

N84-18921



Technical Memorandum 86062

**PRIMER for the
TRANSPORTABLE
APPLICATIONS EXECUTIVE**

**Patricia A. Carlson, Carmen Ana Emmanuelli,
Elfrieda L. Harris and Dorothy C. Perkins**

January 1984

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

PRIMER
for the
TRANSPORTABLE APPLICATIONS EXECUTIVE

Version 1.0

Patricia A. Carlson
Carmen Ana Emmanuelli
Elfrieda L. Harris
Dorothy C. Perkins

January 6, 1984

NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

TAE PRIMER
Table of Contents

CHAPTER 1	OVERVIEW OF TAE FUNCTIONS	
1.1	AN APPLICATIONS EXECUTIVE	1-1
1.2	TAE FUNCTIONS AND COMPONENTS	1-2
1.2.1	Modes (Managerial)	1-4
1.2.2	Procs And Parameters (Operational)	1-4
1.2.3	Help (Informational/Instructional)	1-5
1.3	USING THE PRIMER	1-5
1.3.1	Organization	1-5
1.3.2	Scenarios And Examples	1-6
1.3.3	Document Conventions	1-7
CHAPTER 2	STARTING A TAE SESSION	
2.1	STEP ONE	2-1
2.2	STEP TWO	2-1
2.3	STEP THREE	2-1
CHAPTER 3	GETTING HELP	
3.1	ON-LINE HELP	3-1
3.2	GENERAL HELP	3-2
3.3	SPECIFIC HELP	3-3
3.3.1	Help In Menu Mode	3-3
3.3.2	Help In Tutor Phase	3-4
3.3.3	Help In Command Mode	3-5
3.3.3.1	HELP COMMAND/PROC	3-5
3.3.3.2	HELP-MESSAGE	3-7
3.3.3.3	HELP-PARM	3-7
3.4	TAE CONFIGURATION	3-8
CHAPTER 4	USING MENU MODE	
4.1	MENU DISPLAYS	4-2
4.1.1	Numbered Entries	4-3
4.1.2	Menu Prompt-Line Options	4-3
4.1.3	Menutree	4-5
4.1.3.1	Accessing Menutree From Menu Mode	4-6
4.1.3.2	Accessing Menutree From Command Mode	4-6
4.2	TUTOR DISPLAYS	4-7
4.2.1	Parameters	4-8
4.2.2	Tutor Prompt-Line Options	4-10
4.3	HELP DISPLAYS	4-11
4.3.1	Explanatory Information	4-11
4.3.2	Help Prompt-Line Options	4-12
4.4	MESSAGE DISPLAYS	4-12

CHAPTER 5	USING COMMAND MODE	
5.1	TRANSITION FROM MENU TO COMMAND MODE	5-1
5.2	TAE COMMAND LANGUAGE (TCL)	5-2
5.2.1	Procs	5-2
5.2.1.1	Proc Inputs	5-4
5.2.1.2	Subcommands	5-4
5.2.2	Intrinsic Commands	5-5
5.3	MODE CHANGING	5-7
CHAPTER 6	CREATING A PROC	
6.1	PROCS	6-1
6.2	PROC CONSTITUENTS	6-2
6.3	WRITING A PROC	6-2
6.3.1	Step One: Proc Definition File (PDF)	6-2
6.3.2	Step Two: Help Files	6-5
6.3.3	Step Three: Create "Body" Or Image	6-6
6.3.4	Step Four: Proc Library	6-9
6.4	PRACTICE SESSION	6-9
CHAPTER 7	GLOSSARY	
APPENDIX A	TAE COMMAND LANGUAGE (TCL)	
APPENDIX B	ULOGON AND ULOGOFF	
APPENDIX C	NOSCREEN TUTOR	

FIGURES AND TABLES

Figures

- 3-1: Tutor Display--Menutree
- 3-2: HELP Text--Tutor
- 3-3: HELP Text--Direc, Menutree Proc
- 3-4: Composite for Menutree

- 4-1: Example TAE Menu
- 4-2: GEMPAK Menutree
- 4-3: Example TAE Tutor Screen

- 5-1: Mode Transition Diagram

Tables

- 1-1: TAE Functions and Components
- 1-2: Organization of the TAE Primer

- 3-1: HELP Requests and TAE Responses

- 4-1: TAE Menu Prompt-Line Options
- 4-2: TAE Tutor Prompt-Line Options
- 4-3: TAE Help Prompt-Line Options
- 4-4: TAE Display Information
- 4-5: TAE Prompt-Line Options

- 5-1: TAE Intrinsic Commands
- 5-2: Summary of Mode Traversals

- 6-1: Procedure-Oriented TCL Commands

TAE PRIMER

DESCRIPTION OF CONTENTS

PREFACE

CHAPTER 1: Overview of TAE Functions

Introduces the concept of an applications executive; the three major functions of the Transportable Applications Executive; the organization of this primer.

CHAPTER 2: Starting a TAE Session

Describes a typical logon procedure.

CHAPTER 3: Getting Help

Provides an overview of the steps for obtaining general and specific help. Discusses the basic configuration of the TAE system.

CHAPTER 4: Using Menu Mode

Explains, in more detail, the four categories of visual displays used in Menu Mode. Sample screens illustrate the format and functions of: menu displays, tutor displays, help displays, and message displays.

CHAPTER 5: Using Command Mode

Considers the differences between operating in Menu Mode and operating in Command Mode. Explains the TAE Command Language (TCL) and tells how to formulate command strings.

CHAPTER 6: Creating a Proc

Discusses the major elements of a TAE proc; explains the fundamental steps in creating a proc; provides a practice session for writing an application program.

APPENDIX A: TAE Command Language (TCL)

APPENDIX B: ULOGON and ULOGOFF

APPENDIX C: NOSCREEN Tutor

GLOSSARY

PREFACE

Audience

This primer is for anyone who wants to use Version 1.1 of the Transportable Applications Executive (TAE). You do not need extensive knowledge of computers; however, some experience with basic keyboard arrangements and interactive functions is helpful for Chapters 4 and 5. In addition, Chapter 6 assumes a basic knowledge of programming.

Before starting, you should have:

- o Access to an operating system with TAE.
- o The ability to log in and out.

Since TAE may operate on a variety of computer systems, this presentation does not cover all possible variations in hardware, environment, and applications. For information on host-specific questions, see your system manager. If you do not understand some of the terms used in this document, consult the Glossary.

TAE Support Office

Address requests and comments to:

TAE Support Office
Advanced Systems Development Branch
Code 933
NASA/Goddard Space Flight Center
Greenbelt, MD 20771.

Telephone: 301-344-6034

In addition to answering immediate questions, the Support Office publishes a TAE newsletter, which appears tri-annually (February, June, October). The newsletter contains updates on TAE documentation, information on user conferences, descriptions of TAE at various sites, and brief articles written by TAE users. To have your name placed on the mailing list, contact the TAE Support Office, NASA/Goddard.

