invalid Host Identifiers.

TABLE 5.

Examples of valid and invalid Host Identifiers.

Valid Identifier	Invalid Identifier	Why Identifier is Invalid
TOT_YR_5	5_YR_TOT	Identifiers may not begin with a digit.
LTABLE3	L_TABLE3	Identifiers may not contain spaces.
_GNP_CAP	$GNP/CAP	Identifiers may only contain A-Z, 0-9, and _.
AG_EXPTS	AG:EXPTS	Identifiers may only contain A-Z, 0-9, and _.

User or Manager Access

Host is designed to support two different categories of operators: those using existing Application Modules, and those developing and installing new Application Modules. Host calls the first category of operators *User* and the second category *Manager* and provides access based on these categories to various commands, settings, and data. Whenever you execute Host from the operating system, you must declare yourself a User or Manager. Once within Host, you can change your access level at any time by selecting Access from Host's Main Menu. Figure 1 shows the command structure of Host's Access Menu.

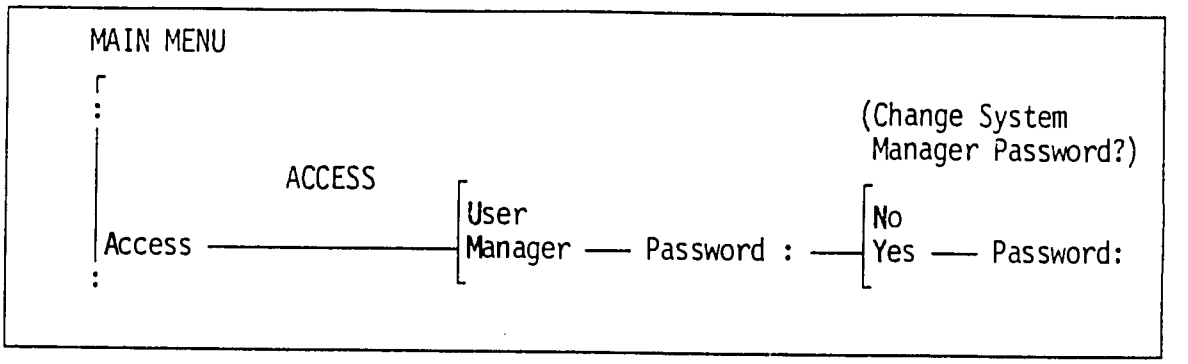


Figure 1. The command structure of Host's Access Menu.

User

If you select User from Host's Access Menu you are given access to all Host commands, settings, and data except those that could disrupt the structure of Host Data Files or the operation of Application Modules. Host does not ask for a password when you request User access. This gives you complete freedom when using existing Application Modules, but minimizes the possibility of unintentional damage to Host Data Files or Application Modules. Specifically, as a Host User, you can not delete Dimension or Variable Definitions from a Host Data File or change parts of these definitions that describe the physical structure of the data, since doing so would destroy at least some of those data. You are also prevented from changing Variable Definition settings used by Host's Data Editor to protect Application Modules from invalid values. If you select a command reserved for Managers, Host displays the following message in the Error Message Space:

Manager access required. Press any key to continue.

Host does not prevent a User from adding new Dimension and Variable Definitions, even though an Application Module can not use new variables unless it has been programmed to do so.

Manager

If you select Manager from Host's Access Menu, you are asked to enter the Manager's password. When you first receive Host, this password is "HOST" in upper case

characters. The Manager's password can be changed by a Manager at any time and is intended to protect Host Data Files from intentional corruption. If you enter the correct password, you are given access to all Host commands, settings, and data except those that could cause unintentional loss of data from Host Data Files. The additional capabilities given a Manager are necessary for developing and maintaining Application Modules.

The Host File System

Host is a large system composed of numerous disk files. Many of these files are part of the Host program itself; these are called *Host System Files* and are provided to you on the Host distribution disks. Appendix A includes a complete list of Host System Files. Other types of $files are created for specific purposes by Host operators. These include Configuration Files, Application Modules, Host Data Files, Library Files, and Instruction files.

A *Configuration File* stores default settings used throughout Host. Each time Host is executed, it establishes system-wide default values by reading the contents of the Configuration File in the current directory. You can easily change the settings in this Configuration File from within Host.

An *Application Module* is a compiled Turbo Pascal program specially constructed to function within Host. Almost all manipulation of data in Host is performed by running Application Modules.

A *Host Data File* contains input and output data used by Application Modules. Each Host Data File also includes a definition of the data it contains.

Host uses three types of *Library Files:* a *Label Library* contains Label Formats for annotating data output and for totaling data categories in tables; a *Table Library* contains Table Definitions specifying how data should be arranged in tables; and a *Graph Library* contains Graph Definitions specifying how data should be depicted in graphs.

Finally, an *Instruction file* contains instructions in sequence for running Application Modules, generating tables, and performing other operations. This sequence of operations can be performed without operator intervention by processing the Instruction file containing them.

The following sections of this chapter describe each of these five file types in more detail.

Configuration Files

A Configuration File contains default settings for values used throughout the Host System. These include default file names and directories for several types of Host files, default output devices for tables and graphs, formatting specifications for printed output, and type of display screen or monitor. The use of these settings is explained in Chapter 4, **CONFIGURING HOST**. Table 6 briefly summarizes Host Configuration Files.

All Configuration Files are assigned the name and extension "SYSTEM.ENV" by Host, so only one Configuration File may exist in any disk directory. Each time Host is

TABLE 6.

A brief summary of Host Configuration Files.

Name	Extension	Contents
SYSTEM	ENV	Contains default settings for various values used throughout the Host System.

executed from DOS, it searches for a Configuration File in the current disk directory. If found, the contents of the Configuration File are read into memory for use during that session of Host. If a Configuration File is not found, one is created containing internal default settings. You can change any of these settings from within Host, then save the changes to disk. Changes are effective immediately after being saved and will also be used by Host the next time it is executed.

You may find it useful to keep several different Configuration Files in separate directories. You can create a Configuration File in any directory not already containing one by executing Host from that directory (see Using the DOS PATH Command on page 2-5). If you create several different Configuration Files in different directories, you can use any one of these Configuration Files by executing Host from the directory containing it. Chapter 13, **ADVANCED TOPICS**, includes a discussion of using Host with multiple directories.

Application Modules

An Application Module is a compiled Turbo Pascal program constructed to function within Host. Nearly all operations involving data in Host Data Files are performed by running Application Modules. Most Application Modules read data from a Host Data File, then use those data in calculations, and, if desired, store results in the same Host Data File. Chapter 9, **RUNNING MODULES**, explains how to run Application Modules. Table 7 summarizes Host Application Modules.

TABLE 7.

A brief summary of Host Application Module files.

Names*	Extension	Contents
	COM	A compiled Turbo Pascal program designed to function within Host and to use data stored in Host Data Files.

* Each Application Module is assigned a file name by the Host user who creates it.

Any task that can be expressed as a Pascal language computer program can be developed into a Host Application Module; Host does not restrict what an Application

Module can do. Possibilities include large scale simulations, interactive training exercises, or computer graphics presentations. Modules can be highly interactive, with decisions made by an operator during execution, or can be run without manual intervention by accepting values stored using Host's Data Editor.

As stated in Chapter 1, **INTRODUCTION**, Host was originally designed for models projecting data forward through time. Application Modules comprising such models are called *Projection Modules*. For these modules, Host provides special support features dealing with the additional dimension of time. All Host facilities accommodate variables having a time dimension.

The process of developing Application Modules is described in the Host Programmer Guide. The Guide you are currently reading assumes you are working with previously developed Application Modules. Each Application Module provided to you should be accompanied by documentation describing its function, construction, requirements, and use.

Each Application Module file is assigned a name by the person who creates it; a file name extension of "COM" is assigned by the Turbo Pascal Compiler. There may be as many Application Modules on disk as available storage space allows and, unless otherwise specified, Host expects to find these files in the default Module Directory (see Chapter 4, **CONFIGURING HOST**, for information on the default Module Directory).

Host Data Files

A Host Data File contains data used by Application Modules. Host makes no distinction between data values used as input to Application Modules and data values output by Application Modules; both types of values are stored in a Host Data File. Table 8 summarizes the components of a Host Data File.

Each Host Data File also includes a definition of the data within it. This definition tells Host the physical structure of the data, including the fundamental type of each variable, and how each variable is dimensioned. In addition, the data definition provides data entry information to Host's Data Editor, value scaling information to Host Application Modules, and output formatting information to Host's Table Generator. The data definition part of a Host Data File is explained in detail in Chapter 5, **DEFINING DATA**.

A Host Data "File" is really a relational database composed of six files. Three of these files contain the data and a definition of the data, while the other three files are indices providing quick access to the information in the first three files. All six files have the same file name; this name is assigned by a Host user when the Host Data File is created. File name extensions for these files are assigned automatically by Host. You can create Host Data Files by using Host's File Management facility or by making copies of existing Host Data Files (see Chapter 12, **MANAGING FILES**).

Although the data, data definition, and indices are stored in separate files, Host usually treats a Host Data File as a single logical unit with a single name. There may be as many Host Data Files on disk as available storage space allows and, unless otherwise specified, Host expects to find these files in the default Data Directory (see Chapter 4, **CONFIGURING HOST**, for information on the default Data Directory).

TABLE 8.

A brief summary of Host Data Files.

Names*	Extension	Contents
Data	DAT	Contains the data for all Variables defined in the data definition part of the Host Data File. Up to 8 megabytes of data can be stored in this file.
	IDX	The index to the data stored in the DAT file.
Data Definition	DIM	Contains Dimension Definitions. Up to 65,535 Dimensions can be defined in this file.
	DDX	The index to the information stored in the Dimension Definition file.
	VAR	Contains Variable Definitions. Up to 65,535 Variables can be defined in this file.
	VDX	The index to the information stored in the Variable Definition File.

* Each Host Data File is assigned a file name by the Host user who creates it.

Library Files

Host uses three types of Library Files: Label Library, Table Library, and Graph Library. Since Host's graphing facility is not complete, the Graph Library is not discussed in this Guide.

Host Libraries are composed of two files: a file containing the Library's information and an index file providing quick access to that information. Library file names and file name extensions are assigned by Host. You can create Host Libraries by using Host's File Management facility or by making copies of existing Libraries (see Chapter 12, MANAGING FILES). Only one Library of each type may be in any disk directory, though you may have several different Libraries of each type by keeping them in separate directories.

Label Library

A Label Library contains Label Formats created by a Host operator. Label Formats can be used to make Tables, Graphs, and Host Data Editor displays easier to understand by attaching text labels to categories, or subdivisions, of a variable. Label Formats

28

can also be used to aggregate categories of a variable during the production of a table or graph. There can be up to 65,535 Label Formats in each Label Library, and each Label Format is assigned a name by the Host user who creates it. Chapter 6, **DEFINING LABELS**, explains the use of Label Formats in detail.

TABLE 9.

A brief summary of Label Library files.

Name	Extension	Contents
LABEL	LIB	Label Formats for attaching text labels to categories or subdivisions, of variables, and for aggregating categories during the production of a table or graph.
LABEL	IDX	The index to the information stored in the file LABEL.LIB

A Label Library is composed of two files: a file named "LABEL.LIB" containing Label Formats, and a file named "LABEL.IDX" used by Host for quick access to Label Formats. Table 9 summarizes the two files comprising the Label Library. Whenever Host needs a Label Library, it looks for one in the default Library Directory (see Chapter 4, **CONFIGURING HOST**, for information on the default Library Directory).

Table Library

A Table Library contains Table Definitions created by a Host user. Each Table Definition is assigned a name by the Host user who creates it and there can be up to 65,535 Table Definitions in each Table Library. A Table Definition tells the Host Table Generator what variables to include in a table and how a table should appear when produced for display, printing, or storage to disk. The data to be displayed in a table are not contained in a Table Definition, but are extracted from a reference Host Data File when the table is generated. Chapter 7, **DEFINING TABLES**, explains the use of Table Definitions in detail.

A Table Library is composed of two files: a file named "TABLE.LIB" containing Table Definitions, and a file named "TABLE.IDX" used by Host for quick access to Table Formats. Table 10 summarizes the two files comprising the Table Library. Whenever Host needs a Table Library, it looks for one in the default Library Directory (see Chapter 4, **CONFIGURING HOST**, for information on the default Library Directory).

Instruction Files

A Host Instruction file contains instructions for running Application Modules, generating tables and graphs, and executing many other tasks. When an Instruction file is processed, its contents are interpreted and acted on by Host in the order in which instructions are encountered within the file. The use of Instruction files within Host is analogous to the use of Batch Files ("BAT") within the DOS operating system; they provide a convenient means of performing a series of operations without operator

TABLE 10.

A brief summary of Table Library files.

Name	Extension	Contents
TABLE	LIB	Table Definitions telling Host's Table Generator what variables to include in a in a table and how a table should appear when produced for display, printing, or storage to disk.
TABLE	IDX	The index to the information stored in the file TABLE.LIB

intervention. Chapter 11, **INSTRUCTION FILES**, describes the use of Instruction files in detail.

Each Instruction file is assigned a file name by the Host user who creates it, but the extension "INS" is assigned automatically by Host. Table 11 summarizes Host Instruction files. You can create Instruction files by using Host's Instruction file facility, by using a standard text editor, or by making copies of existing Instruction files. There may be as many Instruction files on disk as storage space allows and, unless otherwise specified, Instruction files are assumed to be in the default Instruction Directory (see Chapter 4, **CONFIGURING HOST**, for information on the default Instruction Directory).

TABLE 11.

A brief summary of Instruction files.

Names*	Extension	Contents
	INS	Instructions in sequence for running Application Modules, generating tables and performing various other operations without operator intervention.

* Each Instruction file is assigned a file name by the Host user who creates it.

WARNING

As the previous section explains, Host uses many different disk files. While Host is working with a file, part of the file is stored in DOS memory buffers (see Chapter 2, **INSTALLATION**) and other parts of the file may be stored in memory used by Host. When Host is finished with a file, any changes to the file that are stored in memory are written to disk, and any change in the size of the file is recorded in the ap-

propriate disk directory. If Host is interrupted, it may not have a chance to do these important file closing operations; files may be damaged and work may be lost.

When you execute Host, it knows whether the previous Host session ended normally or was interrupted. If the previous session was interrupted while Host was processing an Instruction file, Host displays the messages shown in Screen 5. Host's Instruction File facility is explained in Chapter 11, **INSTRUCTION FILES**. In this example, the interruption may have resulted from the instruction in line 2 of Instruction file "ED2". You should examine Instruction file ED2 in Host's Instruction File Editor to see what was happening when the interruption occurred. Perhaps instruction 2 runs a faulty Application Module or uses a corrupted Host Data File or Library. The interruption may also have been caused by an error in Host, by you rebooting your microcomputer, or by a power failure.

```
User Manager Quit                                              ACCESS
Restrict access to the normal user.
------------------------------------------------------------------------

                              WARNING

  • An error occurred while Host was processing the instruction
    in line 2 of the Instruction file ED2. Files in use at
    that time may not have been properly closed.

  • Refer to the Host User Guide for help in identifying the
    cause of the error and correcting it.



  • Use the arrow keys to highlight your menu choice then press
    [Enter], or press the key corresponding to the first letter
    of your menu choice. Pressing the [Esc] key is the same as
    selecting the choice "Quit".

------------------------------------------------------------------------
Host program system did not terminate normally.
```

Screen 5. Host was interrupted while processing an Instruction file.

If the previous session was interrupted, but not while Host was processing an Instruction file, Host displays the messages shown in Screen 6. Here, you may have been running a faulty Application Module or using a corrupted Host Data File or Library. Again, the interruption may also have been caused by an error in Host, by you rebooting your microcomputer, or by a power failure.

Regardless of what caused an interruption, you should be suspicious of any Host Data Files and Libraries in use when it occurred. As the previous section explains, Host Data Files and Libraries are databases composed of two or more related files. Host can use a Data File or Library only if the database structure is consistent and intact; damaged Data Files and Libraries may produce error messages or strange unpredictable results. Host's Manage Menu, described in Chapter 12, includes facilities for verifying the integrity of Data Files and Libraries. These facilities can detect and repair many of the errors in these databases so that they can be used again by Host.

```
User Manager Quit                                                    ACCESS
Restrict access to the normal user.
------------------------------------------------------------------------------

                                  WARNING

    * The previous Host session was interrupted.  Files in use at
      that time may not have been properly closed.

    * Refer to the Host User Guide for help in identifying the
      cause of the error and correcting it.



    * Use the arrow keys to highlight your menu choice then press
      [Enter], or press the key corresponding to the first letter
      of your menu choice.  Pressing the [Esc] key is the same as
      selecting the choice "Quit".

------------------------------------------------------------------------------
Host program system did not terminate normally.
```

Screen 6. The previous Host session was interrupted.

Using Query Fields

There are many places in Host where you are asked to enter the name of some item. The item may be a Label Format, Dimension, Variable, Table Definition, Graph Definition, or disk file. In any case, the name of the specific item you want may be difficult to remember. In Host, you can display a list of the names, and descriptions if available, of all possible entries by pressing [?]. Host prompts with this feature are called *Query Fields*, and are composed of a blank or *field* to be filled in, preceded by an inverse video "?".

When the item to be selected is a file, the list is presented as a horizontal "Lotus-style" menu in the Menu Space. Screen 7 shows a list of available Host Data Files in this form. If the item to be selected is a Label Format, Dimension, Variable, Table Definition, or Graph Definition, the list is displayed vertically in the Work Space, along with a brief description of each item in the list. Screen 8 shows an example of a vertical list. With either type of list, a cursor is displayed and all the normal cursor movement and paging functions are available. To select any item from one of these lists, first position the cursor on the name of the desired item, then press [Enter]. Pressing [Esc] returns you to the previous screen display without selecting any item in the list.

If there are many items in a list, it can be difficult to locate a specific entry in the list using paging and cursor movement commands. A special search feature is available to help you quickly locate any name in a long vertical list. This feature is similar to those in many text editing programs and is invoked by pressing [Alt]+[S] (^QF). If you issue this command, Host asks you to enter a search pattern. This pattern may be the first character, the first several characters, or the entire name of the desired item. Host then searches the list for the first name in the list matching this pattern. If an exact match is not found, Host locates the next closest name in alphabetical order. The page of the list containing the located name is displayed in

```
                                                                    SELECT FILE
EDDATA1A EDDATA0 LABOR1 EDDATA2 EDDATA1 EDDATA2A LABOR2 EMPLOY HOUSING1 URBANA ▸
Select Data File EDDATA1A.

  ▪ Use the arrow keys to highlight your menu choice then press
    [Enter], or press the key corresponding to the first letter
    of your menu choice.  Pressing the [Esc] key is the same as
    selecting the choice "Quit".
```

Screen 7. A Host menu composed of Data File names.

```
                                                                    SELECT VARIABLES
Contents of file EDDATA1A.VAR.
There are 71 variables in this file.

      Name      Description

  41  NIMIG     Number of immigrants for each region
  42  NOMIG     Number of outmigrants from each region
  43  OG        Overall (migration incl.) pop. growth
  44  ORIGJSS   Original data for junior secondary
  45  ORIGSEC   Original data for old secondary
  46  ORIGSSS   Original data for senior secondary
  47  POP       Population
  48  POP6      Population of Six Year Olds
  49  PROPB     Probability of birth
  50  RATIOPRI  Student-teacher ratio, Nigeria
  51  RJSS      Repeater rates Jr.Secondary - Nigeria
  52  RPOLYH    Repeater rates Polytech.H. - Nigeria
  53  RPOLYM    Repeater rates Polytech.M. - Nigeria
  54  RPREP     Repeater rates College Prep. - Nigeria

    ▪ Use [↓] or [↑] to point to a variable then press [Enter] to select.
```

Screen 8. A variable name selection list.

the Work Space and the cursor is placed on that name.

Using Toggle Fields

There are several places in Host where you are asked to enter a selection from a limited set of possible responses. Sometimes there are only two possibilities, such as "Yes" or "No", while in others there are several, such as "Console", "Printer", or "Plotter". Rather than guessing what the choices are in these situations, you may press [+] or [Space] to display the next possible value, and [-] to display the previous possible value. Host prompts with this feature are called *Toggle Fields* and are composed of a blank, or field to be filled in, preceded by an inverse video "±". You can easily review all possible choices for a Toggle Field by repeatedly pressing [+], [Space], or [-]. Of course, you also may type in an entry for a Toggle Field. If you type in an incorrect entry, Host displays a message in the Error Message Space listing valid entries for that field.

Generating Reports

Host can quickly produce reports of Label Formats, Dimension Definitions, Variable Definitions, Table Definitions, or Graph Definitions. These reports include information about each member of the selected type. For example, if you request a Label Format report, Host generates a report containing information about each Label Format in the Label Library. As Figure 2 shows, you request these reports by first choosing the item to be reported on from Host's Define Menu, then selecting Report from the resulting menu. Host then displays a menu of possible report destinations. Since these commands are the same regardless of the item to be reported on, information in this section applies to all five types of reports.

Reports can be sent to the printer, the screen, or a disk file. No matter what destination is chosen, the information in a report is the same and report generation can be terminated by pressing [Esc]. If you send a report to the printer, it is formatted using Configuration File printer settings and is arranged so that as many items as possible appear on each page (see Chapter 4, **CONFIGURING HOST** for information on printer settings). Reports sent to disk files are also formatted using Configuration File printer settings, resulting in *print-ready* files that can be transferred to another machine for printing, or incorporated into word processing documents.

As Table 12 shows, report disk files are assigned names and extensions by Host according to report type. If a file with the listed name and extension already exists, it is renamed with the extension "BAK" before the new report file is created. If a back-up report file already exists, it is erased before renaming the existing report file with extension "BAK".

Two example reports are presented here. Do not be concerned by unfamiliar terms in these examples; they are explained in following chapters of this Guide. Screen 9 shows part of a Dimension definition report sent to the display screen. Most reports sent to the screen must be divided into two or more display pages. You may move forward or backward through these pages by pressing [PgDn] and [PgUp]. These pages are stored in memory, so the number of pages retained by Host depends on the amount of memory available in your computer. If the entire report can not be stored, you can not always page up to the first part of the report. Figure 3 shows one page of a Label Report that has been sent to the printer. Notice that, using the current Configuration

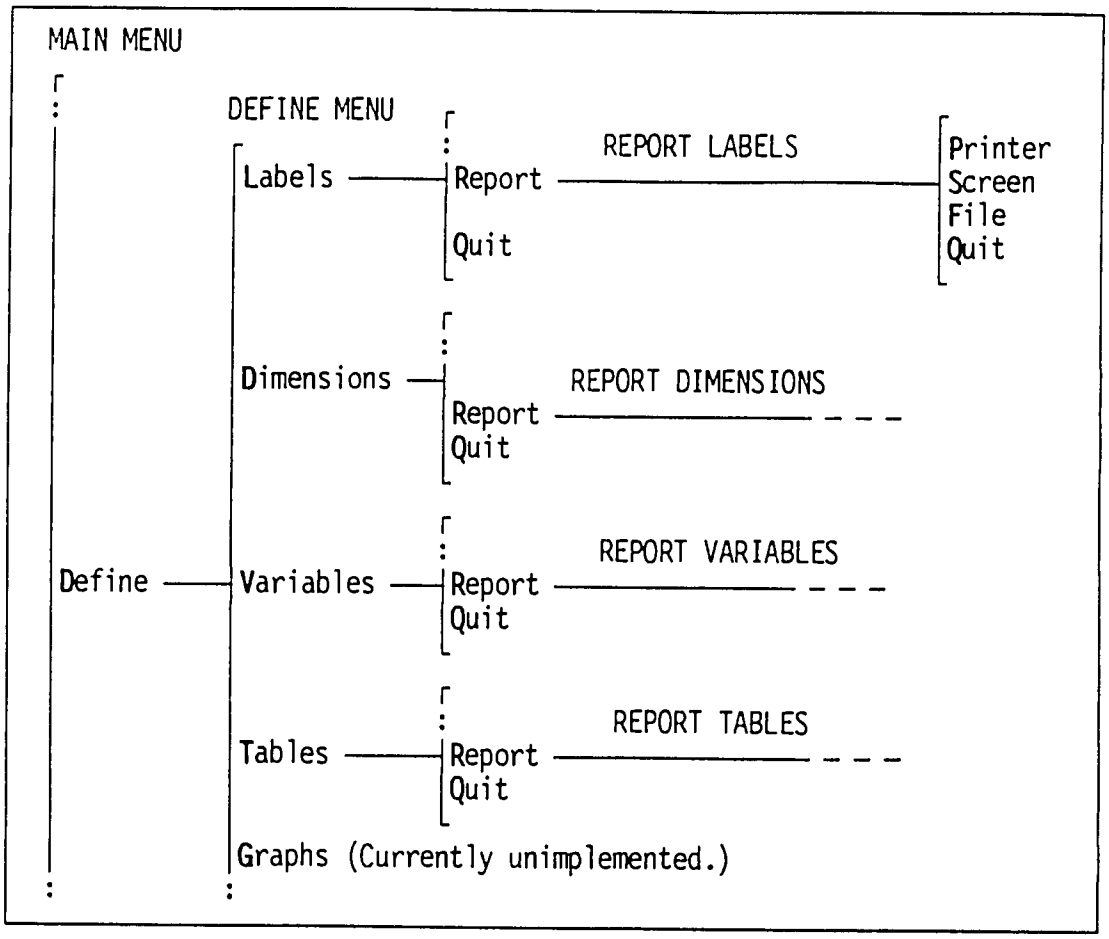


Figure 2. A schematic of Host commands for obtaining reports.

TABLE 12.

Standard Host names for reports sent to disk files.

Contents of Report	Report File Name and Extension*
Label Formats	LBLRPT.TXT
Dimension Definitions	DIMRPT.TXT
Variable Definitions	VARRPT.TXT
Table Definitions	TBLRPT.TXT
Graph Definitions	GPHRPT.TXT

* If a file with this name and extension already exists, it is renamed with the extension "BAK".

```
Dimension Name : AGE
   Description : Age classes

Maximum Number of Levels :       66
Current Number of Levels :       66


          Starting Value :        0
Increment Between Levels :        1

 Maximum Effective Level :       65


LABEL FORMAT:    Name      Description

                 NONE      No Label Format



Dimension Name : CBAGES
   Description : Child bearing ages

Maximum Number of Levels :       40
Current Number of Levels :       40


          Starting Value :        1
Increment Between Levels :        1

 Maximum Effective Level :       40


LABEL FORMAT:    Name      Description

                 NONE      No Label Format



Dimension Name : COLSLT
   Description : Column dimension for life table

Maximum Number of Levels :       10
Current Number of Levels :       10


          Starting Value :        1
Increment Between Levels :        1

 Maximum Effective Level :       10


LABEL FORMAT:    Name      Description

                 NONE      No Label Format
```

Figure 3. A sample page of a printed Label Report.

```
                                                    REPORT DIMENSION
Dimension attributes report for file EDDATA1A.
------------------------------------------------------------------------

Dimension Name : AGE
   Description : Age classes

Maximum Number of Levels :      66
Current Number of Levels :      66


         Starting Value :        0
Increment Between Levels :       1

 Maximum Effective Level :      65


LABEL FORMAT:    Name      Description

                 NONE      No Label Format

     * Press [Esc] to quit; [PgUp], [PgDn] to page up or down.
------------------------------------------------------------------------
```

Screen 9. A dimension report screen display.

File printer settings, Host was able to fit three Label Formats on this page. Each page of a printed report has a heading containing the date, report type, and page number.

Using DOS From Within Host

While working within any computer program, you may often need to execute operating system commands. Host provides you with convenient access to DOS commands through the *DOS command on Host's Main Menu. After you select this command, Host clears your screen and displays the following command prompt:

Type **Host** to return to Host.
C:\HOST>

The first line of this message tells you how to return to Host. The DOS-style prompt below this line always lists the current drive and directory.

You may enter any DOS command after this prompt (e.g., DIR, COPY, ERASE, RENAME), including commands that execute other programs such as Lotus 1-2-3, Turbo Pascal, or your text editor. In this mode, Host passes any command you enter directly to DOS for interpretation and execution so that, although Host is still active and in memory, it appears as if you are in DOS. The amount of memory available to other programs at this point depends on how much memory is installed in your microcomputer, and how much of that memory is occupied by Host and other memory resident programs. If there is not enough memory to execute a command or program, Host will notify you and return to the Main Menu.

After DOS has finish executing a command, Host redisplays the DOS-style prompt so that you can enter another DOS command. When you want to return to Host's Main Menu,

type "host" at the DOS-style prompt as follows:

```
C:\HOST>host[Enter]
```

If you have changed the current drive or directory while issuing DOS commands, Host returns these settings to the values they had before you executed Host's *DOS command.

CHAPTER 4
CONFIGURING HOST

CONFIGURING HOST

Host stores values used throughout the Host System in the Configuration File (see Chapter 3, GETTING STARTED). These values include default file names and directories for several types of Host files, default output devices for tables and graphs, formatting specifications for printed output, and color setting for the display screen. Host assigns all Configuration Files the name "SYSTEM.ENV", therefore only one Configuration File can exist in any disk directory. Each time you execute Host, it looks for a Configuration File in the current disk directory. If one is found, its contents are read into memory for use during that Host session. If Host does not find a Configuration File, it creates one containing internal default settings. You can change any of these settings by selecting the Configure command from Host's Main Menu.

```
Configuration file has been saved.                                  CONFIGURE
FILES Output Directories Printer Monitor Save Quit
Edit Data, Comparison and Instruction file names.
------------------------------------------------------------------------------
FILES                      PRINTER SETTINGS        MONITOR SETTING

Data        : EDDATA1A     Left Margin   :  10     Monochrome : Yes
Comparison  : EDDATA2A     Top Margin    :   6
Instruction : ED2          Bottom Margin :   6
                           Page Length   :  66
OUTPUT DEVICES             Line Length   :  65

Table : Console            Page Eject    : Yes
Graph : Plotter            Setup string  : None

DIRECTORIES

Data        : \TURBO\RUN\DATA\
Module      : \TURBO\RUN\MODULE\
Library     : \TURBO\RUN\LIBRARY\
Instruction : \TURBO\RUN\DATA\

------------------------------------------------------------------------------
                                                                        INS
```

Screen 10. A sample Host configuration facility screen display.

If you select Configure from Host's Main Menu, Host displays a screen showing the current configuration settings arranged in several categories. Screen 10 shows a typical Host configuration screen. To change any of these settings, select the appropriate category from the Configure Menu. This tells Host to place the cursor in the first setting of the selected category. Figure 4 is a schematic showing the command structure of Host's Configuration facility. Each category and setting displayed on the configuration screen is explained in the following sections of this chapter.

```
                    CONFIGURE MENU
MAIN MENU
 :                  Files
 |                  Output
 |Configure —[/]—   Directories
 |                  Printer
 |                  Monitor
 |                  Save
 |                  Quit
 :
```

Figure 4. Commands providing access to Host's default settings.

Default Files

As described in Chapter 3, **GETTING STARTED**, Host uses several types of files for storing different kinds of information. Three of these file types, Host Data Files, Comparison Files, and Instruction files, are given names by the Host operator who creates them. Before Host can do anything requiring files of these types, you must tell it the name of the file you want to use. To save you from having to type one or more file names before each operation, Host allows you to declare default names for a Host Data File, Comparison File, and Instruction file. Host stores these file names in a Configuration File and, unless you explicitly enter other names, uses them whenever files of these types are required.

You can type in default file names or select from existing files using Host's Query Field feature. Several Host Data Files and Instruction files are included on the Host distribution disks and, if you have copied these to your hard disk, you can select some of them as default files. You may also be given Host Data or Instruction files as part of specific Host-based model applications. However, files named as defaults need not already exist; you can create them either before or after changing the configuration settings. If any of the files you name as defaults do not exist, Host tells you when you try to save the configuration settings. Host then gives you the option of saving the settings as they are or editing them further. Any Host facility requiring the default files will also complain if they do not exist and will return to Host's Main Menu. You can create Host Data Files using Host's Manage Menu (Chapter 12, **MANAGING FILES**) and you can create Instruction files using Host's Instruct Menu (Chapter 11, **INSTRUCTION FILES**).

Host Data File

A Host Data File contains input data required by Application Modules and output data produced by Application Modules. These files are described in Chapter 3, **GETTING STARTED**. When a Host Data File is needed for an operation, Host uses the default Host Data File unless another is explicitly specified.

Comparison File

A Comparison File is a Host Data File intended to be used as a "base-run" against which to compare another Host Data File in tables and graphs. When data from a Comparison File are called for in a table or graph, data from this file will be used unless a different Comparison File is explicitly specified.

Instruction File

An Instruction file is a file of instructions telling Host to run Application Modules, generate tables, draw graphs, and do other operations without operator intervention. These files are described in Chapter 3, **GETTING STARTED.** When Host requires the name of a specific Instruction file, it will use the default Instruction file unless you specify otherwise.

Default Output Devices

The default Table and Graph Output Device settings allow you to send all Host generated tables and graphs to a predetermined device. You can change the settings in these fields by typing in an appropriate device name or by using Host's Toggle Field feature. Table output can be directed to the Printer or the Console (display screen), while graph output can be directed to the Printer, Console, or Plotter. Table and graph output can also be stored in disk files, although these may only be specified when tables and graphs are generated.

Table Output Device

Each Host-produced table is sent to the default Table Output Device unless you enter a different destination when a table is generated. The two possible choices for this field are "Printer", or "Console".

Graph Output Device

All graphs generated using Host are sent to the default Graph Output Device unless you enter a different destination when a graph is drawn. The three possible choices for this field are "Printer", "Console", or "Plotter".

Default Directories

Unless you tell it otherwise, Host assumes you want to keep all Host-related files in the current directory. However, Host Data files, Application Modules, Library files, and Instruction files can all be kept in separate disk directories or subdirectories. This allows you to organize your files and directories in the way you find most useful.

Unless other directories are specified when selected file names are entered, Host searches for files as follows:

- Host looks for Host Data Files and Comparison Files in the default Data Directory;
- Host looks for Application Module Files in the default Module Directory;
- Host looks for Label, Table, and Graph Libraries in the default Library Directory; and
- Host looks for Instruction files in the default Instruction Directory.

The initial contents of all four of these entries is the directory, minus disk drive specification, from which you executed Host when the current Configuration File was created.

If you want to change the settings of these directories, move the cursor to the appropriate line and enter the desired directory path. If you include a disk drive specification in these directory paths, Host looks for the corresponding files on the designated disk drive; otherwise, Host looks in the specified directory of the current disk drive. If you are using removable disks or disk cartridges, it is better to omit disk drive specifications. Host will not permit you to leave any of the default directory fields until it contains a valid directory path. If you are not sure how to specify directory paths, refer to the chapters in your PC/MS-DOS manual on File Management and Tree-Structured Directories. Chapter 13, **ADVANCED TOPICS**, includes a discussion of using Host with multiple directories.

Data Directory

The default Data Directory applies to Host Data Files and Comparison Files. Host Data Files are described in Chapter 3, **GETTING STARTED.** Unless you specify a different directory when you enter the names of these files, Host searches for them in the default Data Directory. Host automatically places Label, Dimension, Variable, Table, or Graph report files in this directory (see Generating Reports, page 3-20). If no directory is specified when a table is generated, Host also places Lotus-import and Print-ready files in this directory (see Chapter 10, **GENERATING TABLES**).

Module Directory

The default Module Directory applies to Host Application Module files. Host Application Modules are described in Chapter 3, **GETTING STARTED.** If no other directory is specified when you enter the name of an Application Module, Host looks for the Application Module in the default Module Directory.

Library Directory

The default Library Directory applies to Host Label, Table, and Graph Library files. Host Library files are described in Chapter 3, **GETTING STARTED.** Host facilities requiring these files expect to find them in the default Library Directory. When performing operations allowing you to enter the name of one of these Libraries, Host

looks for the Library you name in the default Library Directory unless you specify a different directory. If any or all of these Libraries do not exist in the default library directory, you can create them using Host's Manage Menu (see Chapter 12, **MANAGING FILES**).

Instruction Directory

The default Instruction Directory applies to Host Instruction files. Instruction files are described in Chapter 3, **GETTING STARTED.** Unless you specify a different directory when you enter the name of an Instruction file, Host searches for it in the default Instruction Directory. You can create Instruction files in the default Instruction Directory, or any other directory, using Host's Instruct Menu (see Chapter 11, **INSTRUCTION FILES**).

Printer Settings

All reports and tables printed by Host are formatted according to the printer settings in the Configuration File. Five of these settings: Left Margin, Top Margin, Bottom Margin, Page Length, and Line Length, control the positioning of print on each page. The relationship of these settings to a printed page is illustrated in Figure 5. Defaults for these settings assume you are using 8½-inch wide by 11-inch long paper, and that your printer is printing 10 characters per inch and 6 lines per inch. The Page Eject printer setting tells Host whether to start each printed page on a separate sheet of paper, or to print without regard to boundaries between sheets of paper. Finally, the Setup String setting can be used to send special print style commands to your printer. Using these seven settings, you can produce Host reports and tables with a wide variety of page sizes and print styles.

Left Margin

The Left Margin setting tells Host the number of character spaces to leave blank on the left side of each page. The default setting of 10 spaces will result in a 1-inch left margin if your printer is printing 10 characters per inch. You should adjust this value if you want a different size left margin or if your printer is printing a different number of characters per inch. For example, if you want a 1-inch left margin, but your printer is printing 17 characters per inch, then you should change the value of this field to 17.

Top Margin

The Top Margin setting tells Host the number of lines to leave blank at the top of each page. The default setting of 6 lines will result in a 1-inch top margin if your printer is printing 6 lines per inch. For example, if you want a 1-inch top margin, but your printer is printing 8 lines per inch, then you should change the value of this field to 8.

Bottom Margin

The Bottom Margin setting tells Host the number of lines to leave blank at the bottom of each page. If your printer is printing 6 lines per inch, then the default

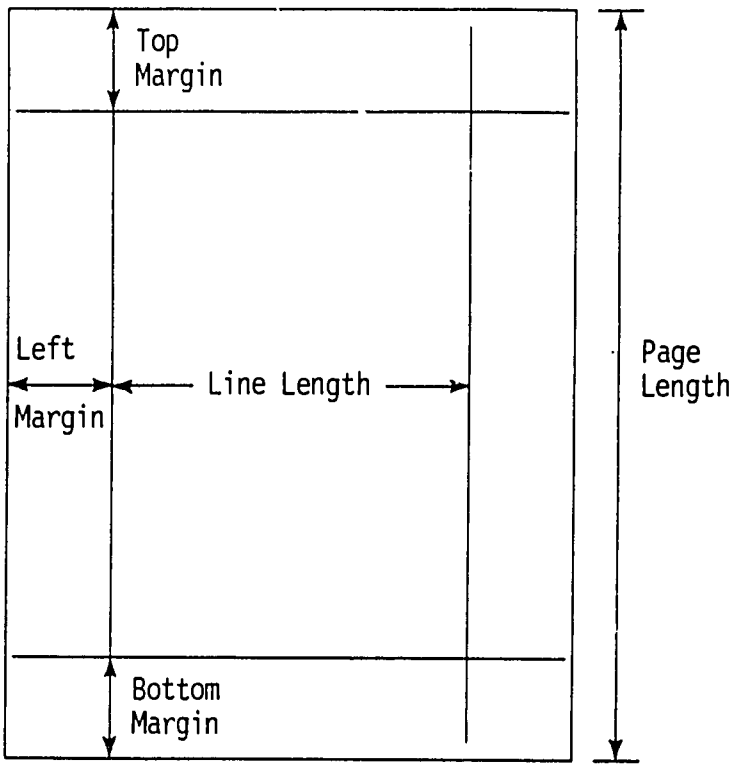


Figure 5. Host printed page layout parameters.

setting of 6 lines will result in a 1-inch bottom margin. If you want a different size bottom margin or if your printer is printing a different number of lines per inch, then you should adjust this value. For example, if you want a 1-inch bottom margin, but your printer is printing 8 lines per inch, then you should change the value of this field to 8.

Page Length

The Page Length setting tells Host the total number of lines, including Top and Bottom Margin, that can be printed on each sheet of paper. The default setting of 66 lines is correct if you are using 11-inch long paper and your printer is printing 6 lines per inch. If you are using a different length of paper or if your printer is printing a different number of lines per inch, then you should adjust this value. For example, if you are using 11-inch long paper, but your printer is printing 8 lines per inch, then you should change the value of this field to 88.

Line Length

The Line Length setting tells Host the maximum number of characters, not including the Left Margin, that can be printed on each line of a report or table. The default setting of 65 characters leaves a 1-inch right margin if you are using using 8½-inch wide paper, the Left Margin is set to 10, and your printer is printing 10 characters per inch. If any of these related factors change (paper width, Left Margin setting, or number of characters per inch), then you should adjust this value. For example, if you are using 14-inch wide paper, the Left Margin is 17 characters, and

your printer is printing 17 characters per inch, then setting the value of this field to 204 would leave a 1-inch right margin.

Page Eject

The Page Eject setting tells Host whether to start each printed page on a separate sheet of paper or to ignore the boundaries between sheets of paper. The default setting of "Yes" tells Host to send a Form-Feed command to the printer after printing every Page Length number of lines. If you have set your printer's top-of-form and Host's Page Length setting correctly, this should cause each printed page of a report or table to begin on a separate sheet of paper. If you change the setting in this field to "No", then Host will print continuously across the boundaries that separate sheets of paper. This can be useful if you need to save paper or decrease the time required to print reports and tables.

Setup String

Host's Setup String allows you to send commands to your printer to change fonts, print styles, and other print characteristics. If you enter commands to your printer into this field as a series, or "string" of characters, they are sent to the printer by Host before printing the first line of any report or table. Host's use of this feature is identical to that of Lotus 1-2-3 spreadsheet software. If you have used a setup string for your printer in Lotus 1-2-3, you can use that same string in Host.

Each brand of printer understands its own set of commands for changing type styles, character sizes, and other print characteristics. Refer to the manual for your printer for a description of these commands. Upper and lower case characters are not considered the same in the Setup String. Also, all spaces and punctuation will be sent to the printer exactly as they appear, so it is important to follow the examples in your printer manual precisely.

Characters that can be displayed on the screen can be entered directly into the Setup String. *Control characters*, those with ASCII codes from 0 to 32, can not be entered directly into the Setup String. These characters must be coded by entering [\] (backslash), followed by a three digit ASCII code. For example, the character generated by pressing [Esc] has an ASCII code of 27 and would therefore be entered as follows:

\027

Characters with ASCII codes above 32 can also be entered this way, although they are usually entered directly. For example, the character generated by pressing [E] has an ASCII code of 69, and could be entered directly as "E", or as follows:

\069

A complete Setup String can be composed of any combination of coded control characters and directly entered non-control characters. For example, the following Setup String will command an Epson or Epson-compatible printer to print 17 characters per inch and 8 lines per inch:

\015\0270

A Setup String can be long and complicated. For example, the following Setup String will command an Epson or Epson-compatible printer to print 12 characters per inch and 8 lines per inch in italics characters, ignoring the printer's paper-out detector:

```
\027:\0270\027%G\0278
```

Monitor Setting

This field tells Host whether to use color in its screen displays. If you are using a color display monitor and you want Host to use preset colors for all screen displays, set the Monochrome setting to "No". If you are using a monochrome display monitor, whether connected to a monochrome display adapter or the composite color output of a color display adapter, set the Monochrome setting to "Yes". Changing this setting does not affect the display unless you save the configuration settings. You will not damage your display hardware by choosing an inappropriate value for this field, but you might cause Host's displays to be difficult to read.

Saving Configuration Settings

Changes to the configuration settings do not affect Host unless you save them to the Configuration File on disk. To save the settings as displayed on the screen, select Save from the Configuration Menu. If all the specified default files exist in their respective default directories, the settings are written to the the Configuration File in the current directory. If one or more default files do not yet exist, Host tells you and gives you the option of saving the settings as they are or editing them further. If changes to configuration settings are saved, they are immediately in effect for the current Host session and will also be read by Host at the start of the next session.