Additional Documentation

After you have become familiar with TAE, you may want to consult other publications which describe the system in more detail. The most comprehensive supplement for a new user is the User's Reference Manual (prepared by Century Computing, Inc., Contract NAS5-27197). If you intend to write your own programs for TAE, the Applications Programmer's Reference Manual (prepared by Century Computing, Inc., Contract NAS5-27197) supplies additional information.

The following annotated list indicates other sources of information you may find useful.

1. Functional Specifications
(Century Computing, NAS5-27197)
Describes the TAE functions.
2. Guide to Writing Menus and Help Texts
(NASA Technical Memorandum No. 84980)
Information on writing and formatting TAE screen text.
3. System Manager's Guide
(Century Computing, NAS5-27197)
Information on installing and maintaining TAE.
4. Utilities Reference Manual
(Century Computing, NAS5-27197)
Directions on using TAE utility features.
5. Release Notes for Version 1.1
(Century Computing, NAS5-27197)
Specific features of Version 1.1

Contact the TAE Support Office for copies.

CHAPTER 1
OVERVIEW OF TAE FUNCTIONS

Chapter Sections

- 1.1 An Applications Executive
- 1.2 TAE Functions and Components
 - 1.2.1 Modes
 - 1.2.2 Procs and Parameters
 - 1.2.3 Help
- 1.3 Using the Primer
 - 1.3.1 Organization
 - 1.3.2 Scenarios and Examples
 - 1.3.3 Document Conventions

TAE was developed by the Information Extraction Division, Applications Directorate, of the NASA/Goddard Space Flight Center. This Division designs interactive image analysis and data management systems to support weather, climate, and earth resources research using satellite data.

1.1 AN APPLICATIONS EXECUTIVE

On an interactive computer system, the user gives directions for the machine's operation. For most applications, the various elements of the system are so complex that the user cannot communicate directly with the operational programs, but must manage the system through an intermediate interface.

An applications executive is a master program which unifies the many subprograms of the system. In general, an executive should facilitate the user's interaction with the

OVERVIEW OF TAE FUNCTIONS
AN APPLICATIONS EXECUTIVE

computer by:

- o Standardizing the user interface to application programs.
- o Shielding the user from the host operating system.
- o Providing a congenial environment.

1.2 TAE FUNCTIONS AND COMPONENTS

TAE contains several parts whose functions and interrelations may be difficult to understand if you are a new user. An analogy will clarify things.

Complex organizations--be they social institutions, physical organisms, or computer systems--are made up of functional domains. Most industrial organizations, for example, have a managerial component (in charge of planning and directing the overall company effort), an operational component (where the nuts-and-bolts activities of the company are performed), and an informational/ instructional component (which disseminates knowledge about the company and provides for training). The TAE system is similar to this three-part industrial organization in that it has managing, operating, and informing capabilities.

To get an idea of TAE as a system, keep in mind three basic concepts:

1. Management mode, refers to the methods by which you address the system's aggregate of resources and services.
2. Procs (processes or procedures) and parameters, refer to the resources and services of the system and how you select a feature and adapt it for a specific situation.

OVERVIEW OF TAE FUNCTIONS
TAE FUNCTIONS AND COMPONENTS

3. On-line help, refers to the various kinds of information-- available interactively--to aid you, either on an individual topic or on the use of the system as a whole.

Educational psychologists advise that we learn a complex process best when we are given a conceptual overview--or a model--of the process. In the case of TAE, this can be done by breaking the system into its major functions and then identifying the part which performs each function.

To build a model for TAE, consider the three major types of activities carried out by the executive system and the components associated with each. Table 1-1 summarizes these relationships.

Table 1-1: TAE Functions and Components

<u>Function</u>	<u>Component</u>	<u>Description</u>
Managerial	Modes	Methods of addressing the operational components which will accommodate the user's individual level of experience with TAE.
Operational	Procs and Parameters	A collection of programs which perform the analytical and house-keeping activities necessary to complete required tasks.
Informational/ Instructional	Help	A means of supplying both supplemental information on a single topic and a systematic teaching tool for the new user.

The next three sections explain more fully these basic TAE components and functions.

OVERVIEW OF TAE FUNCTIONS
TAE FUNCTIONS AND COMPONENTS

1.2.1 Modes (Managerial)

With TAE you may send directions to the computer in one of two modes:

1. Menu Mode, in which you select an operation from a numbered list presented on the screen.
2. Command Mode, in which you communicate with the system through the TAE Command Language (TCL).

Which mode you select depends upon the task you want to perform, how well you know the features of your system, and your own confidence in using TAE.

1.2.2 Procs And Parameters (Operational)

Procs: A proc executes one of two things: (1) a requested application or (2) a utility feature. Application procs usually perform analytical tasks. Utility procs usually perform service functions.

Parameters: Once you invoke the proc you want, you may adapt it to your own use. Each proc has a parameter or set of parameters which may be given different values when the program is run for a specific reason.

Defaulted Parameters: In many cases, a parameter will have a predefined value (based on what is most convenient or most commonly used). This value is the "default"; if you do not replace it with another value, the program executes using this value. Whether you are working in Menu Mode or in Command Mode, you will have the option of entering parameter data or accepting a default value before activating the proc.

OVERVIEW OF TAE FUNCTIONS TAE FUNCTIONS AND COMPONENTS

1.2.3 Help (Informational/Instructional)

To make the system as "self-teaching" as possible, TAE includes on-line explanation. Unlike some systems, help instructions for TAE will not appear of their own accord when the computer detects that the operator is making mistakes at the terminal. Although you can obtain on-line help at all times during the TAE session, you must request it.

Help is available in several different forms on TAE. No matter what mode you are in, the system can provide you with assistance on almost any item. You will find more details on various types under the appropriate section headings in this primer.

Use the help facility to supplement this primer. In most cases, this primer will not reproduce help information since it is already available in a convenient form. However, this document will indicate--when appropriate--that supplemental information is available on-line.

1.3 USING THE PRIMER

This primer covers the basic information needed to operate TAE. It also contains information on how to write TAE programs. Depending on your prior experience, you may want to read this material carefully or just skim it for specific details. To aid you in locating and understanding information, each of the chapters is further broken down into subsections. In addition, a "Quick Reference Summary" appears at the end of each chapter.

1.3.1 Organization

TAE has three functional domains:

1. Informational/Instructional (the explanatory and training information),

OVERVIEW OF TAE FUNCTIONS
USING THE PRIMER

2. Managerial (the user interface with the operating portions of the system),
3. Operational (the application and utility programs of the system).

The organization of this primer reflects the three performance domains of TAE, as Table 1-2 indicates.

Table 1-2: Organization of the TAE Primer

<u>Domain</u>	<u>Chapter</u>	<u>Subject</u>
Instructional	3	On-line Help
Managerial	4	Menu Mode
	5	Command Mode
Operational	6	Procs

The plan is to examine the system on progressively more complex levels by presenting the system through each of the three functional domains.