41

CHAPTER 5.
DEFINING DATA

DEFINING DATA

As Figure 6 shows, a Host Data File is composed of two major components: data and a definition of the data. The data definition part of a Data File is further divided into two major components: *Dimension Definitions* and *Variable Definitions*.

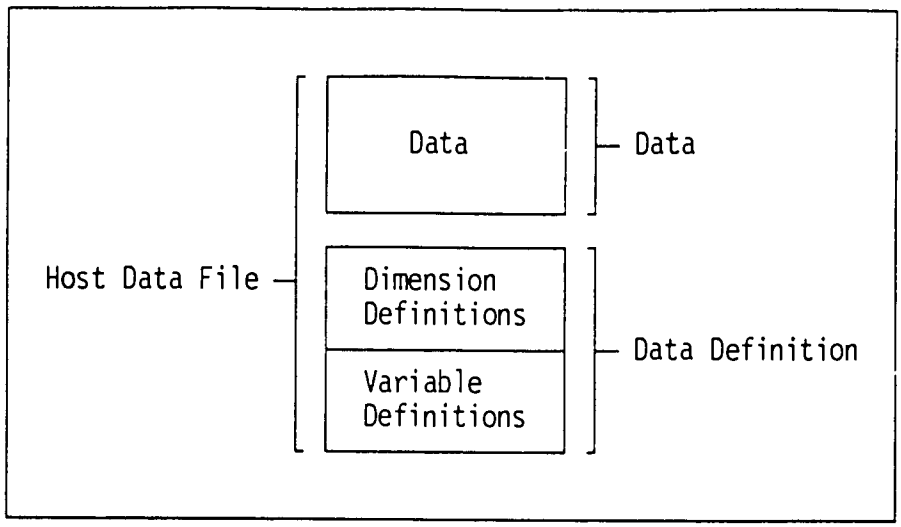


Figure 6. The basic components of a Host Data File.

All data in a Data File are assigned to *variables* and in Host a variable and its data are considered to be the same. In mathematics a variable is something having no fixed quantitative value; in Host, variables may also be used as constants or parameters. Further, Host makes no distinction between variables used as input and variables used as output. Host variables may even contain character rather than numeric data. Application Modules, not Host, determine how Host variables are used.

A Host variable may be subdivided according to one or more *dimensions*. For example, a variable containing a nation's population might be subdivided according to the dimensions age, sex, and geographical region. In Host, and in many computer programming languages, dimensions correspond to subscripts in mathematical equations and dimension levels correspond to subscript values. In the following equation, which calculates new enrolment in the first grade of a school system, the variables NE, e, and EP are all dimensioned by t (time) while e and EP are also dimensioned by a (the age of enrolment). This means there is a separate value of NE for each level of t and a separate value of e and EP for each combination of the levels of a and t. Thus, a dimension partially defines a variable and, as this example shows, a dimension may be shared by two or more variables.

$$NE_t = \sum_{a=amin}^{amax} e_{a,t} * EP_{a,t}$$

$EP_{a,t}$ Eligible population for entering school, i.e., those who did not enter school in previous year.

$e_{a,t}$ Enrolment ratio used to project new enrolment.

NE_t New entrants into the first grade.

Information about the type of data in each variable, how each variable is dimensioned, and how many levels are in each dimension is contained in the data definition part of a Host Data File. In addition to describing the physical structure of the data, this part of the database specifies how data should be entered, scaled, and presented. These dimension and variable *attributes* comprise individual Dimension and Variable Definitions. You can define new dimensions and variables and edit existing Dimension and Variable Definitions through Host's Define Menu.

Host's Dimension and Variable Menus, reached by selecting Define Dimension or Define Variable from Host's Main Menu, include commands to Add, Edit, Delete, List, and Report dimension and variable definitions. Unless you specify otherwise, the Dimension and Variable Definitions used by these commands are in the default Data File. If there is no default Data File, Host asks you to select one before proceeding from the Define Menu to the Dimension or Variable Menu. The File command on the Dimension and Variable Menus allows you to change to another Host Data File at any time. If you select a Data File using the File command, your selection does not become the default Data File, but does remain in effect until you return to the Define Menu or choose a different Data File.

Dimension and Variable Definitions are usually Added to a Data File when someone develops a new Application Module; these modules can not use Host dimensions or variables unless they are programmed to do so. Also, it is important that the structural definition of dimensions and variables in Host match the definitions programmed into the Application Modules that use them (see the Host Programmer Guide). The creation of new dimensions and variables is therefore considered to be the responsibility of an Application Module developer.

Anyone using Host may Edit Dimension and Variable Definitions. However, Host protects certain dimension and variable attributes from being altered. After a dimension or variable is Added to a Data File, attributes defining the physical structure of the data can not be changed. This protects against loss of data and preserves the correspondence between Host's definition of the data and the definition programmed into Host Application Modules. Further, variable attributes restricting data entry through Host's Data Editor can be changed only by a Host Manager. This allows a Host Manager to minimize errors caused by irrational data.

Only Host Managers may Delete variables and dimensions. Also, since the definition of a dimension partially defines variables subdivided by that dimension, a dimen-

sion can only be deleted if it is not needed by a variable. Host's facilities to List dimensions and variables and Report on their definitions are not restricted. They are available to display or print information from the data definition part of any Host Data File.

The remaining sections in this chapter describe each dimension and variable attribute and explain each command for manipulating Dimension and Variable Definitions. When reading these sections, keep in mind that Application Module developers are responsible for Adding new dimensions and variables to Host Data Files. Module developers should refer to the Host Programmer Guide for information about maintaining Data File consistency and for numerous examples of using Host dimensions and variables in Application Modules.

Defining Dimensions

To gain access to Host's Dimension Definition facility, select Define from the Main Menu, then Dimension from the Define Menu. Figure 7 shows the command structure of this facility.

Adding and Editing Dimensions

To create a new dimension, select Add from Host's Dimension Menu, then type in a dimension name conforming to the rules for Host Identifiers. To edit an existing dimension, select Edit from the Dimension Menu, then type in the name of a dimension or select one using Host's Query Field facility. After you have chosen the dimension to Add or Edit, Host displays the current settings of the dimension's attributes in the screen Work Space. If you are defining a new dimension, the attribute settings displayed are default values. Screen 11 is a typical Dimension Definition display showing the definition of a dimension named "AGE".

As in other parts of Host, values you may not change are displayed in dim characters. Some of these values are computed by Host while others are structural attributes protected after a dimension is defined; in either case, Host does not allow you to enter these fields. You can change unprotected dimension attributes using standard Host cursor movement and editing commands. Host protects different dimension attributes according to whether you are Adding or Editing a Dimension Definition and whether you are a User or a Manager. When you are finished viewing or setting attributes, press [/] or [Esc] to display the Add or Edit Dimension Menu. The following sections of this Guide describe each dimension attribute in detail.

Dimension Name

Host displays the name of the current dimension at the top of the screen. You enter this name when Adding a new dimension and you can not change it after a dimension has been defined. When Adding a new dimension, choose a name that reflects the use of that dimension.

Dimension Description

Since it is often difficult to explain the use of a dimension in its name, you may also enter a dimension description. This description can be from 0 to 40 characters long and should briefly describe a dimension's purpose. A dimension's description appears in lists and reports to provide additional information about the dimension.

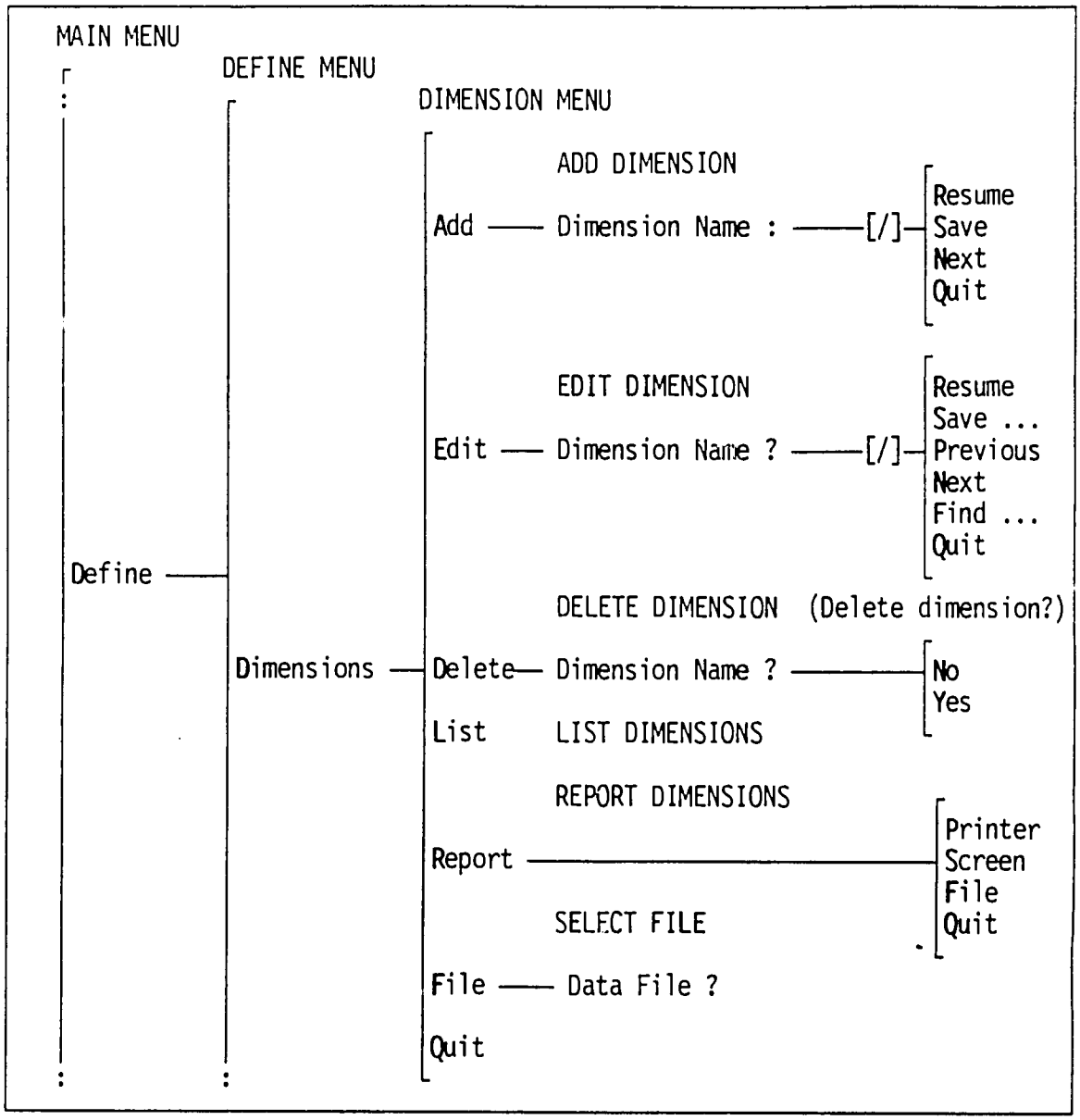


Figure 7. The command structure of Host's Dimension Definition facility.

Both Host Users and Managers may enter or edit the contents of the Dimension Description field. Although you may leave this field blank, an accurate description is useful to anyone using dimensions you create or modify.

Maximum Number of Levels

The Maximum Number of Levels of a dimension is the largest possible number of categories that can be represented by that dimension. For example, if you define a dimension named "GRADES" to represent grade levels in an education system model, you should set this value to the largest number of grade levels in any education system that might be simulated. Host uses the Maximum Number of Levels of each dimension of a variable to determine the amount of disk and memory space required to store that variable.

```
                                                    EDIT DIMENSION
Dimension Name : AGE                                Data File : EDDATA1A
   Description : Age classes___________________________

Maximum Number of Levels :      66
Current Number of Levels :      66

          Starting Level :       8
 Increment Between Levels :      1

* Maximum Effective Level :     65

LABEL FORMAT:    Name      Description

                 NONE      No Label Format

* (Maximum Number of Levels - 1) * Increment Between Levels + Starting Level

                                                          INS
```

Screen 11. A typical Dimension Definition screen display.

Each dimension in a Host Data File may have from 2 to 1024 levels. When Adding a new dimension, the default value of this attribute is 2. The Maximum Number of Levels must be the same in Host's definition of a dimension as it is in Application Modules that use that dimension (see the Host Programmer Guide). You can not change the value of this attribute after initially Adding a dimension to a Host Data File.

Current Number of Levels

Although the Maximum Number of Levels of a dimension defines the largest number of levels a dimension can represent, you may use less than this maximum number. For example, if the Maximum Number of Levels of the dimension "GRADES" in the example above is 20 but you are interested in modeling a primary school system that has only six grade levels, you could set the Current Number of Levels for "GRADES" to 6. Host, and Application Modules using variables dimensioned by "GRADES", would then consider that there are 6 rather than 20 grade levels. The Current Number of Levels can be any number from 1 to the Maximum Number of Levels and you may change this value at any time.

Note that changing the Current Number of Levels of a dimension does not change the amount of disk or memory storage space required by variables that use that dimension. However, reducing a dimension's Current Number of Levels causes data beyond the Current Number of Levels to be ignored by Host. Conversely, increasing a dimension's Current Number of Levels causes Host to use data previously ignored. Since you can not be sure what these data are, it is wise to view them in Host's Data Editor before they are used by Application Modules.

Effective Levels

By default, the levels of all Host dimensions start at 1 and are incremented by 1 up to a maximum of 1024 levels. The *Effective Levels* of a dimension begin at the

dimension's *Starting Level* and are incremented by the dimension's *Increment Between Levels*. This allows the levels of a dimension to be adjusted to values more appropriate to the use of the dimension. The formula for computing an Effective Level is as follows:

Effective Level = (level - 1) * Increment Between Levels + Starting Level

For example, if a dimension named "AGE" has a Maximum Number of Levels of 66, then the levels of AGE range from 1 to 66; if the Starting Level and Increment Between Levels for AGE are both 1, then the Effective Levels of AGE also range from 1 to 66. However, if the Starting Level of AGE is changed to 0, then the Effective Levels of AGE range from 0 to a *Maximum Effective Level* of 65 ((66 - 1) * 1 + 0). For another example, if the time dimension PERIOD has a Maximum Number of Levels of 1024, a Starting Level of 1985, and an Increment Between Levels of 5, then the Effective Levels of PERIOD range from 1986 to a Maximum Effective Level of 7101.

In the absence of a Label Format, Host labels each level of a dimension with its Effective Level. If you assign a Label Format to a dimension, the label specifications in the Label Format apply to the dimension's Effective Levels. Since you may need to know the Maximum Effective Level when Adding a new dimension or when defining a Label Format, Host computes and displays this value automatically.

Starting Level

The Starting Level of a dimension affects the labeling of dimension levels in tables, in graphs, and in Host's Data Editor. By default, and in the absence of a Label Format, Host labels the first level of all dimensions with "1". You can tell Host to begin labeling levels with some other number by changing a dimension's Starting Level. For example, if no Label Format is specified for the time dimension "PERIOD", then setting the Starting Value of PERIOD to 1986 causes Host to label PERIOD level 1 with "1986", PERIOD level 2 with "1987", and so on. You may set a dimension's Starting Level to any integer from 0 to 32767 and you may change this setting at any time. Note that, unless stated otherwise in documentation for an Application Module, the setting of this attribute affects only the labeling of dimension levels.

When you are using Label Formats, make sure that a dimension's Starting Value corresponds to the specifications in the dimension's Label Format (see Chapter 6, DEFINING LABELS). For example, the dimension AGE has a Starting Value of 1 and has been assigned a default Label Format named "AGEBYFIV". If the label specifications in the Label Format AGEBYFIV are as follows:

	Lower Bound	Upper Bound	Size of Interval (Inclusive)	or	Interval Label
1	0	64	5		
2	65	32767			65+

then Host labels data in AGE levels 1 to 4 with "0 - 4", data in AGE levels 5 to 9 with "5 - 9", and so on. If the Effective Levels of AGE should really start at 0, then

data presented in tables that use this Label Format will be mislabeled. You could correct this by changing the label specifications or by changing the Starting Value of the dimension AGE to 0. This type of error is subtle and easy to make until you are familiar with these relationships.

Increment Between Levels

The Increment Between Levels of a dimension affects the labeling of dimension levels in tables, in graphs, and in Host's Data Editor. In the absence of a Label Format, Host labels the levels of a dimension with integers beginning with the dimension's Starting Level and incremented by the dimension's Increment Between Levels. The default Increment Between Levels is 1 so that, if the Starting Level is also 1, Host labels the levels of a dimension with "1", "2", "3", and so on. If you change the Increment Between Levels of the dimension to 5, Host labels the levels of the dimension with "1", "6", "11", and so on. If you then change the Starting Level to 1986, Host labels the the levels with "1986", "1991", "1996", and so on. You may set a dimension's Increment Between Levels to any integer from 1 to 32767 and you may change this setting at any time. Note that, unless stated otherwise in documentation for an Application Module, the setting of this attribute affects only the labeling of dimension levels.

A dimension's Increment Between Levels affects the way Label Formats are applied to that dimension (see Chapter 6, **DEFINING LABELS**). The type of error described in the section above on Starting Level can also occur with the Increment Between Levels. Understanding the relationship between Dimension Definitions and Label Formats is the key to avoiding such errors.

Label Format

A Label Format can be assigned to a dimension by entering its name in the dimension's definition. Label Formats are described in Chapter 6, **DEFINING LABELS.** You can type in the name of any Label Format in the current Label Library or select one using Host's Query Field feature. Host extracts the description of the Label Format from the Label Library and displays it automatically. Whenever Host presents data categorized or subdivided by a dimension, it labels the data using the dimension's Label Format. If desired, you can override this default Label Format in a Table or Graph Definition by specifying a different Label Format.

The Time Dimension PERIOD

When you create a new Host Data File (see Chapter 12, **MANAGING FILES**) Host adds a dimension named "PERIOD" to the Data File automatically. This dimension is used by Host to represent the dimension of time for Projection Modules. Host requires a PERIOD dimension in all Data Files and will not allow you to delete it. If, for some reason, this dimension is lost from a Data File, you should create a PERIOD dimension using Dimension Add.

All variables in Host are classified as *periodic* or *nonperiodic.* Data for nonperiodic variables are stored in a Data File under PERIOD level 0; therefore, Host needs to know only the name of a nonperiodic variable to store or recall its data. Data for periodic variables may be stored in any period from PERIOD Level 1 to the Maximum Number of Levels of PERIOD. In effect, PERIOD is an implied dimension of all periodic variables. To store or recall data for a periodic variable, Host must know both the name of the variable and the PERIOD level of the data. Periodic and non-

periodic variables are discussed further in the section **Adding and Editing Variables** on page 5-11.

The levels of PERIOD may represent seconds, days, years, or any other unit of time. Host sets the Maximum Number of Levels of PERIOD to 1024 and the Current Number of Levels of PERIOD to the highest numbered period that contains data for any variable in the Data File. The Current Number of Levels of PERIOD increases automatically when a period of data is added to a Data File and only data from PERIOD level 1 to the Current Number of Levels of PERIOD appear in tables or graphs. Host never decreases the Current Number of Levels of PERIOD unless you tell it to. For example, if the Current Number of Levels of PERIOD is 1 and you project economic development forward 9 periods from period 1, Host increases the Current Number of Levels of PERIOD from 1 to 10 automatically. If you run the model again using the same Host Data File but projecting forward only 4 periods from period 1, Host does not decrease the Current Number of Levels of PERIOD from 10 to 5. Therefore, though there are data in periods 1 through 10, only data in periods 1 through 5 belong to the most recent projection. There are two ways to prevent data from previous projections from appearing in tables and graphs:

- you can specify which periods you want included in a table or graph by using a period list (see page 10-4, Specifying a Period List); or
- you can change the Current Number of Levels of the dimension PERIOD by editing its Dimension Definition or by using Host's PERIOD Instruction (Host Instructions are explained in Chapter 11, **INSTRUCTION FILES**).

You can set the Starting Level and Increment Between Levels of PERIOD to change Host's labeling of PERIOD. For example, if PERIOD levels represent years and you want the first PERIOD to be labeled "1986", then set the Starting Level of PERIOD to 1986. Projection Modules may use the Increment Between Levels of PERIOD to control the the timing of projections (see the Host Programmer Guide). Continuing the example above, changing the Increment Between Levels of PERIOD from 1 to 5 may cause a Projection Module to store projected data to a Data File every fifth year rather than every year. Host would correctly label the data in PERIOD 1 with "1986", the data in PERIOD 2 with "1991", the data in PERIOD 3 with "1996", and so on. However, if the Increment Between Levels of PERIOD is ignored by a Projection Module, then setting it to a value other than 1 may cause Host to mislabel the data.

As with any other dimension, you may also assign a Label Format to PERIOD (see Chapter 6, **DEFINING LABELS**). Using a Label Format, you can assign a text label up to 15 characters long to any Effective level of PERIOD. Host then uses that label wherever that PERIOD level would appear in displays or printouts.

Neither the Starting Level, the Increment Between Levels, nor the Label Format of the dimension PERIOD change the way period lists are specified when you generate tables and graphs. When specifying levels of the dimension PERIOD for generating tables and graphs or running Application Modules, you should consider levels of PERIOD to start at 1 and to be incremented by 1.

The Next, Previous, and Find Commands

The **Next** command on the Add Dimension Menu provides a convenient way to **Add** several dimensions. The **Next** and Previous commands on the Edit Dimension Menu allow

you to edit Dimension Definitions in sequence. Dimension Definitions are stored alphabetically by name in a Data File. Next allows you to edit the Dimension Definition following the current one in a Data File, and Previous allows you to edit the Dimension Definition before the current one in a Data File.

The Find command on the Edit Dimension menu allows you to locate any Dimension Definition in a Data File quickly. After selecting Find from the menu, Host asks you for the name of the desired dimension. You can type in a name or select one from a list using Host's Query Field feature. If a dimension with that name exists in the Data File, Host displays its attributes on the screen. If the dimension named is not in the Data File, Host tells you and asks you to enter a different name.

Saving Dimensions

There are Save commands on both the Add Dimension and Edit Dimension Menus. When you choose Save from the Add Dimension Menu, Host stores the new Dimension Definition in the Data File under the name you assign to it. When you choose Save from the Edit Dimension Menu, however, Host allows you to save any changes you have made to the Dimension Definition under its original name, or under a different name. This provides a quick, convenient way of creating several similar dimensions. You can Edit a dimension similar to the dimension you want to create, change some of its attributes, then Save the changes under a new name. Often this technique is more efficient than creating a new Dimension Definition using the Add command on the Dimension Menu.

Host may ask whether you want to save changes to a Dimension Definition after you select another Add Dimension and Edit Dimension Menu command. If you choose Next or Quit from the Add Dimension Menu, or choose Next, Previous, Find, or Quit from the Edit Dimension Menu before saving changes to a Dimension Definition, Host gives you an opportunity to save those changes before proceeding.

Deleting Dimensions

Host Managers can delete Dimension Definitions from a Host Data File by choosing the Delete command from the Dimension Menu. After choosing Delete from the Dimension Menu, you may type in the name of the dimension to delete or select it using Host's Query Field facility. Host does not permit you to Delete the reserved dimension PERIOD. Also, Host does not allow you to delete a dimension if that dimension is required by one or more variables; you must first delete all variables that use that dimension. Also, before a Dimension Definition is removed from a Data File, Host requires you to confirm that you want it deleted. If you answer "Yes", Host marks the Dimension Definition so that it is not used in any further operations. To remove deleted dimensions from a Data File and recover the disk space they occupy, you must Pack the Data File as explained in Chapter 12, MANAGING FILES.

Listing Dimensions

You can display a list of dimensions in the current Data File by choosing List from the Dimension Menu. This list is displayed vertically in the screen Work Space and you may view different portions of it by pressing the keys [↑], [↓], [PgUp], or [PgDn]. The number of dimensions in the Data File is displayed above the list. Press [/] or [Esc] to exit from the list and return to the Dimension Menu. Note that you may not select from this list as with the Query Field function, since you are not perform-

ing any operation other than listing. To help you remember this, the mode indicator says "LIST DIMENSIONS" rather than "SELECT DIMENSION" as it does when the Query Field function is invoked.

Reporting Dimensions

You can produce a complete report of all Dimension Definitions in a Data File using the Report command on the Dimension Menu. This report includes the same information and uses the same general layout as Add and Edit Dimension screen displays. After selecting Report from the Dimension Menu, select a destination for the report from the resulting Report Menu. You can send Dimension Definition reports to the Printer, the Screen, or to a File. If you select File, any existing dimension report file is renamed using the file name extension "BAK" and the report is written to the file "DIMRPT.TXT" in the default Data Directory.

Defining Variables

To gain access to Host's Variable Definition facility, select Define from the Main Menu, then Variable from the Define Menu. Figure 8 shows the command structure of this facility.

Adding and Editing Variables

To create a new variable, select Add from Host's Variable Menu, then type in a variable name conforming to the rules for Host Identifiers. To edit an existing variable, select Edit from the Variable Menu, then type in the name of a variable or select one using Host's Query Field facility. After you have chosen the variable to Add or Edit, Host displays the current settings of the variable's attributes in the screen Work Space. If you are defining a new variable, the attribute settings displayed are default values. Screen 12 is a typical Variable Definition display showing the definition of a variable named "POP". Since some attributes are not appropriate for some types of variables, the content and arrangement of this display varies. For example, several attributes are displayed only for numeric variables and dimension name fields are displayed only for variables with one or more dimensions.

As in other parts of Host, values you may not change are displayed in dim characters. Some of these values are computed by Host, others are structural attributes protected after a variable is defined, while still others are reserved for use by Host Managers. In any case, Host does not allow you to enter these protected fields. You can change unprotected variable attributes using standard Host cursor movement and editing commands. Host protects different variable attributes according to whether you are Adding or Editing a Variable Definition and whether you are a User or a Manager. When you are finished viewing or setting attributes, press [/] or [Esc] to display the Add or Edit Variable Menu. The following sections of this Guide describe each variable attribute in detail.

Variable Name

Host displays the name of the current variable at the top of the screen. You enter this name when Adding a new variable and it can not be changed after a variable has been defined. When Adding a new variable, choose a name that reflects the contents

of the variable.

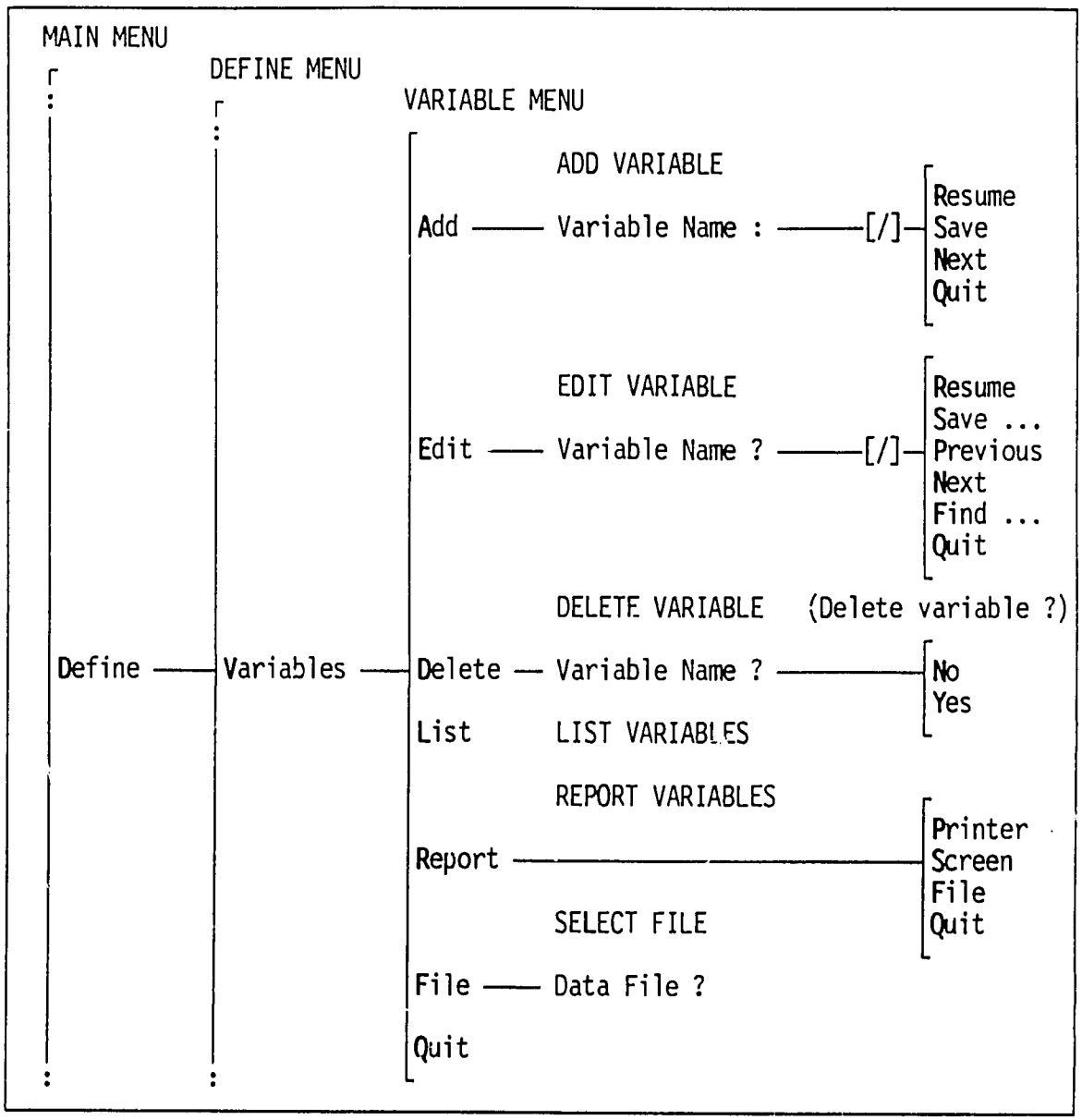


Figure 8. The command structure of Host's Variable Definition facility.

Variable Description

Since it is often difficult to suggest the contents of a variable in its name, you may also enter a variable description. A variable description can be from 0 to 40 characters long and should briefly describe a variable's use or contents. This description is stored as part of the Variable Definition and appears in tables, graphs, lists, and reports to provide additional information about the variable. Both Host Users and Managers may enter or edit the contents of the Variable Description field. Although you may leave this field blank, an accurate description is useful to anyone using variables you create or modify.

```
                                                        EDIT VARIABLE
Variable Name : POP                                  Data File : EDDATA1A
  Description : Population                        Size (bytes) : 2112

        Type : real

Minimum Value : 0.000000000          Unit of Measurement : None
Maximum Value :  1000000000             Scaling Exponent :    6

   Edit Mask : !!!!!!!!!!_________              Periodic : Yes
 Display Mask : ###.######          Number of Dimensions :  3

DIMENSIONS : Name       Description

        1   SEX        Male or female
        2   REGION     Regional division
        3   AGE        Age classes
```

Screen 12. A typical Variable Definition screen display.

Variable Type

Variables must be one of five fundamental Types: *byte, integer, real* (floating-point), *char* (character), or *string* (a sequence of characters). Variables of Type "byte", "integer", or "real" are numeric and may be used in calculations by Application Modules, displayed in tables, and depicted in graphs. Variables of Type "char" or "string" are non-numeric and may by used to communicate option settings to Application Modules, or to record assumptions made when Running Application Modules. Some Application Modules may also manipulate data in "char" or "string" Type variables. Variables of Type "char" or "string" can not be included in Host tables and graphs.

A variable's Type is set when a variable is Added to a Data File; this setting can not be changed after a variable is initially defined. The Type must be the same in Host's definition of a variable as it is in Application Modules that use that variable (see the Host Programmer Guide). When Adding a new variable, the default variable Type is "real".

The Type of a variable affects its size and, for numeric variables, the range of values its data can assume. Table 13 lists the size (in bytes) and value range of Host variable Types. Note that if a numeric variable will contain only whole numbers in the range -32768...32767, you can reduce required storage space by 75 percent if you declare the variable Type "integer" rather than "real". If a numeric variable will always contain positive values in the range 0...255, you can gain a further 50 percent reduction in storage space by declaring the variable Type "byte". Also note that data for a variable of Type "string" require a minimum of 256 bytes and more if the variable has one or more dimensions. Unfortunately, only 20 of these 256 bytes are accessible in Host's Data Editor because of the limitation on the length of the Edit Mask (see page 5-14, *Edit Mask*). We hope to correct this in a future revision of Host.

TABLE 13.

The size and range of Host variable Types.

Type	Size* (bytes)	Numeric Range*
byte	1	0...255
integer	2	-32768...32767
real	8	1.7976E+308...-1.797E+308
char	1	---
string	256	---

* Assumes use of a microcomputer with an Intel 8087/80287 numeric coprocessor.

Minimum and Maximum Values

The Minimum and Maximum Values are used by Host's Data Editor to check the range of numbers entered for a variable. These two variable attributes pertain only to numeric variables and are not displayed for non-numeric variables. A variable's Minimum and Maximum Values can only be changed by a System Manager. When Adding a new variable, the default Minimum and Maximum Values are the limits of the numeric range of the chosen variable Type.

Although the range of possible values numeric data can assume is restricted by the Type of the variable, it may be important to restrict this range further. For example, if a variable of Type "real" is intended to contain only probabilities, you might set its Minimum Value to "0" and its Maximum Value to "1". Host's Data Editor would then reject any attempt to enter values beyond this range. An Application Module using this data could therefore assume that, unless modified by itself or another Application Module, those data are in the range 0 to 1.

System Managers should be aware that data previously entered for a variable are not rechecked when one or both of these attributes are changed; range checking is performed by the Data Editor only as data are entered. Also the values chosen for the Minimum and Maximum value limits should be raw unscaled values. For example, if data for a variable are to be entered in units of millions, the Minimum and Maximum Value limits for the variable should not be entered in millions, but should be entered as unscaled values. Host's Data Editor automatically scales these limits according to the variable's Scaling Exponent and uses these scaled limits to restrict data entry. This concept is discussed further on page 5-18, *Scaling Exponent*. Finally, Host restricts Minimum or Maximum Values entered for variables of Type "byte" or "integer" to whole numbers. Since data entry for variables of these Types is restricted to whole numbers, fractional value range limits could result in a condition that the data entry operator could not satisfy.

Edit Mask

An *Edit Mask* is a sequence of special characters used by Host's Data Editor to check data entries as they are typed. This attribute is set by the System Manager and may not be modified by a User. The default Edit Mask is the one most appropriate for

61

the current variable Type. If you do not intend to modify this variable attribute, you do not need to know how Edit Masks are defined. However, reading the remainder of this section should help you to understand why Host's Data Editor does not allow you to enter certain characters.

Host determines the width of a variable's data entry fields from the number of characters in that variable's Edit Mask. For example, the Edit Mask "%%%%%%" tells the Data Editor to allow a maximum of 6 characters in each data entry field of the variable. An Edit Mask may be up to 20 characters long.

Each character in an Edit Mask tells Host what characters to allow in that position of the data entry field and not all characters in an Edit Mask need be the same. For example, each "%" in the Edit Mask "%%%%%%" tells the Data Editor to permit only the characters "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", and "-" in that character position of the entry field. If any other character is entered in that position of this field, Host beeps and does not display the invalid character. The Edit Mask "%%%%%%" would therefore allow only numeric entries in the range -99999 to 999999 inclusive.

For numeric variables, Host first validates an entry according to the Edit Mask, then attempts to convert the entry into data of the variable's Type. If Host can not complete this conversion because of an error, it places the cursor on the invalid character. Continuing the previous example, if you entered "-45-6", Host places the cursor on the second "-", since even though this entry is correct according to the Edit Mask, it can not be converted into a number. If the entry is a valid number but the number exceeds the range of the target numeric Type, then Host places the cursor after the last character in the entry. For example, the entry "978" for a variable of Type "byte" can be converted into a number, but that number exceeds the range of values that can be represented by a byte (0...255). Finally, after an entry for a numeric variable is validated according to a variable's Edit Mask and has been successfully converted into the target numeric Type, Host checks the entry against the variable's Minimum and Maximum Value limits (see page 5-14, *Minimum and Maximum Values*). This three step validation process ensures that data entered using Host's Data Editor meets a Host Manager's criteria.

TABLE 14.

Edit Mask characters available for numeric variables.

Mask Character	Characters Permitted	Typical Use for Data Entry
!	[0..9,.,E,e,+,-]	Real, i.e., floating-point numbers in normal or scientific notation.
%	[0..9,-]	Positive or negative integers. Number is positive if "-" is absent.
#	[0..9]	Positive integers only, such as byte values.

Host provides numerous masking characters for defining Edit Masks, each masking character implying a different restricted set of valid entry characters. Not all the implied restricted character sets, however, are appropriate for all types of data. For example, the Edit Mask "aaaaaa" would not allow data to be entered for a numeric variable since it prohibits entry of numeric characters. Therefore, Host restricts available masking characters to those appropriate for the current variable's data Type.

Table 14 lists the Edit Mask characters available for use with numeric variables, i.e., variables of Type "byte", "integer", or "real". Table 15 lists the Edit Mask characters available for use with non-numeric variables, i.e., variables of Type "char" or "string". Note that the masking characters listed in Table 14 can also be used in Edit Masks for non-numeric variables. Tables 14 and 15 show the implied set of valid characters that corresponds to each mask character in standard Pascal set notation. In this notation, each set is enclosed in square brackets ("["..."]"). Two such sets separated by a minus sign show that the characters in the set to the right of the minus sign should be removed from the set to the left. Finally, two characters separated by two periods, e.g.,"a..z", means all characters in the standard IBM PC character set from the first character to the second character inclusive.

Display Mask

The Display Mask affects the way data from a variable appear in tables and graphs. Both Host Managers and Users may change a variable's Display Mask to produce output in most appropriate format. When Adding a new variable, the default Display Mask is the one most appropriate for a variable of the current Type. Although eventually intended to provide other capabilities, Display Masks are currently limited to specifying the output field width and, for variables of Type "real", the output precision.

For variables of Type "byte", "integer", "char", and "string", only the character "#" is allowed in the Display Mask and the number of characters in the Display Mask determines the width of the output field. For example, the Display Mask "######" is appropriate for a variable of Type "integer" because the smallest possible integer (-32768) can be represented with 6 characters. If the Minimum and Maximum Values for an integer variable are 0 and 1 hundred respectively, and you do not expect values outside this range, then you should define the Display Mask as "###". You do not need to add extra characters to a display mask to separate columns of numbers in a Host table; Host does this automatically.

If a variable contains data that can not be formatted to fit within the width of the Display Mask, then those data appear as asterisks in tables and graphs. For example, if a variable with the Display Mask "###" contains a data value of -4563, that value appears as "***" in tables and graphs. Although a long text heading might make it appear that such data could be displayed within a column of a table, a variable's Display Mask, not the table column heading, determines the output field width.

These rules are the same for variables of Type "real" except that you may include the character "." to specify the desired output precision. If, for example, you do not expect values for a variable of Type "real" outside the range 0 to 1, but you want data for that variable to appear rounded to the nearest one-thousandth, then use the Display Mask "#.###". Note that Host rounds floating-point values to the precision specified by the Display Mask. If you have a variable of Type "real" whose data values are scaled in millions, then you probably do not want data for that variable to appear with fractional parts in tables and graphs. Here, you could use the Display Mask

TABLE 15.

Edit Mask characters available for non-numeric variables.

Mask Character	Characters Permitted	Typical Use for Data Entry
@	[all]-[/]	Allow any displayable character except the character "/".
*	[all]-[,+,-,/]	Allow all of the characters in the above set except the toggle field control characters.
A	[A..Z,a..z]	Forced alphabetic entry to upper case.
a	[A..Z,a..z]	Forced alphabetic entry to lower case.
&	[A..Z,a..z]	Leave alphabetic entry as entered.
?	[A..Z,a..z]	Character entered is not displayed.
+	[A..Z,a..z,0..9]	Forced alpha-numeric entry to upper case.
<	[!,%,#]	Allow characters valid in real field masks.
=	[%,#]	Allow characters valid in integer masks.
>	[#]	Allow characters valid in byte field masks.
V	[A..Z,a..z,_]	Allow first character of Host identifier.
v	[A..Z,a..z, 0..9,_]	Allow other characters valid in Host identifiers.
F	[A..Z,a..z, 0..9,#..), @,{,},~,!,-]	Allow characters valid in MS/PC-DOS file names. Edit mask for a standard eight character file name would be FFFFFFFF. "a"-"z" are converted to uppercase as they are entered.
D	[A..Z,a..z,0..9, :,.,\,#..),_,@, {,},~,!,-]	Same as above set, but allows entry of characters for disk drive and directory path.
x	[#,.]	Mask set for display masks containing only "#" and "." and of the form "#####.###" implying fieldwidth and number of decimal places.
X	[#]	Mask set for display masks containing only "#" and of the form "#####" implying only the fieldwidth.
L	[0..9,,,A,L,a, l,-]	Comma-delimited strings of positive integers and ALL. Used for period list notation.
M	[!,%,#,@,*,A,a, &,V,v,F,+,D,L, x,X]	Mask set for entry of edit masks themselves should contain all or most of the above case characters.

"######" to round data values for this variable to the nearest whole number. Note that this Display Mask does not provide for data values greater than 999,999.

Unit of Measurement

The Unit of Measurement is a text label that provides additional information about a variable in reports, tables, graphs, and in Host's Data Editor. It is used only for annotation and is intended to record the unit of measurement of numeric variables. Since this attribute is appropriate only for numeric variables, it is not displayed for variables of Type "char" or "string". Both Host Managers and Users may enter or edit the Unit of Measurement. When Adding a new variable, the default value of this attribute is "None" so that Host does not use a Unit of Measurement label for that variable.

A variable's Unit of Measurement can be from 1 to 10 characters long; otherwise there are no restrictions on this attribute. For example, the Unit of Measurement for a variable named "MAIZE" might be "Kilograms" or "Bushels" and the Unit of Measure for a variable named "GNP" might be "Yen", "Rupees", or "$". In the first example above, Host labels the variable MAIZE as "MAIZE (Kilograms)" whenever that variable appears in reports, tables, graphs, or in the Data Editor. A variable's Scaling Exponent (see below) also becomes part of this label if the Scaling Exponent is not 0. For example, if the Scaling Exponent of the variable MAIZE is 6, i.e., the data for MAIZE are stored in millions, then Host labels the variable as "MAIZE (Kilograms/10^6)". You should read this notation as "MAIZE in Kilograms divided by ten raised to the sixth power" or "MAIZE in millions of Kilograms".

Scaling Exponent

The Scaling Exponent tells Host, and anyone using Host's Data Editor, how data for a variable should be scaled. Scaling is the process of dividing data by ten raised to an integer power. For example, for a country with a large population, population data are commonly entered in millions rather than in raw unscaled numbers. In other words, the data are divided by ten to the sixth power. Since fewer digits must be entered when data are expressed in larger units, this simplifies data entry.

However, Host Application Modules expect to process unscaled data. A variable's Scaling Exponent tells Host how to unscale data before providing them to Application Modules. Host also rescales results generated by Application Modules before storing them in a Host Data File. Since Host does these operations automatically according to a variable's Scaling Exponent, Application Modules need no reprogramming to adapt them to different data scaling.

A Scaling Exponent is available only for variables of Type "real", since scaling is appropriate only for floating-point data. When Adding a new variable, the default Scaling Exponent is 0, that is, data for the new variable should be in raw unscaled units. Any Host User or Manager may change a variable's Scaling Exponent. The range of values you may enter for a Scaling Exponent is determined by the configuration of your microcomputer. On microcomputers currently supported by Host (IBM PC and compatible microcomputers with an Intel 8087/80287 numeric coprocessor), you may enter any Scaling Exponent in the range -307 to 307.

Note that the value range checking done by Host's Data Editor uses a variable's Minimum and Maximum Value limits as **scaled by the variable's Scaling Exponent** (see page 5-14, *Minimum and Maximum Values*). This means that the more a variable's Scaling

Exponent departs from 0, the narrower the range of values permitted by the Data Editor during data entry. To use an extreme example, if a variable of Type "real" has Minimum an Maximum Values of -1.797E+308 and 1.7976E+308 respectively, and you set its Scaling Exponent to 307, then the Data Editor permits the entry of values only in the range -0.1797 to 0.17976.