1.3.2 Scenarios And Examples

This text takes you through the basic functions of the TAE system in a step-by-step process. You may follow the scenarios in one of two ways:

- o On-line: by actually performing the tasks discussed in the primer while at the terminal.

OVERVIEW OF TAE FUNCTIONS USING THE PRIMER

- o Off-line: by following the tasks as they are simulated in the text.

The examples in the scenarios have, to some extent, been tailored to your specific application of TAE. In some cases the simulations in the text will represent what you see on your screen. When such adaptation was not possible, the examples and simulated screens come from GEMPAK--an application of TAE to meteorological data.

1.3.3 Document Conventions

This document uses several typographical conventions to ensure ease and consistency of interpretation. Keep these points in mind while reading the examples of user requests and TAE responses.

- o The prompt for Command Mode is site-specific. However, this document assumes "TAE>" as the standard.
- o Many elements of TAE's terminology are also words with everyday meanings (such as "help"). Whenever a technical term might be misread, the word is placed in quotation marks.
- o A vertical ellipsis in an example means that not all the data a user would normally enter is shown.
- o TAE makes no distinction between upper and lower case for user input. That is, HELP is the same as help. This document uses upper case for the sake of readability.
- o Commands and keywords may be abbreviated, even down to a single letter, so long as they are unique. The examples in this presentation use full command and keyword names to ensure clarity.
- o In all examples of user input in this document, TAE responses are underscored. The part not underlined represents the text typed by the user.

OVERVIEW OF TAE FUNCTIONS

QUICK REFERENCE SUMMARY

1.1 An Applications Executive

Many computer activities use multiple programs to perform a task. For some applications, these programs are numerous, complex, and interrelated. Thus, the user needs help in managing the various operations.

An executive system provides an interface between the aggregate of programs in a particular computer application and the user. It gives the user a flexible way to manage the resources of the host system.

1.2 TAE Functions and Components

As a system, TAE contains three functional domains: (1) Informational/Instructional, (2) Managerial, and (3) Operational.

The first domain provides explanation and training on the system through the Help component. The second domain allows the user to give directions to the system through the Menu and Command Mode components. The third domain supplies the analytical and service features of the system through the proc and parameter component.

1.3 Using this Primer

The major divisions of the primer present the three functional domains of the system. Chapters 3, 4, 5, and 6 contain increasingly more complex views of the system, each requiring more expertise on the part of the user. While the presentations in each chapter build upon the previous chapter, each segment is also modularized so that a user may begin at whatever point feels most comfortable.

CHAPTER 2

STARTING A TAE SESSION

2.1 STEP ONE

Use your conventional logon procedure. Most systems have been configured so that the user automatically enters TAE by logging onto the host computer. See your system manager for instructions if this does not happen.

2.2 STEP TWO

Greeting messages from the host system may appear on the screen, followed by a terminal-type prompt similar to:

ENTER TERMINAL TYPE (VT100, VT52, T4027, OTHER):

Respond with the appropriate designation, followed by a carriage return (CR). On systems where the terminal-type is already known or where the type can be determined automatically, this prompt does not appear.

2.3 STEP THREE

Next, TAE identification and system-specific announcements may appear. Following this, TAE will present either the root menu or the Command Language prompt (which may be "TAE" or a local symbol). If you need assistance, type HELP, and TAE will explain what you can do at that point in your processing. Chapter 3 contains a detailed explanation of HELP.

CHAPTER 3

GETTING HELP

Chapter Sections

- 3.1 On-Line Help
- 3.2 General Help
- 3.3 Specific Help
 - 3.3.1 Help in Menu Mode
 - 3.3.2 Help in Tutor Phase
 - 3.3.3 Help in Command Mode
- 3.4 TAE Configuration

Chapter 1 likened TAE to an industrial organization, in that it has informing, managing, and operating features. This chapter approaches the system by examining its ability to inform and to instruct through the on-line help facility. Since the subjects of the help texts are management and operational features, this chapter can also serve as a system summary for TAE.

3.1 ON-LINE HELP

This chapter tells you how to call up the information you need if you run into trouble at any point in your TAE session. TAE is intended for users who are teaching themselves the system, while at the same time doing work-related tasks. Use the next sections to get the kind of help you need, when you need it.

Many users find the HELP component most valuable for calling up information in the middle of a TAE task session. A user might, however, simply browse through parts of the help features without doing any tasks.

GETTING HELP ON-LINE HELP

This browsing capability helps you learn the system because it allows you to get answers to your questions at the time they occur to you. For instance, performing a certain TAE task may suggest questions about associated aspects of the system.

Remember, however, while browsing you may not be able to activate or otherwise operate the topics you read about. For example, you can request information on Command Mode functions without leaving Menu Mode. But you will not be able to activate these functions without switching modes.

3.2 GENERAL HELP

General help texts are linked to the stages of completing a task with TAE. You have already been introduced to two of the three major stages: Menu and Command Mode. The third, Tutor phase, requires some explanation.

As an introduction to Tutor phase, let's review what you already know about TAE's interactions:

- o You may address TAE in one of two modes: menu or command.
- o Your objective in both modes is to access a proc that will do the task you want.
- o Before a proc executes, you have an opportunity to define the parameters.

When you reach the point where you can enter these values you are in the Tutor phase.

At any time in a TAE session--except while the proc is actually running-- you are in one of three phases: Menu, Command, or Tutor. If you simply type "HELP," you will get the general text pertaining to whichever of these three phases you are in at that time.

If your terminal is a CRT recognized by TAE, when a help display is entered, you are presented with the first page of the requested information, and a prompt line. You can then page through the help data, or exit. If your terminal is a hardcopy terminal or an unrecognized CRT, TAE lists the help information line by line, with no break on page boundaries and no prompts.

GETTING HELP SPECIFIC HELP

3.3 SPECIFIC HELP

You can obtain assistance on a specific topic by restricting the HELP request. The next sections tell you how to do this.

3.3.1 Help In Menu Mode

You may request specific help in Menu Mode by using one of three qualifiers:

- HELP n: Help on menu entry number n.
- HELP *: Help on the current menu.
- HELP proc: Help on the named proc/command.

Both "Help n" and "Help *" are menu-specific: they supply commentary on the current menu or on its numbered entries. After you have read the help text and typed "EXIT," you return to your original menu. Once you are back at the menu, you can access the particular item you were seeking help on. You cannot, however, ask for information on topics that are not contained on the immediate menu by using "Help n" or Help *."

"Help proc" allows you to call up information on any proc or command in the TAE system, while remaining in Menu Mode. But you will not be able to activate that particular item unless it is on the menu immediately before you. "Help proc," in Menu Mode, is used for browsing purposes rather than in conjunction with running a specific task.

This potential to browse is an instructional tool of TAE and is especially valuable to the intermediate learner because it allows you to request information on topics other than those presented on the immediate screen display.

For example, "Help proc" allows you to obtain information on topics that reside in Command Mode while you are still in Menu Mode. This is crucial for comparing operation and management features. It also aids in planning a strategy for your session.

GETTING HELP
SPECIFIC HELP

As a second example, "Help proc" also allows you to get information on procs not listed on the current menu, or perhaps not listed on any menu. By using "Help proc", you can read and compare the features of various procs to build up your understanding of your system's capabilities.

3.3.2 Help In Tutor Phase

Assume you have entered the menu number for the proc "Menutree." Your session path moves down to the Tutor phase and a tutor screen appears, prompting you to enter parameters.

Tutor: proc "MENUTREE", library "(TAESO.UTILITY)"

Pg 1.

Make a menu tree for a given menu

<u>parm</u>	<u>description</u>	
MENU	Name of the menu to be listed	"ROOT"
DIREC	Directory containing the menu	"TAESMENU"
OUTPUT	Output destination	"T"

Enter: parm=value, HELP, PAGE, SHOW, RUN, EXIT, SAVE, RESTORE; RETURN to page.
?

Figure 3-1: Tutor Display--Menutree

GETTING HELP SPECIFIC HELP

But you may need assistance before you can supply the parameter values--for example, a more complete definition of the parameter or more precise instructions on entering the target values. Tutor phase has two specific help requests:

Help *: produces information on the current proc.

Help parm: produces the help text on the parameter you specify.

3.3.3 Help In Command Mode

Enter Command Mode by typing "COMMAND." The system will respond with a prompt for more input.

If you type "HELP" by itself, information on all commands available in Command Mode is displayed. This is the general information text.

As in Menu Mode, you can obtain help on a specific topic by restricting the HELP request. To do this, you need to know the basic syntax of the TAE Command Language (TCL). For an explanation of TCL syntax, see Chapter 5 and Appendix A.

Request specific assistance while in Command Mode by using one of three restrictors:

HELP COMMAND/PROC

HELP-MESSAGE

HELP-PARM

An example of each follows. Each illustration uses the full command string. You may, however, enter abbreviations--so long as they are unique.

3.3.3.1 HELP COMMAND/PROC

The HELP request followed by the name of a command or proc calls up information on that particular command or proc. The string format is

TAE>HELP NAME

where NAME is the command or proc designation.

GETTING HELP
SPECIFIC HELP

For example, if you want assistance on the TUTOR command,
you enter:

TAE>HELP TUTOR

TAE responds with

Help: command "TUTOR"

Pg 1+

TUTOR

Function and Format

Function:

Puts TAE in TUTOR mode.

Format:

1. TUTOR PROC=proc-name
2. TUTOR

Enter EXIT or PAGE n (or press RETURN for next page)

?

Help: command "TUTOR"

Pg 2.

Usage Notes

When you specify "TUTOR proc", proc is the name of the proc for which TUTOR data will be displayed.

If you specify "TUTOR" without a proc name, data will be displayed on the proc of the most recent tutor RUN command which was unsuccessful due to parameter specification error. This is called "TUTOR latching".

Enter EXIT or PAGE n (or press RETURN for next page)

?

Figure 3-2: Help Text -- TUTOR

GETTING HELP
SPECIFIC HELP

3.3.3.2 HELP-MESSAGE

TAE can display brief messages to the user. Most often, the text will consist of error messages, which indicate what has gone wrong. But the message can also be used to communicate some information from a running program.

Every message is a one-line comment preceded by a "message key," which you can use to find more explanation about the message. For example, in the message line

```
[TAE-UNRECCMD] Unrecognized command
```

the portion in brackets is the message key.

If you need help on a particular message, the keyword format is

```
TAE>HELP-MESSAGE KEY=message key
```

In order to obtain help on the example message, you would use

```
TAE>HELP-MESSAGE KEY=TAE-UNRECCMD
```

to enter the request using the keyword format.

Using the positional format, you enter

```
TAE>HELP-MESSAGE TAE-UNRECCMD
```

If you do not enter a value for KEY (TAE-UNRECCMD, in the example), or if you enter "?", information is displayed on the message associated with the last error detected by TAE.

3.3.3.3 HELP-PARM

HELP-PARM requests information on the named parameter of a proc. The keyword format is

```
TAE>HELP-PARM PARM=PARAM-NAME PROC=PROC-NAME
```

If you want information on the parameter "direc" of the proc "menutree," you enter

```
TAE>HELP-PARM PARM=DIREC PROC=MENUTREE
```

GETTING HELP
SPECIFIC HELP

Or, using the positional format:

```
TAE>HELP-PARM  DIREC  MENUTREE
```

TAE responds with

Help: parameter "direc", proc "menutree"

Pg 1.

DIREC is the logical or literal name of the
directory where the menu "MENU" is located.

Enter EXIT or PAGE n (or press RETURN for next page)
?

Figure 3-3: Help Text -- Direc, Menutree Proc

3.4 TAE CONFIGURATION

The examples in this chapter illustrate what kind of help is available and how you get the type of assistance you need--at the time you need it. The examples also demonstrate the global features of TAE's configuration: a TAE system is essentially a collection of application and

GETTING HELP
TAE CONFIGURATION

utility procs (and their attendant help texts) which can be addressed either through Menu Mode or Command Mode.

Figure 3-4 gives a composite of the frames associated with the proc "menutree." This configuration should help you to visualize the basic pattern of a typical TAE system.

For on-line practice, try adapting the instructions in Figure 3-4 to call up the screens associated with a proc other than "menutree."