In tables, graphs, and in Host's Data Editor, a nonzero Scaling Exponent is shown in exponential notation following a variable's name. For example, if a variable named "GNP" has a Scaling Exponent of 6, then Host labels the variable as "GNP (/10^9)". You should read this notation as "GNP divided by ten raised to the ninth power" or "GNP in billions." If the Unit of Measurement of the variable GNP is "Dollars", then Host labels the variable as "GNP (Dollars/10^9)", which means "GNP in billions of Dollars."

Periodic

Host was originally designed to support models that project data forward through time. As a consequence, all variables in Host are classified as *periodic* or *nonperiodic*. Whether a variable is periodic or nonperiodic is determined when it is Added to a Data File; you can not change this attribute after initially defining a variable. When Adding a new variable, the default value for this attribute is "Yes", which means the variable is periodic.

The data in periodic variables may change through time so that a periodic variable has an implied time dimension (see page 5-8, **The Time Dimension PERIOD**). In effect, data for a periodic variable may exist in any period from period 1 to period 1024. Periodic variables may contain model parameters or initial assumptions, but typically contain data generated by a Projection Module. Often, periodic variables contain initial values in period 1 and projected data in successive periods. For example, you could define a periodic variable named "P_ENROL" to contain initial enrolment data for primary school in period 1 and projected enrolment data in following periods.

The data in nonperiodic variables are static through time; they do not change from one time period to another. Nonperiodic variables often contain model parameters or initial model assumptions. For example, you could define a nonperiodic variable named "MIGRATE" to contain a square transition matrix that does not change through time. Such a matrix could provide the rates at which people move from one region of a country to another.

Number of Dimensions

All variables in Host may have from 1 to 16 explicitly declared dimensions. If you consider time an implied dimension, then periodic variables in Host may have up to 17 dimensions. The Number of Dimensions is set when a variable is Added to a Data File; this value can not be changed after a variable is initially defined. The Number of Dimensions must be the same in Host's definition of a variable as it is in Application Modules that use that variable (see the Host Programmer Guide). The default Number of Dimensions for a new variable is 0.

A variable with no explicit dimensions can contain only a single datum for each period. Depending on the Type of the variable, the datum may be a character ("char"), a sequence of characters ("string"), or a *scalar value* ("byte", "integer", or "real"). For example, a variable named "XCHANGE", intended to contain a single currency exchange rate, should be defined as having 0 dimensions.

A variable with one dimension is often called a *vector* or a *one-dimensional array*. For example, if you want to Add a variable to contain age-specific fertility rates for a population Projection Module, you could define a variable named "ASFR" as having one dimension — the fertile ages of females in the population. A variable with two or more dimensions is commonly called a *matrix* or a *multi-dimensional array*. An example given earlier in this chapter was a square probability transition matrix commonly used in models incorporating a Markov process. To define such a matrix in Host, you would Add a variable with two identical dimensions. Each cell in this square matrix should contain the probability of moving from one state or level to another in the system being modeled. As with variables having no dimensions, the kind of datum contained in each level or cell of variables having two or more dimensions depends on the Type of the variable.

Dimension Name

If you define a variable as having one or more dimensions, you must enter the names of those dimensions as part of the Variable Definition. These names are entered when a variable is Added to a Host Data File; they can not be changed after a variable is initially defined. The Maximum Number of Levels of each dimension named in a variable's definition affects the variable's structure and size (see page 5-5, *Maximum Number of Levels*). Host extracts the Maximum Number of Levels and Description of each dimension from the Dimension Definition part of the Host Data File.

If you enter a Number of Dimensions greater than 0 for a new variable, Host displays a series of dimension name fields at the bottom of the Work Space. Depending on a variable's Type and Number of Dimensions, there may not be enough room to display the specified number of dimension name fields. If this occurs, you can gain access to additional dimension name fields by using Host cursor movement commands listed in Table 16. You may enter dimension names by typing them in or by using Host's Query Field feature. You may also delete dimension names by using [Alt]+[D] (^Y) and insert blank dimension name fields by using [Alt]+[I] (^N). Note that inserting a blank dimension name field may force Host to drop the last dimension name from the list.

Dimensions named in the definition of a new variable must already be defined as part of the current Host Data File. Therefore, make sure that all required dimensions have been defined using Host's Define, Dimensions, Add facility before you define new variables. Also, the order in which dimension names are listed is important. This order must be the same in Host's definition of a variable as it is in Application Modules that use that variable (see the Host Programmer Guide). When Adding a new variable, you may enter the same dimension name more than once and you may enter fewer dimension names than specified by the Number of Dimensions attribute. When you attempt to save a Variable Definition, Host first removes any blank dimension name fields then, if necessary, corrects the Number of Dimensions value. If adjustments are made, Host displays the corrected Variable Definition and asks whether you want to save the definition as is or edit it further.

Variable Size

Host computes and displays each variable's size and stores this value with a variable's definition and data. The value displayed in the Variable Size field is the number of bytes required to store a nonperiodic variable or a single period of a periodic variable. This value is affected by a variable's Type, Number of Dimensions, and the Maximum Number of Levels of each of its dimensions. After a variable is initially defined, Host protects these structural attributes from being modified so that the variable's size does not change.

61

Exponent departs from 0, the narrower the range of values permitted by the Data Editor during data entry. To use an extreme example, if a variable of Type "real" has Minimum an Maximum Values of -1.797E+308 and 1.7976E+308 respectively, and you set its Scaling Exponent to 307, then the Data Editor permits the entry of values only in the range -0.1797 to 0.17976.

In tables, graphs, and in Host's Data Editor, a nonzero Scaling Exponent is shown in exponential notation following a variable's name. For example, if a variable named "GNP" has a Scaling Exponent of 6, then Host labels the variable as "GNP (/10^9)". You should read this notation as "GNP divided by ten raised to the ninth power" or "GNP in billions." If the Unit of Measurement of the variable GNP is "Dollars", then Host labels the variable as "GNP (Dollars/10^9)", which means "GNP in billions of Dollars."

Periodic

Host was originally designed to support models that project data forward through time. As a consequence, all variables in Host are classified as *periodic* or *nonperiodic*. Whether a variable is periodic or nonperiodic is determined when it is Added to a Data File; you can not change this attribute after initially defining a variable. When Adding a new variable, the default value for this attribute is "Yes", which means the variable is periodic.

The data in periodic variables may change through time so that a periodic variable has an implied time dimension (see page 5-8, **The Time Dimension PERIOD**). In effect, data for a periodic variable may exist in any period from period 1 to period 1024. Periodic variables may contain model parameters or initial assumptions, but typically contain data generated by a Projection Module. Often, periodic variables contain initial values in period 1 and projected data in successive periods. For example, you could define a periodic variable named "P_ENROL" to contain initial enrolment data for primary school in period 1 and projected enrolment data in following periods.

The data in nonperiodic variables are static through time; they do not change from one time period to another. Nonperiodic variables often contain model parameters or initial model assumptions. For example, you could define a nonperiodic variable named "MIGRATE" to contain a square transition matrix that does not change through time. Such a matrix could provide the rates at which people move from one region of a country to another.

Number of Dimensions

All variables in Host may have from 1 to 16 explicitly declared dimensions. If you consider time an implied dimension, then periodic variables in Host may have up to 17 dimensions. The Number of Dimensions is set when a variable is Added to a Data File; this value can not be changed after a variable is initially defined. The Number of Dimensions must be the same in Host's definition of a variable as it is in Application Modules that use that variable (see the Host Programmer Guide). The default Number of Dimensions for a new variable is 0.

A variable with no explicit dimensions can contain only a single datum for each period. Depending on the Type of the variable, the datum may be a character ("char"), a sequence of characters ("string"), or a *scalar value* ("byte", "integer", or "real"). For example, a variable named "XCHANGE", intended to contain a single currency exchange rate, should be defined as having 0 dimensions.

A variable with one dimension is often called a *vector* or a *one-dimensional array*. For example, if you want to Add a variable to contain age-specific fertility rates for a population Projection Module, you could define a variable named "ASFR" as having one dimension — the fertile ages of females in the population. A variable with two or more dimensions is commonly called a *matrix* or a *multi-dimensional array*. An example given earlier in this chapter was a square probability transition matrix commonly used in models incorporating a Markov process. To define such a matrix in Host, you would Add a variable with two identical dimensions. Each cell in this square matrix should contain the probability of moving from one state or level to another in the system being modeled. As with variables having no dimensions, the kind of datum contained in each level or cell of variables having two or more dimensions depends on the Type of the variable.

Dimension Name

If you define a variable as having one or more dimensions, you must enter the names of those dimensions as part of the Variable Definition. These names are entered when a variable is Added to a Host Data File; they can not be changed after a variable is initially defined. The Maximum Number of Levels of each dimension named in a variable's definition affects the variable's structure and size (see page 5-5, *Maximum Number of Levels*). Host extracts the Maximum Number of Levels and Description of each dimension from the Dimension Definition part of the Host Data File.

If you enter a Number of Dimensions greater than 0 for a new variable, Host displays a series of dimension name fields at the bottom of the Work Space. Depending on a variable's Type and Number of Dimensions, there may not be enough room to display the specified number of dimension name fields. If this occurs, you can gain access to additional dimension name fields by using Host cursor movement commands listed in Table 16. You may enter dimension names by typing them in or by using Host's Query Field feature. You may also delete dimension names by using [Alt]+[D] (^Y) and insert blank dimension name fields by using [Alt]+[I] (^N). Note that inserting a blank dimension name field may force Host to drop the last dimension name from the list.

Dimensions named in the definition of a new variable must already be defined as part of the current Host Data File. Therefore, make sure that all required dimensions have been defined using Host's Define, Dimensions, Add facility before you define new variables. Also, the order in which dimension names are listed is important. This order must be the same in Host's definition of a variable as it is in Application Modules that use that variable (see the Host Programmer Guide). When Adding a new variable, you may enter the same dimension name more than once and you may enter fewer dimension names than specified by the Number of Dimensions attribute. When you attempt to save a Variable Definition, Host first removes any blank dimension name fields then, if necessary, corrects the Number of Dimensions value. If adjustments are made, Host displays the corrected Variable Definition and asks whether you want to save the definition as is or edit it further.

Variable Size

Host computes and displays each variable's size and stores this value with a variable's definition and data. The value displayed in the Variable Size field is the number of bytes required to store a nonperiodic variable or a single period of a periodic variable. This value is affected by a variable's Type, Number of Dimensions, and the Maximum Number of Levels of each of its dimensions. After a variable is initially defined, Host protects these structural attributes from being modified so that the variable's size does not change.

TABLE 16.

Cursor movement and editing commands for dimension name lists.

IBM AT Command Key	Function in Host	WordStar Command*
[Ctrl]+[Home]	First dimension name displayed on screen.	^QE
[Ctrl]+[End]	Last dimension name displayed on screen.	^QX
[↑]	Up one dimension name.	^E
[↓]	Down one dimension name.	^X
[PgUp]	Up one page of dimension names.	^C
[PgDn]	Down one page of dimension names.	^R
[Ctrl]+[PgUp]	First dimension name in list.	^QR
[Ctrl]+[PgDn]	Last dimension name in list.	^QC
[Alt]+[D]	Delete current dimension name.	^Y
[Alt]+[I]	Insert blank dimension name above current name.	^N

* WordStar commands are represented using normal WordStar notation. In WordStar notation, holding down the [Ctrl] key while pressing [U] is represented by ^U.

When you are Adding a new variable, Host updates the Variable Size field to reflect changes in the variable's definition. For example, if you are Adding an integer variable to the Data File, Host initially displays "2" in the Variable Size field. Two bytes are required to store a single integer number in most microcomputers. If you then set the Number of Dimensions for that variable to 2, then name the first dimension as "SEX", Host displays "4" in the Variable Size field (assuming the dimension SEX has a Maximum Number of Levels of 2 so that 2 * 2 = 4). If you then name the second dimension as "REGION" and the dimension REGION has a Maximum Number of Levels of 5, Host displays "20" in the Variable Size field (5 * 2 * 2 = 20).

Currently, the maximum size of a variable in Host is 64k bytes. For nonperiodic variables, this is an absolute limit; for periodic variables, this limit applies only to a single period of data. If you exceed the maximum size when Adding a new variable, Host warns you and requires you to reduce the variable's size before continuing. To reduce the size of a variable, you may delete one or more of its dimensions or change its Type.

The Next, Previous, and Find Commands

The Next command on the Add Variable Menu provides a convenient way to Add several variables. The Next and Previous commands on the Edit Variable Menu allow you to edit Variable Definitions in sequence. Variable Definitions are stored alphabetically by name in a Data File. Next allows you to edit the Variable Definition following the current one in a Data File, and Previous allows you to edit the Variable

Definition before the current one in a Data File.

The Find command on the Edit Variable menu allows you to locate any Variable Definition in a Data File quickly. After selecting Find from the menu, Host asks you for the name of the desired variable. You can type in a name or select one from a list using Host's Query Field feature. If a variable with that name exists in the Data File, Host displays its attributes on the screen. If the variable named is not in the Data File, Host tells you and asks you to enter a different name.

Saving Variables

There are Save commands on both the Add Variable and Edit Variable Menus. When you choose Save from the Add Variable Menu, Host stores the new Variable Definition in the Data File under the name you assign to it. When you choose Save from the Edit Variable Menu, however, Host allows you to save any changes you have made to the Variable Definition under its original name, or under a different name. This provides a quick, convenient way of creating a several similar variables. You can Edit a variable similar to the variable you want to create, change some of its attributes, then Save the changes under a new name. Often this technique is more efficient than creating a new Variable Definition using the Add command on the Variable Menu. Note that saving a Variable Definition under a different name does not copy the original variable's data to the new variable. You can, however, save both a variable's definition and its data under a new name using the Save command in Host's Data Editor (see Chapter 8, **EDITING DATA**).

Host may ask whether you want to save changes to a Variable Definition after you select another Add Variable and Edit Variable Menu command. If you choose **Next** or **Quit** from the Add Variable Menu, or choose **Next**, **Previous**, **Find**, or **Quit** from the Edit Variable Menu before saving changes to a Variable Definition, Host gives you an opportunity to save those changes before proceeding.

Deleting Variables

Host Managers can delete Variable Definitions from a Host Data File by choosing the **Delete** command from the Variable Menu. After choosing **Delete** from the Variable Menu, you may type in the name of the variable to delete or select it using Host's Query Field facility. Before a Variable Definition is removed from a Data File, Host requires you to confirm that you want it deleted. If you answer "Yes", Host marks the variable's definition and data so that they are are not used in any further operations. To remove deleted variables from the Data File and recover the disk space they occupy, you must Pack the Data File as explained in Chapter 12, **MANAGING FILES**.

Listing Variables

You can display a list of variables in the current Data File by choosing **List** from the Variable Menu. This list is displayed vertically in the screen Work Space and you may view different portions of it by pressing the keys [↑], [↓], [PgUp], or [PgDn]. The number of variables in the Data File is displayed above the list. Press [/] or [Esc] to exit from the list and return to the Variable Menu. Note that you may not select from this list as with the Query Field function, since you are not performing any operation other than listing. To help you remember this, the mode indicator says "LIST VARIABLES" rather than "SELECT VARIABLE" as it does when the Query Field

function is invoked.

Reporting Variables

You can produce a complete report of all Variable Definitions in a Data File using the Report command on the Variable Menu. This report includes the same information and uses the same general layout as Add and Edit Variable screen displays. After selecting Report from the Variable Menu, select a destination for the report from the resulting Report Menu. You can send Variable Definition reports to the Printer, the Screen, or to a File. If you select File, any existing variable report file is renamed using the file name extension "BAK" and the report is written to the file "VARRPT.TXT" in the default Data Directory.

CHAPTER 6
DEFINING LABELS

DEFINING LABELS

If no Label Format has been assigned, Host always annotates the levels of a dimension with corresponding integer labels. For example, if a dimension named "REGION" is defined in a Host Data File as having two levels, then without a Label Format those levels are labeled "1" and "2" when displayed or printed. Using Host's Label Format facility, however, you can tell Host to substitute text labels of your own choosing for these default integer labels. For example, by defining and assigning an appropriate Label Format, you can cause Host to label the levels of the dimension REGION in the above example as "Urban" and "Rural".

You can assign a default Label Format to any dimension and many different Label Formats may be used with the same dimension. Host's Data Editor, Table Generator, and Graph Generator can all use Label Formats to annotate data from Host Data Files. Host's Label Formats also provide a powerful mechanism for combining data into categories for presentation in tables and graphs. By using different Label Formats in several different Table or Graph Definitions, you can aggregate the same data numerous different ways. Chapter 7, **DEFINING TABLES**, and Chapter 8, **EDITING DATA**, describe fully the use of Label Formats in tables and in the Data Editor.

When the term "level" or "dimension level" is used in this chapter, it really means Effective Level. An Effective Level of a dimension is computed from the actual dimension level using the following formula:

Effective Level = (level - 1) * Increment Between Levels + Starting Level

where Increment Between Levels and Starting Level are parts of the dimension definition (see Chapter 5, **DEFINING DATA**). For example, if a dimension named "AGE" has a Starting Level of zero and an Increment Between Levels of five, then the first Effective Level of AGE is zero, the second Effective Level is five, the third Effective Level is ten, and so on. When using Label Formats, keep in mind that Host applies them to the Effective Levels of dimensions, i.e., the levels of dimensions as modified by their Starting Level and Increment Between Levels.

The Label Library

As explained in Chapter 3, **GETTING STARTED**, a Label Library consists of two files: one named "LABEL.LIB" containing Label Formats, and another named "LABEL.IDX" containing an index to the Label Formats. Since these file names are assigned by Host, there can be only one Label Library in any disk directory and, unless you specify otherwise, Host uses the Label Library in the default Library Directory. If a Label Library does not exist in the default Library Directory, you can create one using Host's Manage facility (see Chapter 12, **MANAGING FILES**). Each Label Library can contain up to 65,535 Label Formats.

Label Formats

A *Label Format* consists of a label name and description and one or more *label specifications*. Each label specification contains the lower bound and upper bound of

the dimension level range to which it applies, and either an interval size for generating equal-sized, automatically labeled subgroups, or a text label to assign to the specified range of levels. Each Label Format is assigned a name by the user who creates it. Any Host user may create, edit, or delete Label Formats. Figure 9 is a schematic of Host commands for manipulating Label Formats.

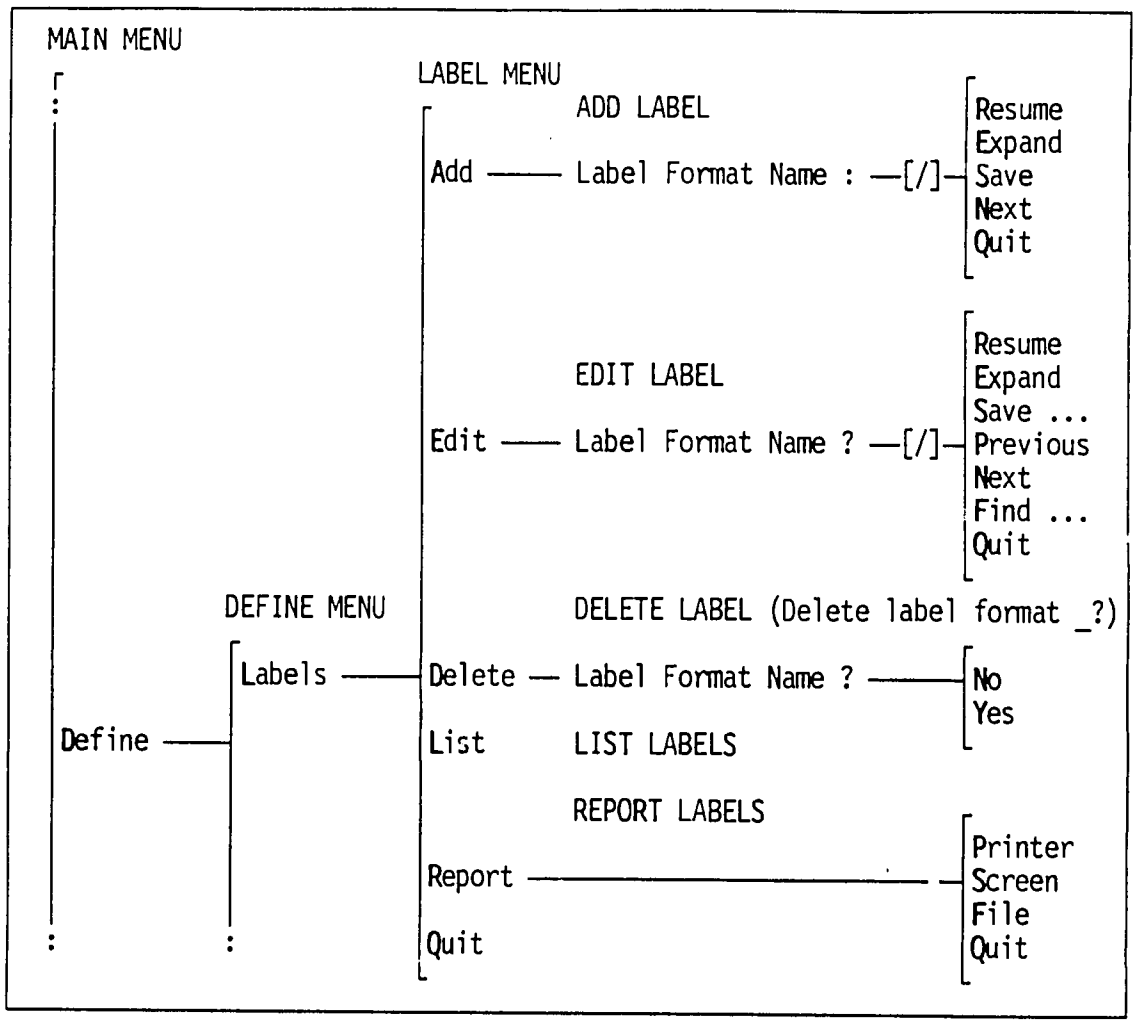


Figure 9. A schematic of Host commands for manipulating Label Formats.

Creating and Editing Label Formats

To create or edit a Label Format, first select Add or Edit from the Label Menu, then enter the name of a new or existing Label Format. Once you have entered a Label Format name, Host displays the contents of the Label Format. Screen 13 shows a Label Format Editor display of a Label Format named "SEX". In this example, there are only two label specifications, each assigning a text label to a single dimension level. The information displayed on this screen is described in the following sections.

```
                                                                    EDIT LABEL
Label Format Name : SEX
     Description : 1=Male 2=Female
____________________________________________________________________________
     Lower      Upper    Size of Interval        Interval
     Bound      Bound    (Inclusive)      or     Label

  1    1          1                               Male
  2    2          2                               Female


____________________________________________________________________________
                                                                    INS
```

Screen 13. An example Label Format Editor display.

The Label Format Description

A Label Format description should briefly describe the Label Format or its purpose. This description, which can be up to forty characters long, is stored as part of the Label Format definition and appears in lists and reports to provide additional information about the Label Format. Although you may leave this field blank, some description will be useful to anyone using Label Formats you create.

Label Specifications

A label specification tells Host to apply a text label to a dimension level, or range of dimension levels. A Label Format may be composed of from one to fifty label specifications. Each label specification consists of a lower bound, an upper bound, and an interval size or a text label.

The lower bound and upper bound establish the range of dimension levels to which the label specification applies. They must be integers in the range 0...32767, and the upper bound must be greater than or equal to the lower bound. In addition, the lower bound of each label specification after the first must be strictly greater than the upper bound of the previous label specification.

The interval size tells Host to divide the specified range of levels into groups of that size and to automatically assign an appropriate text label to each group. The interval size must be a positive integer and must evenly divide the interval established by the lower and upper bounds.

The Interval Label, which may be any text string up to fifteen characters long, annotates the specified dimension level range when Host displays or prints data. If you do not enter an interval size, Host generates a default text label for the range

established by the upper and lower bounds. You can accept this default label or enter a text label of your own choosing.

There are three types of label specifications that can be included in Label Formats to assign text labels to dimension levels: explicit, grouping, and computed.

Explicit An explicit label specification assigns a text label to a single dimension level.

Grouping A grouping label specification assigns a single text label to all levels of a dimension within a specified range.

Computed A computed label specification automatically generates a series of equal-sized grouping label specifications.

Explicit Label Specifications

In an *explicit* label specification, the lower bound is set equal to the upper bound, thus assigning the specified text label to a single dimension level. Host supplies a default integer text label, but you may substitute any text. Note that no interval size may be specified for this type of label specification since the "interval" defined by the lower and upper bounds consists of a single value. Both label specifications in Screen 13 above are of the explicit type.

Grouping Label Specifications

In a *grouping* label specification, the upper bound is greater than the lower bound and the text label assigned applies to all dimension levels between the lower and upper bounds, inclusive. Host creates a default text label for grouping label specifications of the form

LowerBound -- UpperBound

but you may substitute text of your own choosing. Screen 14 on the next page shows the definition of the Label Format WORKAGES. All three label specifications for this Label Format are grouping specifications. The value in the upper bound field in the third label specification is the largest integer that can be represented in the microcomputer running Host, thus the interval applies to the dimension levels 65 and above. No additional label specifications can be created below one with this value as its upper bound. You do not have to remember this number when you want to specify this type of open-ended interval. If you press [+] when the cursor is in the upper bound field, Host supplies this value for you. Note that you can not place non-consecutive dimension levels in the same group. Thus the Label Format defined below in screen 15 does **not** have the effect of creating two groups of levels.

Computed Label Specifications

A *computed* label specification is a convenient way to generate a series of equal-sized grouping specifications. Host creates an appropriate text label automatically for each generated specification. Although the generated specifications do not appear on the Label Format Editor screen, you can display them by using the Expand command on the Add or Edit Label Menu. To create a computed label specification, specify an upper bound greater than the lower bound and an interval size by which to subdivide the specified range of levels. The interval size must divide evenly into the range estab-

```
                                                                    EDIT LABEL
Label Format Name : WORKAGES
     Description : Agreggates 0-15, 16-45 and 46+__________
____________________________________________________________________________
     Lower      Upper    Size of Interval        Interval
     Bound      Bound    (Inclusive)      or     Label

  1    0          15                              0 - 15
  2   16          45                              16 - 45
  3   46       32767                              46 +


____________________________________________________________________________
                                                                        LMS
```

Screen 14. A Label Format illustrating grouping label specifications.

```
                                                                    ADD LABEL
Label Format Name : JOBCLASS
     Description : 1,3,4 = Laborer 2,5 = Professional
____________________________________________________________________________
     Lower      Upper    Size of Interval        Interval
     Bound      Bound    (Inclusive)      or     Label

  1    1          1                               Laborer
  2    2          2                               Professional
  3    3          3                               Laborer
  4    4          4                               Laborer_______
  5    5          5                               Professional


____________________________________________________________________________
```

Screen 15. (Warning!) A Label Format with no grouping label specifications.

lished by the upper and lower bounds. Screen 16 shows the definition of a Label Format named "AGEBYFIV". The first label specification is an example of a computed specification. The second is an example of a grouping specification. Like the last label specification in the WORKAGES Label Format above, it specifies an open interval of 65

and above. This example illustrates that label specifications within a Label Format need not all be of the same type.

```
                                                                    EDIT LABEL
Label Format Name : AGEBYFIV
     Description : 0 to 4 = 0 - 4...)= 65 = 65+___________
-----------------------------------------------------------------------------
     Lower      Upper    Size of Interval        Interval
     Bound      Bound    (Inclusive)      or     Label

  1    0         64        5
  2   65      32767                             65 +


                                                                    INS
```

Screen 16. A Label Format illustrating a computed label specification.

Editing Label Specifications

In addition to the within-field editing commands listed in Table 3, page 3-6, Host provides you with commands for deleting and inserting entire label specifications. You can delete a label specification by first placing the cursor in the row containing the label specification to be removed, then pressing [Alt]+[D] (^Y). You can insert a label specification above the one containing the cursor by pressing [Alt]+[I] (^N). However, Host will not allow you to insert a label specification where one can not be placed logically based on upper bound and lower bound field values. For example, you can not insert a label specification if the upper bound of the label specification above the current one is not at least 2 less than the lower bound of the current label specification. Also, you can not insert a label specification if the current label specification is the first one in the Label Format and its lower bound is 0. You can move freely among all fields within the current row, but you can not move up or down until the label specification on that row is consistent with label specification rules.

The Next, Previous, and Find Commands

The Next command on the Add Label Menu allows you to create a new Label Format after you have completed the current one. The Next and Previous commands on the Edit Label Menu provide a way of editing Label Formats in sequence. Label Formats are stored alphabetically by name in a Label Library. Next allows you to edit the Label Format following the current one in the Label Library, and Previous allows you to edit the Label Format before the current one in the Label Library.

The Find command on the Edit label menu allows you to locate any Label Format in the Label Library with ease. After selecting Find from the menu, you are asked for the name of the desired Label Format. You can enter this name directly or select one using Host's Query Field feature. If a Label Format with that name exists in the Label Library, Host reads it into the Label Format Editor. If the Label Format named is not found in the Label Library, Host tells you and asks you to enter a different name.

Saving Label Formats

There are Save commands on both the Add Label and Edit Label Menus. When you choose Save from the Add Label Menu, Host stores your new Label Format in the Label Library under the name you assign to it. When you choose Save from the Edit Label Menu, however, Host allows you to save any changes you have made to the Label Format under its original name, or under a different name. This provides a quick, convenient way of creating several similar Label Formats. You can Edit a Label Format similar to the one you want to create, change some of its settings, then Save the changes under a new name. Often this technique is more efficient than creating a new Label Format using the Add command on the Label Menu.

Host may also ask whether you want to save changes to the current Label Format when you select another Add Label or Edit Label Menu command. For example, if you choose Next or Quit from the Add Label Menu, or choose Next, Previous, Find or Quit from the Edit Label Menu before saving changes to a Label Format, Host gives you an opportunity to save those changes before proceeding.

Using Label Formats

You can assign a default Label Format to any dimension by using Host's Dimension Definition Facility (see Chapter 5, **DEFINING DATA**). Host will then use this Label Format whenever data from variables subdivided by that dimension are displayed in the Data Editor or presented in tables and graphs. It is possible, however, to use different Label Formats for the same dimension in numerous different Table and Graph Definitions (see Chapter 7, **DEFINING TABLES**). Note that only explicit label specifications are used in the Data Editor. When the Data Editor encounters grouping or computed label specifications, it substitutes integer labels for the dimension levels.

You can use grouping label specifications in Table and Graph Definitions to present the same data in many different forms. For example, both the AGEBYFIV and WORKAGES Label Formats shown above are intended to be applied to the levels of the AGE dimension defined in Chapter 5, **DEFINING DATA**. They provide different ways of grouping data subdivided by AGE for presentation in Host tables and graphs. Chapter 7, **DEFINING TABLES**, contains several example tables illustrating the use of these Label Formats.

Expanding Label Formats

The Expand option on the Add and Edit Label Menus allows you to see how a Label Format containing one or more computed label specifications will be interpreted. This feature generates all dimension level intervals and corresponding labels for each label specification. You can view an expanded Label Format this way, but you can not modify it. Screen 17 shows the expanded version of the AGEBYFIV Label Format defined above.

```
                                                          EXPANDED LABELS
Label Format Name : AGEBYFIV
     Description : 0 to 4 = 0 - 4...)= 65 = 65+

     Lower      Upper     Interval
     Bound      Bound     Label

  1  0          4         0 - 4
  2  5          9         5 - 9
  3  10         14        10 - 14
  4  15         19        15 - 19
  5  20         24        20 - 24
  6  25         29        25 - 29
  7  30         34        30 - 34
  8  35         39        35 - 39
  9  40         44        40 - 44
 10  45         49        45 - 49
 11  50         54        50 - 54
 12  55         59        55 - 59
 13  60         64        60 - 64
 14  65       32767       65 +

                                                                INS
```

Screen 17. An example screen showing an expanded Label Format.

Deleting a Label Format

You can delete a Label Format from a Label Library by using the **Delete** command on the Label Menu. After you select **Delete** from this menu, you are asked to type in or select the name of the Label Format to be deleted. Before Host deletes the Label Format from the Label Library, you must confirm your choice by selecting "Yes" from the resulting menu. Host then marks the Label Format as deleted so that it is not used in any further operations. To remove it physically from the Label Library and recover disk space, you must Pack the Library as discussed in Chapter 12, **MANAGING FILES**.

Listing Label Formats

You can display a list of all Label Formats in the Label Library by choosing **List** from the Label Menu. The list is displayed vertically in the screen Work Space and you may view different portions of it by pressing the [↑] and [↓] keys, or the [PgUp] and [PgDn] keys. Host displays the number of Label Formats in the Label Library above the list. Press [/] or [Esc] to exit from the list and return to the Label Menu. Note that you may not select from this list as with the Query Field function, since you are not performing any operation other than listing. The mode indicator says "LIST LABELS" rather than "SELECT LABELS" as it does when the Query Field function is invoked.

Producing Label Reports

You can produce a complete report of the Label Formats in a Label Library by using the **Report** command on the Label Menu. This report includes the same information and uses the same general layout as the Label Editor screen displays. After selecting

Report from the Label Menu, select a destination for the report from the resulting Report Menu. You can send Label Reports to the printer, to the screen, or to a file. If you select File, any existing Label Report file is renamed using the file name extension "BAK" and the report is written to a file named "LBLRPT.TXT" in the default Data Directory.

CHAPTER 7
DEFINING TABLES

DEFINING TABLES

Host has a table production facility capable of displaying data from one or more Host Data Files in numerous useful formats. Each Host table is generated from a *Table Definition* consisting of a Table Name and Title, a *variable list* naming the variables to be included in the table, and *dimension settings* containing information on how the data are to be arranged in the table. These definitions are stored in the Table Library, located in the default Library Directory. Host's *Table Generator* generates a table by combining a Table Definition from the Table Library with data from a Host Data File. Chapter 10, **GENERATING TABLES**, tells how to invoke the Table Generator to produce a table on the printer, on the screen, or in a disk file. This Chapter explains the settings stored in Table Definitions and describes Host's facilities for creating, editing, listing, reporting, and deleting these Definitions.

The Table Library

As explained in Chapter 3, **GETTING STARTED**, a Table Library consists of two files: one named "TABLE.LIB" that contains Table Definitions, and another named "TABLE.IDX" that contains an index to the Table Definitions. Since these file names are assigned by Host, there can be only one Table Library in any disk directory. If a Table Library does not exist in the Library Directory, one can be created using Host's Manage facility (see Chapter 12, **MANAGING FILES**). Each Table Library can contain up to 65,535 Table Definitions.

Table Definitions

Any user may create, edit, delete, list, or report Table Definitions. Figure 10 shows the relationships among Host commands for manipulating Table Definitions. If you have named a default Host Data File, then choosing Tables from Host's Define Menu displays the Table Menu illustrated in Figure 10. Here, the default Host Data File serves as the *reference Data File* for adding and editing Table Definitions. Host extracts information about the variables you want included in the table from the reference Data File. If there is no default Data File, Host asks you to select a reference Data File before proceeding to the Table Menu. You can select a new reference Data File at any time using the File command on the Table Menu (see Selecting a Reference Data File on page 7-21).

Each Table Definition is assigned a name by the user who creates it. Table Definitions do not contain any data and are, to some extent, independent of specific Host Data Files. When you create a Table Definition, information about each variable in the table is read by Host from the reference Host Data File. This information, which includes the variable name, the number and names of related dimensions, and whether the variable is periodic, is stored as part of the Table Definition. This Definition can then be used with any Host Data File having variables with the same names and structure as those in the reference Host Data File.

Depending on the structure of the variables included in the Table Definition and the way the data are presented, a table consists of one or more *subtables*. Each subtable contains a different cross-section of the data stored in the table's variable or

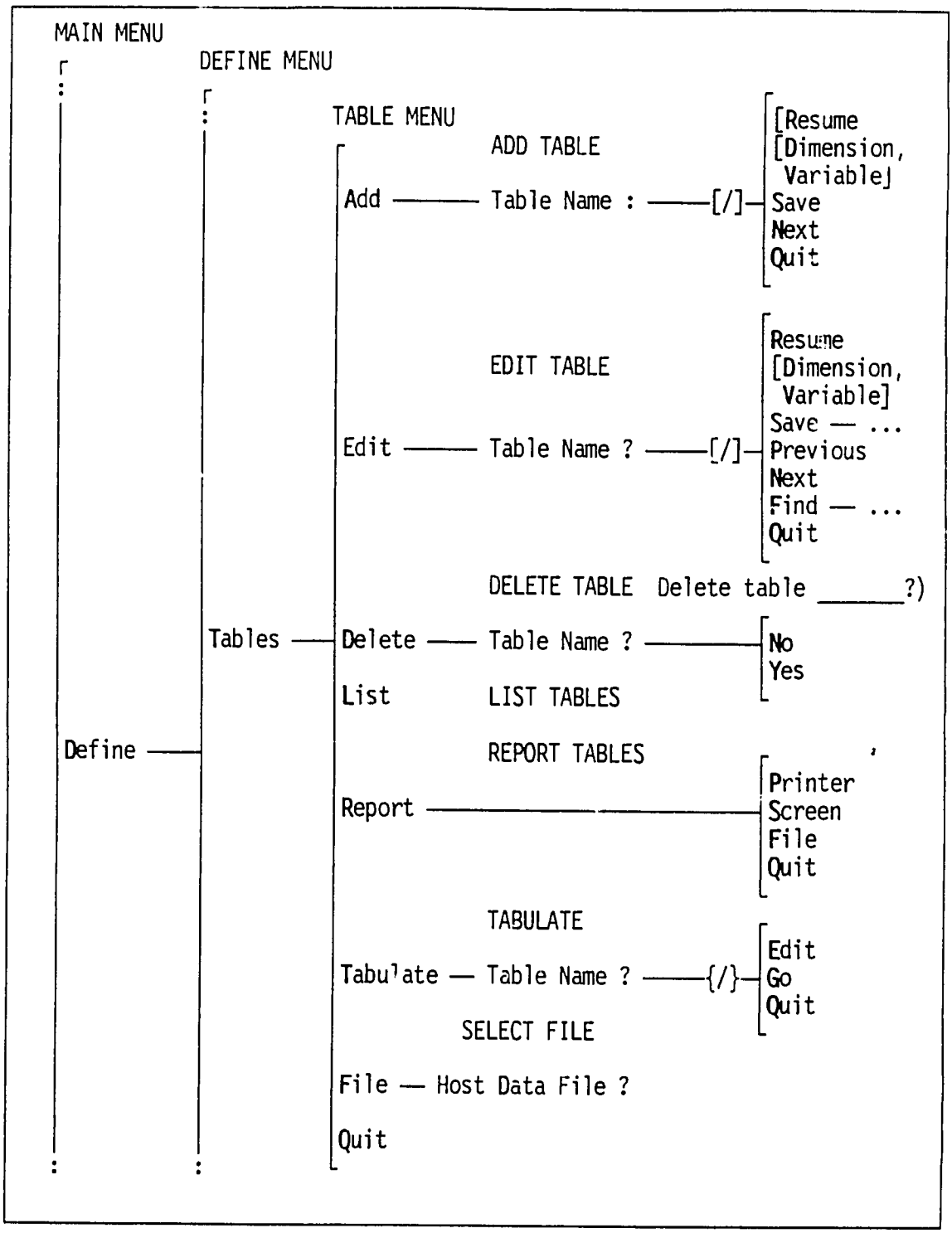


Figure 10. A schematic of Host commands for defining and generating tables.

variables. A single Table Definition can therefore produce a large quantity of tabular output.

Creating and Editing Table Definitions

To create a new Table Definition, or to edit an existing Table Definition, first select Add or Edit from the Table Menu, then enter a table name conforming to the rules for Host Identifiers. After you have entered the name of the Table Definition to create or edit, Host displays a *Table Definition Editor* screen and places the cursor on the first character of the Table Title field. Whether you are editing an existing Table Definition or creating a new one, the Table Definition Editor functions the same way from this point on. All fields in the Table Definition Editor are accessible to both Managers and Users.

The Table Definition Editor always displays one of two screens: a variable specification screen or a dimension specification screen. When creating or editing a Table Definition, you always begin with the variable specification screen.

```
                                                          ADD TABLE
Table Name : TOTALPOP                              Data File : EDDATA1A
Title1 : Total Population_____________________________________________

No.  Variable          No.  Variable          No.  Variable

 1   POP               17                      33
 2                     18                      34
 3                     19                      35
 4                     20                      36
 5                     21                      37
 6                     22                      38
 7                     23                      39
 8                     24                      40
 9                     25                      41
10                     26                      42
11                     27                      43
12                     28                      44
13                     29                      45
14                     30                      46
15                     31                      47
16                     32                      48
```

Screen 18. An example screen showing the creation of a Table Definition.

Screen 18 shows the variable specification screen of a Table Definition named TOTAL-POP. In this example, the cursor is in the Table Title field, the contents of Title1 is "Total Population", and a variable named POP has been included in the variable list.

To move to the dimension specification screen, first display the Add or Edit Table menu by pressing [/] or [Esc], then choose the Dimension entry. To return to the variable specification screen from the dimension specification screen, display the Add or Edit Table menu again and choose the Variable entry.

The Table Title

You may specify a Title to be displayed whenever the table you are defining is produced. A Table Title appears on the first screen or page of output when the table is generated, and each line of the Title is centered over the body of the table. The Title field appears on both the variable specification and dimension specification screens, and the Title may be edited from either screen.

The Table Title may be up to 240 characters in length, but for screen editing it is divided into four 60-character *segments*. Only one of these segments is displayed in the Title field at a time. The first segment of the Title is used as a table description when you request a list of tables from a Query field. You can move among the four Title segments by using the [Tab] key. Pressing [Tab] displays the next Title segment, and pressing [Shift]+[Tab] displays the previous Title segment. The inverse video "↔" displayed left of the Title field is a reminder that [Tab] and [Shift]+[Tab] operate this way.

The four 60-character segments do not determine any division of the Title into separate output lines when the table is produced. For example, a Title entered as follows

```
Title1: This
Title2: is
Title3: a
Title4: Title
```

would appear like this when the table is generated:

```
This is a Title
```

To show where a new line of the Title should begin, you may enter a backslash [\]. For example, a title entered as follows

```
Title1: This is
Title2: a
Title3: Title\Displayed\
Title4: On Three Lines
```

would appear like this when the table is generated:

```
This is a Title
   Displayed
On Three Lines
```

Host also divides the Title wherever necessary to make it fit within the Line Length stored in the Configuration File. The Title may be divided into as many as ten output lines.

The Variable List

The variable specification screen lists the names of variables to be included in the table. Screen 18 on page 7-4 shows the variable specification screen for a Table Definition named TOTALPOP. Screen 19 below shows the variable specification screen for a Table Definition named MANYVAR.

```
                                                          ADD TABLE
Table Name : MANYVAR                                  Data File : AG
Title1 : Table Illustrating the Use of Multiple Variables

No.  Variable          No.  Variable          No.  Variable

 1   AGGCOST           17                      33
 2                     18                      34
 3                     19   COSTIMP            35
 4                     20                      36
 5                     21                      37
 6                     22                      38
 7                     23                      39
 8                     24                      40   POP
 9                     25                      41
10 ? CALCAP__          26                      42
11                     27                      43
12                     28                      44
13                     29   FOODBILL           45
14                     30                      46
15                     31                      47
16                     32                      48
```

Screen 19. An example variable specification screen of a Table Definition.

General Guidelines

At least one variable name must be included in a Table Definition, and up to 48 variable names may be listed. Only variables of type byte, integer or real may appear in tables. When more than one name is included in the variable list or the same name is included more than once, data belonging to these variables appear in separate columns of the table and these columns are placed in the same order as the names in the list. All variables named must have exactly the same dimensions (though these need not be in the same order). Further, the variables in the list must be all periodic or all non-periodic. The variables need not all be of the same type (byte, integer, or real).