GETTING HELP
TAE CONFIGURATION

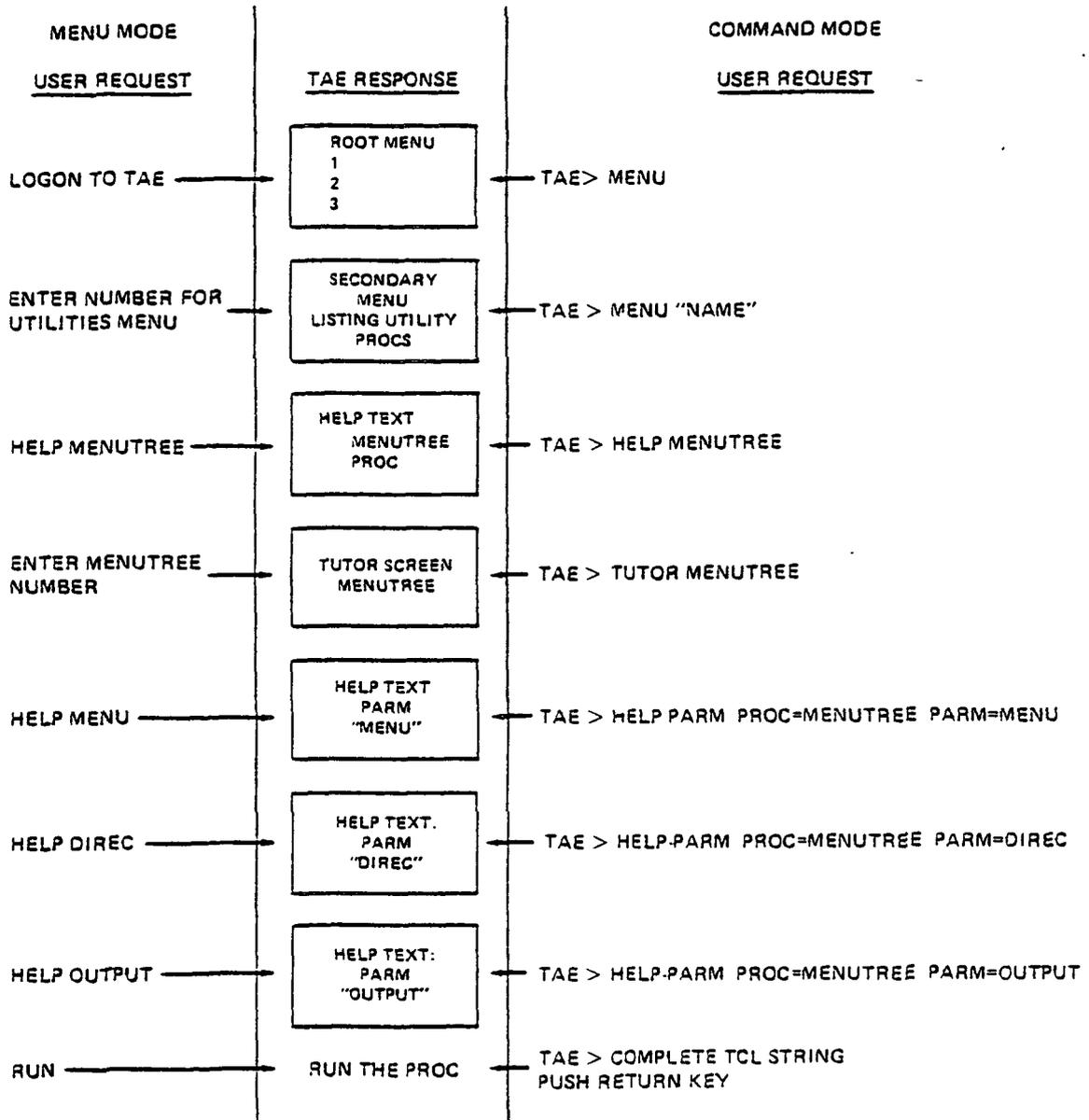


Figure 3-4: Composite for Menutree

GETTING HELP

QUICK REFERENCE SUMMARY

3.1 On-line Help

To provide an effective interface, an executive must unify a number of complex operations and present the control of a particular task to the user through a flexible and easily-learned system. The executive must be self-contained: that is, it should be able to inform the user of its resources and give instructions for their use.

"Help" on a selected topic may be obtained by a user in any mode. The types of help information available are

- o Information on a proc or a TAE command
- o Information on a menu
- o Information on the operation of a mode
- o Information on the parameters of a proc
- o Information on a TAE message

3.2 General Help

To make getting help as easy as possible, the designers have linked all information to a simple request--HELP. This request may be modified or not, depending on the kind of help you want at that particular phase of your session. In short, TAE is designed so that the help text is timely and useful.

You may enter the HELP request at any point in a session: in many cases, the information you receive is associated with the phase of TAE you are in. This is especially true for the unrestricted HELP request.

3.3 Specific Help

Although there are relatively few help requests, their syntax and distribution might be confusing. Use Table 3-1 to review the HELP request formats and the TAE responses.

GETTING HELP

Table 3-1: Help Requests and TAE Responses

REQUEST	RESPONSE		
	Menu	Command	Tutor
HELP	General explanation on management features in the three major phases. Instructions on getting more specific information.		
HELP *	The current menu	N/A	The current proc
HELP n	Current menu entity designated by "n."	N/A	N/A
HELP proc	The named proc or command	The named proc or command	N/A
HELP COMMAND	The named proc or command	The named proc or command	N/A
HELP-PARM	N/A	The named parameter	N/A
HELP-MESSAGE	N/A	The named message key	N/A
HELP parm	N/A	N/A	The named parameter

You probably have noticed that you can access the same help text in several different ways. Such redundancy is a planned feature of TAE; it ensures that the user can reach required information from several directions, using what knowledge of the system he or she has available at the time.

3.4 TAE Configuration

As a system, TAE consists of three basic parts:

- o A set of application and utility programs that perform specific task operations.

GETTING HELP

- o A complement of help texts that provide information on individual topics.
- o A dual management capability that accommodates a variety of user styles and levels of experience.

During any specific TAE session, you will be located in one of three phases:

- o Menu Phase--which guides you to a target proc through a controlled sequence of numbered choices.
- o Command Phase--which uses the TAE Command Language to package all information needed to invoke, adapt, and activate a proc on a single command line.
- o Tutor Phase--which allows you to input parameter values within a proc.

CHAPTER 4
USING MENU MODE

Chapter Sections

- 4.1 Menu Displays
 - 4.1.1 Numbered Entries
 - 4.1.2 Menu Prompt-Line Options
 - 4.1.3 Menutree
- 4.2 Tutor Displays
 - 4.2.1 Parameters
 - 4.2.2 Tutor Prompt-Line Options
- 4.3 Help Displays
 - 4.3.1 Explanatory Information
 - 4.3.2 Help Prompt-Line Options
- 4.4 Message Displays

In Menu Mode, TAE communicates with the user by means of four types of displays: (1) menu, (2) tutor, (3) help, (4) messages.

In all four types, the screen shows two different kinds of material:

- o Content: the instructions, information, or data you need to know at that time.
- o Prompt-Line Options: a set of choices which allow you to manage your task session.

USING MENU MODE
MENU DISPLAYS

4.1 MENU DISPLAYS

Unless you are using a hardcopy terminal, a TAE menu is normally a CRT display containing numbered entries (arranged vertically on the screen) and menu prompt-line options (arranged horizontally at the bottom of the screen near the prompt line).

On most TAE systems, you log on as a menu user and are given a "root" menu and a menu prompt. The prompt is a question mark (?), appearing at the lower, left-side of the screen. You then have two choices of action:

- o You can make a selection from the numbered entries.
- o You can give a prompt line option.

Menu: "ROOT", Library "TAE\$MENU"

TAE DELIVERY SKELETON ROOT MENU

- 1) Demonstration Entry: No menu exists for this entry.
- 2) Submit Suggestions, Complaints, Comments to TAE Developers (SUGGEST)
- 3) Menu of TAE Utilities (UTIL)
- 4) Menu of TAE Demonstration Programs (DEMO)
- 5) Menu of TAE Test Programs (TESTS)

Enter: selection number, HELP, BACK, TOP, MENU, COMMAND, or LOGOFF
?

Figure 4-1: Example TAE Menu

USING MENU MODE
MENU DISPLAYS

4.1.1 Numbered Entries

By selecting a particular number on a TAE menu, you access one of two things:

- o a menu--containing another level of numbered entries.
- o a proc--a program which can be executed.

On most systems, a convention is used to indicate which one it will be. For example, in GEMPAK:

- o If the numbered entry's name is placed in brackets the selection will give you another menu.
- o If the numbered entry's name is placed in parentheses, the selection will give you a proc.

After some observation, you should be able to detect what convention--if any--your system uses.

If a menu entry is selected, the new menu is displayed; if a proc entry is selected, TAE moves into the tutor phase for that proc.

4.1.2 Menu Prompt-Line Options

The menu mode interface simplifies things for you by restricting the number and kinds of decisions you can make at a given time. However, you must also be able to perform certain basic management functions whenever you need to. These include: moving back up the menu hierarchy; making a transition to Command Mode; accessing help; and logging off the system. Six menu prompt-line options allow you to perform these routine management functions during your session. Table 4-1 summarizes these six options and the TAE responses.

USING MENU MODE
MENU DISPLAYS

Table 4-1: TAE Menu Prompt-Line Options

<u>USER ENTRY</u>	<u>TAE RESPONSE</u>
BACK	Returns to the previous menu.
MENU menu-name	Activates the named menu. The "menu-name" specification may be a simple menu file name or a host-dependent qualified file specification. The new menu is added to the menu path so that a BACK from this menu returns the user to the previous menu. If no menu name is given, the current menu is repainted.
TOP	Returns to the root menu.
COMMAND	Causes TAE to exit menu mode and prompt for a TCL command.
HELP	Displays procedural information on the operation of TAE menus and procs. HELP can have one of three modifiers.
HELP *	Displays information on the current menu.
HELP proc	Displays help information on the named proc or command.
HELP n	Displays help information on entry n of the current menu.
LOGOFF	Logs the user off the system.

The prompt line options may be abbreviated, even down to a single letter, as long as their uniqueness is retained. Entries may be in lower or upper case; that is, "menu" is equivalent to "MENU."

USING MENU MODE
MENU DISPLAYS

4.1.3 Menutree

"Menutree" is a service or housekeeping routine. Running the proc produces a branching, vertical-tree listing of all menus, submenus, and procs--starting at whatever point you specify. Figure 4-2 is an example of a portion of the menutree from GEMPAK.

```

ROOT-----|
            | - GEM$MENU:
            | GEMPAK----|
            |             | - GEM$MENU:
            |             | SURFACE----|
            |             |             | - SFLIST
            |             |             | - SFMAP
            |             |             | - SFPLOT
            |             |             | - GEM$MENU:
            |             |             | SFOBJA----|
            |             |             |             | - SFBNSSET
            |             |             |             | - TGCNTR
            |             |             |             | - SFBNGRID
            |             |             |             | - SFBNANL
            |             |             |             | - SFCNTR
            |             |             |
            |             |             | - SFCNTR
            |             |             | - SFFLAMES
            |             |
            |             | - GEM$MENU:
            |             | SNFILES----|
            |             |             | - SNTERM
            |             |             | - SNSUBSET
            |             |             | - SNCOMBIN
            |             |             | - GEM$MENU:
            |             |             | SNDIAG----|
            |             |             |             | - SDADD
            |             |             |             | - SDLIST
    
```

Figure 4-2: GEMPAK Menu Tree

Notice that each time you reach a menu from another menu, you move down a level on a branch. In addition, you can move back through the branch by using the menu prompt-line options BACK and TOP.

USING MENU MODE MENU DISPLAYS

Although this ability to move forward and backward through the levels is essential to operating TAE, you also run the risk of conceptually "losing your place" in the system because of this flexibility. Since "menutree" generates a graphic representation of the system's levels, it supplies you with a roadmap of the system and guides you to your target proc.

4.1.3.1 Accessing Menutree From Menu Mode

On most TAE systems, the initial menu lists a main category of "Utility Procs." By selecting this numbered option, you will get a secondary menu, with a listing of the subcategories of utility features. Menutree may be listed in the subset of one of these categories; however, this is not a standard arrangement for all systems. If you have difficulty locating "menutree," see your system manager.

When you reach the numbered entry for menutree, you can invoke the proc and enter the tutor phase. At this point, TAE presents you with a tutor screen and prompts you to supply parameters.

4.1.3.2 Accessing Menutree From Command Mode

If you have used only Menu Mode prior to now, following this short segment will let you become familiar with Command Mode in a relatively simple case.

To run "menutree," switch from Menu Mode to Command Mode by using the menu prompt-line option "COMMAND." TAE will give you a ready prompt. Respond with

```
TAE>MENUTREE MENU=      DIREC=      OUTPUT=
```

and define the parameters for MENU, DIREC, and OUTPUT as follows:

- o MENU specifies the starting menu name. If omitted, the root menu is the default value.
- o DIRECTORY is the logical or literal name of the directory where MENU is found. The default for this parameter is system specific.

USING MENU MODE
MENU DISPLAYS

- o OUTPUT directs the listing of the menutree to a file (F), to the line printer (P), or to the terminal (T). If omitted, the menu tree is displayed at the terminal.

After "menutree" has run, TAE will return the command prompt to the screen. Switch back to Menu Mode by typing "MENU."

4.2 TUTOR DISPLAYS

A TAE tutor screen is normally a CRT display containing guides for entering parameter values (arranged in tabular form in the central area of the screen) and tutor prompt-line options (arranged horizontally across the bottom of the screen near the prompt line).

Figure 4-3 gives the format of a standard tutor screen.

```
Tutor: proc "TESTDATA", Library "(CARLSON)"                               Pg 2.
      Test of new image data
      parm      description
      -----
FORMAT      Format of the file: BSQ or BIL      "BSQ"

THRESHLD      Probability threshold                2.459830e-04

Enter: parm=value, HELP, PAGE, SHOW, RUN, EXIT, SAVE, RESTORE; RETURN to page.
?
```

Figure 4-3: TAE Tutor Screen

USING MENU MODE
TUTOR DISPLAYS

If you are working at a hardcopy terminal or an unrecognized CRT, see Appendix B for instructions on entering parameters through the NOSCREEN tutor method.

4.2.1 Parameters

The tutor screen presents you with the first page of a (possibly) multi-page display showing information on each variable of the selected proc (see Figure 4-3). For each parameter, the tutor screen displays three items:

- o the name of the parameter;
- o descriptive text;
- o the current value, if one exists.

In most cases, you establish the "current value" of a parameter in one of three ways:

- o **Explicit:** You can explicitly set the value of the parameter during the tutor session.
- o **Default:** You can accept the value listed initially on the tutor screen. Some--but not all--parameters have an initial, default value.
- o **Restored:** You can restore parameter values that were saved on a disk file in the same or a previous tutor session.