Host removes blank entries from the variable list before a Table Definition is stored, but saves non-blank entries in the order they are listed. Figure 11 shows a table named MANYVAR including data from several different variables in the same Host Data File. Note that each column of the table contains data from a different variable and that each row of the table contains data from a different level of the dimension PERIOD. Also, Host has labeled data from the variables AGGCOST and COSTIMP as "(Local $)". This is because the definitions of these two variables specify that they contain data in units of "Local $" (see Chapter 5, **DEFINING DATA**). Screen 19 shows the variable specification screen of the corresponding Table Definition.

Table Illustrating the Use of Multiple Variables

PERIOD/	AGGCOST (Local $)	CALCAP	COSTIMP (Local $)	FOODBILL	POP
1985	5761706	2021	97215	5858921	5696
1986	5945524	2042	85869	6031393	5835
1987	6132690	2064	89727	6222417	5977
1988	6322916	2086	93803	6416719	6122
1989	6514662	2108	98103	6612765	6269
1990	6705057	2130	102623	6807679	6416
1991	6894741	2153	107388	7002129	6564
1992	7085602	2176	112434	7198036	6715
1993	7274516	2200	117763	7392278	6867
1994	7459428	2223	123384	7582812	7019
1995	7636823	2247	129301	7766124	7169
1996	7809533	2272	135566	7945099	7320
1997	7986575	2296	141713	8128288	7470
1998	8191519	2321	146911	8338431	7620
1999	8391092	2347	152247	8543339	7768
2000	8585675	2372	157746	8743415	7915

Figure 11. A table produced using the variable list shown in Screen 19.

Guidelines for Comparison Tables

The same variable name may appear up to three times in the variable list, and these occurrences do not have to be consecutive. If there are second and third occurrences of a variable name in the variable list, the table produced is called a *Comparison Table.* The first occurrence refers to the variable in the reference Host Data File ("V" in the formula below). The second occurrence refers to that variable in a second Host Data File called a Comparison File ("V_c" in the formula below). The third occurrence requests a column of percentage differences calculated by the formula

$$(V_c - V)/V_c * 100$$

Note that the variables to be compared must have the same dimensions, and must both be periodic or non-periodic, but need not be the same type (byte, integer or real). In addition, the attributes of all dimensions of the corresponding variables must match, except for default Label Formats, or the Table Definition can not be applied to the two Host Data Files. If these conditions are not met, the Table Generator displays an error message and returns without generating the table.

Figure 12 on the next page shows a Comparison Table including a column for a percentage difference calculation. Data are shown for the years 1985-1987. The asterisks in rows 2 and 7 of the subtable for 1985 are displayed because the percentage difference calculation is meaningless when the data values to be compared are both 0.

In a Comparison Table, Host labels data from the Comparison Data File with "(Comp)" and percentage differences, if they are requested, with "(% Diff)". Notice that the Title and row labels in this table are in Spanish. Other languages, German or French for example, could easily be substituted by creating a different Label Format for the dimension SUMMROWS and by editing the Title field in this Table Definition.

Estadisticas resumen para cada periodo

PERIOD : 1985

SUMMROWS/	SUMMARY	SUMMARY(Comp)	SUMMARY(% Diff)
Poblacion	5696.000	5696.000	0.000
Tasa crec. pob.	0.000	0.000	*********.***
Costo prod. ag.	5761706.355	5761706.355	0.000
Costo imps.	97215.000	97215.000	0.000
Gasto ag. tot.	5858921.355	5858921.355	0.000
Bal. com. ag.	335184.000	335184.000	0.000
Tasa crec. ingr	0.000	0.000	*********.***
Razon de depend	0.110	0.110	0.000
Calor. per cap.	2020.788	2020.788	0.000

PERIOD : 1986

SUMMROWS/	SUMMARY	SUMMARY(Comp)	SUMMARY(% Diff)
Poblacion	5835.000	5840.000	-0.086
Tasa crec. pob.	0.024	0.025	-3.472
Costo prod. ag.	5945524.056	5951307.042	-0.097
Costo imps.	85868.746	85911.697	-0.050
Gasto ag. tot.	6031392.802	6037218.739	-0.097
Bal. com. ag.	359639.254	359596.303	0.012
Tasa crec. ingr	0.020	0.020	0.000
Razon de depend	0.102	0.102	0.036
Calor. per cap.	2042.154	2042.154	0.000

PERIOD : 1987

SUMMROWS/	SUMMARY	SUMMARY(Comp)	SUMMARY(% Diff)
Poblacion	5977.000	5988.000	-0.184
Tasa crec. pob.	0.024	0.025	-3.972
Costo prod. ag.	6132690.309	6145577.616	-0.210
Costo imps.	89727.075	89822.872	-0.107
Gasto ag. tot.	6222417.385	6235400.488	-0.208
Bal. com. ag.	369306.875	369211.078	0.026
Tasa crec. ingr	0.020	0.020	0.000
Razon de depend	0.103	0.103	0.077
Calor. per cap.	2063.797	2063.797	0.000

Figure 12. An example Comparison Table with a percentage difference column.

Using the Variable Specification Screen

Each variable name field on the variable specification screen is a Query Field, so you can either type in variable names or select them from a list. If you press [?] to display a list of available variable names, all variables in the reference Host Data File are listed. As noted before, however, the table may use data from any Host Data File having the same variables and structure as the reference Host Data File.

There are several cursor movement and editing commands available to make specification of variables easier. Table 17 on the next page contains a list of commands recognized by the Table Definition Editor. To delete a variable name from the list, place the cursor on that name and press [Alt]+[D] (^Y). To insert a variable name in the list above the current variable name, press [Alt]+[I] (^N). You may use the cursor movement keys, or their WordStar equivalents, to move among the variable name fields on the screen. To move into the Title field, first place the cursor at the top of the first column of the variable list by pressing [Ctrl]+[PgUp], then press the [↑] key. To return to the variable list press [Enter] or the [↓] key.

The Table Dimensions

When a Table Definition is created, its dimensions are taken from the first variable listed that is found in the reference Host Data File. Table dimension settings can be edited using the dimension specification screen. Screen 20 on page 7-11 is a sample screen showing dimension status settings for the Table Definition named TOTALPOP. These are default settings assigned by Host. In this example, the cursor is on the Use field for the dimension PERIOD. The dashes displayed in several fields show that they are meaningless under the current Use settings. Fields displayed in bright characters may be modified, but fields displayed in dim characters can not be changed under the current Use settings.

The meaning of each field or setting on the dimension specification screen is explained briefly below.

Use	determines the way a dimension is used in a table.
Label Format	determines how dimension levels are identified, as well as what groupings of data values are used for totals and summations.
Combine Status	determines whether dimension levels are shown individually in addition to group subtotals, or only in label groups.
Total Status	requests summation across all levels of a dimension.
Percent Status	requests the calculation of Row, Column, or Nested percentages for inclusion in a table.

Use

The Dimension Use field is a toggle field, so you can scan through available choices using [+], [Space], or [-]. With certain exceptions, Use may be any of the following:

TABLE 17.

Table Definition cursor movement and editing commands.

IBM AT Command Key	Function in Host	WordStar Command*
[Esc]	Same as "Quit" on menus. Same as [/] in fields, but restores original field contents before displaying menu. (Similar, not identical, to WordStar ^U.)	^U
[Home]	Beginning of field.	^QS
[End]	End of field.	^QD
[Ctrl]+[Home]	Top of current column.	^QE
[Ctrl]+[End]	Bottom of current column.	^QX
[↑]	Up one field.	^E
[↓]	Down one field.	^X
[←]	Left one field.	^S
[→]	Right one field.	^D
[Ctrl]+[←]	First field in row (Dimension Screen).	^A
[Ctrl]+[→]	Last field in row (Dimension Screen).	^F
[Ins]	Insert/overtype toggle.	^V
[Del]	Delete character under cursor.	^G
[←]	Delete character to left of cursor.	[←]
[Ctrl]+[PgUp]	Top of first column.	^QR
[Ctrl]+[PgDn]	Bottom of last column.	^QC
[Tab Left]	Previous Title section.	None
[Tab Right]	Next Title section.	None
[Alt]+[D]	Delete current variable.	^Y
[Alt]+[E]	Erase field.	^QY
[Alt]+[I]	Insert a variable above current variable.	^N
[Alt]+[S]	Search for name or partial name (List Facility).	^QF
[Alt]+[U]	Restore original entry ("undo").	None

* WordStar commands are represented using normal WordStar notation. In WordStar notation, holding down the [Ctrl] key while pressing [U] is represented by ^U.

Row. Only one dimension may be selected as a *Row* dimension. If no Row dimension is chosen, the resulting table consists of a single row of values. If you choose a dimension as the Row dimension, a separate row of the table is created for each level the dimension. If a Label Format is assigned to the Row dimension, the contents of the

```
                                                          ADD TABLE
Table Name : TOTALPOP                                Data File : EDDATA1A
Title1 : Total Population
------------------------------------------------------------------------
Dimension   Use            Label Format   Combine      Total   Percent

SEX         Sum over       -------        ---          ---     ---
REGION      Sum over       -------        ---          ---     ---
AGE         Sum over       -------        ---          ---     ---
PERIOD      Unnested       NONE           ---          ---     ---
```

Screen 20. An example dimension specification screen of a Table Definition.

Label Format may affect the way the rows of the table are constructed. The number and nature of the values depend on whether a Column dimension, a Nested dimension, or both are included in the Table Definition.

Figure 13 shows a table with no Row dimension. Figures 14-20 on the following pages illustrate tables with Row dimension AGE. In Figure 14, the first page of the table POPBYAGE is shown. There is no Label Format assigned to the Row dimension AGE. Figures 15-20 show the use of different Label Formats for this dimension. The effect of the Label Format is explained in detail in a following section.

```
                    Total Population

PERIOD :  1980

    POP
(/10^6)
-------
  4.400
```

Figure 13. A table produced using default dimension Use settings.

Population by AGE
No Label Format

PERIOD : 1980

AGE/	POP (/10^6)
0	0.159
1	0.155
2	0.152
3	0.149
4	0.145
5	0.142
6	0.139
7	0.136
8	0.132
9	0.129
10	0.125
11	0.122
12	0.117
13	0.112
14	0.105
15	0.099
16	0.093
17	0.087
18	0.083
19	0.079
20	0.076
21	0.072
22	0.069
23	0.067
24	0.065
25	0.063
26	0.061
27	0.059
28	0.057
29	0.056
30	0.054
31	0.053
32	0.051
33	0.049
34	0.048
35	0.046
36	0.044
37	0.043
38	0.041
39	0.040
40	0.039
41	0.038
42	0.037
43	0.035

Figure 14. A table showing AGE used as a Row dimension.

Column. Only one dimension may be selected as a *Column* dimension. If the Table Definition has more than one variable or multiple occurrences of the same variable, no Column dimension may be specified. When no Column dimension is selected, the table has a separate column for each variable included in the Table Definition. Figure 11 on

Population by AGE
Five-Year Age Groups

PERIOD : 1980

AGE/	POP (/10^6)	AGE %
0 - 4	0.760	17.27
5 - 9	0.677	15.39
10 - 14	0.581	13.21
15 - 19	0.440	10.01
20 - 24	0.348	7.90
25 - 29	0.296	6.72
30 - 34	0.255	5.80
35 - 39	0.215	4.88
40 - 44	0.183	4.15
45 - 49	0.153	3.48
50 - 54	0.128	2.90
55 - 59	0.105	2.39
60 - 64	0.086	1.96
65 +	0.173	3.94
Total	4.400	100.00

Figure 15. A table using the Label Format AGEBYFIV for the Row dimension AGE.

page 7-7 and Figure 12 on page 7-8 are tables having multiple variables as the columns.

When a Column dimension is chosen but no Label Format is assigned to it, a separate column of the table is created for each level of the Column dimension. If a Label Format is assigned to the selected Column dimension, the contents of the Label Format may affect the way the columns of the table are constructed. Figures 18 and 19 on pages 7-18 and 7-19 illustrate tables with PERIOD as the Column dimension. No Label Format has been assigned to the Column dimension of this table.

Nested. The term *Nested* dimension refers to a third dimension that may be displayed within one subtable of a table. Only one dimension may be selected as a Nested dimension. When a Nested dimension is selected, all levels of the dimension are displayed within each level of the Column dimension. If no Column dimension is chosen, then all levels of the Nested dimension are displayed within each variable column. If a Label Format is assigned to the selected Nested dimension, the contents of the Label Format may affect the way the nested sub-columns of the table are constructed. Figure 19 on page 7-19 illustrates a table with SEX as the Nested dimension.

Unnested. There is no restriction on the number of Unnested dimensions that may be included in a Table Definition. As an extreme case all table dimensions could be Unnested dimensions. A separate subtable of the table is produced for each level of an Unnested dimension. If there is more than one Unnested dimension, a separate subtable

is produced for each combination of levels of these dimensions. Thus in the extreme case above, a Table Definition with one variable name in the variable list would lead to a separate subtable for each data value in the variable.

If a Label Format is assigned to an Unnested dimension, the contents of the Label Format may affect the way the table subtables are constructed. Unnested is the default Use for the reserved dimension PERIOD. Thus by default a separate subtable is generated for each level of PERIOD. Figure 20 on page 7-20 shows the first page of a table with both PERIOD and SEX as Unnested dimensions. On the page shown, the levels of PERIOD run from "1980" to "1981" and the level of SEX varies from "Male", to "Female", to "Total", and back to "Male". As always, the levels of the first Unnested dimension vary most rapidly when a table is generated, the levels of the second Unnested dimension vary next most rapidly, and so on.

Sum-over. There is no restriction on the number of Sum-over dimensions that may be included in a Table Definition. This is the default Use setting for all dimensions except PERIOD, which can never be a Sum-over dimension. When Sum-over Use is selected, the values in the table are produced by adding across all levels of the Sum-over dimension. Since PERIOD is the time dimension, it is rarely appropriate to sum over its levels. When all dimensions of a Table Definition except PERIOD are assigned the Sum-over Use, the values in the resulting table are variable totals for each level of the PERIOD dimension. Since Label Format and other dimension settings have no meaning for Sum-over dimensions, these fields are filled with dashes when the Use setting is Sum-over. The table in Figure 13 is an example of a table with all dimensions except PERIOD having the Use Sum-over. The table in Figure 14 shows a table with the dimension AGE as the Row dimension and dimensions REGION and SEX as Sum-over dimensions.

If you assign Row, Column or Nested Use to a dimension when that Use is already assigned to some other dimension, the other dimension is reassigned its default Use. By default, all dimensions except PERIOD (if present) have Use Sum-over, and the dimension PERIOD has Use Unnested. This produces totals for each variable in the table, with a separate subtable for each level of PERIOD present in the Host Data File. Figure 13 on page 7-11 shows a table named TOTALPOP that was generated using the default settings shown in screen 20, also on page 7-11. This is the simplest form of table Host can produce. Notice in this example that Host has labeled data from the variable POP as "(\10^6)". As in Host's Data Editor, this means that the data shown have been divided by ten to the sixth power, i.e., are expressed in millions. The exponent "6" is the Scaling Exponent in the definition of the variable POP (see Chapter 5, **DEFINING DATA**).

Label Format and Combine Status

The Label Format field names the Label Format to be applied to each dimension in the Table Definition. Since this setting is not appropriate for Sum-over dimensions, dashes are displayed in the Label Format column of these dimensions. The default value for this field is the default Label Format for that dimension in the Host Data File. The Label Format field is a Query Field, and any Label Format in the Label Library may be assigned to any dimension. To remove a Label Format from a dimension, clear the Label Format field by pressing [Alt]+[E] (^QY) or enter "NONE".

The Combine Status setting affects the way the levels of a dimension are displayed when a Label Format with Grouping or Computed Label Specifications has been assigned (see the discussion of Grouping and Computed Label Specifications on page 6-5). The two possible settings for this field are "Yes" and "No". If a Label Format is assigned to a dimension and the Combine Status is "Yes", then all Effective Levels

of the dimension corresponding to a single label are summed to produce table entries (see page 5-4, **Adding and Editing Dimensions**, for an explanation of Effective Levels). When the entry for Combine is "No", the table generated has an entry for each level of the dimension, and also subtotal entries corresponding to the distinct Label Format categories.

Figure 15 on page 7-13, Figure 16 below, and Figure 17 on page 7-16 show the effect of the Label Format and Combine Status settings. The Table Definition used to generate the table in Figure 15 is produced by changing the Label Format setting for the AGE dimension in the Table Definition named POPBYAGE to AGEBYFIV, the Percent setting to "Yes", and saving the result under the name POPBYAG1. Changing the same Label Format from AGEBYFIV to WORKAGES and saving the result under the name POPBYAG2, creates the Table Definition used to generate the table in Figure 16. The table in Figure 17 is produced by changing the Combine Status for the dimension AGE in the Table Definition POPBYAG2 to "No", making appropriate changes in the Title, and saving the result as POPBYAG3. Only the first page of the resulting table is shown. Definitions of the example Label Formats AGEBYFIV and WORKAGES are shown in Chapter 6, **DEFINING LABELS**.

```
                    Population by AGE
                 Grouped by Work Status

PERIOD :  1980

AGE/                 POP
                 (/10^6)
                 -------
Below Work Age     2.018
   Working Age     2.209
Above Work Age     0.173
         Total     4.400
```

Figure 16. A table using the Label Format WORKAGES for the Row dimension AGE.

For the dimension PERIOD and for Sum-over dimensions, the Combine setting has no meaning. Note that if there is a Nested dimension, neither it nor the Column dimension may have the Combine Status set to "No".

Total Status

The Total field requests the calculation of dimension totals for any non-Sum-over dimension except PERIOD (since PERIOD is the time dimension, it seldom makes sense to sum over its levels). The possible settings for this field are "Yes" and "No".

When a total is requested for a dimension, then a total computed by summing over all levels of the dimension is displayed following the other levels of the dimension.

Population by AGE
Grouped by Work Status
Illustrating the Combine=No Option

PERIOD : 1980

AGE/	POP (/10^6)
Below Work Age	
0	0.159
1	0.155
2	0.152
3	0.149
4	0.145
5	0.142
6	0.139
7	0.136
8	0.132
9	0.129
10	0.125
11	0.122
12	0.117
13	0.112
14	0.105
SubTotal	2.018
Working Age	
15	0.099
16	0.093
17	0.087
18	0.083
19	0.079
20	0.076
21	0.072
22	0.069
23	0.067
24	0.065
25	0.063
26	0.061
27	0.059
28	0.057
29	0.056
30	0.054
31	0.053
32	0.051
33	0.049
34	0.048
35	0.046
36	0.044
37	0.043
38	0.041

Figure 17. A table using Combine set to "No" for the Row dimension AGE.

If the dimension is the Row dimension, then the total is the last row of the table; if the dimension is the Column dimension, then the total is the last column of the table; If the dimension is the Nested dimension, then the total follows the last level of this dimension within every column of the table. If the dimension is an Unnested dimension, then a total subtable is produced following the subtables corresponding to

all other levels of the dimension. The tables in Figures 15, 16, 18, 19 and 20 show Row totals; the table in Figure 19 shows Nested totals; the table in Figure 20 shows an Unnested total.

Percent Status

The Percent field requests the calculation of percentages for Row, Column and Nested dimensions. Since Totals are never computed for the PERIOD dimension, however, percentages may not be requested for this dimension. The two possible settings for this field are "Yes" and "No".

If a percentage is requested for a dimension, the total for that dimension is also included in the table. When the Percent setting is "Yes" for a dimension, the corresponding Total setting is automatically set to "Yes" and can not be changed. The percentage computed is the ratio of the value corresponding to the current level (or sum of levels) to the dimension total multiplied by 100. Row percentages are displayed to the right of data values; Column and Nested percentages are displayed below the data values. Figure 15 shows a table generated from a Table Definition requesting that percentages be calculated for the Row dimension AGE.

Figures 18-20 on the following pages illustrate further the Use concepts discussed above.

The Next, Previous, and Find Commands

The Next command on the Add Table Menu allows you to create a new Table Definition after you have completed the current one. The Next and Previous commands on the Edit Table Menu provide a way of editing Table Definitions in sequence. Table Definitions are stored alphabetically by name in a Table Library. Next allows you to edit the Table Definition following the current one in a Table Library, and Previous allows you to edit the Table Definition before the current one in a Table Library.

The Find command on the Edit Table menu allows you to locate any Table Definition in a Table Library with ease. After selecting Find from the menu, you are asked for the name of the desired Table Definition. You can type in a name or select one from a list using Host's Query Field feature. If a Table Definition with that name exists in the Table Library, Host reads it into the Table Definition Editor. If the Table Definition named is not found in the Table Library, Host tells you and asks you to enter a different name.

Saving Table Definitions

There are Save commands on both the Add Table and Edit Table Menus. When you choose Save from the Add Table Menu, Host stores your new Table Definition in the Table Library under the name you assign to it. When you choose Save from the Edit Table Menu, however, Host allows you to save any changes you have made to the Table Definition under its original name, or under a different name. This provides a quick, convenient way of creating a several similar Table Definitions. You can Edit a Table Definition like the Table Definition you want to create, change some of its settings, then Save the changes under a new name. Often this technique is more efficient than creating a new Table Definition using the Add command on the Table Menu.

The Save commands are also available in several other Add Table and Edit Table Menu commands. If you choose Next or Quit from the Add Table Menu, or choose Next, Previous, Find or Quit from the Edit Table Menu before you save changes to a Table

100

Population by AGE
Grouped by Work Status
Illustrating the Use of PERIOD as the Column Dimension

POP (/10^6): Population (Millions)

SEX : Male

AGE/PERIOD	1980	1981	1982	1983	1984
Below Work Age	0.983	0.994	1.002	1.009	1.015
Working Age	1.057	1.076	1.098	1.122	1.146
Above Work Age	0.088	0.090	0.091	0.092	0.094
Total	2.128	2.160	2.192	2.223	2.255

AGE/PERIOD	1985	1986	1987	1988	1989
Below Work Age	1.022	1.028	1.034	1.040	1.045
Working Age	1.172	1.198	1.224	1.250	1.276
Above Work Age	0.095	0.096	0.097	0.099	0.100
Total	2.289	2.322	2.356	2.388	2.421

AGE/PERIOD	1990
Below Work Age	1.051
Working Age	1.302
Above Work Age	0.101
Total	2.454

SEX : Female

AGE/PERIOD	1980	1981	1982	1983	1984
Below Work Age	1.036	1.043	1.050	1.055	1.060
Working Age	1.152	1.173	1.195	1.218	1.242
Above Work Age	0.085	0.086	0.088	0.089	0.090
Total	2.272	2.302	2.332	2.362	2.392

AGE/PERIOD	1985	1986	1987	1988	1989
Below Work Age	1.065	1.071	1.076	1.081	1.087
Working Age	1.267	1.292	1.318	1.342	1.368
Above Work Age	0.091	0.092	0.093	0.094	0.096
Total	2.424	2.456	2.487	2.518	2.550

Figure 18. A table showing PERIOD used as a Column dimension.

Population by AGE, PERIOD and SEX
Working Age Groups

POP (/10^6): Population (Millions)

AGE/PERIOD BY SEX	1980 Male	1980 Female	1980 Total
Below Work Age	0.983	1.036	2.018
Working Age	1.057	1.152	2.209
Above Work Age	0.088	0.085	0.173
Total	2.128	2.272	4.400

AGE/PERIOD BY SEX	1981 Male	1981 Female	1981 Total
Below Work Age	0.994	1.043	2.037
Working Age	1.076	1.173	2.249
Above Work Age	0.090	0.086	0.176
Total	2.160	2.302	4.462

AGE/PERIOD BY SEX	1982 Male	1982 Female	1982 Total
Below Work Age	1.002	1.050	2.052
Working Age	1.098	1.195	2.293
Above Work Age	0.091	0.088	0.179
Total	2.192	2.332	4.524

AGE/PERIOD BY SEX	1983 Male	1983 Female	1983 Total
Below Work Age	1.009	1.055	2.063
Working Age	1.122	1.218	2.340
Above Work Age	0.092	0.089	0.181
Total	2.223	2.362	4.585

AGE/PERIOD BY SEX	1984 Male	1984 Female	1984 Total
Below Work Age	1.015	1.060	2.075
Working Age	1.146	1.242	2.388
Above Work Age	0.094	0.090	0.184
Total	2.255	2.392	4.647

Figure 19. A table showing SEX used as a Nested dimension.

```
                    Population by AGE
          PERIOD and SEX Unnested By Dimensions

SEX : Male     PERIOD :  1980

AGE/               POP
                (/10^6)
                -------
Below Work Age   0.983
   Working Age   1.057
Above Work Age   0.088
         Total   2.128


SEX : Female   PERIOD :  1980

AGE/               POP
                (/10^6)
                -------
Below Work Age   1.036
   Working Age   1.152
Above Work Age   0.085
         Total   2.272


SEX : Total    PERIOD :  1980

AGE/               POP
                (/10^6)
                -------
Below Work Age   2.018
   Working Age   2.209
Above Work Age   0.173
         Total   4.400


SEX : Male     PERIOD :  1981

AGE/               POP
                (/10^6)
                -------
Below Work Age   0.994
   Working Age   1.076
Above Work Age   0.090
         Total   2.160
```

Figure 20. A table showing the use of more than one Unnested dimension.

Definition, Host gives you an opportunity to save those changes before proceeding with the selected command.

Deleting Table Definitions

You can delete a Table Definition from a Table Library by choosing the Delete command from the Table Menu. After choosing Delete from the Table Menu, you may type in the name of the Table Definition to delete or select it using Host's Query Field facility. Before a Table Definition is removed from the Table Library, you must confirm the deletion by selecting "Yes" from the resulting menu. Host then marks the Table Definition so that it is not used in any further operations. To remove it physically from the Table Library and recover disk space, you must Pack the Library as explained in Chapter 12, MANAGING FILES.

Listing Table Definitions

You can display a complete list of Table Definitions in the current Table Library by choosing List from the Table Menu. This list is displayed vertically in the screen Work Space and you may view different portions of it by pressing the keys [↑], [↓], [PgUp], or [PgDn]. The number of Table Definitions in the Table Library is displayed above the list. Press [/] or [Esc] to exit from the list and return to the Table Menu. Note that you may not select from this list as with the Query Field function, since you are not performing any operation other than listing. To help you remember this, the mode indicator says "LIST TABLES" rather than "SELECT TABLE" as it does when the Query Field function is invoked.

Producing Table Definition Reports

You can produce a complete report of all Table Definitions in a Table Library using the Report command on the Table Menu. This report includes the same information and uses the same general layout as Table Editor screen displays. After selecting Report from the Table Menu, select a destination for the report from the resulting Report Menu. You can send Table Definition reports to the Printer, the Screen, or to a file. If you select File, any existing table report file is renamed using the file name extension "BAK" and the report is written to the file "TBLRPT.TXT" in the default Data Directory.

Generating Tables

Since it is often useful to generate tables immediately after creating or modifying Table Definitions, a Tabulate command is included in the Table Menu. This command is identical to the Tabulate command on Host's Main Menu and is completely explained in Chapter 10, GENERATING TABLES.

Selecting a Reference Data File

As stated earlier in this chapter, Host extracts information about the variables included in a Table Definition from the reference Data File. Normally Host uses the default Data File as the reference Data File for adding and editing Table Definitions. You can change this at any time by choosing the File selection on Host's Table Menu.

Host then asks you for the name of the Host Data File you want to use as the reference Data File for Table Definitions. If you press [Esc] here, Host returns you to the Table Menu with the reference Data File unchanged. To proceed, you can use Host's Query Field Feature to select any Data File in the default Data Directory, or you can enter the name of any Host Data File in your system. If you do not supply a directory path, Host assumes the Data File is in the default Data Directory. The new reference Data File remains in effect until you leave the Table Menu or until you change it by using the File command again.

CHAPTER 8
EDITING DATA

EDITING DATA

Host's Data Editor can be used to enter new data, edit existing data, and view data generated by Application Modules. Figure 21 shows the commands available in the Data Editor.

```
MAIN MENU
 :
 |                                                  Resume
 |                                                  Area
 |          EDIT DATA                               Period
 |                                                  Interpolate
 |Edit —— Data File Name ? —— Variable Name ?——[/]—Save ...
 |                                                  Dimension
 |                                                  Variable ...
 |                                                  File ...
 :                                                  Quit
```

Figure 21. The command structure of Host's Data Editor.

Selecting a Data File

After you choose Edit from Host's Main Menu, Host asks for the name of the Data File containing the variables you want to edit. You may select any Data File in the default Data Directory from a horizontal "Lotus-style" menu. If there is a default Data File, its name is the first selection on the menu and can be chosen by pressing [Enter]. If instead of entering a Data File name you press [/] or [Esc], Host gives you an opportunity to Resume selecting a Data File or Quit and return to the Main Menu.

Selecting a Variable

After you select a Data File, Host asks you the name of the variable you want to edit. You may type in a variable name or select from a list of variables in the Data File by using Host's Query Field facility. If instead of entering a variable name you press [/] or [Esc], Host gives you an opportunity to Resume selecting a variable, choose a different File, or Quit and return to the Main Menu.

Once you have selected a Data File and variable, Host reads the variable's definition from the Data File. Host also reads any required Label Formats from the current Label Library. If your microcomputer does not have enough memory to edit the variable, Host tells you at this point and gives you an opportunity to select a dif-

ferent variable. Otherwise, Host reads the variable's data from the Data File and displays as much of the data as possible in the Work Space. For periodic variables, the first period of data displayed always corresponds to PERIOD level 1.

If Data is not Found in the Data File

If the Data Editor does not find data for a variable in the Data File, it provides appropriate default values and displays the following message in the Message Space:

Data not found in file; default values have been supplied.

You should always receive this message when entering data for a new variable created using Host's Define, Variable, Add commands. If the variable is periodic, this message applies only to the current period of data. As you view different periods of a periodic variable, the presence or absence of this message tells you whether a period of data is really in the Data File or is composed of default values provided by the Data Editor.

For numeric variables, the default value is zero. For character and character string variables, the default value is the ASCII NUL character (^@ or decimal 0) which is displayed on the screen as a blank space. When you first change one of these default values for a variable, the message shown above disappears from the top of the display screen. This indicates that Host has sensed the change.

Saving Editor-supplied Default Values

For nonperiodic variables, default values provided by the Data Editor are always saved to the Data File when you select Save from the Edit Menu. However, unless you specifically ask it to, Host's Data Editor does not save a period of data for a periodic variable when it consists entirely of default values. Thus, Host attempts to minimize the amount of disk space occupied by unnecessary zero or NUL data values. If you want to save an entire period of Editor-supplied default values, that period must be the one displayed when you issue the Save command. To save several such periods of data, issue the Save command after displaying each period of data on the screen.

The Data Editor Display

The Menu Space

When you are not selecting or executing a function from the Edit Menu, the name and description of the current variable are displayed in the Menu Space. For variables of Type "real" (floating-point numbers) with data stored as scaled rather than raw values, the variable name is followed by scaling notation. For example:

```
Variable Name : POP (/10^6)
  Description : Population
```

shows that the current variable is named "POP" and that its data should be entered in millions (population divided by ten to the sixth power). Using Host's Define, Variable, Edit commands to view the Variable Definition of POP would show that its

Scaling Exponent is six. A Unit of Measurement may also be included in this notation. For example:

Variable Name : GNP (Dollars/10^9)
  Description : Gross National Product

shows that the current variable is "GNP" (Gross National Product) and that its data should be in billions of Dollars. If a variable is not scaled but has a Unit of Measurement, this notation includes only the Unit of Measurement. For example:

Variable Name : XCHANGE (US$->Baht)
  Description : Currency exchange rate

shows that the current variable is "XCHANGE" and that it contains a currency exchange rate from US$ to Baht (the unit of currency used in Thailand).

The Work Space

Host displays the data of the current variable in the Work Space. Each data entry field in the Data Editor is the width of the current variable's Edit Mask (see page 5-14, *Edit Mask*). For variables of Type "real", Host displays as many significant digits as possible, resorting to scientific notation only when necessary. If a value can not be formatted to fit within the width of the Edit Mask, Host displays asterisks ("*") in the data entry field.

Since Host variables can have from zero to seventeen dimensions (sixteen explicit dimensions and an implied seventeenth dimension for periodic variables), the Data Editor has a variety of screen display arrangements. Host displays data for variables with no dimensions as a single value in the upper left corner of the Work Space. Screen 21 shows a Data Editor display of a periodic variable named "CALCAP" which has no explicit dimensions. The value shown is the per capita calorie consumption for 1985.

If a variable has one or more dimensions, the top row of the Work Space displays information about these dimensions. This area of the Data Editor display is called the *Dimension Line.* Screen 22 shows the display of a nonperiodic variable named "CALUNIT" which has one explicit dimension named "PRODUCTS". The dimension PRODUCTS has a default Label Format that Host uses to annotate the rows of data. The data in this variable are the calorie values per unit of agricultural production for several crops. Note that the data entry cursor is on the entry for "Plantano" and that "Tabaco", "Cafe", and "Cacoa" have no calorie values.

Since a screen display is two-dimensional, it can display only two dimensions of data. Data entries displayed on different rows of the Work Space belong to different levels of the *row dimension* and data entries displayed in different columns belong to different levels of the *column dimension.* The names of these two dimensions are displayed on the left side of the Dimension Line in the following format:

<row dimension>/<column dimension>

Screen 23 shows a nonperiodic variable named "LTABLE" (a Life Table), that has two explicit dimensions, "ROWSLT" and "COLSLT". In this example, only six columns of the data can be displayed in the Work Space window, which has been "scrolled" to the

```
                                                                  EDIT
Variable Name : CALCAP                                  Data File : AG
  Description : Consumo per capita de calorias
---------------------------------------------------------------------
 PERIOD : 1985
---------------------------------------------------------------------
2020,2084

F3-Previous Period  F4-Next Period  F5-First Period  F6-Last Period
```

Screen 21. A Data Editor display of a periodic variable with no dimensions.

```
                                                                  EDIT
Variable Name : CALUMIT                                 Data File : AG
  Description : Calorias por unidad de prod. agric.
---------------------------------------------------------------------
PRODUCTS
---------------------------------------------------------------------
Arroz        165000.00
Habichuelas  150000.00
Maiz         160000.00
Yuca         50000.000
Mamey        50000.000
Batatas      50000.000
Platano      30000.000
Tabaco       0.0000000
Azucar       160000.00
Cafe         0.0000000
Cacao        0.0000000
Gan. vacuno  110000.00
Gan. porcino 105000.00
Pollos       80000.000
Leche        30000.000
Trigo        165000.00
Otros        150000.00
---------------------------------------------------------------------
                                                          NUM SCROLL
```

Screen 22. A Data Editor display of a one-dimensional nonperiodic variable.

right by pressing [→] to display columns "4" through "9". Note that the row dimension ROWSLT has a default Label Format while COLSLT, the column dimension, does not.

EDIT

Variable Name : LTABLE
Description : Life table
Data File : EDDATA1A

ROWSLT/COLSLT

	4	5	6	7	8	9
Male Life Ex.	58.000000	60.000000	62.000000	64.000000	66.000000	68.000000
0	0.0842000	0.0776700	0.0712300	0.0648000	0.0586200	0.0524600
1 - 4	0.0569900	0.0494100	0.0423900	0.0359300	0.0300200	0.0246700
5 - 9	0.0152300	0.0131100	0.0111600	0.0094000	0.0078000	0.0063000
10 - 14	0.0079400	0.0068500	0.0058500	0.0049500	0.0041200	0.0033000
15 - 19	0.0107100	0.0091900	0.0078000	0.0065400	0.0054100	0.0044000
20 - 24	0.0150900	0.0129300	0.0109600	0.0091800	0.0075900	0.0061600
25 - 29	0.0184000	0.0159000	0.0136000	0.0115000	0.0096700	0.0079500
30 - 34	0.0223300	0.0194000	0.0168400	0.0143900	0.0121500	0.0101000
35 - 39	0.0272700	0.0241500	0.0212200	0.0184600	0.0158700	0.0134700
40 - 44	0.0319000	0.0280500	0.0250300	0.0229400	0.0201000	0.0175400
45 - 49	0.0390500	0.0364600	0.0331600	0.0299400	0.0268000	0.0237600
50 - 54	0.0521700	0.0482300	0.0443400	0.0405000	0.0367200	0.0330000
55 - 59	0.0744200	0.0693000	0.0643700	0.0593600	0.0543000	0.0494100
60 - 64	0.1090900	0.1026000	0.0960700	0.0894000	0.0828300	0.0761300
65 +	0.3301300	0.3313400	0.3144300	0.3170000	0.3095600	0.3017300

END SCROLL

Screen 23. A Data Editor display of a two-dimensional nonperiodic variable.

For variables with more than two dimensions, Host chooses the dimension with the largest Current Number of Levels to be the row dimension and the dimension with the next largest Current Number of Levels to be the column dimension. This ensures that the largest possible two-dimensional slice of data is displayed in the Work Space. The names and levels of other dimensions are listed after the row/column label in the following format:

⟨dimension name⟩ : ⟨level⟩

For periodic variables, the time dimension PERIOD is always the first dimension on the Dimension Line after the row/column label; Host will not select PERIOD to be the row or column dimension. If there are more dimensions than can be displayed on the Dimension Line, Host displays as many as possible and places an arrowhead symbol at the end of the Dimension Line to show that there are more in that direction.

Screen 24 shows a Data Editor display for a periodic variable named "POP" (Population in ten thousands) that has dimensions named "AGE", "SEX", and "REGION". Host has chosen AGE to be the row dimension and SEX to be the column dimension; both the time dimension PERIOD and the dimension REGION have been placed on the Dimension Line. Note that the dimensions SEX and REGION have default Label Formats so that the levels of SEX are labeled "Male" and "Female", and the level of REGION is labeled "Urban". Note also that the dimension AGE apparently has a Starting Level of zero and no default Label Format.

```
                                                                    EDIT
 Variable Name : POP (/10^4)                          Data File : EDDATA1A
   Description : Population
 ------------------------------------------------------------------------
 AGE/SEX  PERIOD : 1988  REGION : Urban
 ------------------------------------------------------------------------
          Male     Female
 0  170.00000 195.00000
 1  186.00000 197.00000
 2  192.60000 199.00000
 3  197.00000 199.00000
 4  199.00000 199.00000
 5  198.00000 198.00000
 6  195.00000 191.00000
 7  188.00000 184.00000
 8  177.00000 174.00000
 9  164.00000 162.00000
 10 150.00000 149.00000
 11 135.00000 136.00000
 12 121.00000 125.00000
 13 111.00000 116.00000
 14 103.00000 110.00000
 15 96.000000 104.00000
 F3-Previous Period  F4-Next Period  F5-First Period  F6-Last Period
                                                          NUM
```

Screen 24. A Data Editor display of a three-dimensional periodic variable.

Using Label Formats

If a dimension does not have a default Label Format, the Data Editor labels its levels with integer labels. Otherwise, the Data Editor replaces these integer labels according to any explicit label specifications in the default Label Format. In an explicit label specification, the lower bound and upper bound of the label interval are equal. For example, if the default Label Format for a dimension named "SEX" is as follows:

	Lower Bound	Upper Bound	Size of Interval (Inclusive)	or	Interval Label
1	1	1			Male
2	2	2			Female

then when the level of SEX is one, Host displays the label "Male" and when the level of SEX is two, Host displays the label "Female". Since it can not combine data from several dimension levels, Host's Data Editor does not use grouping or computed label specifications. Any dimension level that does not have a corresponding explicit label in the dimension's default Label Format is labeled with an integer label.

Although the row labels in Screen 23 may appear to be produced by grouping or computed label specifications, they have been produced using explicit label specifications. A portion of the default Label Format of the dimension ROWSLT in Screen 23 appears as follows:

	Lower Bound	Upper Bound	Size of Interval (Inclusive)	or	Interval Label
1	1	1			Male Life Ex.
2	2	2			0
3	3	3			1 - 4
4	4	4			5 - 9
:	:	:			:

As this example shows, Label Formats provide powerful data labeling capabilities. Refer to Chapter 6, **DEFINING LABELS**, for a detailed discussion of Label Formats.

Moving Through the Data

The Data Entry Cursor

Host displays a data entry at each intersection of the row and column dimension levels. These intersections are called *cells* and Host highlights the data entry in the current cell with an inverse or colored video block. This is combined with a small blinking rectangle showing the current character position within that entry. The combination of the highlighted block and the blinking rectangle is the data entry cursor. When data for a newly selected variable are first displayed on the screen, Host places the data entry cursor in the upper left corner of the Work Space.

Changing Row and Column Dimension Levels

You can change the current level of the row or column dimensions by moving the data entry cursor. Table 18 lists Data Editor commands for moving the cursor and editing data entries. Note that several commands in the Data Editor are different than elsewhere in Host. This allows cursor movement commands in Host's Data Editor to be identical to those of Lotus 1-2-3.

If there are more data than can be displayed in the Work Space, the Work Space acts as a window through which you view a portion of the data. You can move this window to view different areas of data by using the cursor movement commands listed in Table 18. Host beeps if you try to move the cursor beyond the range of the data; for example, if you press [↑] when you are already on the top row of data. However, if you try to move beyond the boundaries of the Work Space window, Host repositions the window automatically so the cursor remains on the screen.

Host's Data Editor has an "End Mode" identical to that in Lotus 1-2-3. This provides a fast, convenient way to move around within one or two-dimensions of data displayed in the Work Space. If you press [End], the Data Editor displays "END" at the bottom of the screen, then waits for you to press [End], [↑], [↓], [←], [→], or [Home]. If you press [End] again, Host leaves End Mode without changing the position of the cursor. Pressing a directional arrow moves the cursor to the last item of data in that direction. For example, pressing [End] then [→] moves the cursor to the last data entry in the current row, and pressing [End] then [↑] moves the cursor to the first data entry in the current column. Pressing [End] then [Home], moves the cursor to the last entry in the last column of data. After all End Mode commands, Host erases

TABLE 18.

Data Editor Cursor movement and editing commands.

IBM AT Command Key	Function in Host	WordStar Command*
[Esc]	Same as [/] in fields, but restores original field contents before displaying menu (Similar, not identical, to WordStar ^U).	^U
[Home]	First row of first column of data.	^QS
[End]	End Mode.	^QD
	END MODE	
[End]	Leave End Mode.	^QD
[↑]	Top of current data column.	^E
[↓]	Bottom of current data column.	^X
[←]	Beginning of current data row.	^S
[→]	End of current data row.	^D
[Home]	Last row of last column of data.	^QS
[↑]	Up one data row.	^E
[↓]	Down one data row.	^X
[←]	Left one data column.	^S
[→]	Right one data column.	^D
[Ctrl]+[←]	Left one window full of data.	^A
[Ctrl]+[→]	Right one window full of data.	^F
[Ins]	Insert/overtype toggle.	^V
[Del]	Delete character under cursor.	^G
[←]	Delete character to left of cursor.	[←]
[PgUp]	Up one window full of data.	^C
[PgDn]	Down one window full of data.	^R
[Ctrl]+[PgUp]	Top of current data column.	^QR
[Ctrl]+[PgDn]	Bottom of current data column.	^QC
[Ctrl]+[Home]	Top of data in window.	^QE
[Ctrl]+[End]	Bottom of data in window.	^QX
[Tab Left]	Left one window full of data.	None
[Tab Right]	Right one window full of data.	None
[Alt]+[E]	Erase current data entry.	^QY
[Alt]+[U]	Restore previous data entry ("undo").	None

* WordStar commands are represented using normal WordStar notation. In WordStar notation, holding down the [Ctrl] key while pressing [U] is represented by ^U.

the END indicator and leaves End Mode automatically.

The [Scroll Lock] key also functions as in Lotus 1-2-3. If you press [Scroll Lock], Host displays "SCROLL" at the bottom of the screen. Pressing [Scroll Lock] again turns Scroll Lock off and erases the SCROLL indicator. While the SCROLL indicator is displayed, Host cursor movement commands move the Work Space window, not the row-column position of data entry cursor,i.e., the cursor stays in the same data cell. However, as when Scroll Lock is off, Host does not allow the cursor to move out of the Work Space. If you move the window so the cursor is at an edge of the Work Space, further movement in that direction forces Host to adjust the cursor position.

Changing Other Dimension Levels

To change the levels of third, fourth, fifth, and other dimensions, first display the Edit Menu by pressing [/] or [Esc]. If you then select **Dimension** from the Edit Menu, Host places the cursor on the level of the first dimension in the Dimension Line. This is the implied time dimension PERIOD if the variable you are editing is periodic. You can move the cursor to other dimensions on the Dimension Line by pressing [←], [→], [Home], [End], [Ctrl]+[←], or [Ctrl]+[→] the same way you would in a horizontal "Lotus-style" menu. If there are more dimensions than can be displayed on the Dimension Line, Host places an arrowhead indicator at one or both ends of the Dimension Line to show that there are more in that direction.

The level of each dimension is a Host Toggle Field, so there are several ways to change this value: you can use [+], [Space], or [-] to "toggle" through available levels, or you can type in the desired setting directly. For example, if the current setting of the dimension named "REGION" is "Urban", you can change it to "Rural" by typing "rural" (upper or lower case is not important) or by pressing [Space] until Host displays "Rural". When a dimension has many levels, it is often more convenient to type in the desired level directly than to toggle to it. For example, if the level of PERIOD is "1986" and you want view data for PERIOD "1995", it is easier to type in "1995" than to toggle forward nine levels. Note that if Host does not display a different level when you attempt to toggle a dimension setting, then the Current Number of Levels of the dimension is set at one (see page 5-4, **Adding and Editing Dimensions**).

You may change the level of one, several, or all dimensions on the Dimension Line. When you are satisfied with the dimension settings, press [Enter] to display data corresponding to those levels and to return the cursor to the Work Space. If you do not want to save changes to dimension settings, press [Esc] or [/]; Host then asks whether you want to cancel or save the changes.

Changing the PERIOD Dimension Level

The time dimension PERIOD is always the first dimension on the Dimension Line for periodic variables. Although you can view different periods of data by using the Dimension command on the Edit Menu, as described above, there is an easier way. Function keys are assigned to the "Next Period", "Previous Period", "First Period", and "Last Period" of data so that pressing these keys causes Host to display data for the desired period. Table 19 lists these keys and their functions. If you are editing a periodic variable, Host displays a function key status-line at the bottom of the Work

Space to remind you that these commands are available.

TABLE 19.

Data Editor function keys for changing the level of PERIOD.

IBM AT Command Key	Function in Data Editor
[F3]	Display next period of data.
[F4]	Display previous period of data.
[F5]	Display first period of data.
[F6]	Display last period of data.

The "First Period" of data corresponds to PERIOD level one, though the attributes of the dimension PERIOD may tell Host to label this period with something other than "1". "Last Period" is the last period in the current Data File that contains data for any variable, not just the current variable. This period corresponds to the Current Number of Levels of the time dimension PERIOD (see page 5-4, **Adding and Editing Dimensions**). Pressing [F4] when you are viewing the "First Period" of data also displays the "Last Period" of data. However, pressing [F3] always displays the next period of data unless the current period corresponds to PERIOD level 1024; here, pressing [F3] displays the "First Period" of data. This allows you to append new periods of data to a periodic variable regardless of the Current Number of Levels of PERIOD. If you save new periods of data to the Data File, Host alters the Current Number of Levels of PERIOD to reflect the change.

Entering Data

While moving the data entry cursor is the same in Host as in Lotus 1-2-3, typing in data entries is not. Like 1-2-3, first you move the cursor to the cell you want to change, then you type in the desired entry. Unlike 1-2-3, changes to an entry do not appear at the top of the screen; each character you type appears at the position of the blinking cursor within the highlighted cell.

If the variable you are editing is numeric, Host erases the current cell automatically when you begin to type in a new entry. Regardless of a variable's Type, you can erase the current cell manually by pressing [Alt]+[E]. If you leave a cell empty, Host fills the cell with a default value appropriate for the variable's Type. Default values for different variable Types are explained in the section titled **If Data is not Found in the Data File** on page 8-3.

The Data Editor checks each character you type according to the current variable's Edit Mask (see page 5-14, *Edit Mask*). If you type a character not permitted by the Edit Mask, Host beeps and does not display the character. There are several ways to correct mistakes while typing an entry: you can erase the character under the blinking cursor by pressing [Del] (^G); you can erase the character to the left of the blinking cursor by pressing [BackSpace]; or you can erase the entire entry by pressing [Alt]+[E] (^QY). Until you leave a cell by pressing [Enter], or [/], or by

116

moving to a different cell, you can also use [Alt]+[U] to return the entry to the way it was when you entered the cell.

A new entry is not really stored in the current cell until you press [Enter] or move to a different cell. Pressing [/] also stores the entry in the current cell, then displays the Edit Menu. Pressing [Esc] also displays the Edit Menu, but first restores the original cell entry just as if you pressed [Alt]+[U].

When you enter a new value for a numeric variable, Host first verifies that the entry is a number. If it is not, Host places the cursor on the first invalid character so that you can correct it. Once Host verifies that the entry is a number, it checks its value against the variable's Minimum and Maximum Values (see page 5-14, *Minimum and Maximum Values*). Host will not let you leave a cell of a numeric variable until the entry in that cell is a number within the variable's Minimum Value to Maximum Value range.

Note that Host does not store new data entries to disk automatically; entries are stored in memory until you write them to the Data File by selecting Save from the Edit Menu. Host will not, however, discard new entries without asking whether you want to save them before continuing. Of course if you suspect that you made mistakes entering data, you can choose not to Save the new entries to the Data File.

The Edit Menu

If the Edit Menu is not displayed in the Menu Space, you can display it by pressing [/] or [Esc]. Screen 25 shows data for the same variable as Screen 23, but the name and description of the variable have been replaced by the Edit Menu. The Edit Menu offers the commands Resume, Area, Period, Interpolate, Save, Dimension, Variable, File, and Quit. The Resume command returns you to editing data and places the data entry cursor at its previous row-column position. Selecting Quit from the Edit Menu returns you to Host's Main Menu. The advanced functions Area, Period, and Interpolate are not yet completed (see **Appendix D**, PLANNED ENHANCEMENTS TO HOST). Attempting to choose any of these three selections in this version of Host results in the following message:

Sorry, that selection is not active on this menu.

The Dimension command is discussed above in the section **Changing Other Dimension Levels** on page 8-10. The selections Save, Variable, and File are discussed in the following sections.

Saving Data

Select Save from the Edit Menu to save any changes you have made to the data of the current variable. After you select Save from the Edit Menu, Host asks you for the name of the variable to which the data should be saved. Host always saves data to the current Data File, however, you can choose to save data under the current variable name or under a different variable name.

When asking you the name under which to save the data, Host displays the current variable name as a default. You can accept this default by pressing [Enter]; Host then saves any changes to the data under the current variable name. If the variable is

EDIT MENU

Resume Area Period Interpolate Save Dimension Variable File Quit
Resume editing current variable.

ROWSLT/COLSLT

	4	5	6	7	8	9
Male Life Ex.	58.000000	60.000000	62.000000	64.000000	66.000000	68.000000
0	0.0842000	0.0776700	0.0712300	0.0648000	0.0586200	0.0524600
1 - 4	0.0569900	0.0494100	0.0423900	0.0359300	0.0300200	0.0246700
5 - 9	0.0152300	0.0131100	0.0111600	0.0094000	0.0078000	0.0063000
10 - 14	0.0079400	0.0068500	0.0058500	0.0049500	0.0041200	0.0033000
15 - 19	0.0107100	0.0091900	0.0078000	0.0065400	0.0054100	0.0044000
20 - 24	0.0150900	0.0129300	0.0109600	0.0091000	0.0075900	0.0061600
25 - 29	0.0184000	0.0159000	0.0136000	0.0115000	0.0096700	0.0079500
30 - 34	0.0223300	0.0194000	0.0168400	0.0143900	0.0121500	0.0101000
35 - 39	0.0272700	0.0241500	0.0212200	0.0184600	0.0158700	0.0134700
40 - 44	0.0319000	0.0288500	0.0258300	0.0229400	0.0201000	0.0175400
45 - 49	0.0390500	0.0364600	0.0331600	0.0299400	0.0268000	0.0237600
50 - 54	0.0521700	0.0482300	0.0443400	0.0405000	0.0367200	0.0330000
55 - 59	0.0744200	0.0693000	0.0643700	0.0593600	0.0543000	0.0494100
60 - 64	0.1090900	0.1026000	0.0960700	0.0894000	0.0828300	0.0761300
65 +	0.3381300	0.3313400	0.3144300	0.3170000	0.3095600	0.3017300

Screen 25. A Data Editor display showing the Edit Menu.

periodic, Host saves the period of data displayed in the Work Space window regardless of whether it has been changed (see page 8-3, **Saving Editor-supplied Default Values**).

To save data under a new name, you should enter that name at the prompt. If the name you enter is not that of an existing variable, Host adds the variable to the Data File by saving the definition and data of the current variable under the new name. This is a convenient way to create several variables with identical Variable Definitions. You can Edit a variable having the same definition as the variable you want to create, change some of or all its data, then Save the changes under a new name. You can also create new variables from existing variables using Host's Define, Variable, Add commands, but this copies only the definition, not the data of the existing variable.

If you enter the name of an existing variable, the situation is more complex. Host can not save data from one variable to another unless both have identical Variable Definitions. Therefore if you enter the name of an existing variable, Host compares its definition to that of the current variable. If their Variable Definitions are different, Host asks you to enter a different variable name; if the Variable Definitions are the same, Host asks whether you want to overwrite any existing data with the data of the current variable. If you answer "No", Host asks you to enter a different variable name, otherwise Host saves the data under the specified name, replacing any data that might have existed for that variable.

Host may also ask whether you want to save changes to data when you select another Edit Menu command. If you choose Variable or File from the Edit Menu before saving changes to data of the current variable, Host gives you an opportunity to save those changes before proceeding.

Selecting Another Variable

Choose the Variable selection on the Edit Menu if you want to edit another variable in the current Host Data File. If you made changes to the data of the current variable, Host asks you whether you want to save those changes before continuing. Host gives you an opportunity to save the data, as described in the section above, if you answer "Yes". After you have saved or declined to save any changes to the current variable, Host asks you to enter the name of the variable you want to edit. You may type in a variable name or select from a list of variables in the current Data File by using Host's Query Field feature. From this point, you can not return to editing the previous variable except by reselecting that variable. This is because Host must reallocate some of the memory space used when editing a variable while allowing you to select a new variable. If instead of entering a variable name you press [/] or [Esc], Host gives you an opportunity to Resume selecting a variable, choose a different File, or Quit and return to the Main Menu.

Selecting Another Data File

Choose File from the Edit Menu if you want to edit variables stored in a different Data File. As with the Variable command, if you made any changes to the data of the current variable, Host asks you whether you want to save those changes before continuing. If you answer "Yes", Host gives you an opportunity to save the data as described in the section Saving Data. After you have saved or declined to save any changes to the current variable, Host displays a menu of Data Files in the default Data Directory. If there is a default Data File, its name is the first selection on the menu. As with the Variable command, you can not return from this point to editing the previous variable except by reselecting that Data File and variable. If instead of entering a Data File name you press [/] or [Esc], Host gives you an opportunity to Resume selecting a Data File or Quit and return to the Main Menu. If you select a Data File, Host asks you the name of the variable you want to edit as explained in the section Selecting a Variable on page 8-2.

CHAPTER 9
RUNNING MODULES

- 120

RUNNING MODULES

To run any Application Module from within Host, select Run from Host's Main Menu. Figure 22 shows the command structure of the Run facility.

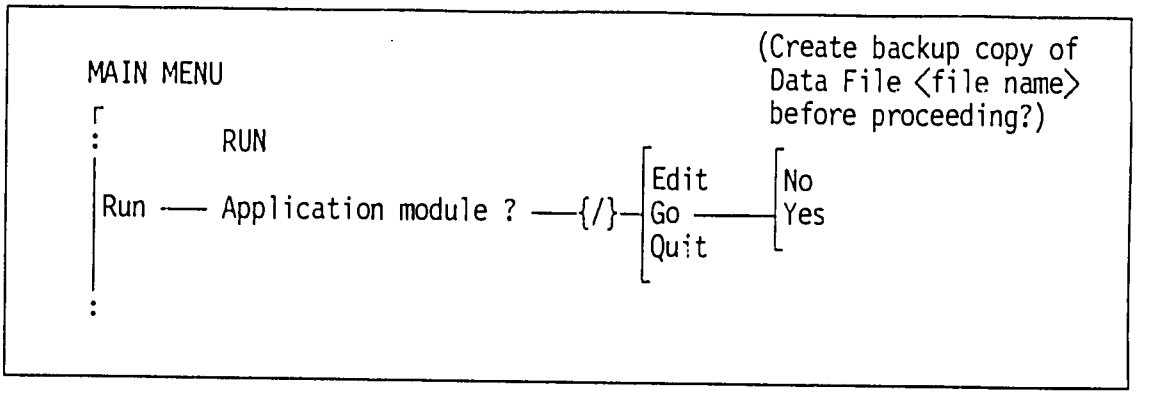


Figure 22. Commands required to run a single Application Module.

Selecting an Application Module

After you select Run from Host's Main Menu, Host asks you for the name of the Application Module to be executed. You can type in the name of any Application Module in your system or choose from those in the default Module Directory by using Host's Query Field feature. If you enter the name of an Application Module without preceding the name with a directory path, Host looks for the Application Module in the default Module Directory.

In addition to the name of the Application Module, Host must know several other things before it can run the module. If must know which Host Data File contains the data the module is to use. It also needs to know the Starting Period and Number of Periods to project for Projection Modules. For other types of Application Modules, the Period settings may be ignored, or they may be used to supply other information to the module. You should consult the documentation provided with a module for details on the use of these settings with that particular module.

Once you have selected an Application Module, the current settings for these items are displayed in the Work Space. You may accept these values and run the application immediately (Go), or change them (Edit) before running the module. Each of these settings is explained in the following sections.

Selecting the Host Data File

Most Application Modules read initial data from and write result data to the Host Data File whose name appears in the Data File field. Initially, the entry in this

field is the name of the default Data File, but you can change this by entering the name of any Data File in your system. If there is no default Data File, this field is blank and you must enter the name of a Host Data File before you can run an Application Module. The variables and dimensions required by the selected module must match existing variables and dimensions defined in the chosen Data File or the module will halt, display an error message, and return control to Host. Host looks for the Data File in the default Data Directory unless you enter an explicit directory path preceding the Data File name.

Specifying the Starting Period

The Starting Period value is intended to be used by Projection Modules as the period from which to begin projecting data. The default starting time period for projections is 1. Some modules may not function correctly if you choose a starting period other than 1; this depends on the function and implementation of the module. If an Application Module is not a Projection Module, it may ignore this setting or use it for other purposes. Consult any documentation provided with the module you intend to run before changing this value. The maximum value of this field is 1023, but starting period and number of periods to project are interrelated such that starting period plus number of periods must not exceed 1024.

Specifying the Number of Periods

The Number of Periods value is intended to be used by Projection Modules as the number of periods to project data forward from the Starting Period. The default number of time periods to project forward is 1. You will almost always want to specify a value greater than 1 when running a Projection Module. Some modules, depending on their purpose and implementation, may yield nonsensical results if asked to project a large number of periods. If an Application Module is not a Projection Module, it may ignore this setting or use it for other purposes. Consult any documentation provided with the module you intend to run before entering a large value into this field. The maximum value of this field is 1023, but starting period and number of periods to project are interrelated such that starting period plus number of periods must not exceed 1024.

Running the Application Module

If you have selected Edit from the Run Menu to change the displayed settings, you can redisplay the Run Menu by pressing [/] or [Esc]. You can select Go from the Run Menu to run the module with the current settings, Edit to further edit the settings, or Quit to return to Host's Main Menu. If you choose Go, Host gives you an opportunity to make a back-up copy of the specified Data File before running the selected Application Module. This allows you to save the contents of the Data File before any changes are made by the Application Module. After you have copied or declined to copy the specified Data File, Host passes the current values of the displayed settings, then control of the microcomputer over to the Application Module. At the top of the screen, in the Menu Space, Host displays the instruction that would have achieved the same purpose as this interactive procedure had it been encountered in an Instruction file or entered at the DOS prompt. See Chapter 11, **INSTRUCTION FILES**, for an explanation of running Application Modules using Instruction files. Chapter 13, **ADVANCED TOPICS**, describes how to run Application Modules directly from DOS.

132

Once the Application Module is in control, it can do whatever it has been programmed to do, including prompting you for information, manipulating data in the selected Host Data File, and displaying information in the Work Space. A module may even take over the entire screen to display graphics images. While the module is in control of the microcomputer, if the display is still in text mode, Host flashes the Mode Indicator "RUN".

When the Application Module has completed its task, Host asks if you wish to run another Application Module. If you answer "Yes", you can run another module by repeating the process described above. The settings that were established in the prior run are displayed in the Work Space. You may edit these settings to re-run the same module, or to begin running other Application Modules. If you respond "No", Host returns you to its Main Menu.

CHAPTER 10
GENERATING TABLES

GENERATING TABLES

Host's *Table Generator* combines a Table Definition from the Table Library with data from a Host Data File to produce a table. This table can be directed to your display screen, to your printer, or to a Print-ready or Lotus-import file. Chapter 7, **DEFINING TABLES**, explains the settings stored in Table Definitions and describes creating, editing, listing, reporting, and deleting these Definitions. This chapter explains how to produce a table by calling Host's Table Generator.

Host's Table Generator can be executed from Host's menus, from within Host Instruction files, and directly from the DOS prompt. See Chapter 11, **INSTRUCTION FILES**, for an explanation of generating tables from within Instruction files. Chapter 13, **ADVANCED TOPICS**, describes how to generate tables directly from DOS. To generate tables from Host's menus, select the Tabulate command from the Main Menu or Table Menu. Figure 23 shows commands required to generate a table, starting from Host's Main Menu. The commands to do the same operation from the Table Menu are identical.

```
MAIN MENU
:
|
|     TABULATE
|                                          Edit
| Tabulate — Table Name ? ————————{/}—Go
|                                          Quit
:
```

Figure 23. Commands required to generate a table from Host's Main Menu.

Selecting a Table

After selecting Tabulate (from the Main Menu or Table Menu), Host asks you to enter the name of a Table Definition. You should enter the name of a Table Definition in the Table Library, or select one using Host's Query Field feature. The desired Table Library must reside in the default Library Directory.

After you have chosen a Table Definition, the Table Generator must know several other things before it can produce the table. It must know which Host Data File contains the data to be tabulated and which to use as the Comparison File if one is required. It also needs to know the destination for the resulting table and which periods of data should be included in it. After you have selected a Table Definition, the default settings for these items are displayed in the Work Space. You may accept

these values and generate the table immediately (Go) or change them (Edit) before producing the table. These entries are explained in the following sections.

Selecting a Host Data File

Initially, the entry in this field is the default Host Data File, but you can change this by entering the name of any Host Data File in your system. If there is no default Host Data File, you must supply the name of a Host Data File before generating the table. The variables and dimensions named in the Table Definition must match existing variables and dimensions in the chosen Host Data File or the table can not be produced. The Host Data File is assumed to be in the default Data Directory unless you enter a directory path with the file specification.

Selecting a Comparison File

Initially, the entry in this field is set to the default Comparison File, but you can change this by entering the name of any Host Data File in your system. The variables and dimensions stored in the Table Definition must match existing variables and dimensions in the chosen Host Data File or the table can not be produced. The Comparison File is assumed to be in the default Data Directory unless you enter a directory path with the file specification. If a Comparison File is specified but there are no comparison variables in the tables, the file is ignored. If a Comparison File is required but none is named, the Table Generator issues an error message and returns without producing the table.

TABLE

Currently executing :
TABLE T=SUMMARY1 D=AG O=COM P=ALL

Estadisticas resumen para cada periodo

SUMMARY: Resumen anual de la proy. agricola

PERIOD/SUMMROWS	Poblacion	Tasa crec. pob.	Costo prod. ag.	Costo imps.
1985	5696.000	0.000	5761706.355	97215.000
1986	5835.000	0.024	5945524.056	85060.746
1987	5977.000	0.024	6132690.309	89727.075
1988	6122.000	0.024	6296479.067	94049.521
1989	6269.000	0.024	6425029.476	99054.140
1990	6416.000	0.023	6519699.033	104009.027
1991	6564.000	0.023	6094740.770	107307.773
1992	6715.000	0.023	7005601.525	112434.123
1993	6867.000	0.023	7274515.521	117762.509
1994	7019.000	0.022	7459420.100	123303.077
1995	7169.000	0.021	7636023.000	129300.692

* Press [Esc] to quit; [PgUp], [PgDn] to page up or down.

Screen 26. A sample screen showing output of the Table Generator.

Specifying an Output File

Table output may be sent to the display screen, printer, or one of two types of disk files. Initially, the entry in this field is the default Table Output Device, and thus is either "CON" (console or display screen) or "PRN" (printer). Screen 26 on the previous page shows a typical table output display produced by directing table output to the console. Figure 24 shows an example page produced by sending table output to a printer. You may change the output device to "CON" or "PRN", or direct output to one of two types of output files: a *Lotus-import* file or a *Print-ready* file.

A Lotus-import file is an ASCII text file formatted according to the import requirements of Lotus 1-2-3 spreadsheet software. Specifically, all text labels in such a file are enclosed in double-quotes and no line in the file is longer than 240 characters. When producing such a file, print margin and page eject settings are ignored. A file with this format can be imported, or read into a Lotus 1-2-3 Worksheet for further numerical or graphical analysis (see Importing Table Output into Lotus 1-2-3, page 10-8).

A Print-ready file contains the Setup String (stored in the Configuration File), ASCII Form-Feed characters, and the table output. You might want to save your tables in Print-ready files to transfer them to other machines for printing or to include them in word processing documents. After saving a table to a Print-ready file, you can print it by entering the following command at the DOS prompt:

C>copy <file name> prn[Enter]

where <file name> represents the table Output File named when the table was generated. If you are familiar with PC/MS-DOS, you may also know several other commands that accomplish the same task.

To request a Lotus-import file, enter a file name followed by the extension "PRN". For example, entering "POPBYFIV.PRN" tells Host's Table Generator to format table output for importing into Lotus 1-2-3 and store it in the file POPBYFIV.PRN. An Output File named using any other file name extension will be formatted as a Print-ready file. File output is directed to the default Data Directory unless an explicit path is given with the Output File name. The section Importing Table Output into Lotus 1-2-3, on page 10-8, describes how to read a Lotus-import file into a Lotus worksheet.

Specifying a Period List

The period list tells Host's Table Generator which periods of data to include in the table. This list is useful only for tables made up of periodic variables. A period list is the word "ALL", or a list of integer *period values* and *period intervals* separated by commas.

A period value is an integer from 1 to 1024 inclusive. Note that these values are levels, not Effective Levels, of the time dimension PERIOD (see Chapter 5, DEFINING DATA). For example, if the Starting Value of the dimension PERIOD is defined as 1986, then a "1" in the period list tells Host to include data from the PERIOD labeled as "1986". Further, if the Increment Between Levels of the dimension PERIOD is defined as 5, then a "2" in the period list corresponds to the PERIOD labeled "1991", a "3" to the PERIOD labeled "1996", and so on.

Hoja de balance alimenticio

FBS: Hoja de balance alimenticio

PERIOD : 1985

PRODUCTS/FBS	Prod.	Export.	Import.	Consumo	Cal. p.c.
Arroz	5537	0	895	6432	510
Habichuelas	685	0	146	831	60
Maiz	0	0	3401	3401	262
Yuca	1784	26	0	1758	42
Names	316	0	0	316	8
Batatas	1127	127	0	1000	24
Platano	901	0	0	901	13
Tabaco	836	484	0	352	0
Azucar	22029	17440	0	4589	353
Cafe	1325	566	0	759	0
Cacao	510	510	0	0	0
Gan. vacuno	1073	0	0	1073	57
Gan. porcino	350	0	0	350	18
Pollos	1528	0	22	1550	60
Leche	189	0	200	389	6
Trigo	0	0	3126	3126	248
Otros	4000	0	1000	5000	361

PERIOD : 1986

PRODUCTS/FBS	Prod.	Export.	Import.	Consumo	Cal. p.c.
Arroz	5728	0	940	6668	517
Habichuelas	712	0	146	858	60
Maiz	0	0	3505	3505	263
Yuca	1834	26	0	1808	42
Names	325	0	0	325	8
Batatas	1156	127	0	1029	24
Platano	927	0	0	927	13
Tabaco	875	508	0	366	0
Azucar	22721	17963	0	4757	357
Cafe	1370	583	0	787	0
Cacao	520	520	0	0	0
Gan. vacuno	1123	0	0	1123	58
Gan. porcino	364	0	0	364	18
Pollos	1594	0	22	1616	61
Leche	185	0	220	405	6
Trigo	0	0	3260	3260	253
Otros	4043	0	1100	5143	362

Figure 24. A typical page of output from the Table Generator.

A period interval has the form

first-last

where first and last are period levels. A period interval includes all periods from first to last inclusive. If first is omitted, all periods up to and including last belong to the interval; if last is omitted, all periods from first on belong to the interval. If first and last are omitted all periods are included in the interval, thus "-" is a synonym for ALL. Overlaps in groups are ignored, so each period listed is included in the table only once. Table 20 shows some valid period lists and, for each entry, the periods included in the resulting table, as well as some invalid period lists.

TABLE 20.

Examples of valid and invalid period list entries.

Valid Period List Entry	Periods Included in the Table	Invalid Period List Entry*
1,,,1,2-5,8-7,10	1,2,3,4,5,7,8,10	-1-,2-5,8-7,10
-	All periods	--
-5, 8-10	1,2,3,4,5,8,9,10	--5, 8-10
1-3,5,-	All periods	1-3-5,-

* Each of these period list entries is invalid because it contains more than one dash in a period interval.

Generating the Table

You can display the Tabulate Menu at any time by pressing [/] or [Esc]. You may then select Go to generate the table using the current settings, Edit to further edit the settings or Quit to return to Host's Main Menu. If you choose Go, Host passes the current values of the displayed settings, then control of the microcomputer over to the Table Generator. In the Menu Space, Host displays the instruction that would have achieved the same effect had it been encountered in an Instruction file or entered at the DOS prompt.

When the Table Generator is active, it displays several status messages in the Message Space, then sends the resulting table to the requested output destination. While working, the Table Generator displays the flashing Mode Indicator "TABLE" for tables directed to the console, "STORING TABLE" for tables directed to files, and "PRINTING TABLE" for tables directed to the printer.

If you generate a table to the console, you may page up and down though the table by using the [PgUp] and [PgDn] keys. Pages of the table are stored in memory, so the number of pages retained by Host depends on the amount of memory available in your microcomputer. If the entire table can not be stored, you can not always page up to the first part of the table.

If you press [Esc], the Table Generator immediately stops producing a table. If output is directed to a Lotus-import or Print-ready file, the Output File is closed. If the output file is "PRN", no further data is sent to the printer, but the size of your printer's memory buffer determines how quickly printing stops.

When the Table Generator has completed its task, or has been interrupted, Host asks if you wish to generate another table. If you respond "Yes", you can generate another table by repeating the process described above. If you respond "No", Host returns you to the menu from which you selected Tabulate.

Problems Generating Tables

When Host's Table Generator begins executing, it does a series of checks. If no errors are encountered during these checks, the Table Generator reads the data of the variables named in the Table Definition from the specified Host Data File and produces a table according to the Table Definition dimension settings. The following paragraphs describe the errors you may encounter when trying to produce a table.

Incompatible Data File

Host's Table Generator first verifies that all variables named in the Table Definition are defined in the specified Host Data File, then it verifies that each variable has the dimensions listed in the Table Definition dimension settings. If PERIOD is among the dimensions of the table, the Table Generator verifies that all variables are periodic; otherwise it checks to see that all variables are non-periodic. If any variables in the specified Host Data File fail to meet the above conditions, the Table Generator displays an error message and returns without producing a table. This type of error occurs when you specify the wrong reference Data File for the table.

Insufficient Memory

If the above conditions are met, the Table Generator next computes the amount of memory required to produce the table. If there is not enough memory available, the Table Generator displays the error message

Insufficient memory to generate table <table name>.

and returns without producing a table. Tables requiring Unnested Totals and Subtotals as well as tables displaying variables of very large size may cause this type of error.

Table Too Wide

Lastly, if there is enough memory to produce the table, the Table Generator verifies that there is enough room to print or display the table. The Table Generator can split most tables having too many rows or columns to fit on a single page or screen, but there are two types of tables that the Table Generator may not be able to split into sufficiently small pieces: tables having a Nested dimension and tables

having a column dimension with Combined status "No". The rule is that the Table Generator does not split the levels of a Nested dimension within a single level of a column dimension or a single variable, and it does not split the levels of a column dimension having Combined status "No" within a single label group.

For tables directed to the printer or to a Print-ready file, the minimal groups of levels described above must fit within the Line Length stored in the Configuration File. For tables directed to Lotus-import files, these levels must fit within the 240-character Line Length limit. For tables directed to the screen, the lines "wrap around" if these levels do not fit within the screen width. However, the Table Generator will not wrap a line onto more than three screen lines. For tables directed to the printer or to a Print-ready or Lotus-import file the Table Generator issues the error message

Line length is too short to permit generation of <table name>.

if there is insufficient space, then returns without producing the table. For tables directed to the screen, the Table Generator issues the error message

Table exceeds screen width; will try wrapping lines . . .

if there is insufficient space to fit the minimum group of levels on a single line of the screen. If the required space exceeds the three-line limit, the Table Generator returns without producing the table.

Whenever a line length error occurs, you can either increase the Line Length in the Configuration File (for tables directed to the printer or to a Print-ready file), or you can change the Table Definition so that the problem does not occur. Usually a Nested dimension having more than two or three levels should be changed to an Unnested dimension. Another possibility for tables having only one variable is to change the Nested dimension to the column dimension and make the column dimension an Unnested dimension. If the label groups of a dimension with Combine setting "No" contain large numbers of levels, then the dimension should probably be the row dimension or an Unnested dimension rather than the column dimension.

Importing Table Output into Lotus 1-2-3

As explained on page 10-4, Specifying an Output File, table output can be stored in a Lotus-import file by directing it to a disk file with the extension "PRN". Host's Table Generator creates a file that conforms to Lotus 1-2-3 requirements for such files. To read the contents of an Import file into any Lotus Worksheet, first enter the Lotus 1-2-3 program, then open a Lotus Worksheet and make sure you are in the Lotus "READY" mode. Next, set the Lotus default data directory to the directory containing the Import file by issuing the following command sequence: /Worksheet, Global, Default, Directory. There is no need to save this setting (Update) unless you want to retain it for future Lotus sessions. Finally, tell Lotus to read the Import file by issuing the command sequence: /File, Import, Numbers. Lotus should display the name of the import file in a menu containing the names of all files in the default data directory that have a "PRN" extension. Lotus reads any file you select from this menu into the current Worksheet, beginning at the current cursor position.

After the selected Import file has been read into the Worksheet, some text labels may appear truncated or overlapped and some numbers may be formatted differently from the original Host table. All these problems are easily corrected using the Lotus commands /Worksheet, Global, Column-Width to increase the column width, /Move to move text labels, and /Range, Label-Prefix or /Range, Format to change the display format of text or numeric entries. All the above changes affect only the Worksheet display, not the actual content of the Worksheet. Regardless of whether you make any adjustments in the Worksheet, the contents of the table may now be numerically or graphically analyzed using Lotus 1-2-3.

CHAPTER 11
INSTRUCTION FILES

INSTRUCTION FILES

Host Instruction files provide a convenient way of running Application Modules, generating tables, and executing other functions without operator intervention. An *Instruction file* is a standard ASCII text file containing DOS operating system commands and Host instructions. Each line of an Instruction file contains a DOS command or an instruction telling Host to perform some task, such as run an Application Module or generate a table. When Host is commanded to process an Instruction file, it executes the instructions in sequence from the first to the last instruction in the file. Instructions not recognized by Host are passed to DOS for interpretation and execution.

You can create and modify Instruction files using Host's Instruction File Editor, or by using any standard text editor. You can assign any valid file names to your Instruction files; Host automatically assigns the extension "INS" and stores the files in the default Instruction Directory. If you create an Instruction file using a text editor, you should assign it the extension "INS" and place it in the default Instruction Directory, so that Host can recognize the file as an Instruction file. You may have as many Instruction files as your disk storage space allows.

You can process Instruction files from within Host using the Process command on the Instruct Menu or you can execute them directly from DOS. See Chapter 13, **ADVANCED TOPICS**, for details on processing Instruction files from DOS.

Figure 25 shows the command structure of Host's Instruction file facility. Both Users and Managers have access to all commands in this facility.

```
MAIN MENU
 :
 |            INSTRUCT     Create —— Instruction file to create :
 |                         Edit ——— Instruction file to edit ?
 |                         Backup —— Instruction file to copy ?
 | Instruct ———————————    Process — Instruction file to process ?
 |                         Delete —— Instruction file to delete ?
 |                         Quit
 :
```

Figure 25. The command structure of Host's Instruction file facility.

Each Host instruction has the following format

INSTRUCTION argument-specification ... argument-specification

where INSTRUCTION is the instruction name, and argument-specification has the form argument=<value>, with no embedded spaces. The angle brackets ("<"...">") are used to delimit the value in this notation and are not part of the instruction. If an argument specification is enclosed in square brackets ("["..."]"), the argument is optional, otherwise the argument is required. The order of argument specifications in an instruction line is not important. Spaces are used to separate the instruction and its argument specifications, and many spaces are allowed wherever one space is allowed. Blank instruction lines are permitted, but an instruction may not be divided across two or more lines. Instructions may be in any combination of upper and lower case characters.

The instruction names are

RUN	run an Application Module
SUSPEND	suspend processing of Instruction file
TABLE	generate a table
GRAPH	generate a graph
BASE	set the base year for the dimension PERIOD
INCR	set the increment between levels of dimension PERIOD
PERIOD	set the current levels of dimension PERIOD
REM	comment line (not interpreted)

The complete syntax for each of these instructions is explained in detail in a following section. Unless you intend to create and edit Host Instruction files using some other text editing program, however, it is not necessary to memorize the syntax of each instruction. You can request Host's Instruction File Editor to prompt you for the information required by any of these instructions and to compose a correctly formulated instruction from your responses.

In addition to the special Host instructions described above, you can also include DOS operating system commands in an Instruction file. This includes commands to manipulate disk files (e.g., DIR, COPY, ERASE, RENAME), as well as commands to execute other programs such as Lotus 1-2-3 or SPSS/PC+™. In general, DOS commands should be entered exactly as you would type them at the DOS prompt. The few changes and restrictions are listed in Table 21.

Any instruction not recognized by Host is passed to DOS for interpretation and execution. Host remains in memory throughout this process, so the amount of memory available to other programs at this point depends on how much memory is installed in your microcomputer, and how much of that memory is occupied by Host and other memory resident programs. If there is not enough memory to execute a command or program, Host will notify you and return to the Main Menu.

Screen 27 shows a typical Instruction File Editor display. This screen display shows an Instruction file named "ED2" being edited. The current line number ("7") is displayed in the Menu Space along with the Instruction file name. Below that, in this example, is a prompt for the period list to be used in the TABLE instruction being constructed interactively on line 7. The contents of the Instruction file are displayed in the Host Work Space. A brief help message, followed by a function key help line, is displayed at the bottom of the Work Space.

TABLE 21.

Restrictions to DOS command use in Host Instruction files.

DOS Batch Files	Host Instruction files
replaceable parameters	not implemented
FOR %%variable IN (set) DO command	will work in Instruction file if if %%variable is changed to %variable
GOTO :label	not implemented
SHIFT	not implemented

Line 1 of the Instruction file specifies running of an Application Module named "populate" starting in PERIOD one and projecting for five PERIODS. Line 2 sets the Starting Level of the PERIOD dimension to 1980, and line 3 sets the Increment Between Levels of PERIOD to one. Lines 4 through 6 request generation of tables named "prim", "jss", and "sss", respectively. In each case output is sent to the console ("con") and "all" periods of data are included in the table. Line 7, contains part of a Host TABLE instruction that is being constructed interactively. This instruction tells Host to generate a table using the Table Definition "oldsec" and to store the table in a Lotus-import file named "oldsec.prn" in the "\lotus" subdirectory of the current disk drive. Note that the answer to the prompt at the top of the screen determines which periods of data are included in the table. Finally, since the last two lines of this Instruction file are not Host instructions, Host passes them to the PC/MS-DOS operating system. These lines tell DOS change to the "\lotus" subdirectory and to execute the "lotus" program.

Creating and Editing Instruction Files

To create a new Instruction file from the Instruct Menu, select the **Create** option and enter the name of the Instruction file. To edit an existing Instruction file, select the **Edit** option on the Instruct Menu and either type in the name of the file you wish to edit or use the Query Feature to obtain a list of all Instruction files in the default Instruction Directory. Once the name of the Instruction file to create or edit is entered, Host treats these two functions the same way.

After an Instruction file is named, the Instruction File Editor screen appears. This screen places you on the first line of the Work Space and allows you to enter text as you would in most text editors. The [Ins], [Del], and cursor movement keys provide basic editing capabilities. In addition, you may use [Alt]+[I] (^N) to insert a line above the current one and [Alt]+[D] (^Y) to delete the current line. The top of the screen displays the file name and the number of the line you are editing.

If you prefer, you can enter each instruction using the Instruction File Editor's basic text editing capabilities. Instructions entered or modified this way are not checked for correct syntax, so it is possible to create an Instruction file in Host that is not syntactically correct. Host's Instruction File Processor, described in the section Processing an Instruction File on page 11-10, checks the syntax of each line

```
                                                                EDIT FILE
Instruction File Name : EDZ          Line Number : 7
Periods : ALL_____________________________________________________
run m=populate s=1 n=5
base b=1980
incr i=1
table t=prim o=con p=all
table t=jss o=con p=all
table t=sss o=con p=all
table t=oldsec o=\lotus\oldsec.prn
cd \lotus
lotus

* Use [↓] or [↑] to point to a line, enter text to edit or press a function key.
F3-RUN  F4-TABLE  F5-SUSPEND  F6-GRAPH  F7-BASE  F8-INCR  F9-PERIOD  F10-REM
```

Screen 27. A typical Host Instruction File Editor display

of the Instruction file before executing any instructions. To avoid syntax errors and speed the editing process, you can use interactive prompting provided by Host's Instruction File Editor. This facility helps you build each instruction quickly and correctly, eliminating the the need to memorize instruction syntax.

The Interactive Prompting Facility

To construct instructions using Host's interactive prompting facility, press a function key as shown in the function key help line at the bottom of the screen. Each of the function keys [F3] through [F10] corresponds to a Host instruction. Table 22 lists these keys and their functions.

After you press one of these function keys, Host prompts you for the values of all required arguments and some optional arguments of the selected instruction. As you respond to each prompt, the instruction is composed and displayed for you in the current line of the Instruction file. After you have supplied all necessary values, the completed instruction line is displayed, and a prompt appears asking you whether the instruction you have entered is what you want. If you answer "No", Host re-prompts you for each argument value. When making corrections in this interactive mode, press [Enter] to retain the displayed setting or enter a new value when desired.

This process of selecting a function key then responding to the associated prompts can be repeated to build the instruction sequence you want. Note that the Instruction File Editor does not prompt for values of all arguments. Host Data File (D) and Comparison File (C) arguments are not required, since all Host Instructions use the default Data File and Comparison File if these arguments are not supplied. If you do not want default values to be used when an instruction is exec ted, you must explicitly specify other values by using basic text editing commands to insert the necessary argument specifications.

TABLE 22.

Function keys used in the Instruction File Editor.

IBM AT Command Key	Host Instruction	Explanation of Instruction
[F3]	RUN	run an Application Module
[F4]	SUSPEND	suspend processing of Instruction file
[F5]	TABLE	generate a table
[F6]	GRAPH	generate a graph
[F7]	BASE	set the base year for the dimension PERIOD
[F8]	INCR	set the increment between levels of PERIOD
[F9]	PERIOD	set the current levels of dimension PERIOD
[F10]	REM	comment line (not interpreted)

You may alternate freely between using interactive prompting and basic text editing commands to enter or modify instructions. Many of the interactive prompts require entry of file names or Host Identifiers. In these cases, you can select from a list of available choices using Host's Query Field facility.

Corrections can be made to instruction lines by first moving the cursor to the desired line, then using basic text editing commands. Alternatively you can move to a line and replace its contents by using the interactive prompting facility. As stated above, when making corrections using interactive prompting, press [Enter] to retain the displayed value or enter a new value when desired. Note that the values of the arguments in the original instruction do not appear as defaults for the new instruction.

Saving Host Instruction Files

You can save new Host Instruction files or changes to existing Instruction files using the Save command on the Edit Menu. After entering the Save command, Host prompts you for a name under which to save the file. You can save the file under its original name or save it under a different name. Host automatically supplies the "INS" extension and stores the file in the default Instruction Directory. The ability to save changes to an Instruction file under a new name allows you to create a several similar Instruction files quickly. You can Edit an Instruction file similar to the file you wish to create, change as many instructions as you like, and Save the changes under a new name.

Host Instruction File Syntax

The following syntax diagrams and explanations are included for reference.

RUN M=<module> [D=<data file>] [S=<starting period>] [N=<number of periods>]

This instruction tells Host to run the named Application Module.

M= Application Module name. This is a required argument. If an extension is supplied, it is used, and otherwise the extension "COM" is assumed. If a path is desired, it must be supplied. Otherwise the Application Module is assumed to reside in the default Module Directory.

D= Host Data File name. This argument is optional. If this argument is not included in the instruction line, Host uses the default Data File.

No file name extensions are allowed. If a path is desired, it must be supplied, otherwise the Host Data File is assumed to be in the default Data Directory.

S= Period at which to begin projecting. This is an optional argument, and defaults to a value of one. Most Application Modules expect initial data to be stored in period one, so the first time a Application Module is run, this argument should usually be one (S=1). Consult the documentation provided with the Application Module being run before using a value other than one.

N= Number of periods to project. This is an optional argument, and its default is one. Consult the documentation provided with the Application Module being run to determine what values of this argument are reasonable.

SUSPEND

This instruction has no arguments. Its effect is to suspend the processing of the Instruction file and allow the user to return to the interactive portion of Host. It is then possible to resume processing of this file by reselecting the Process option on the Instruct Menu.

TABLE T=<table> [D=<data file>] [C=<comp data file>] [O=<output>] [P=<period list>]

This instruction tells Host to generate the named table.

T= Table Definition name. This is a required argument. The named Table Definition must be in the Table Library.

D= Host Data File name. This argument is optional. If this argument is not supplied in the instruction line, Host uses the default Data File.

No file name extensions are allowed. If a path is desired, it must be supplied, otherwise the Host Data File is assumed to be in the default Data Directory.

C= Comparison File name. This is an optional argument. If this argument is not supplied, Host uses the default Comparison File. The same rules for extensions and paths stated for the D= argument apply to this argument as well.

O= Output file name for table. This is an optional argument, and defaults to the output device named in the Configuration File.

If the file is the printer ("PRN") or console ("CON"), then the table is directed to the appropriate device. Otherwise a file of the given name is created and the table is written to that file. If a file of the given name

already exists, it is overwritten.

A file extension of "PRN" designates creation of a file that can be imported into Lotus 1-2-3 (/File, Import, Numbers) for further analysis or for graphing. Any other extension specifies creation of a disk file containing exactly the same information as would have been sent to the printer if output had been directed to "PRN" (printer). See Chapter 10, **GENERATING TABLES**, for a complete discussion of Lotus-import and Print-ready files.

Output is directed to the default Data Directory unless another directory path is supplied.

P= Period list. This is an optional argument listing the periods of data to include in the table. The default value is "ALL". A complete discussion of period list syntax is given in Chapter 10, **GENERATING TABLES**.

GRAPH

This instruction is not yet implemented.

BASE B=<base year> [D=<data file>]

This instruction tells Host to set the Starting Value of the time dimension PERIOD in the Host Data File to the value of **base year**. This causes period level one to be labeled as **base year** whenever that level of PERIOD appears in displayed or printed output.

B= Base year value of time dimension PERIOD.

D= Host Data File name. This argument is not required; if it is omitted, Host uses the default Data File in the default Data Directory.