The next three sections give an example of each.

Example: Explicit

To enter an explicit value for a parameter, use the tutor prompt-line option "PARM=VALUE." Type the parameter name (or abbreviation), followed by an equals sign, followed by the value(s). On the sample tutor screen from Figure 4-3, to change the "THRESHLD" parameter, enter

```
?THRESHLD="value"
```

USING MENU MODE
TUTOR DISPLAYS

The number you type will appear on the screen to the right of the description of THRESHLD.

Example: Default

Some parameters have a default value. If a parameter has such a value, it is displayed on the parameter grid when the tutor screen first appears.

If you do not change this value, TAE assumes you have accepted the parameter as defined. For example, on the sample screen, the "format" parameter has a default of "BSQ."

To change the value, use the same procedure as you did to enter an explicit value.

Example: Restored

A "restored" parameter refers to situations where the user saves parameter sets and reinstitutes them at a later time. Two examples show why the "restore" capability is useful.

First, some procs have multiple parameters, and entering the same values for a subsequent session is time-consuming. Or, as a second example, you may have a standard set of values which you consistently use to run a cluster of procs. In such a case, it is easier to save these values rather than having to enter them at every session.

A set of parameters can be transferred to a disk file at any time during a tutor session by typing "SAVE," plus a file name, after the desired parameter values are entered. If you do not give a file name, the parameter set is saved using the current proc name as the file name. The SAVE feature does not require that all parameters have values.

To restore these values, either during the same tutor session or a subsequent one, type "RESTORE," plus the file name. If the file name is omitted, TAE assumes that the file name is the current proc name.

USING MENU MODE
TUTOR DISPLAYS

4.2.2 Tutor Prompt-Line Options

Tutor displays have a set of options listed at the bottom of the screen. You have already been introduced to three: PARM=VALUE, SAVE, and RESTORE. These options allow you to perform management functions, such as: scrolling between pages of a tutor display; updating a parameter value; requesting detailed information for a parameter; and saving parameter values on a disk. Table 4-2 summarizes these tutor options and the TAE responses.

Table 4-2: TAE Tutor Prompt-Line Options

<u>User Entry</u>	<u>TAE Response</u>
parm=value	Assigns a new value to the parameter with the name "parm."
parm(i)=value	This option is similar to the one above, except only the value of the ith element of a multi-valued parameter is set.
HELP	Displays help information on the tutor phase.
HELP parm	Detailed text describing the parameter "parm" is displayed.
HELP *	Text describing the current proc is displayed.
PAGE parm	Displays the page containing parameter "parm."
PAGE n	Displays page n.
Carriage Return	The display moves to the next page.
RUN	Executes the proc.
EXIT	Terminates the tutor session without running the proc.
SAVE filename	Saves the current parameter values in the disk file "filename." If

USING MENU MODE
TUTOR DISPLAYS

"filename" is omitted, the current
proc name is used.

RESTORE filename Restores the parameter values from
the disk file "filename." If
"filename" is omitted, the current
proc name is used.

RESTORE LAST The parameter values are restored
from the most recent tutor RUN.

The RUN option activates the proc. When the proc
terminates, TAE prompts you to return to a menu or to the
TCL, as appropriate. When returning to a menu, the screen
displayed is the one from which the proc was selected.

4.3 HELP DISPLAYS

A typical TAE help text may have several screens, each of
which contains explanatory information (usually in sentence
form and appearing in the top part of the frame) and help
options (arranged horizontally across the bottom of the
screen).

4.3.1 Explanatory Information

The help texts on your system have been written by two
different sources. Some are indigenous to TAE and supply
information on management and operational functions common
to all TAE applications. Others were written by local
programmers and contain information on procs that are
site-specific.

Despite these different origins for help texts, the screen
format and the sequencing of information follows similar
patterns. Generally, a series of panels will present
information on a specific topic in the following order:

- o Identification/Introduction
- o Definition(s)/Illustrations(s)
- o Qualification(s)/Extension(s)

USING MENU MODE
HELP DISPLAYS

4.3.2 Help Prompt-Line Options

Four screen options aid you in managing a help session.

Table 4-3: TAE Help Prompt-Line Options

Carriage return:	The display moves to the next page. The first page is the "next" page after the last page.
PAGE n:	Page number "n" is displayed.
EXIT:	Exit from the help display and return to the parent mode.
None of the above:	Any other character string, except those listed above, is interpreted as a command from the previous mode.

The first three actually appear on the display. The "none of the above" allows you to enter information and action requests, without obliterating the current screen. This feature allows you to read and act on the help screen for a tutor display without erasing the frame before you can enter data. For example, when using the tutor option "HELP parm," you can read about the details of the parameter and--while still being able to see the help text on the screen--set the parameter value.

4.4 MESSAGE DISPLAYS

A message display, consisting of a single comment line, appears on the current display screen, which can be a menu, tutor, or help frame. The user may then choose from the options appearing on the prompt-line of that particular screen.

A message in TAE is a line of information preceded by a "message key." The message key is used to find more information about the message. The format of the message is "[facility-id]." The "facility" is the function within TAE that generated the message. The "id" is the message

USING MENU MODE
MESSAGE DISPLAYS

identification within the facility.

A typical example of a TAE message is

[TAE-UNRECCOMD] Unrecognized command

The normal position for the message is immediately over the screen option line.

 QUICK REFERENCE SUMMARY

Managing TAE through Menu Mode involves four types of displays: menu, tutor, help, and message. With the exception of the message display, each category has a full-screen format containing two important elements: (1) the information of the display and (2) the requests for acting on that information and for managing the display session. Table 4-4 indicates the type of information presented in each category. Table 4-5 summarizes the various prompt-line options for each type of display.

Table 4-4: TAE Display Information

<u>Display</u>	<u>Elements</u>	<u>Purpose</u>
Menu	Numbered entries, arranged vertically on the screen	The numbered entries control the selection of the next session level. When you choose one of these numbers, TAE responds either with the subsequent menu level or with a proc tutor display.
Tutor	The name, description, and current value of each parameter in the proc	Assists in specifying the parameters for a proc.
Help	Information, presented in a timely and economical fashion	Provides user-oriented instructions for various TAE concepts and components.
Message	A single line comment	Tells when an error has been made or conveys information from a running program.

Table 4-5: TAE Prompt-Line Options

<u>Category</u>	<u>Option</u>	<u>TAE Response</u>
Menu	Back	Returns to previous menu
	Menu	Repaints the current menu
	Top	Returns to the initial menu
	Command	Switches to Command Mode
	Help	Displays the requested help text
	Logoff	Logs the user off the system
	Tutor	Parm=value
Help		Displays the requested help text
Page parm		Displays the page containing the named parameter
Page n		Displays page n
Carriage Return		The display moves to the next page
Run		Executes the proc
Exit		Terminates the tutor session without running the proc
Save filename		Saves the current parameter value in the specified disk file
Restore filename		Restores parameter values from the specified disk file
Restore last		Restores parameter values from the more recent tutor run
Help		Carriage Return

USING MENU MODE

Page n Displays page number n

Exit Exit from the help display

Message Will accept the prompt-line options listed for
the screen on which the message appears.