INCR I=<increment> [D=<data file>]

This instruction tells Host to set the Increment Between Levels of the time dimension PERIOD in the specified Host Data File to the value of **increment**. This causes each level of the dimension PERIOD to to be labeled as

(level - 1) * increment + base year

whenever it appears in displayed or printed output.This is the formula for the Effective Level of the PERIOD dimension, where **base year** is the Starting Value of the dimension. See Chapter 5, **DEFINING DATA**, for a discussion of Effective Level of a dimension.

I= Increment Between Levels of the time dimension PERIOD.

D= Host Data File name. This argument is not required; if it is omitted, Host uses

the default Data File in the default Data Directory.

PERIOD L=<level> [D=<data file>]

This instruction tells Host to set the Current Number of Levels of the time dimension PERIOD in the specified Host Data File to the value of **level**. This causes any Host Data File data beyond that period value to be excluded from tables and graphs.

L= Current Number of Levels of the time dimension PERIOD.

D= Host Data File name. This argument is not required; if it is omitted, Host uses the default Data File in the default Data Directory.

REM <text>

This instruction designates a comment. It can be used to annotate Instruction files and is ignored when an Instruction file is processed.

Backing-Up an Instruction File

To back-up an Instruction file, select the Backup command from the Instruct Menu. Enter the name of the file to be copied, or select the name using the Query Field feature. This file is called the *source* file. After you have entered the name of the source file, Host prompts you for the name of the file to receive the copy. This file is called the *target* file. You may enter any valid DOS file name for the target file. If you do not precede the target file name with a directory path, Host assumes you want to place the file in the default Instruction Directory. The file name extension "INS" is assigned automatically by Host.

The target file should not have the same name as the source file unless you are placing it on a different drive or in a different directory than the source file, since a file can not be copied onto itself. If you inadvertently designate the same file for both source and target, Host warns you of the error and allows you to name a different target file. You should not name an existing file as the target file unless you want Host to replace its contents. If this occurs, Host warns you of the danger and asks you if you want to replace the contents of that file before proceeding with the copy.

Once the target file name has been entered as described above, Host checks to see if there is enough disk space to make the copy. If there is not enough space, Host reports the error and asks you to enter a different target file name. You should name a file on a drive with more available space, or insert a new diskette or cartridge with enough space for the copy.

Processing an Instruction File

To process an Instruction file, select the Process command from the Instruct Menu. Enter the name of the desired Instruction file, or select the name using the Query Field feature. Figure 26 summarizes the possible paths through the Instruction File Processor under different circumstances.

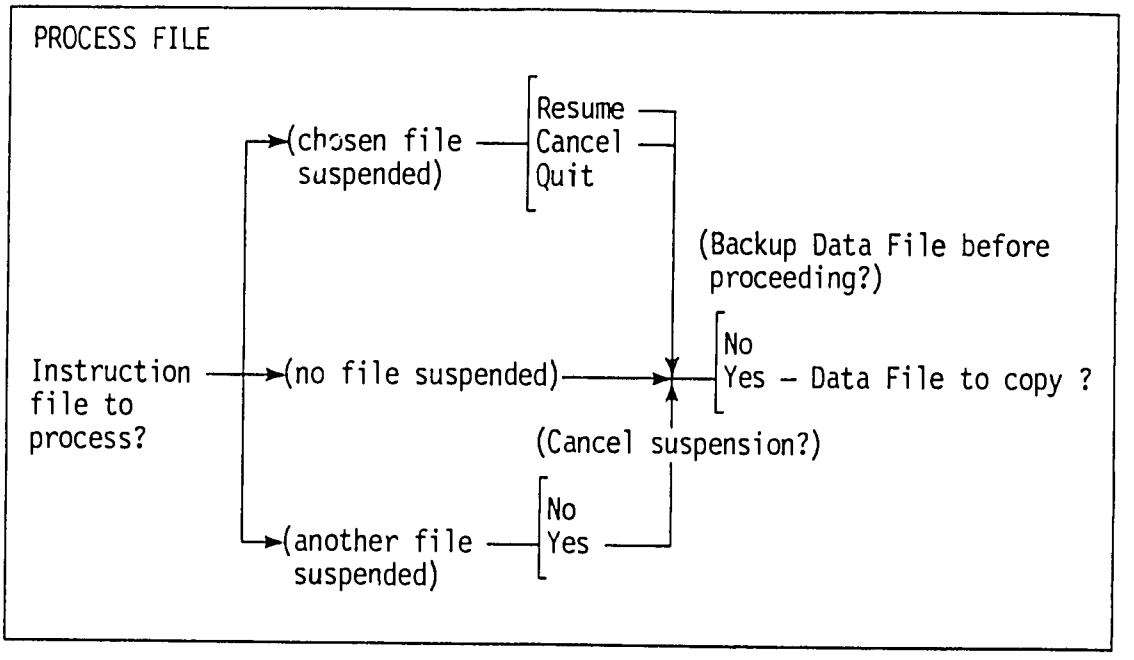


Figure 26. A summary of paths through the Instruction File Processor.

If the current Instruction file is suspended, Host asks you if you wish to resume processing at the point of suspension, or if you want to cancel the suspension and begin processing with the first instruction in the Instruction file. If some other Instruction file is suspended, Host gives you an opportunity to cancel the suspension before beginning processing of the current file.

Before processing the Instruction file, Host gives you an opportunity to make back-up copies of any Data Files that may be altered by instructions in the Instruction file. If you want to back-up one or more Host Data Files answer "Yes". Host then asks you to enter the name of each source Data File and target Data File until you press [Enter] without entering a source Data File name.

After you have responded to the back-up prompt, the Instruction File Processor reads the Instruction file and verifies that each line is a correct Host Instruction. If Host detects a syntax error, it displays a brief error message in the Error Message Space and more detailed explanation in the Work Space; then Host returns to the Instruct Menu. Screen 28 shows an example of the messages Host displays after discovering an instruction syntax error. In this example, the problem is an invalid argument ("Q") in line 1 of Instruction file "ED2". Here, you should choose Edit from the Instruct Menu, enter the Instruction file name "ED2", then correct the error in line 1.

```
                                                          INSTRUCT MENU
Create Edit Backup Process Delete Quit
Edit an existing Instruction file.
------------------------------------------------------------------------

                              WARNING

    * A syntax error was found while Host was processing the
      instruction in line 1 of the Instruction file ED2.

    * Refer to the Host User Guide for help in identifying the
      cause of the error and correcting it.



    * Use the arrow keys to highlight your menu choice then press
      [Enter], or press the key corresponding to the first letter
      of your menu choice.  Pressing the [Esc] key is the same as
      selecting the choice "Quit".

------------------------------------------------------------------------
Invalid argument (Q).
```

Screen 28. A syntax error was found while processing an Instruction file.

If no syntax errors are found, Host begins to execute the instructions. If the Instruction file is not suspended, processing begins in line one. If the Instruction file has been suspended and you have chosen to Resume processing the file, Host begins with the line following the last line executed before the the file was suspended. While the file is being processed, Host displays each instruction in the Menu Space as it is executed. Brief status messages may also be displayed to tell you what steps are being executed.

If Host encounters an error while executing an instruction, it displays a brief error message in the Error Message Space and a more detailed explanation in the Work Space. In addition, Host suspends the Instruction file at the instruction where the error occurred, then returns to the Instruct Menu. Screen 29 shows an example of the messages Host displays when it encounters an error executing an instruction. In this example, the error occurred while Host was executing line 1 of the Instruction file "ED2". The problem is that Host could not find the time dimension PERIOD in the Data File used by that instruction. You could correct this error by Adding a Dimension Definition for PERIOD to the Data File as described in Chapter 5, **DEFINING DATA**.

Any tables that have their output directed to the screen appear in the Work Space, just as they would if you had used the Tabulate command on Host's Main Menu or Table Menu. You may use [PgUp] and [PgDn] to page through displayed tables, or [Esc] to exit the displayed table and resume processing of the Instruction file.

Deleting an Instruction File

To delete an Instruction file, first select the Delete command from Host's Instruct Menu. Next, select the Host Instruction file you want to delete by entering an Instruction file name and optional directory path, or by using Host's Query Field feature. Host assumes you do not want to delete the default Instruction file, so there

```
Instruction file ED2 is currently suspended in line 1.          INSTRUCT MENU
Create Edit Backup Process Delete Quit
Create a new Instruction file.
------------------------------------------------------------------------------

                                  WARNING

    * A run-time error occurred while Host was processing the
      instruction in line 1 of the Instruction file ED2.

    * Refer to the Host Programmer Guide for help in identifying
      the cause of the error and correcting it.

    * Use the arrow keys to highlight your menu choice then press
      [Enter], or press the key corresponding to the first letter
      of your menu choice.  Pressing the [Esc] key is the same as
      selecting the choice "Quit".

------------------------------------------------------------------------------
Could not find dimension PERIOD.
```

Screen 29. A run-time error occurred while processing an Instruction file.

is no default Instruction file name entry for this field. If you do not precede the Instruction file name with a directory path, Host looks for the file in the default Instruction Directory. To prevent you from accidentally erasing files, Host requires that you confirm your selection before it will delete an instruction file.

CHAPTER 12
MANAGING FILES

MANAGING FILES

Host provides you with a variety of general file management functions. Figure 27 shows the command structure of Host's File Management facility. You gain access to these functions by selecting Manage from Host's Main Menu. From the resulting Manage Menu, you can choose to manage Host Data Files (Data) or Library Files (Library), or run one of Host's special utility programs (Utility). As with all the Host menus, Quit or [Esc] returns you to the previous menu.

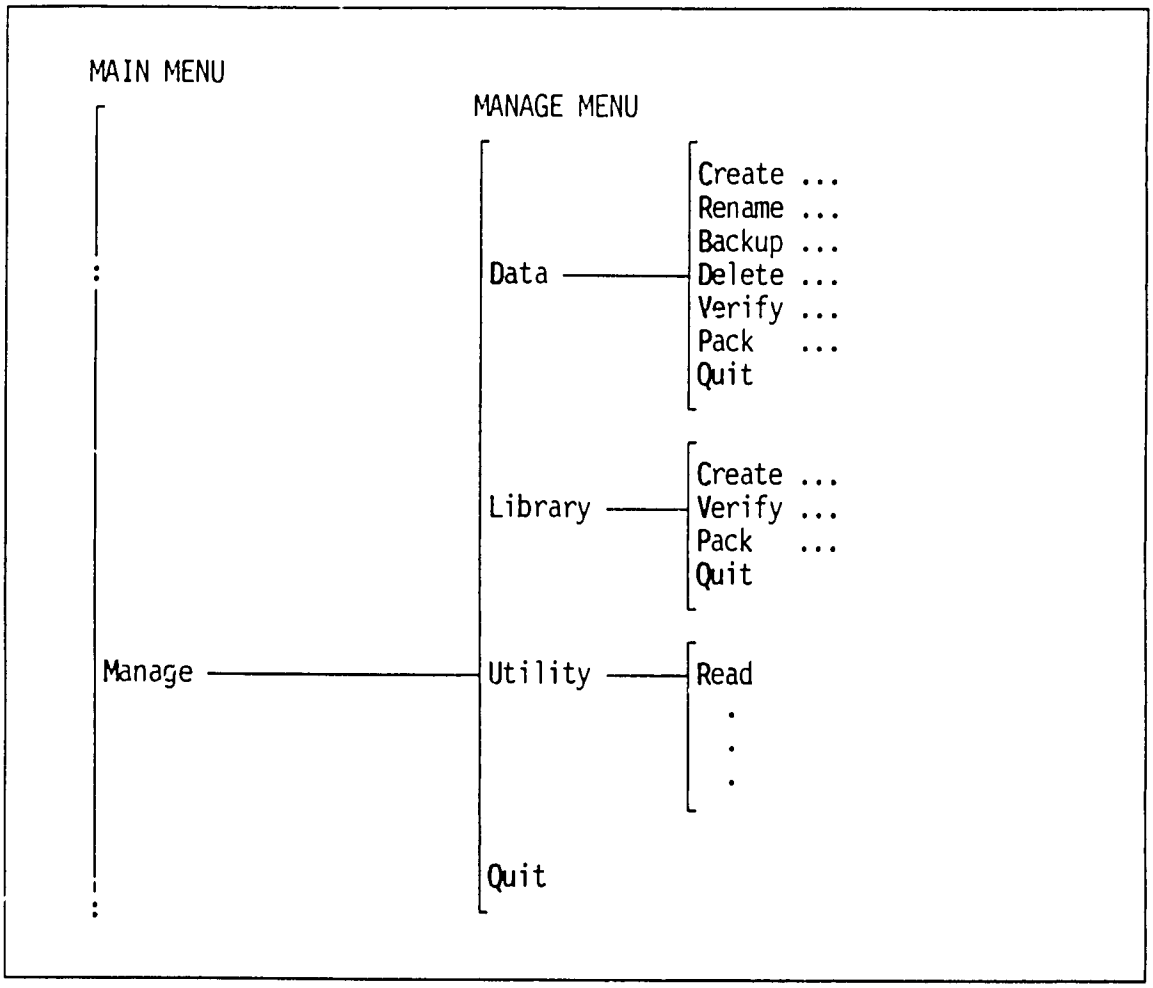


Figure 27. The command structure of Host's File Management facility.

Many of the commands in Host's File Management facility create a temporary or permanent Host Data File or Library, including Create, Backup, Verify, and Pack. If there is not enough storage space for these files on your disk, Host reports the problem and the approximate amount of disk space required. There are several ways to make more disk space available. You can direct the permanent or temporary Data File or Library to a disk drive that has more storage space by preceding its file name with a

disk drive designation. You can also free some space on the current disk by deleting some unnecessary files. To delete non-Host files or Host Libraries, return to Host's Main menu, use the *DOS or Quit command to gain access to PC/MS-DOS, then issue an ERASE or DEL command as described in your DOS manual. To delete unnecessary Host Data Files, return to the Manage Menu, select Data, then select Delete from the Manage Data Menu. To delete unnecessary Host Instruction files, return to Host's Main Menu, select the Instruct command, then select Delete from the Instruct Menu. Once there is enough free disk space, execute the appropriate File Management commands again to do the desired operation.

Managing Host Data Files

File management functions available for Host Data Files include

Create
Rename
Backup
Delete
Verify
Pack

All these functions, except Create of course, can be performed on any existing Host Data File. The following sections describe these functions in detail.

Creating a Host Data File

To create a new Host Data File, first choose the Create option on Host's Manage Data Menu, then type in the name of the Host Data File to create. If you do not precede the Data File name with a directory path, Host places the Data File in the default Data Directory. You should not supply a file name extension, since Host automatically supplies standard extensions. If there is no default Host Data File and you create a new Data File in the default Data Directory, the new Data File becomes the default, otherwise the default Data File remains unchanged.

Renaming a Host Data File

You can change the name of a Host Data File by choosing Rename from the Manage Data Menu and selecting a Data File to rename. You can type in a Data File name and an optional directory path, or select a Data File from those in the default Data Directory by using Host's Query Field feature. If you do not specify a directory path, Host looks for the Data File in the default Data Directory. Next Host asks you for the new Host Data File name. You can not supply a directory path at this point, since renaming a Data File can not move it from one directory to another. If there is already a Host Data File with the new name in the directory containing the file to be renamed, Host warns you and asks if you want to overwrite that Data File. If you rename the default Host Data File or Comparison File, Host automatically changes the corresponding default name in the Configuration File.

Backing Up a Host Data File

To make copies of important Host Data Files, choose the Backup option on Host's Manage Data Menu then select the Data File to be copied. The Data File to be copied is called the *source* file. Host initializes this entry to be the default Data File and you can accept this default by pressing [Enter]. If you want to copy any other Data File, type in a Data File name and an optional directory path or select a Data File from those in the default Data Directory by using Host's Query Field feature. If you do not supply a directory path, Host looks for the source file in the default Data Directory. Once you have selected the source file, enter the name of the file to receive the copy. This Data File is called the *target* file. The target file name can be any valid file name, and may be preceded by a directory path. As always, Data File extensions should not be specified, since they are supplied by Host. If no directory path is specified, the target file is created in the default Data Directory.

Since a file can not be copied onto itself, you can not give the target file the same name as the source file unless you are placing it on a different drive or in a different directory than the source file. If you inadvertently name the same file for both source and target, Host warns you of the error and allows you to choose a different target file name. You should not name an existing file as the target file unless you want Host to overwrite that file. If you name an existing file as the target file, Host warns you of the danger and requires you to confirm that you want to overwrite that file before it proceeds to make the copy.

Once the target file name has been entered, Host checks to see if there is enough disk space for the duplicate Data File. If there is not enough space, Host reports the problem and the approximate amount of disk space required, then asks you to enter a different target file name.

Deleting a Host Data File

To delete a Host Data File, first select the Delete command from Host's Manage Data Menu. Next, select the Host Data File you want to delete by entering a Data File name and optional directory path, or by using Host's Query Field feature. Host assumes you do not want to delete the default Data File, so there is no default Data File name entry for this field. If you do not precede the Data File name with a directory path, Host looks for the file in the default Data Directory. To prevent you from accidentally erasing files, Host requires that you confirm your selection before it will delete a Data File. If you delete the default Host Data File or Comparison File, Host resets that default to "NONE" in the Configuration File. You should then name a new default using the Configure command on Host's Main Menu.

Verifying a Host Data File

As mentioned in Chapter 3, **GETTING STARTED**, a Host Data File is really a database composed of six related files. Host can use a Data File only if the structure of this database is consistent and intact. Structural damage to a Data File can result from errors in the programming of an Application Module or from physical or magnetic damage to the disk surface. Such damage can also occur if a Host session is interrupted by a power failure or by you rebooting your microcomputer. The result is a Data File that produces error messages or strange and unpredictable results. If you suspect a Data File has been damaged, you can verify its structure using the Verify command on the Manage Data Menu. You may also choose to correct any errors encountered and to store

the corrected version of the Data File under a different name or in a different directory. Figure 28 summarizes the possible paths through Host's Data File verification facility.

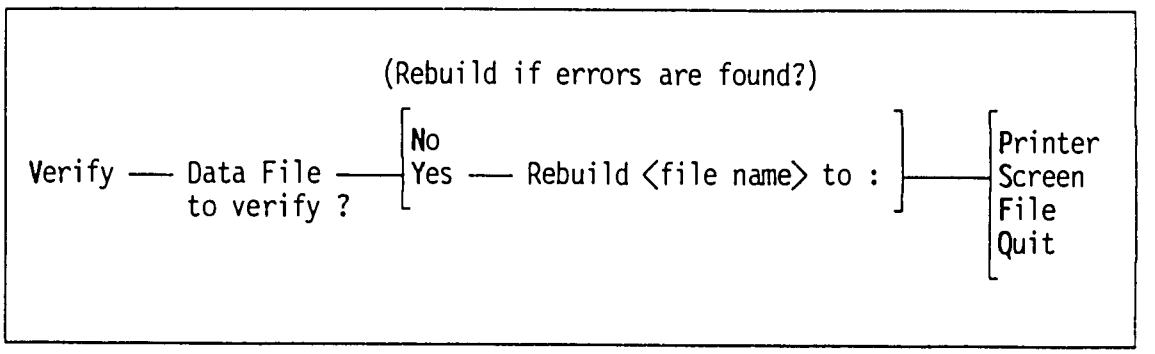


Figure 28. A summary of paths through Host's Data File verification facility.

To verify a Host Data File, first choose the Verify option on the Manage Data Menu, then select the Data File to be verified. Host initializes this entry to be the default Data File and you can accept this default by pressing [Enter]. If you want to verify some other Data File, type in a Data File name and optional directory path or select a Data File from those in the default Data Directory by using Host's Query Field feature.

After you have entered a Data File name, Host asks you whether you want to rebuild the Data File if any errors are found. If you answer "No", Host reports any errors encountered, but does not attempt to correct them. If you answer "Yes", you place Host's Verify facility in *Rebuild Mode*. In Rebuild Mode, Verify reports errors and trys to correct them while constructing a "rebuilt" version of the Data File. Host's Verify facility tests the structural integrity of a Data File in great detail, so, depending on the size of a Data File and the speed of your disk drive, verification may take a long time. Since rebuilding requires more time and disk space, you may want to verify without rebuilding just to check the integrity of a Data File.

If you choose to rebuild while verifying a Data File, Host asks you for the name to give the rebuilt version of the Data File. The default is to rebuild the Data File "in place", that is, to give the rebuilt Data File the same name as the original Data File. You can accept this default by pressing [Enter]. If you want to give the rebuilt version of the Data File a different name or place it in a different directory than the original Data File, change the default entry. Host places the rebuilt file in the default Data Directory if the Data File name is not preceded by a directory path.

If there is not enough disk space to rebuild the selected Data File, Host reports the problem and the approximate amount of disk space required. Host then gives you an opportunity to specify a different name for the rebuilt Data File.

Whether or not you rebuild while verifying a Data File, Host displays a message in the Work Space describing each error encountered. When you are rebuilding a Data File, Host also displays a message in the Work Space describing the action taken for each error. Before beginning to verify, Host displays a menu giving you the option to send these messages to the Screen only, to the screen and to the Printer, or to the

screen and to a File. If you choose Printer, Host will also print any messages on the printer according to the Printer Settings in the Configuration File. If you choose File, then each message displayed is also written to a file with the same name as the Data File being verified, but with the file name extension "VRF". Host places this file in the same directory as the Data File being verified.

After you answer the above preliminary questions, Host begins to verify and, if requested, rebuild the selected Data File. If Host discovers an error in the structure of the Data File, it displays a message describing the error. If you are rebuilding and verifying the Data File, Host corrects each problem encountered depending on the type of error and displays a message describing the action taken. Screen 30 is an example showing a Host Data File named "EMPLOY" being verified and rebuilt.

```
Choose the correct name for the Dimension.                          VERIFY
REGION2 REGION Neither Quit
Correct Dimension name is "REGION".
------------------------------------------------------------------------
05 Dec 86  VERIFY REPORT for EMPLOY                                Page  1
  1 ERROR : Dimension name is "AGE", index key is "AGE ".
    ACTION: Changed index key to "AGE".
  2 ERROR : Dimension name is "JSS", index key is "K#S".
    ACTION: Changed index key to "JSS".
  3 ERROR : Index entry for Dimension LAGES is wrong.
    ACTION: Location of LAGES corrected in index file.
  4 ERROR : Dimension name is "REGION2", index key is "REGION".
```

30. An example showing a Host Data File being verified and rebuilt.

ıen there is only one possible correction, Host corrects the error autc
rebuilt version of the Data Fiie. When there are two or more ways ı
ror, Host asks you to select which action should be taken from a L
Finally, there are some errors that Host can easily detect but can r
: in Rebuild Mode. For example, Host can not recreate missing Dime
e Definitions because it does not know what they contained; fortuna
:reate these definitions "manually" using Host's Dimension and Variabl
acilities. Regardless of the type of error, Host gives you a chanc
ng and rebuilding the file whenever an error is found. A detailed e:
h possible error and correction is included in Appendix F, VERIFY E
You should refer to this Appendix whenever you encounter errors while
Data File.

When Host finishes verifying a Data File, it tells you by displaying a message in the Message Space. If you chose the Rebuild Mode of Verify, Host then displays a message in the Message Space telling you it has finished rebuilding the Data File.

Regardless of whether any errors were found in the original Data File, Host constructs a "rebuilt" copy of the Data File. If you gave the rebuilt Data File the same name as the original Data File, Host asks your permission to replace the original Data File. If you answer "No", Host erases the rebuilt copy of the Data File and leaves the original Data File unchanged. Otherwise, Host erases the original Data File and replaces it with the rebuilt Data File. If you gave the rebuilt Data File a different name than the original or placed it on a different disk or in a different directory, you now have two versions: an original Data File and new rebuilt Data File.

Packing a Host Data File

When you delete Dimension Definitions, Variable Definitions, or data from a Host Data File, Host marks them as deleted, so that they are ignored in any further operations, but does not physically remove them from the disk. These deleted items occupy space that you might need for storing programs and data. If you want to recover this disk space, you must *pack* a Host Data File by removing items marked as deleted. Depending on the size and contents of a Data File and the speed of your disk drive, this process may take a long time, so be prepared wait while Host packs a Data File.

To pack a Host Data File, first choose the Pack option on the Manage Data Menu, then enter the name of a Data File. Host initializes this entry to be the default Data File and you can accept this default by pressing [Enter]. If you want to pack some other Data File, type in a Data File name and optional directory path or select a Data File from those in the default Data Directory by using Host's Query Field feature. If you do not precede the Data File name with a directory path, Host looks for the Data File in the default Data Directory.

Next, Host asks you for the name of the resulting packed Data File. The default is to pack the Data File "in place", that is, to give the packed Data File the same name as the original Data File. You can accept this default by pressing [Enter]. If you want to give the packed version of the Data File a different name or place it in a different directory than the original Data File, change the default entry. If you do not specify a directory path, Host places the packed file in the default Data Directory.

If there is not enough disk space to pack the selected Data File, Host reports the problem and an estimate of the amount of disk space required. Host then gives you an opportunity to specify a different name for the packed Data File.

If there is enough disk space, Host makes a copy of the selected Data File, removing all items marked as deleted. Host packs the Dimension Definition component of the Data File first. If no errors are detected in the Dimension Definitions, Host then packs the Variable Definition data components of the Data File. If Host discovers inconsistencies while packing, it warns you and recommends that you Verify the structure of the Data File before attempting to Pack it again.

When Host finishes packing a Data File, it tells you by displaying a message in the Message Space. If you gave the packed Data File the same name as the original Data File, Host asks you permission to replace the original Data File. If you answer "No", Host erases the packed version of the Data File and leaves the original Data File unchanged. Otherwise, Host erases the original Data File and replaces it with the packed Data File. If you gave the packed Data File a different name than the original, you now have two versions: an original unpacked Data File and new packed Data File.

If any items are marked as deleted, a packed Data File is smaller than the original Data File. Even if no items are marked as deleted, a packed Data File may be slightly smaller as the result of reorganization. In either case, a packed Data File contains the same active information as did the original Data File.

Managing Host Libraries

File management functions available for Libraries include:

Create
Verify
Pack

Functions to rename, back-up, and delete Libraries are not included. Since each Library is named by Host, only one Library of each type can exist in any given directory. You can copy a Library to another directory or disk drive by using the *DOS or Quit command on Host's Main Menu, then issuing a DOS COPY command as described in your DOS manual. Each Host Library consists of two files with the same name but separate extensions, "LIB" and "IDX", so be sure to copy both files. You will seldom want to delete an entire Library. If you do want to delete a Host Library, use the *DOS or Quit command on Host's Main Menu, then issue a DOS ERASE or DEL command as described in your DOS manual.

Creating a Library

To create a Library, first choose Create from the Manage Library Menu, then specify the Library you want to create. You can type in a Library name or select from a list of Libraries by using Host's Query Field feature. Host checks for the specified Library in the default Library Directory unless you specify some other directory path. If a Library with that name already exists, Host warns you and asks if you want to overwrite it. If you answer "No", Host returns to the Manage Library Menu without overwriting the Library. If you answer "Yes", or if no Library with that name exists, Host creates the Library and its index using standard file names and extensions.

Verifying a Library

As described in Chapter 3, **GETTING STARTED**, a Host Library is a database composed of two files, one containing items stored in the Library and the other containing an index to those items. Host can use a Library only if the references in the index file are consistent with the information stored in the item file. Damage to one or both of these files can result from physical or magnetic damage to the disk surface. Such damage can also occur if a Host session is interrupted by a power failure or by you rebooting your microcomputer. This may cause a Library to produce error messages or strange, unpredictable results. If you suspect that a Library is damaged, you can verify its structure using Host's Verify command on the Manage Library Menu. You may also choose to correct any errors encountered and to store the corrected version of the Library in a different directory or on a different disk. Host's Library verification facility is nearly identical to Host's Data File verification facility. Figure 29 summarizes the possible paths through Host's Library verification facility.

To verify a Host Library, first choose the Verify option on the Manage Library Menu, then select the Library to be verified. There is no default entry for this

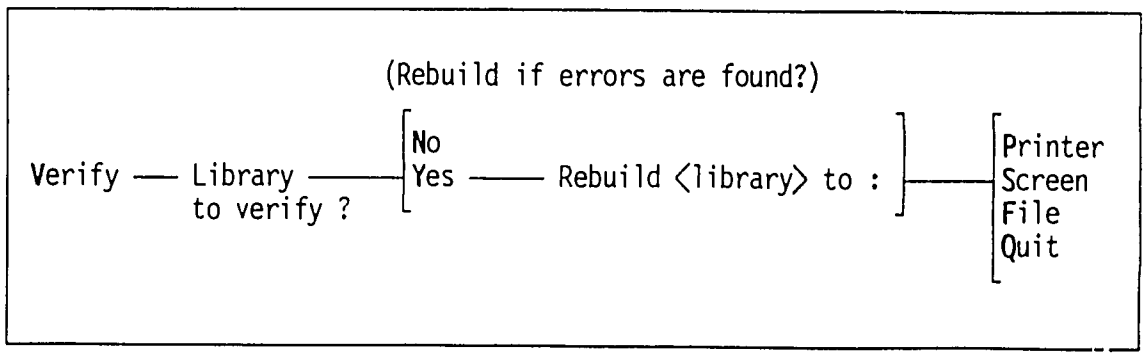


Figure 29. A summary of paths through the Library verification facility.

field. You can type in a Library name and an optional directory path, or select a Library from those in the default Library Directory by using Host's Query Field feature. If you do not specify a directory path, Host looks for the selected Library in the default Library Directory.

After you have entered a Library name, Host asks you whether you want to rebuild the Library if any errors are found. If you answer "No", Host reports any errors encountered, but does not attempt to correct them. If you answer "Yes", you place Host's Verify facility in Rebuild Mode. In Rebuild Mode, Verify reports errors and trys to correct them while constructing a "rebuilt" version of the Library. Host's Verify facility tests the structural integrity of a Library in great detail, so, depending on the size of a Library and the speed of your disk drive, verification may take a long time. Since rebuilding requires more time and disk space, you may want to verify without rebuilding just to check the integrity of a Library.

If you choose to rebuild while verifying a Library, Host asks you for the name to give the rebuilt version of the Library. The default is to rebuild the Library "in place", that is, to give the rebuilt Library the same name as the original Library. You can accept this default by pressing [Enter]. If you want to place the rebuilt version of the Library in a different directory or on a different disk than the original Library, change the default entry. Host places the rebuilt Library in the default Library Directory if the Library name is not preceded by a directory path.

If there is not enough disk space to rebuild the selected Library, Host reports the problem and the approximate amount of disk space required. Host then gives you an opportunity to specify a different disk drive or directory for the rebuilt Library.

Whether or not you rebuild while verifying a Library, Host displays a message in the Work Space describing each error encountered. When you are rebuilding a Library, Host also displays a message in the Work Space describing the action taken for each error. Before beginning to verify, Host displays a menu giving you the option to send these messages to the Screen only, to the screen and to the Printer, or to the screen and to a File. If you choose Printer, Host will also print any messages on the printer according to the Printer Settings in the Configuration File. If you choose File, then each message displayed is also written to a file with the same name as the Library being verified, but with the file name extension "VRF". Host places this file in the default Library Library directory.

After you answer the above preliminary questions, Host begins to verify and, if requested, rebuild the selected Library. If Host discovers an error in the structure of the Library, it displays a message describing the error. If you are rebuilding and verifying the Library, Host corrects each problem encountered depending on the type of error and displays a message describing the action taken. Screen 31 is an example showing a Host Library being verified and rebuilt.

```
Indicate whether to save or delete Label Format IBS.                    VERIFY
Save Delete Quit
Save Label Format IBS; create index entry.
------------------------------------------------------------------------------
85 Dec 86  VERIFY REPORT for LABEL library                          Page   1
  1 ERROR : Label Format name is "AGEBYFHV", index key is "AGEBYFIV".
    ACTION: Changed Label Format name to "AGEBYFIV".
  2 ERROR : Label Format INPUTS is marked as deleted, but has an index entry.
    ACTION: Label Format INPUTS saved.
  3 ERROR : Label Format name is "CHOICES", index key is "CHOYCES".
    ACTION: Changed index key to "CHOICES".
  4 ERROR : There is no index entry for Label Format "IBS".
```

Screen 31. An example showing a Host Library being verified and rebuilt.

When there is only one possible correction, Host corrects the error automatically in the rebuilt version of the Library. When there are two or more ways to correct the error, Host asks you to select which action should be taken from a Lotus-style menu. Finally, there are some errors that Host can easily detect but can not easily correct in Rebuild Mode. For example, Host can not recreate missing Label Formats or Table Definitions because it does not know what they contained; fortunately, you can recreate these definitions "manually" using Host's Label Format and Table Definition facilities. Regardless of the type of error, Host gives you a chance to stop verifying and rebuilding the Library whenever an error is found. A detailed explanation of each possible error and correction is included in **Appendix F**, VERIFY ERROR MESSAGES. You should refer to this Appendix whenever you encounter errors while verifying a Host Library.

When Host finishes verifying a Library, it tells you by displaying a message in the Message Space. If you chose the Rebuild Mode of **Verify**, Host then displays a message in the Message Space telling you it has finished rebuilding the Library. Regardless of whether any errors were found in the original Library, Host constructs a "rebuilt" copy of the Library. If you gave the rebuilt Library the same name as the original Library, Host asks your permission to replace the original Library. If you answer "No", Host erases the rebuilt copy of the Library and leaves the original Library unchanged. Otherwise, Host erases the original Library and replaces it with the rebuilt Library. If you placed the rebuilt Library on a different disk or in a

different directory, you now have two versions: an original Library and new rebuilt Library.

Packing a Library

When you delete Label Formats, Table Definitions, or Graph Definitions from a Host Library, Host marks them as deleted, so that they are ignored in any further operations, but does not physically remove them. These deleted items occupy space that you might need for storing programs and data. If you want to recover this disk space, you must *pack* a Library by removing items marked as deleted. Depending on the size and contents of a Library and the speed of your disk drive, this process may take a long time, so be prepared wait while Host packs a Library.

To pack a Host Library, first choose the Pack option on the Manage Library Menu, then select a Library. There is no default entry for this field. You can type in a Library name and optional directory path, or select a Library from those in the default Library Directory by using Host's Query Field feature. If you do not precede the Library name with a directory path, Host looks for the Library in the default Library Directory.

Next, Host asks you for the name of the resulting packed Library. You can not give the packed version of a Library a different name than the original Library, but you can place it in a different directory or on a different disk. The default is to pack the Library "in place", that is, to replace the existing Library with the packed Library. You can accept this default by pressing [Enter]. If you want to place the packed Library in a different directory or on a different disk than the original Library, change the default entry. If you do not specify a directory path, Host places the packed file in the default Library Directory.

If there is not enough disk space to pack the selected Library, Host reports the problem and an estimate of the amount of disk space required. Host then gives you an opportunity to specify a different disk drive or directory for the packed Library.

If there is enough disk space, Host makes a copy of the selected Library, removing all items marked as deleted. If Host discovers inconsistencies during this process, it warns you and recommends that you Verify the structure of the Library before attempting to Pack it again. When Host finishes packing a Library, it tells you by displaying a message in the Message Space. If you placed the packed Library on the same disk and in the same directory as the original Library, Host asks you permission to replace the original Library. If you answer "No", Host erases the packed version of the Library and leaves the original Library unchanged. Otherwise, Host erases the original Library and replaces it with the packed Library. If you placed the packed Library in a different directory or on a different disk than the original Library, you now have two versions: an original unpacked Library and new Packed Library.

If any items are marked as deleted, a packed Library is smaller than the original Library. Even if no items are marked as deleted, a packed Library may be slightly smaller as the result of reorganization. In either case, a packed Library contains the same active information as did the original Library.

Using the Management Utilities

Host's Utility Menu allows the Host support staff to add functions to Host without changing Host internally. Selections on this menu are Turbo Pascal programs written by the Host support staff to do a variety of tasks. For example, the Read utility included with this version of Host is a simple program allowing you to view or print the contents of any ASCII text file. There is no limit to the selections that may appear on the Utility Menu, and the choices will change as new utility programs are developed and old ones are modified.

Any program in Host's home directory, or in the DOS PATH, with a name of the form "$???????.CHN" or "$???????.COM" appears on this menu with the leading dollar sign and the file name extension removed. You can execute any program on this menu by selecting its name. When the program has completed its task, it returns to Host's Utility Menu.

Read

A Read utility program is included on Host's Utility Menu. It allows you to display or print the contents of any ASCII text file. To execute this program, select Utility from the Manage Menu, then choose Read. Next, the Read program prompts you for the name of the ASCII text file to be read. You can precede the name of this file with a directory path; otherwise, Read looks for the file in the current directory. Read then asks if you want to send the contents of the file to the Screen or the Printer. If you send the contents of the file to the screen, you can page forward or backward through the file using the [PgUp] and [PgDn] keys. When you have finished viewing or printing the contents of the file, press [Esc] to return to Host's Utility Menu.

ADVANCED TOPICS

Executing Host Instructions from DOS

Advanced Host users may want to run Application Modules, produce tables and even process entire Instruction files directly from DOS. The instructions available include:

RUN	run an Application Module
SUSPEND	suspend Instruction file processing
TABLE	generate a table
BASE	set the base year for the dimension PERIOD
INCR	set the increment between levels of dimension PERIOD
PERIOD	set the current levels of dimension PERIOD

These instructions can be executed from Host's Home Directory, or from any other directory if Host's Home Directory is in the DOS environment PATH (if you are not familiar with the DOS environment or the DOS PATH command, consult your operating system manual). The syntax for executing these instructions at the DOS prompt is identical to that used in Host Instruction files.

Using a Configuration File

When you execute a Host instruction from DOS, its behavior is affected by the presence or absence of a Host Configuration File. If there is a Configuration File in the current directory, the instruction uses the default file and directory settings contained in that file unless these settings are overridden by command-line parameters. Further, the TABLE instruction uses the default printer settings stored in the Configuration File. All syntax diagrams presented in the sections below assume a Configuration File is present in the current directory. If no Configuration File is present, there is no default Data File and all directories default to the current directory. Printer settings default to those appropriate for printing on 8½-inch wide by 11-inch long paper at 10 characters per inch and 6 lines per inch. Since there is no default Data File in the absence of a Host Configuration File, you must supply the D=<file name> argument to each of these instructions except SUSPEND. The file name must be the name of a Host Data File. You may precede the file name with a directory path, but you should not specify a file name extension.

The Error File HOSTERR.TXT

When you execute Host instructions from the DOS prompt or include them in a DOS "BAT" file, all error messages are written to a file named "HOSTERR.TXT". Host looks for this file in the current directory. If an error occurs and no such file exists, Host creates one and writes the error message to the new file. If the file "HOSTERR.TXT" does exist in your current directory, error messages are appended to the contents of the file. These error messages have the following format:

MM/DD/YY HH:MM:SS ProgramName ErrorMessage

where **MM/DD/YY** is the date the error occurred in the form month/day/year, **HH:MM:SS** is the time the error occurred in the form hour:minute:second, **ProgramName** is the name of the program in which the error occurred, and **ErrorMessage** is the error message describing the error that occurred. The line below is a typical HOSTERR.TXT entry.

06/26/86 16:50:39 RUN.COM Application Module name argument is required.

You can see what is in HOSTERR.TXT by using the DOS TYPE command as follows:

C:\>**type HOSTERR.TXT**[Enter]

If you are familiar with DOS, you may know several other commands that display or print the contents of a file. You may also view the contents of HOSTERR.TXT by using any standard text editing program.

There are two reasons for sending error messages to a file rather than to the screen when instructions are executed directly from DOS.

- Host's screen display does not appear when instructions are executed from DOS, so the arrangement of the display when an error occurs is unpredictable. There is no logical "Error Space" that can be used for all error messages.
- Error messages are preserved on disk so that you can refer to them at any time. This is particularly important when Host instructions are placed in a DOS "BAT" file that may be run unattended, since error messages displayed on the screen by one instruction might be erased by the next instruction.

Two types of errors are written to the HOSTERR.TXT file: *syntax errors* and *run-time errors*. If you omit a required parameter or specify parameters incorrectly in a Host instruction, you have made a syntax error. Run-time errors are errors encountered during the execution of a instruction. Both types of errors cause execution of the current instruction to halt. In addition to writing an error message to the file HOSTERR.TXT, the instruction encountering the error displays the following message

The file HOSTERR.TXT contains some messages.

Host Instruction Syntax

Each Host instruction has the following format

INSTRUCTION argument-specification ... argument-specification

where **INSTRUCTION** is the instruction name, and **argument-specification** has the form **argument=<value>**, with no embedded spaces. The angle brackets ("<"...">") are used to delimit the **value** in this notation and are not part of the instruction. If an argument specification is enclosed in square brackets ("["..."]"), the argument is optional, otherwise the argument is required. The order of argument specifications in an in-

struction line is not important. One or more spaces are used to separate an instruction and its argument specifications, but an instruction may not be divided across two or more lines. Instructions may be in any combination of upper and lower case characters.

Running Application Modules from DOS

There are two ways to run an Application Module directly from DOS. For example you could run the POPULATE Application Module by entering the following command:

C:\>**run m=populate s=1 n=5**[Enter]

The formal syntax for this method is as follows:

RUN M=<module> [D=<data file>] [S=<starting period>] [N=<number of periods>]

You could also run the POPULATE Application Module by entering the following abbreviated command:

C:\>**populate s=1 n=5**[Enter]

The formal syntax for the abbreviated method is as follows:

<**module**> [D=<data file>] [S=<starting period>] [N=<number of periods>]

The arguments and values for both methods are explained below.

module The name of a Host Application Module. When the first syntax is used, this is a required argument. If an extension is supplied, it is used, and otherwise the extension "COM" is assumed. If a path is desired, it must be supplied. Otherwise the Application Module is assumed to reside in the default Module Directory. When the second syntax is used, **module** is the name of the program DOS is to execute. Here, module must be in the current directory or in the DOS environment PATH.

D= Host Data File Name. This argument is required if no Configuration File exists in the current directory.

No file name extensions are allowed. If a path is desired, it must be supplied, otherwise the Host Data File is assumed to be in the default Data Directory.

S= Period at which to begin projecting. This is an optional argument, and defaults to a value of 1. Most Application Modules expect initial data to be stored in period 1, so the first time an Application Module is run, this argument should usually be 1 (S=1). Consult the documentation provided with an Application Module before using a value other than 1.

N= Number of periods to project. This is an optional argument, and its default is 1. Consult the documentation provided with the Application Module being run to determine what values of this argument are reasonable.

The first syntax corresponds exactly to that recognized by the Host Instruction File Processor. The second abbreviated syntax is slightly simpler. Table 23 summarizes the default files and directories used by the RUN instruction.

TABLE 23.

Default files and directories used by the RUN command.

File or Directory	Default Value if	
	Configuration File Present	No Configuration File Present
Host Data File	Default Data File	NONE: User must supply D=file name parameter
Directory location of Data File	Default Data Directory	Current Directory
Application Module	NONE: User must supply M=module parameter	NONE: User must supply M=module parameter
Directory location of Application Module	Default Program Directory	Current Directory

Producing Tables from DOS

The syntax for producing a table from DOS is

TABLE T=<table> [D=<data file>] [C=<comp data file>] [O=<output>] [P=<period list>]

where

T= Table Definition name. This is a required argument and the Table Definition named must be in the Table Library.

D= Host Data File name. This argument is required if there is no Configuration file in the current directory.

No file name extensions are allowed. If a path is desired, it must be supplied, otherwise the Host Data File is assumed to be in the default Data Directory.

C= Comparison File name. If the Table Definition named is a Comparison table and there is no Configuration File in the current directory, this argument is required. The same rules for extensions and paths stated for the D= argument apply to this argument as well.

O= Output file name for table. This is an optional argument, and defaults to the output device named in the Configuration File if one exists in the current directory. Otherwise the output file default to "PRN".

If the file is the printer ("PRN") or console ("CON"), then the table is directed to the appropriate device. Otherwise a file of the given name is created and the table is written to that file. If a file of the given name

already exists, it is overwritten.

A file extension of "PRN" designates creation of a file that can be imported into Lotus 1-2-3 (/File, Import, Numbers) for further analysis or for graphing. Any other extension specifies creation of a disk file containing exactly the same information as would have been sent to the printer if output had been directed to "PRN" (printer). See Chapter 10, **GENERATING TABLES**, for a complete discussion of Lotus-import and Print-ready files.

Output is directed to the default Data Directory unless another directory path is supplied.

P= Period list. This is an optional argument listing the periods of data to include in the table. The default value is "ALL". A complete discussion of period list syntax is given in Chapter 10, **GENERATING TABLES**.

Table 24 summarizes the default files and directories used by the TABLE instruction.

TABLE 24.

Default files and directories used by the TABLE command.

File or Directory	Default Value if	
	Configuration File Present	No Configuration File Present
Host Data File Comparison File	Default Data File Default Comparison File	NONE: User must supply D=file name parameter, C=file name parameter
Directory location of Data & Comparison Files	Default Data Directory	Current Directory
Output	Default Output Device	PRN (printer)
Directory location of File Output	Default Data Directory	Current Directory

Executing the BASE Instruction from DOS

The syntax for executing a BASE instruction from DOS is

BASE B=<base year> [D=<data file>]

where

B= Base year value of time dimension PERIOD.

D= Host Data File name. This argument is required if there is no Configuration File in the current directory.

Executing the INCR Instruction from DOS

The syntax for executing a INCR instruction from DOS is

INCR I=<increment> [D=<data file>]

where

I= Increment Between Levels of the time dimension PERIOD.

D= Host Data File name. This argument is required if there is no Configuration File in the current directory.

Executing the PERIOD Instruction from DOS

The syntax for executing a PERIOD instruction from DOS is

PERIOD L=<level> [D=<data file>]

where

L= Current Number of Levels of the time dimension PERIOD.

D= Host Data File name. This argument is required if there is no Configuration File in the current directory.

Executing the SUSPEND instruction from DOS

The syntax for the SUSPEND instruction is

SUSPEND

Processing Instruction Files from DOS

Any Host Instruction file can be processed from DOS by simply renaming the file extension from "INS" to "BAT" and typing the file name at the DOS prompt. In addition, DOS batch file commands can be added to the file when it is executed in this mode. Finally, parameters may be specified at the DOS command line by using the DOS replaceable parameter (%) facility within the batch file.

For example, the Instruction file ED1.INS in Figure 30 could be renamed ED1.BAT and run from the DOS command line by typing **ED1**. Note that the Host Instruction File Processor automatically changes the directory back to the one from which Host has been executed before processing Host instructions. The DOS batch file interpreter does not

do this, so you need to insert DOS CD (Change Directory) commands into your "BAT" file to return to the current directory before each Host instruction following a directory change. The "**cd \host**" command following each "lotus" command in Figure 30 below shows how you can force the DOS batch file interpreter to return to Host's Home Directory.

```
rem     run the POPULATE Application Module
run     m=populate d=eddata1 s=1 n=5
suspend
rem     set the base and increment before producing tables
base    b=1980 d=eddata1
incr    i=1 d=eddata1
rem     produce the table POP1 in a Lotus-import file
table   t=pop1 d=eddata1 o=\lotus\pop1.prn p=all
rem     execute DOS CD command to get into the Lotus directory, then
rem     run Lotus 1-2-3
cd \lotus
lotus
rem     for "BAT" file only: change directory back to Host's Home
rem     Directory
cd \host
rem     run the NEDUC Application Module
run     m=neduc d=eddata1 s=1 n=5
suspend
rem     produce the table PRIM in a Lotus-import file
table   t=prim d=eddata1 o=\lotus\primtbl.prn p=all
rem     execute DOS CD command to get into the Lotus directory, then
rem     run Lotus 1-2-3
cd \lotus
lotus
rem     for "BAT" file only: change directory back to Host's Home
rem     Directory
cd \host
```

Figure 30. An Instruction file modified to run as a "BAT" file from DOS.

To avoid having the name of the Host Data File specified in the batch file, all occurrences of "d=eddata1" could be replaced with "d=%1". The modified Instruction file, renamed "ED1.BAT" is shown in Figure 31. The instructions can now be executed for the Data File "EDDATA1" by typing the following command at the DOS prompt:

C:\>**ED1 EDDATA1**[Enter]

and you can execute the same set of instructions for the Data File "EDDATA2" by typing the following command at the DOS prompt:

C:\>**ED1 EDDATA2**[Enter]

```
rem     run the POPULATE Application Module
run     m=populate d=%1 s=1 n=5
suspend
rem     set the base and increment before producing tables
base    b=1980 d=%1
incr    i=1 d=%1
rem     produce the table POP1 in a Lotus-import file
table   t=pop1 d=eddata1 o=\lotus\pop1.prn p=all
rem     execute DOS CD command to get into the Lotus directory, then
rem     run Lotus 1-2-3
cd \lotus
lotus
rem     for "BAT" file only: change directory back to Host's Home
rem     Directory
cd \host
rem     run the NEDUC Application Module
run     m=neduc d=%1 s=1 n=5
suspend
rem     produce the table PRIM in a Lotus-import file
table   t=prim d=eddata1 o=\lotus\primtbl.prn p=all
rem     execute DOS CD command to get into the Lotus directory, then
rem     run Lotus 1-2-3
cd \lotus
lotus
rem     for "BAT" file only: change directory back to Host's Home
rem     Directory
cd \host
```

Figure 31. An Instruction file using DOS replaceable parameters.

Warnings

The Host Instruction File Processor can not handle the full range of DOS batch file commands. Those not supported by the Host Instruction File Processor must be removed before you rename a "BAT" file to an "INS" file. Chapter 11, **INSTRUCTION FILES** lists the DOS batch file commands that are not supported by Host. Also, because the Host instructions disable the [Ctrl]+[Break] facility during execution, you cannot stop a batch file during execution of these instructions. Attempting to do so may cause the DOS batch file interpreter to go into an infinite loop, requiring you to warm boot your system.

Using Host with Multiple Directories

Host is designed to allow you to take advantage of the DOS tree-structured directory system. Host Data files, Application Modules, Library files, and Instruction files can all be kept in separate disk directories or subdirectories. Default settings for these directories can be stored in Host Configuration Files. See Chapter 4, **CONFIGURING HOST**, for details on the use of default directories in Host.

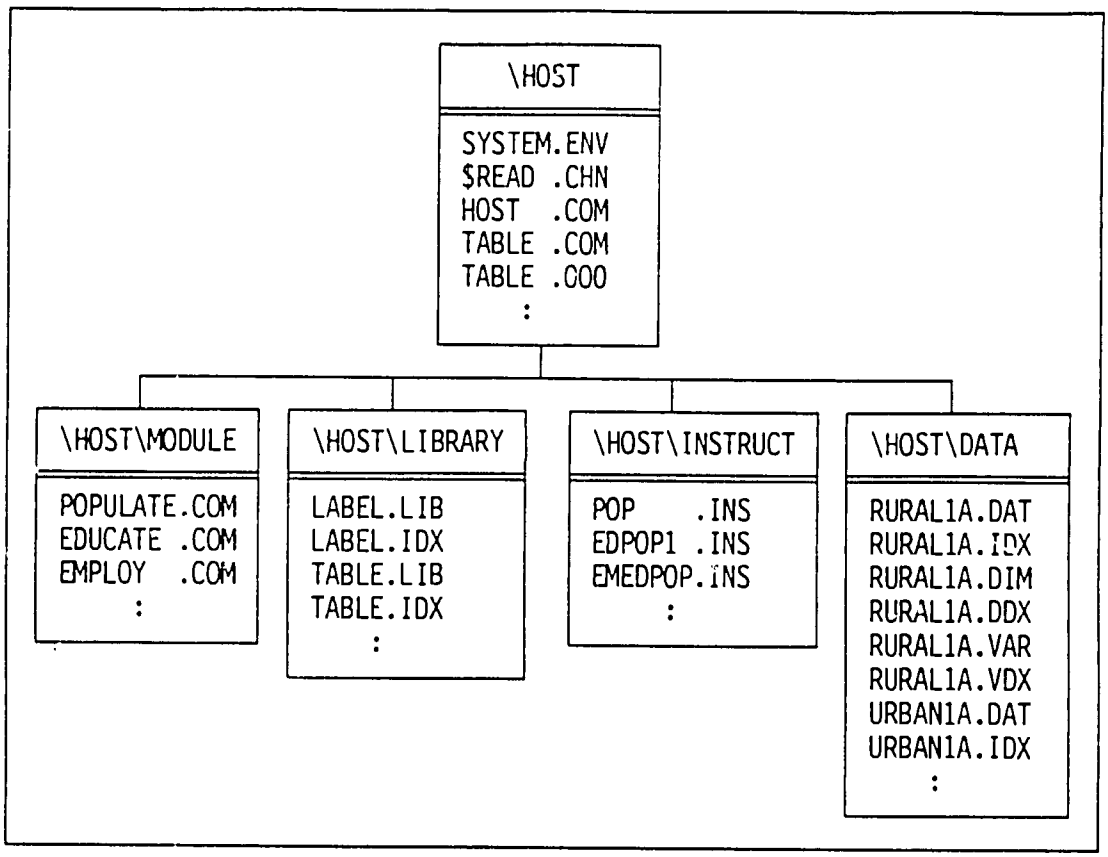


Figure 32. A possible multiple-directory configuration for use with Host.

Figure 32 shows an example multiple-directory configuration that could be used with Host. In this example, all Application Modules are in a subdirectory named "MODULE", Libraries are in a second subdirectory named "LIBRARY", Instruction files are in a third subdirectory named "INSTRUCT", and Host Data Files are in a fourth subdirectory named "DATA". The Configuration File is in Host's Home Directory. In this example you would execute Host from its Home Directory. Screen 32 shows the corresponding Host configuration.

```
                                                                    CONFIGURE
Files Output Directories Printer Monitor Save Quit
Edit printer settings.
------------------------------------------------------------------------------
FILES                      PRINTER SETTINGS        MONITOR SETTING

Data        : RURAL1A      Left Margin   : 18      Monochrome : Yes
Comparison  : None         Top Margin    :  6
Instruction : EDPOP1       Bottom Margin :  6
                           Page Length   : 66
OUTPUT DEVICES             Line Length   : 65

Table : Printer            Page Eject    : Yes
Graph : Plotter            Setup string  : None

DIRECTORIES

Data        : \HOST\DATA\
Module      : \HOST\MODULE\
Library     : \HOST\LIBRARY\
Instruction : \HOST\INSTRUCT\

------------------------------------------------------------------------------
```

Screen 32. Configuration settings for multiple-directory system in figure 32.

Figure 33 shows another example multiple-directory configuration that could be used with Host. In this example, Host Data Files for several subnational Projection Modules are placed in separate subdirectories. Each of these subdirectories contains data for one region of the country. The common Application Modules and Libraries reside in subdirectories shared by the regional projections. Each subdirectory containing regional Data Files contains its own Configuration File. In each Configuration File, the default Data Directory and the default Instruction Directory are set to the current subdirectory. The default Module Directory is "\HOST\MODULE" and the default Library Directory is "\HOST\LIBRARY" in all Configuration Files. To work with data for a specific region, you would use the DOS CD (Change Directory) command to change to that subdirectory, then run Host by typing "**host**" at the DOS prompt. Host reads the Configuration File in the current directory—the directory for the appropriate region—and uses the default directory settings it contains. Remember that Host's Home Directory must be in the DOS environment PATH in order for you to execute Host from the regional subdirectories.

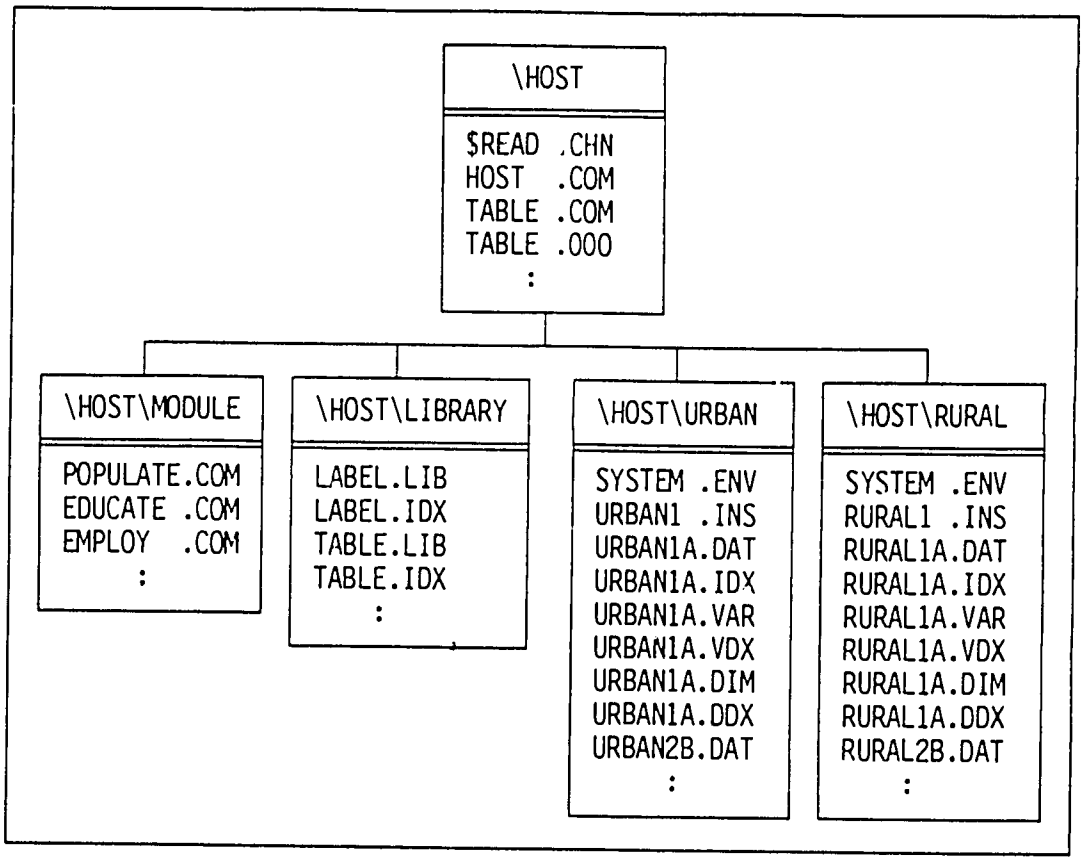


Figure 33. A multiple-directory configuration with regional subdirectories.

APPENDIX A.

APPENDIX A.

HOST SYSTEM FILES

Name	Extension	Function
BASE	COM	Responsible for interpreting and executing the BASE instruction when that instruction is executed from DOS (See Executing Host Instructions from DOS, page 13-2).
CONTASKS	CHN	Provides facility for setting the Host's configuration values and saving those values to the Configuration File. Corresponds to the Configure selection on Host's Main Menu.
DIMTASKS DIMTASKS DIMTASKS	CHN 000 001	Provides facilities for adding, editing, deleting, listing and reporting the dimension attribute definitions in a Host Data File. Corresponds to the Dimensions selection on Host's Describe Menu.
DUMMY	COM	A utility program executable from DOS. Removes all executable statements from any Turbo Pascal source code file. Intended for use by Application Module developers as a means to improve the compilation time of an Application Module while simply checking the module's syntax.
EDTTASKS EDTTASKS EDTTASKS EDTTASKS	CHN 000 001 002	Provides facilities to assign and edit data values in a Host Data File. Corresponds to the Edit selection on Host's Main Menu.
FMTTASKS FMTTASKS FMTTASKS FMTTASKS	CHN 000 001 002	Provides facilities to add, edit, delete, list and report Label Formats in a Label Library. Corresponds to the Labels selection on Host's Describe Menu.
GENMOD	COM	A utility program executable from DOS. Generates a Turbo Pascal source code file containing requested constant, type and variable definitions and conforming to the structure required by Host for Application Modules. Intended for use by Application Module developers to accelerate the process of module development while reducing errors.
HOST	COM	The main program of the Host System. Controls most other programs in the System and corresponds to Host's

		Main Menu.
INCR	COM	Responsible for interpreting and executing the INCR instruction when that instruction is executed from DOS (see Executing Host Instructions from DOS, page 13-2).
INSTASKS INSTASKS INSTASKS INSTASKS	CHN 000 001 002	Provides facilities for creating, backing up, processing, editing and deleting Instruction Files. Corresponds to the Instruct selection on Host's Main Menu.
MANTASKS MANTASKS MANTASKS	CHN 000 001	Provides facilities for creating, backing up, deleting, renaming, verifying and packing Host Data Files and libraries. Also provides access to certain Host utility programs. Corresponds to the Manage selection on Host's Main Menu.
PERIOD	COM	Responsible for interpreting and executing the PERIOD instruction when that instruction is executed from DOS (see Executing Host Instructions from DOS, page 13-2).
PROTASKS	COM	Called to process Instruction Files from within Host. Corresponds to the Process selection under Instruct on Host's Main Menu.
RUN	COM	Responsible for interpreting and executing the RUN instruction when that instruction is executed from DOS (see Executing Host Instructions from DOS, page 13-2).
RUNTASKS	CHN	Provides a facility for running one Application Module at a time from within Host. Corresponds to the Run selection on Host's Main Menu.
SUSPEND	COM	Responsible for interpreting and executing the SUSPEND instruction when that instruction is executed from DOS (see Executing Host Instructions from DOS, page 13-2).
TABLE TABLE TABLE TABLE	COM 000 001 002	Called to produce a table by combining a Table Definition with a specific Host Data File. Corresponds to the Go selection under Tabulate on Host's Main Menu and under the Tabulate selection of Host's Table Menu.
TABTASKS	CHN	Provides a facility for generating one table at a time from within Host. Corresponds to the Tabulate selection on Host's Main Menu and the Tabulate selection on Host's Table Menu.
TBLTASKS TBLTASKS TBLTASKS TBLTASKS	CHN 000 001 002	Provides a facility for adding, editing, deleting, listing, reporting table table definitions, as well as generating tables. Corresponds to the Tables selection on Host's Describe Menu.
UPGRADE	COM	A utility program to upgrade Host Configuration and Instruction files from Version 1.01 to Version 2.00 of Host.

VARTASKS	CHN	Provides facilities for adding, editing, deleting, listing and reporting the variable attribute definitions in a Host Data File. Corresponds to the Variables selection on Host's Describe Menu.
VARTASKS	000	
VARTASKS	001	
VARTASKS	002	
VARTASKS	003	
VERIFY	CHN	Provides a facility for verifying and rebuilding Host Libraries and Data Files. Corresponds to the Verify selections on Host's Manage Data and Manage Library Menus.
VERIFY	000	
VERIFY	001	

Dollar Utility Files

Host's Utilities Menu (located under the Manage selection on Host's Main Menu) searches the disk directory in which the DOS operating system located HOST for files with names beginning with "$" and extensions of "CHN" or "COM". Host adds the names of all such files to the Utilities Menu (First stripping the "$" prefix and file name extension, then down-casing the remaining characters and capitalizing the first character of the resulting string).All such Dollar Utility programs have disk file names following the pattern $???????.CHN, or $???????.COM where "???????" is the Utilities Menu selection corresponding to that program.

When a utility is selected from the menu, Host passess control to the selected utility program.It is the responsibility of the utility program, when it terminates, to return control to Host.This facility permits the Host Support Staff to provide additional functions to Host without making internal changes to the Host System.

Name	Extension	Contents
$READ	CHN	A Host Dollar utility program which provides a facility for viewing and printing any ASCII text file.

Application Module Include Files

When an Application Module is developed, certain support procedures and functions must be added to the module for it to perform correctly under Host. These procedures and functions are provided in the form of Turbo Pascal source code files which must be included into the module by using the Turbo Pascal Include compiler directive.The contents and usage of these files are described in detail in the Host Module Developer's Guide (currently being drafted).If you do not intend to develop new Application Modules, you do not need these files.

Name	Extension	Contents
HOST1	BOX	Host specific support procedures and functions t included in Application Modules.
HOST2	BOX	
ACCESS3	BOX	ISAM B+ Tree file management facilities from the International Turbo Toolbox. These procedures a functions have been modified for use with Host.
ADDKEY	BOX	
GETKEY	BOX	

* If you do not intend to execute Application Modules or generate tables or process Instruction Files from the DOS operating system, the following files are not necessary:

BASE	COM
INCR	COM
PERIOD	COM
RUN	COM
SUSPEND	COM

* If you do not intend to develop new Application Modules, the following files are not necessary:

DUMMY	COM
GENMOD	COM
HOST1	BOX
ACCESS3	BOX
ADDKEY	BOX
GETKEY	BOX
HOST2	BOX

* Once you have upgraded your Configuration and Instruction files using the UPGRADE utility, the file UPGRADE.COM is no longer needed.

APPENDIX B.

APPENDIX B.

USING HOST WITH FLOPPY DISKS AND RAM DISKS

Although the design of Host assumes the availability of a fast, large capacity storage device, such equipment may not always be available. For example, you may want to apply a model in the field using a transportable or lap-top microcomputer. Though it is increasingly common for such machines to have a 10 megabyte or larger fixed hard disk, they are still not readily available. Current lap-top microcomputers typically have one or two floppy disk drives, each with 360K to 800K of storage capacity. Current transportable computers nearly always have two 360K floppy disk drives. Some of these machines may also have sufficient RAM to allocate to a RAM disk.

In any case, it is not likely that you will have sufficient disk storage space on these types of microcomputers to place all the Host program files on a single disk. Therefore, you will be forced to place a subset of the Host program files on each of several disks. The problem is how to divide these program files among several disks such that Host operates smoothly and correctly, even though it operates very slowly. Since we can not make specific recommendations for all possible hardware arrangements, you need to understand a few things about how Host functions. The following sections, along with information in **Appendix A**, HOST SYSTEM FILES, should help you arrange Host's program files for your specific hardware configuration.

Many Host commands require that a Host program file be read into memory from disk before the command can be executed. When this occurs, what happens depends on which version of the PC/MS-DOS operating system you are using. If you are using DOS version 2.00 or higher, Host looks for the required program file in Host's Home Directory, that is, the disk directory from which the operating system loaded the main "HOST.COM" program file. In earlier versions of DOS, this information is not available, so Host will expect to find the required program file in the current disk directory. Some software utilities are available that correct this deficiency in these versions of PC/MS-DOS.

If Host fails to find a required program file, it will issue an error message similar to the following:

Can not find DIMTASKS.CHN in A:\

where "DIMTASKS.CHN" is the missing program module that Host is expecting to find in the directory "A:\". Sometimes, you may be able to return to the operating system from Host to correct the situation. In other cases, Host may not be able to continue without the missing program file and you must restart your microcomputer to regain control.

In addition, nearly all Host program files must have access to the Configuration File "SYSTEM.ENV". When each program file is loaded from disk, it searches the current directory for this file. If it can not be found, the program module will immediately return control to the operating system.

It is possible to divide Host among several separate removable disks, (or a removable disk and a RAM disk) so that none of the problems described above occur. For a two floppy disk drive microcomputer, the procedure is described in the following paragraphs.

First, make a Host "Master Disk" by copying the following files onto a newly formatted disk in drive A:

HOST.COM
CONTASKS.CHN

Next, place another formatted disk in drive "B". If these files are available, copy the Host Label Library, Table Library, and Graph Library that you want to use to the disk in drive "B". If want to use any existing Host Instruction files or Data Files, copy those to the disk in drive "B" also.

After copying any Host Libraries, Instruction files, or Data Files to drive "B", first make sure that you place the Host "Master Disk" in drive "A", then make drive "A" the current drive. You should now be able to startup Host by typing "**host**" at the operating system prompt. When Host fails to find the Configuration File "SYSTEM.ENV" in drive "A", it will create one containing preset default settings.

Next, press any key to pass through Host's title screen and request User access by pressing [Enter]. Select Configure from Host's Main Menu, then change the default Data Directory, Module Directory, and Library Directory to "B:\". This assumes that you will place the Libraries, Instruction files, and Data Files you want to use on a disk inserted into the "B" drive. Do not set the default Data File, Comparison File, or Instruction File until you have placed a disk containing these files in drive "B". While you are in Configure, change the values of the printer and display settings to match your hardware, then save this configuration and Quit Host.

Now copy the "HOST.COM" and "SYSTEM.ENV" files from the disk in drive "A to several other formatted disks that you will use to store other Host program files. Copy groups of Host program files onto these disks according to function by referring to the information in Appendix A. Label each disk so that you know what program files it contains. For example, you could make a disk for defining and generating tables by copying the following files onto it:

TABTASKS.CHN
TBLTASKS.CHN
TBLTASKS.000
TBLTASKS.001
TBLTASKS.002
TABLE.COM
TABLE.000
TABLE.001
TABLE.002

You should then label this disk "Defining and Generating Tables", or something similar. You could make a disk for executing projection modules, editing data, and generating tables by copying the following files onto it:

RUNTASKS.CHN
TABTASKS.CHN
TABLE.COM
TABLE.000
TABLE.001
TABLE.002

The appropriate division of Host program files among the disks depends on what you want to do and the storage capacity of each disk.

Now, after entering Host, you must make sure that the appropriate Host program disk is in drive "A" before executing any command. If, when attempting to execute some command, you receive a message stating that a required Host program file can not be found, just insert the disk containing that file into drive "A" and repeat the command. Notice that this arrangement means changes to the Configuration File on one disk are not read by program files on other disks unless the altered "SYSTEM.ENV" file is recopied to those disks.

APPENDIX C.

APPENDIX C.

KNOWN BUGS

bug : an unexpected defect, fault, flaw, or imperfection.

— *Webster's New Collegiate Dictionary*, 1980

Bugs Corrected in Host Version 2.00

The following is a list of the bugs in Host Version 1.01 that have been detected by or reported to the Host support staff. All these bugs have been corrected in Host Version 2.00.

- In tables having two or more unnested dimensions, the unnested totals were sometimes wrong or missing.

- In tables having PERIOD as the row dimension, data in the current level of PERIOD sometimes were not displayed if there were no data in the previous level of PERIOD. Conversely, zeros or spurious values sometimes were displayed when there were no data in the current level of PERIOD, but there were data in a previous level of PERIOD.

- The Table Definition Editor sometimes allowed a column dimension to be assigned when there were two or more variables in the table variable list and sometimes refused to allow assignment of a column dimension when there was only one variable in the table variable list. This happened when variables were added to a table variable list that had only one variable, or when all variables but one were removed from a table variable list that had more than one variable.

- A run-time error FF (Heap-Stack Collision) occurred when you left the Instruct Menu after encountering a fatal error while processing a RUN command.

- When entering the names of Host Data Files and Instruction files it was possible to specify a directory path without specifying a file name. When this occurred, Host displayed the following error message:

  Unable to find in <directory path>.

  instead of the following, more appropriate error message:

  No file name.

Known Bugs in Version 2.00

As of this writing, the Host support staff know of no bugs in the Host System. Of course we know some bugs must remain; we just haven't found them.

150

APPENDIX D

APPENDIX D.

PLANNED ENHANCEMENTS TO HOST

Significant enhancements are planned for future versions of Host. Though there are many improvements being considered, only some of these are listed below.

Internal Graph Definition and Generation Facility

Although any Application Module can be programmed to display data in graph form and tables can be imported into Lotus 1-2-3 for graphing, Host was designed to have an internal graphing facility. Building graphing directly into Host will allow efficient cycling through making assumptions, running projections, and analyzing results, and will make maximum use of the data definition and labeling information available in the Host database. We are preparing a prototype Host graph definition and generation facility analogous to the existing table definition and generation facility. We also hope to add a Host instruction that can import graphics images produced by other software packages. This capability, when used in Host Instruction files, will make it possible to present automatic graphic "slide-shows" of any data stored in a Host Data File.

Advanced Data Editor Functions

Also designed, but unimplemented in the current version of Host, are some advanced functions in the Data Editor. These include the following:

Area Functions

- Copying a selected area of data from one location to another within a variable.
- Filling any area of data within a variable with a specified constant value.
- Scaling any area of data within a variable (dividing, multiplying, adding or subtracting) by a constant value.

Period Functions

- Copying one period of data to another period within a variable.
- Filling a period of data within a variable with a specified constant value.

- Scaling any period of data within a variable (dividing, multiplying, adding or subtracting) by a constant value.
- Deleting any period of data within a variable from the Host Data File.

Interpolation

- Interpolating over a dimension within any subspace of a multidimensional variable.

Data Import Facility

Host needs a way to import data already in machine readable form into variables stored in a Host Data File. Host's ability to handle multi-dimensional data complicates this problem, but it can be solved. We plan to develop a utility that can recall data stored in several popular formats (ASCII, DIF, dBASE DBF and Lotus 1-2-3 WKS) and store those data in Host Data Files.

Advanced Module Support Routines

Numerous Turbo Pascal routines have been developed by the Host programming staff that would be useful to Host users writing certain types of modules. We plan to make some of these available in the future as additional HOST?.BOX source code include files. The first two of these will contain routines useful for developing highly interactive modules and graphics display modules. Example modules using these routines will also be provided.

Concise Dimension and Variable Report Formats

The report formats currently used by Host for Dimension and Variable Definitions include all possible attributes. This is fine for some purposes, but includes more information than necessary to document the structure of dimensions and variables used by an Application Module. We plan to provide an alternative format that lists only structural attributes in a concise tabular format for use in documenting Application Modules.

Application Module Access to Multiple Host Data Files

Though our original Host design provided an Application Module access to a minimum of two Host Data Files, circumstances have lead us to concentrate on access to a single Data File. Our original Host design also includes indirect references to variables in other Host Data Files in a way completely transparent to Host Application Modules. These two unimplemented capabilities would add great flexibility to Host-based applications and we plan to complete them as soon as possible. Until then, though this capability is not directly supported by Host, the Host Programmer Guide shows how easy it is for a module to gain access to more than one Data File.

APPENDIX E.

APPENDIX E.

RUNNING HOST ON THE WANG PROFESSIONAL COMPUTER

To run Host on the Wang Professional Computer you must have both an Intel 8087-2 numeric coprocessor and the Wang IBM Monochrome Emulation Card (Wang Part Number PC-PM101). The IBM Monochrome Emulation Card includes software required to access the features of this optional cicuit board. If these two options are installed, you can place the Wang PC in IBM emulation mode by performing the following operations:

1. Boot the Wang PC with the Wang Disk Operating System.

2. Select the "Other" option from the operating system's main menu.

3. When asked for a file name, enter "loadibm". This tells the operating system to execute the LOADIBM program from the IBM Monochrome Emulation Card software. If this program is not in the current directory you must preface "loadibm" with the directory path to the program.

4. Following the instructions displayed by the LOADIBM program, place an IBM PC-DOS system disk in disk drive A:, then press [Exec].

If you are successful in following the above procedure, you can now run Host as if using an IBM PC microcomputer.

Although the Wang IBM Monochrome Emulation Card should include an alternate set of key caps, you might find that these have not been installed on the PC you are using. The following table lists Wang keyboard equivalents for a few crucial IBM PC keyboard keys.

IBM AT Command Key	Wang Professional Computer Command Key
[Enter]	[Exec] or [Return]
[Esc]	[Back Tab]
[PgUp]	[Prev]
[PgDn]	[Next]
[Ins]	[Insert]
[Num Lock]	[Go To]
[Scroll Lock]	[Cancel]

185

APPENDIX F

APPENDIX F.

VERIFY ERROR MESSAGES

You can Verify Host Data Files and Libraries by choosing Verify from the Manage Data and Manage Library Menus respectively. The following is an alphabetized list of error messages that may be issued by Host's file verification facility. Following each message is a more detailed explanation of the error and, where appropriate, a discussion of the possible actions you may take to correct the error. Each error message is shown in bold print and optional phrases in these messages are enclosed in square brackets ("["..."]"). Portions of the messages that may vary, such as names, values, and items are enclosed in angle brackets ("<"..."><"). For example, "<variable name>" represents the name of a variable and "<item>" represents one of the following:

Data [for period <value> of]
Dimension Definition
Label Format
Table Definition
Variable Definition

- **Computed size of <variable name> is <value> bytes; stored size is <value> bytes.**

Verify recomputes the size of each variable by multiplying together the Maximum Number of Levels for each dimension of the variable and the number of bytes required to store a single value of the variable's Type. If this value does not equal the size value stored in the Variable Definition, then Verify issues the above error message.

In Rebuild Mode Verify displays a menu with the choices Computed, Stored, Neither, and Quit. Choose Quit if you want to stop verification. If the computed value of size is correct, then choose Computed from this menu and Verify will change the Variable Size stored in the Variable Definition to the computed value. If the stored value of size is correct, or if neither value is correct, then choose Stored or Neither from the menu; in either case Verify can not correct the problem. The problem may be an incorrect variable Type, an incorrect value for the Maximum Number of Levels of one or more dimensions, or an incorrect value for the Number of Dimensions of the variable. Verify does not attempt to change any of these values, but continues verification.

- **<item> <name> is marked as deleted, but has an index entry.**

Verify searches sequentially through the index files to the data, Dimension Definitions, Label Formats, Table Definitions, and Variable Definitions to check that

each entry has a corresponding data record that is not marked as deleted. If the data or record is marked as deleted, one of two possible errors has occurred. Either the data or record should be marked as deleted and the index entry has been retained by mistake, or the data or record is incorrectly marked as deleted and the index entry should be retained. Verify has no way of knowing which error occurred, so it issues the above error message.

In Rebuild Mode Verify displays a menu with the choices Save, Delete, and Quit. Choose Quit if you want to stop verification. If the data or record is incorrectly marked as deleted, choose Save and Verify will mark the data or record as active. If the data or record is correctly marked as deleted, choose Delete and Verify will remove the corresponding entry from the index file.

- **<item> name is "<key value 1>"; index key is "<key value 2>".**

Verify compares each data key and the name of each Dimension Definition, Label Format, Table Definition, and Variable Definition with the key value stored with the corresponding index entry. If these two values do not agree, Verify issues the above error message.

In Rebuild Mode Verify displays a menu consisting of the choices <key value 1>, <key value 2>, Neither and Quit. If you want to stop verification, choose Quit. If one of the displayed values is correct, choose the appropriate menu selection and Verify will correct the value in error. If you choose Neither, Verify does not change either value, but continues verification.

- **Error adding entry for <key value> to index.**

This message shows that an error has occurred while attempting to add an index key to one of the rebuilt files. Nothing can be done to correct this error within Verify, but Host gives you an opportunity to halt verification at this point.

- **Index entry for <item> <name> is wrong.**

Verify compares the actual location of each period of data and each Dimension Definition, Label Format, Table Definition, and Variable Defintion to the location stored in the corresponding index entry. If the location stored in the index entry does not match the actual location, Verify issues the above error message. In Rebuild Mode Verify automatically corrects this error.

- **Size of <name> stored in Variable Definition is <value> bytes; size stored with data for period <value> is <value> bytes.**

Verify compares the size stored with each period of data for a variable with the size stored in the corresponding Variable Definition. If these two values do not match, Verify issues the above error message.

In Rebuild Mode Verify determines the location of the next period of data stored in the Data file and uses this information to compute the number of 128-byte blocks being used to store the period of data in question. Verify then compares this number to the number of blocks that would be required to store the variable if the size stored with the Variable Definition is correct. If these numbers match, Verify tells you that the size stored in the Variable Definition is probably correct and gives you an opportunity to correct the size stored with the data.

If these block sizes do not match, Verify next compares the actual number of blocks being used to the number of blocks that would be required to store the variable if the size stored with the data is correct. If these numbers match, Verify tells you that the size stored with the data is probably correct and gives you an opportunity to correct the size stored in the Variable Definition. If these block sizes do not match, Verify can not correct the problem, and does not alter the size stored in either location. In this case you are given an opportunity to stop verification.

- **There is no index entry for <item> "<name>".**

Verify searches sequentially through the data file, Dimension Definitions, Label Formats, Table Definitions, and Variable Definitions to make sure that each period of data and each record which is not marked as deleted has a corresponding index entry. If there is no entry, one of two possible errors has occurred. Either the data or record should be marked as deleted, and the index entry removed, or the record is correctly marked as active, and the corresponding index entry should be created. Verify has no way of knowing which error has occurred, so it issues the above error message.

In Rebuild Mode Verify displays a menu with the choices Save, Delete, and Quit. If you want to stop verification, choose Quit. If the data or record is correctly marked as active, choose Save, and Verify will create an index entry. If the record is incorrectly marked as active, choose Delete, and Verify will mark the data or record as deleted.

- **There is no <item> <name> matching index entry.**

Verify searches sequentially through the index files to the data, Dimension Definitions, Label Formats, Table Definitions, and Variable Definitions to see that each entry has a corresponding period of data or record. If the index entry does not have a matching period of data or record, Verify issues the above error message.

In Rebuild Mode Verify can only resolve this error by deleting the entry in the rebuilt index. This enables you to recreate the missing record or period of data within Host after verification is completed or has been stopped. Verify gives you the opportunity to stop verification at this point.

- **Unable to verify stored size of <variable name>.**

If one or more dimensions referenced by a variable are missing, Verify is unable to recompute the variable size because this calculation requires the Maximum Number of Levels of each dimension. Since Verify is unable to compare the computed size with the size stored in the Variable Definition, it issues the above warning message. This problem can not be corrected within Verify, but after verification is completed or stopped, you can select Define, Dimension, Add from Host's Main Menu to recreate the missing Dimension Definitions. In Rebuild Mode Verify gives you an opportunity to stop verification at this point.

- **Variable Definition for <name> has been marked deleted.**

Verify issues this error message whenever the Variable Definition whose name matches the name stored with an active period of data is marked as deleted. In Rebuild Mode Verify displays the menu Save, Delete, Quit. If you want to stop verification, choose Quit. If you want to recover the deleted Variable Defintion, choose Save, and Verify will mark the record as active and create a corresponding index. If you do not want to save the record, but do want to continue verification, choose Delete.

- **Variable Definition for <name> has no index entry.**

The Variable Definition corresponding to a period of data has no corresponding index entry. This error can not occur if you are in Rebuild Mode since it is detected and corrected when Verify is checking the Variable Definition portion of the Data File. If you are not in Rebuild Mode, this error can occur, and is the same error that is described above under the message "There is no index entry for <item> '<name>'." This error can not be corrected until you re-Verify the Data File in Rebuild Mode.

- **Variable Definition for <name> is missing.**

Verify issues this error message whenever it is unable to find a Variable Definition whose name matches the name stored with a period of data. This error can not be corrected within Verify, but after verification is completed or stopped, you can select Define, Variable, Add from Host's Main Menu to recreate the missing Variable Definition. In Rebuild Mode Verify gives you an opportunity to stop verification at this point.

- **Variable <name> references deleted dimension <name>.**

A dimension referenced in a Variable Definition is marked as deleted. If you are in Rebuild Mode, Verify displays the menu Save, Delete, Quit. If you want to stop verification, choose Quit. If you want to recover the deleted Dimension Defintion, choose Save, and Verify will mark the record as active and create a corresponding index entry. If you do not want to save the record, but do want to continue verification, choose Delete.

- **Variable <name> references missing dimension <name>.**

A dimension referenced in a Variable Definition is missing. This error can not be corrected within Verify, but once verification is completed or stopped, you can select Define, Dimension, Add from Host's Main Menu to recreate the missing Dimension Definition. In Rebuild Mode Verify gives you an opportunity to stop verification at this point.

- **Variable <name> references unindexed dimension <name>.**

A dimension referenced in a Variable Definition has no corresponding index entry. This error can not occur if you are in Rebuild Mode since it is detected and corrected when Verify is checking the Dimension Definition portion of the Data File. If you are not in Rebuild Mode, this error can occur, and is the same error which is described above under the message "There is no index entry for <item> '<name>'." This error can not be corrected until you re-Verify the Data File in Rebuild Mode.

GLOSSARY

GLOSSARY

ASCII. American Standard Code for Information Interchange. ASCII, which is the standard character coding scheme used on most computers, assigns a unique numeric code to each available character.

Application Module. A compiled Turbo Pascal program specially constructed to function within Host.

Attributes. The individual characteristics that define Host dimensions and variables. *See also* **Dimension Definition** and **Variable Definition.**

Backup. Host command used to make copies of Data Files and Instruction files. *See also* **Source File**, **Target File.**

Byte. A byte is the basic unit of data storage in computers. One byte can contain a single character of data or a numeric value in the range 0 to 255. Byte is one of five basic Host variable Types. *See also* **Integer**, **Real**, **Char**, and **String.**

Cell. An intersection of the row and column dimension levels displayed in the Work Space window by Host's Data Editor. In the Data Editor, each cell is occupied by a data entry for the current variable.

Char. A single character, e.g, "A", "Q", or "&". There are 128 characters in the ASCII character set, the set of characters used on most microcomputers. IBM PC, XT, and AT microcomputers extend the ASCII character set by an additional 128 characters. Char is one of five basic Host variable Types. *See also* **Byte**, **Integer**, **Real**, and **String.**

Column Dimension. A dimension whose levels differentiate the columns of a Host table or Data Editor display.

Comparison File. A Host Data File intended to be used as a "base-run" against which to compare another Host Data File in tables and graphs.

Comparison Table. Table produced when there are second and third occurrences of a variable name in the variable list of a Table Definition. The first occurrence refers to the variable in the reference Host Data File ("V" in the formula below). The second occurrence refers to that variable in a second Host Data File called a Comparison File ("V_c" in the formula below). The third occurrence requests a column of percentage differences calculated by the formula

$$(V_c - V)/V_c * 100$$

Computed Label Specification. One of three basic types of label specifications that can be used in Host Label Formats. A computed label specification is a convenient way to generate a series of equal-sized grouping specifications. Host creates an appropriate text label automatically for each generated specification. To create a computed label specification, specify an upper bound greater than the lower

bound and an interval size by which to subdivide the specified range of levels. *See also* **Explicit Label Specification** and **Grouping Label Specification.**

CON. A mnemonic that means "Console" if entered as the Output Device when you are generating tables and graphs. *See also* **PRN.**

Configuration File. File containing default settings used throughout Host. Host gives all Configuration files the name "SYSTEM.ENV".

Console. The display screen attached to your microcomputer.

Control Characters. Characters with ASCII codes from 0 to 32. These characters must be coded by entering [\] followed by a three digit ASCII code when used in the Setup String in the Configuration File. *See also* **ASCII.**

Cursor. A special symbol, color, or shape used to show position on the display screen.

Cursor Selection. One of two methods of selecting a command or item from a Host menu. A selection is made by moving the Menu Cursor to the desired item using the cursor movement keys, then pressing [Enter]. *See also* **Key Selection.**

Data File. *See* **Host Data File.**

Default. In Host, a value, option, or selection assumed in the absence of any other.

Default Data Directory. Directory where Host expects to find or create Data Files if no explicit directory path is supplied with a Data File name.

Default Graph Output Device. Each Host-produced graph is sent to the default Graph Output Device unless you enter a different destination when a graph is generated. You can use the Configure selection on Host's Main Menu to set this device to the "Printer", "Plotter", or "Console".

Default Instruction Directory. Directory where Host expects to find or create Instruction files if no explicit directory path is supplied with an Instruction file name.

Default Library Directory. Directory where Host expects to find or create Label, Table and Graph libraries if no explicit directory path is supplied with a library name.

Default Module Directory. Directory where Host expects to find Application Modules if no explicit directory path is supplied with a module name.

Default Table Output Device. Each Host-produced table is sent to the default Table Output Device unless you enter a different destination when a table is generated. You can use the Configure selection on Host's Main Menu to set this device to either the "Printer" or the "Console".

Dimension. A set of categories or levels used to address or specify a single value in a variable's aggregate data space.

Dimension Definition. A set of attributes that define a dimension in Host. These attributes include a dimension's Name, Description, Maximum Number of Levels,

Current Number of Levels, Starting Level, Increment Between Levels, and default Label Format. Dimension definitions are part of the data definition portion of a Host Data File. Dimension definitions are stored in files with the extension "DIM". Indices to these definitions are stored in files with the extension "DDX".

Dimension Line. The top two rows of the Work Space in Host's Data Editor when the variable being edited has one or more dimensions. The Data Editor uses this area of the screen to display the names of the row and column dimensions, and the names and current levels of any other dimensions.

Dimension Settings. The component of a Table Definition that contains information on how data are to be arranged in the table.

Display Mask. An attribute included in the Variable Definition of each Host variable. A variable's Display Mask affects the way data from that variable appear in tables and graphs.

Edit Mask. An attribute included in the Variable Definition of each Host variable. An Edit Mask is a sequence of special characters used by Host's Data Editor to check data entries as they are typed.

Effective Level. Computed from the actual dimension level using the following formula:

Effective Level = (level - 1) * Increment Between Levels + Starting Level

where Increment Between Levels and Starting Level are parts of a dimension definition. *See also* **Increment Between Levels** and **Starting Level.**

Error Message Space. The lower left corner of the Host screen display. This space is reserved for short error messages. When a potentially serious error has occurred, the short message displayed in this space may be supplemented by information in the Work Space.

Explicit Label Specification. One of three basic types of label specifications that can be used in Host Label Formats. In an explicit label specification, the lower bound is set equal to the upper bound, thus assigning the specified text label to a single dimension level. *See also* **Grouping Label Specification** and **Computed Label Specification.**

Field. In Host, a field is a space or blank to be filled in following a question or prompt on the screen.

Graph Library. Contains Graph Definitions specifying how data should be depicted in graphs.

Grouping Label Specification. One of three basic types of label specifications that can be used in Host Label Formats. In a grouping label specification, the upper bound is greater than the lower bound and the text label assigned applies to all dimension levels between the lower and upper bounds, inclusive. Host creates a default text label for grouping label specifications of the form

LowerBound - UpperBound

but you may substitute text of your own choosing. *See also* **Explicit Label Specification** and **Computed Label Specification.**

Home Directory. The directory containing the Host System Files.

Host Data File. Contains input and output data used by Application Modules. Each Host Data File also includes a definition of the data it contains. *See also* **Application Module.**

Host System File. A disk file that is part of the Host program itself. *See* **Appendix A,** HOST SYSTEM FILES.

Identifier. The name of a Host Label Format, Dimension, Variable, Table Definition, or Graph Definition. All identifiers must begin with a letter (A-Z) or underline character (_); the first letter may be followed by any combination of letters (A-Z), digits (0-9), or underline characters (_); and the identifier may be from one to eight characters long.

Increment Between Levels. An attribute included in the Dimension Definition of each Host dimension. In the absence of a Label Format, Host labels the levels of a dimension with integers beginning with the dimension's Starting Level and incremented by the dimension's Increment Between Levels. A dimension's Increment Between Levels also affects the way Label Formats are applied to that dimension.

Instruction File. Contains instructions in sequence for running Application Modules, generating tables, and performing other operations without operator intervention.

Integer. A member of the set of positive whole numbers (1,2,3,...), negative whole numbers (-1,-2,-3,...), or zero (0). An IBM PC, XT, or AT microcomputer can manipulate integer values in the range -32768 to 32767. Integer is of five basic Host variable Types. *See also* **Byte, Real, Char,** and **String.**

K. One K (Kilobyte) is equal to 1024 bytes. *See also* **Byte.**

Key Selection. One of two methods of selecting a command or item from a Host Menu. A selection is made by pressing the key corresponding to the first letter of the desired item. Menu items are scanned from left to right, and the first item beginning with the letter entered is selected. *See also* **Cursor Selection.**

Label Format. Consists of a label name and description and one or more label specifications. Host's Data Editor, Table Generator, and Graph Generator can all use Label Formats to annotate data from Host Data Files. Host's Label Formats also provide a powerful mechanism for aggregating data into categories for presentation in tables and graph. *See also* **Label Specification** and **Label Library.**

Label Library. Contains Label Formats for annotating data output and for totaling data categories in tables.

Label Specification. A Host Label Format is composed of one or more label specifications. Each label specification tells Host to apply a text label to a dimension level, or range of dimension levels. A label specification includes the lower bound and upper bound of the dimension level range to which it applies, and either an interval size for generating equal sized, automatically labeled subgroups, or a text label to assign to the specified range of levels. There are three types of label specifications that can be included in Label Formats to assign text labels to dimension levels: explicit, grouping, and computed. *See also* **Label Format**, **Explicit Label Specification**, **Grouping Label Specification**, and **Computed Label Specification**.

Library Files. *See* **Label Library**, **Graph Library**, and **Table Library**.

Lotus Import File. An ASCII text file formatted according to the import requirements of Lotus 1-2-3 spreadsheet software. All text labels in such a file are enclosed in double-quotes and no line in the file is longer than 240 characters. A file with this format can be imported, or read into a Lotus 1-2-3 Worksheet for further numerical or graphical analysis *See also* **Print-ready File.**

Manager. One of two categories of Host operators. Managers develop and install new Application Modules. A password must be entered to enter Host as a Manager. *See also* **User.**

Manager Access. The access level associated with Host Managers. Managers are given access to all Host commands, settings, and data except those that could cause unintentional loss of data from Host Data Files. *See also* **User Access.**

Matrix. In mathematics, a rectangular array of numerical or algebraic quantities treated as an algebraic entity. In Host, it is a variable with two or more dimensions. *See also* **Multi-dimensional Array.**

Maximum Value. An attribute included in the Variable Definition of each Host variable of Type "byte", "integer", or "real". A variable's Minimum and Maximum Values are used by Host's Data Editor to check the range of numbers entered for that variable. *See also* **Minimum Value.**

Megabyte. One megabyte is equal to 1,048,576 (about one million) bytes or 1024K bytes. *See also* **Byte** and **K.**

Menu Space. The second two lines of the Host screen. When not used for displaying other information, this area is used for displaying "Lotus-style" horizontal menus.

Menu Cursor. A highlighted block that shows you the current selection on a Host menu.

Message Space. The upper left corner of the Host screen. This space is reserved for messages about what the program is doing, or has done.

Minimum Value. An attribute included in the Variable Definition of each Host variable of Type "byte", "integer", or "real". A variable's Minimum and Maximum Values are used by Host's Data Editor to check the range of numbers entered for that variable. *See also* **Maximum Value.**

Mode Indicator. One or two words displayed in reverse video in the Mode Space to show you where you are within the Host system. This is useful for orientation and for

reporting the location of errors within the program.

Mode Space. The upper right corner of the Host screen. This space contains the mode indicator.

Module. *See* **Application Module** and **Projection Module.**

Multi-dimensional Array. A variable with two or more dimensions. *See also* **Matrix.**

Nested. Refers to a dimension Use within tables. A Nested dimension is a third dimension that may be displayed within one subtable of a table. When a Nested dimension is selected, all levels of the dimension are displayed within each level of the Column dimension. If no Column dimension is chosen, then all levels of the Nested dimension are displayed within each variable column. *See also* **Unnested,** and **Sum-over.**

Nonperiodic. All variables in Host are classified as periodic or nonperiodic. The data in periodic variables may change through time so that a periodic variable has an implied time dimension. The data in nonperiodic variables are static through time; they do not change from one time period to another.

One-dimensional Array. A variable with only one dimension. *See also* **Vector.**

Operator Access. The access level assigned to a Host Operator when Host is entered from the operating system, or through the Access command on Host's Main Menu. *See also* **Manager Access** and **User Access.**

Pack. A Host command used to compress a Host Data File or Library by removing records marked as deleted.

PERIOD. *See* **PERIOD Dimension.**

PERIOD Dimension. A dimension used by Host to represent the dimension of time for Application Modules that project data through time. PERIOD is an implied dimension of all periodic Host variables. When you create a new Host Data File Host adds a dimension named "PERIOD" to the Data File automatically. Host requires the dimension PERIOD in all Data Files and will not allow you to delete it.

Periodic. All variables in Host are classified as periodic or nonperiodic. The data in periodic variables may change through time so that a periodic variable has an implied time dimension. The data in nonperiodic variables are static through time; they do not change from one time period to another.

Period Interval. A shorthand notation for a collection of period values. A period interval has the form

first-last

where **first** and **last** are period levels. A period interval includes all periods from **first** to **last** inclusive. If **first** is omitted, all periods up to and including **last** belong to the interval; if **last** is omitted, all periods from **first** on belong to the interval. If **first** and **last** are omitted all periods are included in the interval. *See also* **Period Value** and **Period List.**

198

Period List. Used to tell Host's Table Generator which periods of data to include in a table. A period list is the word "ALL", or a list of integer period values and period intervals separated by commas. *See also* **Period Value** and **Period List.**

Period Value. An integer from 1 to 1024 inclusive. *See also* **Period Interval** and **Period List.**

Print-ready File. File containing the Setup String stored in the Configuration File, ASCII Form-Feed characters and table, graph or report output. Such files can transferred to other machines for printing or included into word processing documents. *See also* **Lotus-Import File.**

PRN. A mnemonic that means "Printer" if entered as the Output Device when you are generating tables and graphs. *See also* **CON.**

Projection Module. An Application Module that projects data through the time dimension PERIOD. *See also* **Application Module.**

Protected Field. A data entry field is protected if it is possible to view it, but not to modify its contents. *See also* **Unprotected Field.**

Query Field. Host prompt composed of a blank to be filled in, preceded by an inverse video "?". This prompt permits you to display a list of names, and descriptions if available, of all possible entries by pressing [?]. You can select an item from this list by highlighting it with a cursor, then pressing [Enter].

RAM. An acronym for Random Access Memory. All memory used by Host in a microcomputer is Random Access Memory.

RAM Disk. A portion of RAM managed by a program that simulates the operation of a physical disk drive. *See also* **RAM.**

Real. A numeric value that may contain a fractional part. IBM PC, XT, and AT microcomputers using an Intel 8087/80287 numeric coprocessor can manipulate real values in the range -1.797E+308 to 1.7976E+308. Real is one of five basic Host variable Types. *See also* **Byte**, **Integer**, **Char**, and **String.**

Rebuild. A subcommand of the Host Verify command used to reconstruct damaged Host Data Files and Libraries. *See also* **Verify.**

Reference Data File. A Host Data File used by the Table Definition Editor for obtaining information about the variables and dimensions of a table.

Row Dimension. A dimension whose levels differentiate the rows of a Host table or Data Editor display.

Run-time Error. An error that occurs while running a Host Application Module or processing a Host Instruction file.

Scalar Value. In general, a quantity, such as mass, length, time, or temperature completely specified by a number on an appropriate scale. In Host, it is the value of a numeric variable with no dimensions.

Scaling Exponent. An attribute included in the Variable Definition of each Host variable that is Type "real". The Scaling Exponent tells Host, and anyone using

199

Host's Data Editor, how data for a variable should be scaled. Scaling is the process of dividing data by ten raised to an integer power. Since fewer digits must be entered when data are expressed in larger units, this simplifies data entry.

Setup String. A character string composed of any combination of coded control characters and directly entered non-control characters. Host uses the Setup String to send commands to a printer to change fonts, print styles, and other print characteristics before printing the first line of any report, table, or graph.

Size. *See* **Variable Size.**

Source File. A Host Data File or Instruction file to be copied by a **Backup** command. *See also* **Target File, Backup.**

Starting Level. An attribute included in the Dimension Definition of each Host dimension. In the absence of a Label Format, Host labels the levels of a dimension with integers beginning with the dimension's Starting Level and incremented by the dimension's Increment Between Levels. A dimension's Starting Level also affects the way Label Formats are applied to that dimension.

String. A sequence of characters, e.g, "GH%hj8$" and "Occupation Class". String is one of five basic Host variable Types. *See also* **ASCII, Byte, Integer, Real,** and **Char.**

Subtable. A cross-section of the data stored in a table's variable or variables. A table consists of one or more such subtables, depending on the structure of the variables included in the Table Definition and the way the data are presented.

Sum-over. Refers to a dimension Use within tables. This is the default Use setting for all dimensions except PERIOD, which can never be a Sum-over dimension. When Sum-over Use is selected, the values in the table are produced by adding across all levels of the Sum-over dimension. *See also* **Nested** and **Unnested.**

Syntax Error. In computer programming, the type of error that occurs when a program statement or structure violates the rules governing programs written in a specific computer language. In Host, it is the type of error that occurs if you omit required parameters or specify parameters incorrectly in a Host Instruction.

Table Definition. Consists of Table Name and Title, a variable list naming the variables to be included in the table, and dimension settings containing information on how the data are to be arranged in the table. *See also* **Table Library.**

Table Generator. A Host facility that combines a Table Definition from the Table Library with data from a Host Data File to produce a table.

Table Library. Contains Table Definitions specifying how data should be arranged in tables.

Target File. The file to receive a copy of a Host Data File or Instruction file in a **Backup** command. *See also* **Source File, Backup.**

Toggle Field. Host prompt composed of a blank to be filled in, preceded by an inverse video "±". Successive choices for a Toggle Field can be displayed by pressing [+], [Space], or [-].

Toggle Indicator Space. The lower right corner of the Host screen. This space is used for displaying the status of the toggle keys [Ins], [Caps Lock], [Num Lock] and [Scroll Lock]. In the Data Editor, it is also used to display the status of the [End] key.

Type. *See* **Variable Type.**

Unit of Measurement. An attribute included in the Variable Definition of each Host variable that is Type "byte", "integer", or "real". The Unit of Measurement is a text label up to ten characters long that provides additional information about a variable in reports, tables, graphs, and in Host's Data Editor. It is used only for annotation and is intended to record the unit of measurement of numeric variables. *See also* **Variable Definition** and **Variable Type.**

Unnested. Refers to a dimension Use within tables. A separate subtable of a table is produced for each level of an Unnested dimension. If there is more than one Unnested dimension, a separate subtable is produced for each combination of levels of these dimensions. *See also* **Nested** and **Sum-over.**

Unprotected Field. A data entry field is unprotected if it is possible to modify the contents of the field. *See also* **Protected Field.**

User. One of two categories of Host operators. Users use existing Application Modules. *See also* **Manager.**

User Access. The access level associated with Host Users. Users have access to all Host commands, settings, and data except those that could disrupt the structure of Host Data Files or the operation of Application Modules. *See also* **Manager Access.**

Variable. All data in a Host Data File are assigned to variables such that a variable and its data are considered to be the same. In mathematics, a variable is something having no fixed quantitative value; in Host, variables may also be used as constants or parameters. Further, Host makes no distinction between variables used as input and variables used as output. Host variables may even contain character rather than numeric data.

Variable Definition. A set of attributes that define a variable in Host. These attributes include a variable's Name, Description, Size, Type, Minimum Value, Maximum Value, Scaling Exponent, Unit of Measurement, Edit Mask, Display Mask, whether it is Periodic or Nonperiodic, Number of Dimensions, and the name of each dimension. Variable definitions are part of the data definition portion of a Host Data File. Variable definitions are stored in files with the extension "VAR".

Variable list. The component of a Table Definition containing the names of the variables to be included in the table.

Variable Size. An attribute included in the Variable Definition of each Host variable. This attribute is the size of a variable in bytes and is computed by Host based on a variable's Type, Number of Dimensions, and the Maximum Number of Levels of each of its dimensions.

Variable Type. An attribute included in the Variable Definition of each Host variable. Each Host variable must be one of five fundamental Types: byte, integer,

real (floating-point), char (character), or string (a sequence of characters).

Vector. In computer programming, a one-dimensional array. In Host, it is a variable with only one dimension. *See also* **One-dimensional Array.**

Verify. A Host command used to check the structural integrity of a Host Data File or Library. *See also* **Rebuild.**

Work Space. The center part of Host's display screen, bounded at top and bottom by horizontal lines. This space is usually used for data entry and display, or for other helpful information.

INDEX

203

INDEX

access level, 3-2, 3-10
accuracy, 2-4
accurate, 1-6, 5-5, 5-12
adapter, 2-2 to 2-3, 4-9
Add,
  Dimension Menu, 5-9 to 5-10
  Label Menu, 6-7 to 6-8
  Table Menu, 7-17 to 7-18
  Variable Menu, 5-21 to 5-22
  advanced, 1-3 to 1-5, 8-12, D-2
aggregate, 3-15, 6-2
alphabetical, 3-18,
alphanumeric, 1-5
analysis, 1-3, 10-4, 10-9, 11-8, 13-6
annotate, 1-4, 3-11, 6-2, 8-4, 11-9
Application Module, 1-2, 1-4, 3-8, 3-10 to 3-13, 3-17, 4-5, 5-3 to 5-4, 5-7 to 5-8, 5-14, 9-2 to 9-4, 11-2 to 11-4, 11-6 to 11-7, 12-4, 13-2 to 13-5, 13-8 to 13-9, A-2 to A-4, D-2, E-3
arguments, 11-5 to 11-7, 13-4
ASCII, 4-8, 8-3, 10-4, 11-2, 12-12, A-4, E-3
attributes, 2-2, 5-3 to 5-4, 5-10 to 5-11, 5-14, 5-20, 5-22, 7-7, 8-11, E-3
AUTOEXEC.BAT, 2-5 to 2-6, 3-3
automating module execution and table generation, 1-3 to 1-4

back-up, 3-20, 9-2 to 9-3, 11-9, 11-10, 12-8
backslash, 4-8, 7-5
base run, 1-3
batch, 1-3, 13-7 to 13-9
Bernoulli Box, 2-4
blinking, 3-5, 8-8, 8-11
boldface, 1-5
boot, 2-4 to 2-6, 3-17, 12-4, 12-8, 13-9
bottom margin, 4-7
brackets, 1-5, 5-16, 11-3, 13-3, F-2
bright, 3-5, 7-9
buffer, 2-5, 3-16, 10-7
bugs, C-2 to C-3

calculated, 1-3, 7-7, 7-17
calculations, 1-2, 2-4, 3-12, 5-13
cancel, 8-10, 11-10
cartridge, 2-2 to 2-4, 11-9
categories, 1-4, 3-10 to 3-11, 3-14 to 3-15, 4-2, 5-5, 6-2, 7-15
categorized, 5-8
cell, 5-20, 8-8, 8-10, 8-11, 8-12
char, 5-13, 5-14, 5-16, 5-18, 5-19
chip, 2-4
color, 2-2 to 2-3, 3-5, 4-2, 4-9
  display, 2-2 to 2-3
column,
  dimension, 8-4, 8-6, 8-8, 10-8, C-2
  width, 10-9
combine, 6-2, 7-2, 7-9, 7-14 to 7-16, 8-7, 10-8, A-3
commas, 10-4
Compaq, 2-3
compare, 1-3, 4-4, 7-7, F-5
comparison, 1-3, 10-3
Comparison,
  File, 4-3 to 4-4, 7-7, 10-2 to 10-3, 11-5, 11-7, 12-3 to 12-4, 13-5 to 13-6, B-3
  Table, 7-7 to 7-8
compatible, 1-5, 2-2, 3-5, 5-18
compiler, 1-6, 3-13, A-4
computed label specification, 6-5, 6-7
computer, 1-2, 1-6, 3-12 to 3-13, 3-20, 3-23, 5-2
CON, 10-4, 11-7, 13-5
CONFIG.SYS, 2-2, 2-4 to 2-6
configuration, 2-4, 3-4, 4-2 to 4-3, 4-9, 5-18, 13-10 to 13-12, A-2, B-2 to B-3
Configuration File, 3-11 to 3-12, 3-20, 4-2 to 4-3, 4-5 to 4-6, 4-9, 7-5, 10-4, 10-8, 11-7, 12-3 to 12-4, 12-6, 12-9, 13-2, 13-4 to 13-7, 13-11, A-2, B-2 to B-3, C-4
conflicts, 2-4
consecutive, 7-7
console, 2-5 to 2-6, 3-20, 4-4, 10-4, 10-6, 11-4, 11-7, 13-5
construct, 11-5, 1-3, 12-5, 12-9
control characters, 4-8
coprocessor, 2-4, 5-14, 5-18, F-2
Current Number of Levels, 5-6, 5-9, 8-6, 8-10 to 8-11, 11-9, 13-7

204

current,
  directory, 1-5
  drive, 1-5

damage, 3-10, 4-9, 12-4, 12-8
danger, 11-9, 12-4
data definition, 1-2, 3-13 to 3-14, 5-2 to 5-4, D-2
database, 1-2, 3-13, 3-17, 5-3, 12-4, 12-8, D-2
DBMS, 1-2
decimal, 5-17, 8-3
default,
  Comparison File, 10-3, 11-7
  Data Directory, 3-13, 4-5, 5-11, 5-23, 6-10, 7-22 to 7-23, 8-2, 8-14, 9-3, 10-3 to 10-4, 11-7 to 11-9, 12-3 to 12-5, 12-7, 13-4 to 13-6, 13-11, B-3
  Data File, 5-3, 7-2, 7-22, 8-2, 8-14, 9-3, 11-5, 11-7 to 11-9, 12-3 to 12-5, 12-7, 13-2, B-3
  Graph Output Device, 4-4
  Instruction Directory, 3-16, 4-5, 4-6, 11-2, 11-4, 11-6, 11-9, 11-12, 13-11
  Instruction file, 4-4, 11-11 to 11-12
  Label Format, 5-7 to 5-8, 6-2, 6-8, 7-14, 8-4, 8-6 to 8-7
  Library Directory, 3-15, 4-5 to 4-6, 6-2, 7-2, 10-2, 12-8 to 12-9, 12-11, 13-11
  Module Directory, 3-13, 4-5, 9-2, 11-7, 13-4, 13-11
  Table Output Device, 4-4, 10-4
  values, 1-4, 3-11, 5-4, 5-11, 8-3, 11-5
Define Menu, 3-7, 3-20, 5-3 to 5-4, 5-11, 7-2
desk-top, 2-4
destination, 3-20, 4-4, 5-11, 5-23, 6-10, 7-22, 10-2, 10-6
destroy, 3-10
detect, 3-4, 3-17, 12-6, 12-10
devices, 2-4, 3-11, 4-2
diagnose, 1-6
differences, 1-5, 2-2, 3-2, 3-7, 7-7, 7-8
digit, 3-8, 3-9, 4-8, 5-18, 8-4
dim, 3-5, 5-1, 5-11, 7-9
Dimension,
  Definition, 3-14, 5-4 to 5-6, 5-9 to 5-11, 5-20, 6-8, 11-11, 12-7, F-2 to F-3, F-6
  Line, 8-4, 8-6, 8-10
  Menu, 5-4, 5-10, 5-11
  dimension,
  level, 6-2 to 6-5, 6-8, 8-7
  settings, 7-2, 7-9, 7-14, 8-10, 10-7
disaggregated, 1-2
disk space, 2-3 to 2-4, 5-10, 5-22, 6-9, 7-22, 8-3, 11-9, 12-2 to 12-5, 12-7, 12-9, 12-11
diskette, 1-6, 2-4, 11-9
Display Mask, 5-16, 5-18
display,
  adapter, 2-2, 4-9
  attributes, 2-2
document, 1-4, 1-5, 2-2, E-3
documentation, 1-6, 3-13, 5-7 to 5-8, 9-2 to 9-3, 11-7, 13-4
Dollar Utility, A-4

Edit,
  Dimension Menu, 5-4, 5-9 to 5-10
  Label Menu, 6-5, 6-7 to 6-8
  Mask, 5-13 to 5-17, 8-4, 8-11
  Menu, 8-3, 8-10, 8-12 to 8-14, 11-6
  Table Menu, 7-17 to 7-18
  Variable Menu, 5-11, 5-21 to 5-22
Effective Level, 5-7, 6-2, 11-8
efficiency, 3-8
eject, 4-6, 4-8
emulation, 2-3, F-2
Enhanced Graphics Display Adapter, 2-3
enhancements, D-2
environment, 13-2, 13-4, 13-11
Epson, 4-8 to 4-9
erase, 3-20, 8-11, 13-3
Error Message Space, 3-4, 3-10, 3-20, 11-10 to 11-11
error message, 1-5 to 1-6, 3-2, 3-4, 3-17, 7-7, 9-3, 10-3, 10-7 to 10-8, 11-10 to 11-11, 12-4, 12-8, 13-2 to 13-3, B-2, C-2, F-2 to F-5
errors, 1-4 to 1-6, 3-4, 3-17, 5-3, 5-8, 10-7, 11-5, 11-11, 12-4 to 12-10, 13-3, A-2, F-3 to F-4
escape, 1-5
explicit, 5-19, 8-4, 9-3, 10-4
  label specification, 6-5, 8-7
exponent, 7-14
exponential, 5-19
extension, 1-5, 3-11, 3-13, 3-16, 3-20 to 3-21, 5-11, 5-23, 6-10, 7-22, 10-4, 10-8, 11-2, 11-6 to 11-9, 12-3 to 12-4, 12-6, 12-8 to 12-9, 12-12, 13-2, 13-4 to 13-7, A-4

failure, 3-17, 12-4, 12-8
faulty, 3-17

field width, 5-16
file name extensions, 1-5, 3-14, 11-7, 13-4 to 13-5
floating-point, 5-13, 5-16, 5-18, 8-3
floppy disk, 1-4, 2-2, B-2, B-3
fonts, 4-8
foreign, 2-4
format, 5-16, 6-3, 7-2, 8-4, 8-6, 10-4, 10-9, 11-3, 13-2 to 13-3, E-3
formula, 5-7, 6-2, 7-7, 11-8
French, 7-8

Genoa, 2-3
German, 7-8
Graph Library, 3-11, 3-14, 4-5, B-3
graphics, 1-2 to 1-3, 3-13 to 3-15, 4-2, 4-4, 5-7 to 5-9, 5-12 to 5-13, 5-18 to 5-19, 6-2, 6-8, 9-4, 11-8 to 11-9, 13-6, D-2, E-3
grouping, 6-5 to 6-6, 6-8, 8-7
grouping label specification, 6-5 to 6-6, 6-8
groups, 6-4 to 6-5, 7-9, 10-6, 10-8, B-3

hardware, 1-4, 2-2, 2-4, 3-2, 4-9, B-2, B-3
heading, 3-23, 5-16
help, 1-3 to 1-4, 1-6, 3-18, 5-11, 5-15, 5-22, 7-22, 11-3, 11-5, B-2
Hercules, 2-3
highlighted, 3-5, 3-7 to 3-8, 8-8, 8-11
Home Directory, 2-4, 3-3, 13-2, 13-8, 13-11, B-2
Host System Files, 1-4, 3-11
HOSTERR.TXT, 13-2, 13-3

IBM, 1-5, 2-2 to 2-3, 2-5, 3-5, 5-16, 5-18, 11-6, F-2
Identifier, 3-9, 5-17
improvements, 1-5 to 1-6, D-2
increment, 11-3, 11-6, 11-8, 13-2, 13-8 to 13-9
Increment Between Levels, 5-7 to 5-9, 6-2, 10-4, 11-4, 11-8, 13-7
input, 1-2, 2-5 to 2-6, 3-11, 3-13, 4-3, 5-2
insert, 1-5, 2-3, 3-5 to 3-6, 5-20, 6-7, 7-9 to 7-10, 11-4 to 11-5, 11-9, 13-8, C-4, F2
Instruct Menu, 4-3, 4-6, 11-2, 11-4, 11-7, 11-9 to 11-11, 12-3, C-2
Instruction,
  File Editor, 3-5, 3-17, 11-2 to 11-6
  File Processor, 11-4, 11-10, 13-4,
File Processor (cont.), 13-7, 13-9
  file, 1-3, 3-8, 3-11, 3-15 to 3-17, 4-3 to to 4-4, 4-6, 9-3, 10-6, 11-2 to 11-7, 11-9 to 11-12, 13-2, 13-7 to 13-9
integer, 1-2, 5-7 to 5-8, 5-13 to 5-19, 5-21, 6-2 to 6-8, 7-6 to 7-7, 8-7, 10-4
Intel, 2-4, 5-14, 5-18, F-2
interactive, 1-2 to 1-3, 1-5, 3-13, 9-3, 11-5 to 11-7, E-3
interruption, 3-17
interval, 6-3 to 6-6, 8-7, 10-6
Interval Label, 6-4
invalid, 3-8 to 3-10, 5-15, 8-12, 10-6, 11-10
IOMEGA, 2-4
italicized, 1-5
italics, 4-9

Kaypro, 2-3
key,
  notation, 1-5
  selection, 3-7 to 3-8
keyboard, 1-3 to 1-5, 2-2, 3-5, F-2
  mnemonics, 1-5

Label,
  Format, 3-8, 3-15, 3-18, 3-20, 5-7 to 5-9, 6-2 to 6-9, 7-8 to 7-11, 7-13 to 7-15, 12-10, F-2, F-3
  Library, 3-11, 3-14 to 3-15, 3-20, 5-8, 6-2, 6-7 to 6-9, 7-14, 8-2, A-2, B-3
  Menu, 6-3, 6-8 to 6-10
labeling, 1-2 to 1-3, 5-7 to 5-9, 8-8, D-2
labels, 1-3, 3-14 to 3-15, 5-7 to 5-8, 5-18 to 5-19, 6-2, 6-5, 6-8, 7-8, 8-7, 10-4, 10-9
language, 1-2, 3-12, 5-2, 7-8
lap-top, 2-4, B-2
largest, 5-5 to 5-6, 6-5, 8-6
left margin, 4-6
length, 3-9, 4-7, 5-13, 7-5, 10-8
Library Files, 3-11, 3-14, 12-2, A-3
line length, 4-6, 4-7, 7-5, 10-8
locate, 1-3, 1-6, 3-8, 3-18, 5-10, 5-22, 6-8, 7-17
Lotus, 3-23, 4-8, 8-8, 8-10 to 8-11, 10-4, 10-8 to 10-9, 11-3, 11-8, 13-6, 13-8 to 13-9, D-2, E-3
Lotus-import file, 4-5, 10-2, 10-4, 10-7, 10-8, 11-4, 11-8, 13-6, 13-8, 13-9

lower,
  bound, 6-2, 6-4 to 6-5, 6-7, 8-7
  case, 3-9, 4-8, 5-17, 8-10, 11-3, 13-4

Main Menu, 3-2 to 3-4, 3-10, 3-23, 4-2 to 4-3, 5-3 to 5-4, 5-11, 7-22, 8-2, 8-12, 8-14, 9-2 to 9-4, 10-2, 10-6, 11-3, 11-11, 12-2 to 12-4, 12-8, A-2 to A-4, B-3, F-5 to F-6
manage, 1-3, 12-2
Manage,
  Data Menu, 12-3 to 12-5, 12-7
  Library Menu, 12-8, 12-11
  Menu, 3-17, 4-3, 4-6, 12-2 to 12-3, 12-12
Manager, 1-4, 3-2, 3-10 to 3-11, 5-3 to 5-4, 5-11, 5-14 to 5-15, 5-18
  access, 3-10
matrix, 5-19 to 5-20
maximum, 2-4, 4-7, 5-6, 5-15, 5-21, 9-3, D-2
Maximum,
  Effective Level, 5-7
  Number of Levels, 5-5 to 5-9, 5-20 to 5-21, F-2, F-5
  Value, 5-14 to 5-15, 5-18, 8-12
memory, 2-2, 2-5, 3-12, 3-16, 3-20, 3-23, 4-2, 5-5 to 5-6, 8-2, 8-12, 8-14, 10-6 to 10-7, 11-3, B-2
  space, 5-5, 8-14
Menu Space, 3-4, 3-7 to 3-8, 3-18, 8-3, 8-12, 9-3, 10-6, 11-3, 11-11
Message Space, 3-4, 8-3, 10-6, 12-6 to 12-7, 12-10 to 12-11
microcomputer, 1-2 to 1-5, 2-2, 2-4 to 2-6, 3-3, 3-5, 3-17, 3-23, 5-14, 5-18, 6-5, 8-2, 9-3 to 9-4, 10-6, 11-3, 12-4, 12-8, B-2 to B-3, F-2
minimize, 5-3, 8-3
Minimum Value, 5-14, 8-12
mistake, 8-11 to 8-12, F-3
Mode,
  Indicator, 3-4, 9-4, 10-6
  Space, 3-4
model, 1-2 to 1-3, 2-5, 3-13, 4-3, 5-5, 5-9, 5-19 to 5-20, B-2
modify, 2-2, 2-4, 5-5, 5-12, 5-15, 6-8, 7-22, 11-2, 11-6
module, 1-2 to 1-3, 2-3, 3-13, 5-3, 9-2 to 9-4, 13-4, A-2, A-4, B-2, E-3
module developer, 1-2
monitor, 2-2, 3-11, 4-9
monochrome, 2-2, 4-9
Monochrome Display Adapter, 2-2, 2-3
MS-DOS, 2-2 to 2-4

multi-dimensional, 1-2 to 1-3, 5-20, E-3

nested, 7-9, 7-11, 7-13 to 7-17, 7-20, 10-7 to 10-8
non-consecutive, 6-5
non-document, 2-4, 2-6
non-periodic, 7-6 to 7-7
non-numeric, 5-13 to 5-17
nonperiodic, 5-8 to 5-9, 5-19 to 5-21, 8-3 to 8-6, 10-7
Norton, 3-4
notation, 1-5, 3-6, 3-9, 5-15 to 5-19, 5-21, 7-10, 8-3 to 8-4, 8-9, 11-3, 13-3
Number of,
  Dimensions, 5-19 to 5-21, F-2
  Periods, 9-2 to 9-3
numeric, 2-4, 5-2, 5-11, 5-13 to 5-16, 5-18, 8-3, 8-11 to 8-12, 10-9, F-2
  coprocessor, 2-4, 5-14, 5-18, F-2

one-dimensional, 5-20, 8-5
open, 2-4 to 2-5, 6-6, 10-8
open-ended, 6-5
operating system, 1-2 to 1-6, 2-2, 2-4, 3-2, 3-10, 3-15, 3-23, 11-2 to 11-4, 13-2, A-4, B-2 to B-3, B-5, F-2
organization of User Guide, 1-3
output, 1-2, 3-8, 3-11, 3-13, 4-2 to 4-4, 4-9, 5-2, 5-16, 7-4 to 7-5, 10-3 to 10-8, 11-4, 11-7 to 11-8, 11-11, 13-5 to 13-6
  device, 10-4, 11-7, 13-5
over-type, 2-3, 3-5
overlapped, 10-9
overwrite, 8-13, 12-3 to 12-4, 12-8

pack, 5-10, 5-22, 6-9, 7-22, 12-2 to 12-3, 12-7 to 12-8, 12-11, A-3
page,
  eject, 4-6, 4-8
  length, 4-6 to 4-8
Paradise, 2-3
Pascal, 1-2 to 1-3, 1-6, 3-9, 3-11 to 3-13, 3-23, 5-16, 12-12, A-2, A-4, E-3
password, 3-10 to 3-11
PATH command, 1-5, 2-2, 2-5, 3-3, 3-12, 12-12, 13-2, 13-4, 13-11
PC-DOS, 2-2 to 2-4, F-2
percent, 5-13, 7-7 to 7-9, 7-15, 7-17
performance, 2-5
period, 8-10 to 8-12, 9-2, 10-6, 11-7 to 11-8, 13-4, 13-6, D-2

PERIOD Dimension, 8-10
period list, 5-9, 10-4
periodic, 1-2, 5-8, 5-19 to 5-21, 7-2, 7-6 to 7-7, 8-3 to 8-7, 8-10 to 8-11, 8-13, 10-4, 10-7
permitted, 2-4, 5-19, 11-3
plotter, 3-20, 4-4
presentation, 1-2, 1-4, 6-2, 6-8, 3-13
print, 2-2, 4-6, 4-8, 4-9, 5-4, 10-4, 10-7, 12-6, 12-9, 12-12, 13-3, F-2
  spoolers, 2-2
print-ready file, 3-20, 4-5, 10-2, 10-4, 10-7, 10-8, 11-8, 13-6
printer, 3-20, 3-23, 4-6 to 4-9, 6-10, 7-2, 10-2, 10-4, 10-6 to 10-8, 11-7 to 11-8, 12-6, 12-9, 13-2, 13-5 to 13-6, B-3
printing, 3-6 to 3-7, 3-15 to 3-16, 3-20, 4-6 to 4-8, 10-4, 10-7, 12-12, 13-2, A-4
printouts, 5-9
PRN, 10-4, 10-7 to 10-8, 11-7 to 11-8, 13-5 to 13-6
problems, 1-4, 1-6, 2-2, 3-4, 10-9, B-3
programming, 1-2, 5-2, 12-4, E-3
projecting, 3-13, 5-9, 9-3, 11-4, 11-7, 13-4
projection, 1-2, 5-9, C-4
Projection Module, 5-9, 5-19 to 5-20, 9-3
protect, 3-10 to 3-11
protected, 5-4, 5-11

Query Field, 4-3, 5-4, 5-8, 5-10, 5-11, 5-20, 5-22, 6-8, 6-9, 7-9, 7-14, 7-17, 7-22, 7-23, 8-2, 8-14, 9-2, 10-2, 11-6, 11-9, 11-10, 11-11, 12-3, 12-4, 12-5, 12-7, 12-8, 12-9, 12-11

RAM, 1-4, 2-2 to 2-3, B-2, B-3
range, 1-2, 5-7, 5-13 to 5-19, 6-3 to 6-5, 8-8, 8-12, 13-9
real, 1-2, 5-13 to 5-19, 7-6 to 7-7, 8-3, 8-4
rebuild, 12-5 to 12-6, 12-9, 12-10
recommend, 1-4, 2-2, B-2
record, 1-6, 5-13, 5-18, F-3 to F-6
recover, 5-10, 5-22, 6-9, 7-22, 12-7, 12-11, F-5 to F-6
reference Data File, 7-2, 7-22, 7-23, 10-7
registering your copy of Host, 1-6
relational, 3-13
removable, 4-5, B-3
renaming, 3-20 to 3-21, 5-11, 5-23, 6-10, 7-22, 12-3, 12-8, 13-7 to 13-9, A-3
repair, 3-17
report, 3-7, 3-20, 3-23, 4-5, 4-7 to 4-8, 5-11, 5-23, 6-9, 6-10, 7-2, 7-22, A-2, E-3
Report Menu, 5-11, 5-23, 6-10, 7-22
reporting problems, 1-6
reports, 3-20 to 3-21, 4-6, 4-8, 5-4, 5-11 to 5-12, 5-18, 5-23, 6-4, 7-22, 11-9, 12-2, 12-4 to 12-5, 12-7, 12-9, 12-11
resident, 2-2, 2-5, 3-23, 11-3
restriction, 5-18, 7-13, 7-14, 11-3
results, 1-2 to 1-3, 3-12, 3-17, 5-18, 8-12, 9-3, 12-4, 12-8, D-2
right margin, 4-7 to 4-8
root, 2-4 to 2-5
row dimension, 8-4, 8-6, 10-8, C-2
rules, 1-5, 3-8 to 3-9, 5-4, 5-11, 5-16, 6-7, 7-4, 11-7, 13-5
run-time, 11-12, 13-3, C-2

saving, 4-3, 4-9, 5-10, 5-22, 6-8, 7-15, 8-13, 10-4, A-2
scalar, 1-2, 5-19
scale, 1-2, 3-13
Scaling Exponent, 5-14, 5-18 to 5-19, 7-14, 8-4, D-2, E-3
scenarios, 1-3
scientific notation, 5-15, 8-4
search, 2-4, 2-5, 3-12, 3-18, 4-5, 4-6, A-4, B-2, F-2, F-4
segment, 7-5
sequence, 1-3 to 1-4, 1-6, 2-6, 3-4, 3-11, 3-16, 5-10, 5-13 to 5-14, 5-19, 5-21, 6-7, 7-17, 10-8, 11-2, 11-5
series, 1-3, 3-15, 4-8, 5-20, 6-5, 10-7
session, 2-5, 3-12, 3-17 to 3-18, 4-2, 4-9, 12-4, 12-8
setup string, 4-6, 4-8 to 4-9, 10-4
size, 3-16, 4-6 to 4-7, 5-13 to 5-14, 5-20 to 5-21, 6-3 to 6-5, 10-7, 12-5, 12-7, 12-9, 12-11, F-2, F-4 to F-5
Size of Interval, 5-7, 8-7 to 8-8
slide-shows, 1-3, D-2
software, 1-2 to 1-3, 2-4, 2-6, 4-8, 10-4, B-2, D-2, F-2
source file, 11-9, 12-4
spaces, 3-9, 4-6, 4-8, 11-3, 13-3 to 13-4
Spanish, 7-8

Sperry, 2-3
spreadsheet, 4-8, 10-4
SPSS/PC+, 11-3
Starting,
  Level, 5-7 to 5-9, 6-2, 8-6, 11-4
  Period, 9-2 to 9-3
storage, 2-3, 2-5, 3-13, 3-15 to 3-16, 5-6, 5-13, 11-2, 12-2, B-2, C-4
string, 1-2, 4-6, 4-8 to 4-9, 5-13 to 5-14, 5-16, 5-18 to 5-19, 6-4, 8-3, 10-4
sub-columns, 7-13
subdirectory, 1-4, 2-4, 4-4, 11-4, 13-9 to 13-12
subdivide, 3-14 to 3-15, 6-5
subgroups, 6-3
subsystem, 1-2
subtable, 7-2, 7-7, 7-13 to 7-16
subtotals, 7-9
sum-over, 7-14 to 7-15
suspend, 11-3, 11-6 to 11-7, 11-10 to 11-11, 13-2, 13-8 to 13-9
symbol, 1-5, 2-5, 3-5, 3-8, 8-6
syntax, 11-3 to 11-11, 13-2 to 13-7, A-2
  error, 11-10 to 11-11, 13-3

tab, 7-5, F-2
Table,
Definition, 3-8, 3-15, 3-18, 7-2 to 7-22, 10-2 to 10-3, 10-7 to 10-8, 11-4, 11-7, 12-10, 13-5, A-3, C-2, F-2 to F-3
  Generator, 3-13, 3-15, 6-2, 7-2, 7-7, 10-2 to 10-8
  Library, 3-11, 3-14 to 3-16, 7-2, 7-17, 7-22, 10-2, 11-7, 13-5, B-3
  Menu, 7-2, 7-4, 7-17, 7-22, 7-23, 10-2, 11-11, A-3
tabular, 1-4, 7-4, E-3
target file, 11-9, 12-4
text file, 10-4, 11-2, 12-12, A-4
Toggle Field, 3-20, 4-4, 8-10
toggle indicator space, 3-4
top margin, 4-6
top-of-form, 4-8
total, 1-4, 3-11, 4-7, 7-4, 7-9, 7-14 to 7-17, C-2
tree-structured directories, 1-5, 13-9
Turbo Pascal, 1-2 to 1-3, 1-6, 3-9, 3-11 to 3-13, 3-23, 12-12, A-2, A-4, E-3
two-dimensional, 8-4, 8-6
typographical conventions, 1-5

undo, 3-6, 7-10, 8-9
Unit of Measurement, 5-18 to 5-19, 8-4
Unnested, 7-13 to 7-14, 7-16 to 7-17, 7-21, 10-7 to 10-8, C-2
unprotected, 5-4, 5-11
upper bound, 6-2, 6-4 to 6-5, 6-7, 8-7
upper case, 3-8 to 3-10, 5-17
User, 1-3 to 1-4, 1-6, 3-2, 3-10, 5-4, 5-11, 5-14, 5-18, 13-5 to 13-6, B-3
user support, 1-6
utilities, 1-4, 2-2, 3-4, 12-2, 12-12, A-4, B-2

valid, 3-9, 3-20, 4-5, 5-15 to 5-17, 10-6, 11-2, 11-9, 12-4
Variable,
  Definition, 3-10, 3-14, 5-11 to 5-13, 5-20 to 5-23, 8-3, 12-6, 12-7, F-2 to F-6
  Menu, 5-3, 5-11, 5-22 to 5-23
  Size, 5-20 to 5-21, F-2
  Type, 5-13
vector, 5-20
verification, 1-5, 12-5, 12-8 to 12-9, F-2 to F-6
verify, 2-2, 2-4, 3-3, 12-4 to 12-10, A-4, F-5 to F-6
version, 2-2 to 2-4, 6-8, 8-12, 12-5 to 12-12, A-3, B-2, C-2 to C-3, D-2

Wang, 1-5, 2-3, F-2
warning, 3-16, 6-6, F-5
window, 3-6, 8-4, 8-8 to 8-10, 8-13
WordStar, 1-3, 1-5, 2-4, 2-6, 3-5 to 3-6, 3-9, 5-21, 7-9 to 7-10, 8-9
  notation, 1-5
Work Space, 3-4
write, 1-3, 1-6, 8-12, 9-2
Wyse, 2-3

↔, 7-5
±, 3-20

209