CHAPTER 5

USING COMMAND MODE

Chapter Sections

- 5.1 Transition from Menu to Command Mode
- 5.2 TAE Command Language (TCL)
 - 5.2.1 Procs
 - 5.2.2 Intrinsic Commands
- 5.3 Changing Modes

The previous chapter dealt with Menu Mode and covered menu, tutor, help, and message displays. This chapter guides you in applying the knowledge you have gained of TAE to a more powerful method of addressing the system.

5.1 TRANSITION FROM MENU TO COMMAND MODE

Even if you have only operated TAE in Menu Mode up to this point, you will be able to make the transition to Command Mode by reading this chapter and practicing for a few sessions at the terminal. Once you become accustomed to Command Mode, it will probably be your preferred way of interacting with TAE.

You enter Command Mode by typing "COMMAND" at any time after logging on to the system.

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

5.2 TAE COMMAND LANGUAGE (TCL)

The object of your TAE session in Command Mode is the same as it was in Menu Mode: to reach and run a proc. Command Mode allows you to (1) access a particular program, (2) supply the variables, and (3) initiate the run--using a single command line. To use this more efficient method, you must package your input information in the TAE Command Language (TCL).

The simplest and most common form of a TCL line consists of a command name followed by a variables list.

TAE>command name variables list.

The command name may be either (1) a proc or (2) an intrinsic command.

A proc may be an application or a utility program. Application procs are host-dependent. Utilities procs are service features and may or may not be system-specific. Intrinsic commands are used to manage the session. These commands are common to all TAE systems, hence the term "intrinsic."

5.2.1 Procs

Procs are the operational heart of the system: they are the processes and procedures which carry out the tasks you want to complete.

Keep in mind three things about procs:

- o A proc can be either a process or a procedure hence the term "proc." A process is a program: that is, a pre-existing disk file of executable code activated by TAE. A procedure (or "command procedure") is a program written in the TAE Command Language (TCL).
- o While operating either in Menu Mode or in Command Mode, you will not be aware if the proc is a process or a procedure: both types of procs are activated using similar features.

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

- o You can activate procs either in Menu Mode or in Command Mode. However, procs do not have to be located in a menu in order to be accessed through Command Mode.

After practicing for a few sessions in Menu Mode, you now have a fairly good idea of what kinds of application programs and utility features are available on your system. You also have gained some familiarity with the terminology.

However, if you are a new user, you should not expect to remember the exact function of all the programs, what they are called, where they are located, how they are invoked, and what their parameters are. When you feel you need more guidance for operating in Command Mode, you can:

- o Access the proc "menutree" to refresh your memory of the available programs and the terminology.
- o Access HELP for specific entities.
- o Switch to Menu Mode (by typing MENU) for more practice in accessing and running procs.
- o Access the tutor phase for the proc (by typing TUTOR plus the proc name) and enter the parameters.

For many users, a paper copy of the system's menutree acts as a handy catalogue of procs and provides orientation throughout the TAE Command session. You may find it useful to keep a copy of your system's menu tree close at hand while in Command Mode.

You invoke a proc by entering its name immediately after the command prompt. For example, to call up the proc COPY, start the command line with

TAE>COPY

This identifies the proc to TAE.

However, you still need to define the values on which the proc will run. For example, in the proc COPY, two variables--"FROM" and "TO"--must be defined in order for the program to execute successfully. The next section describes how to define the variables in a proc.

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

5.2.1.1 Proc Inputs

Variables must have specific values for the proc to run. These specific values are either supplied by the system or determined by the user in the form of proc inputs.

In the proc COPY, for example, the variables "TO" and "FROM" can be specified on the command line invoking the proc by appending the values to the parameter names:

```
TAE>COPY FROM=FILE1 TO=FILE2
```

You may also enter the values by position, thus eliminating the need for typing in the variable name.

```
TAE>COPY FILE1 FILE2
```

After you have entered as much information on the command line as you intend to supply, push the return key to activate the proc.

You may have trouble remembering all the information you must enter on a command line for a specific proc. For instance, you may not remember all the parameter names or you may not remember the order of the parameters. In some instances, you may not realize that you have entered the command line incorrectly until TAE responds with a message, such as

```
[TAE-MISPAR] Missing Parameters
```

Error comments may contain enough information to jog your memory. But it could take several tries before you get the correct format. This trial-and-error approach may be too slow for you.

If you cannot remember either the parameter names or the parameter positions, you should enter Tutor phase and establish the values for proc by that method.

5.2.1.2 Subcommands

Some procs can perform several different but related operations, each of which requires a different set of parameters. These functions are called "subcommands." For example, a magnetic tape utility proc, called TAPE, might have the following subcommands and associated parameters:

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

- o COPY: input tape unit, output tape unit,
number of files
- o REWIND: input tape unit
- o SPACE: input tape unit, number of files.

A subcommand is specified as a suffix to the proc name, separated with a hyphen. No spaces are allowed between the command name and the hyphen.

```
TAE> TAPE-REWIND UNIT=5
```

```
TAE> TAPE-SPACE UNIT=5 FILES=-3
```

5.2.2 Intrinsic Commands

The format for a TCL command line is

```
TAE> command name variables list
```

Instead of entering a proc name, you may enter an intrinsic command as the first element in a TCL line.

An intrinsic command causes the system to perform a standard, predefined operation. These programs aid the user in performing basic management tasks during a TAE session. As an analogy, you have already been introduced to menu prompt-line options (some of which assist the user in moving backward and forward in the menu hierarchy) and tutor prompt-line options (some of which assist in entering variables for running procs). An intrinsic command is such a tool.

Table 5-1 gives some commonly used intrinsic commands.

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

Table 5-1: TAE Intrinsic Commands

<u>Intrinsic Command</u>	<u>Description</u>
DEFC*MD	Define command
DELC*MD	Delete command definition
DISA*BLE-LOG	Disable session logging
DISP*LAY	Display variable value
ENA*BLE	Enable logging or script
EX*IT	Terminate TAE, remain in host command language
H*ELP	Obtain help information
LET	Assign a value to a variable
LO*GOFF	Terminate TAE, logout of host
ME*NU	Display menu
SETLIB	Set library search order
SH*OW	Show status of libraries
TU*TOR	Display tutor screen

Every intrinsic command name may be truncated up to the asterisk (as shown on the list). You can obtain more information on each of these intrinsic commands by typing "HELP," followed by the command name.

As is the case with a proc name, the intrinsic command name may have a subcommand, which follows the command name and is separated from it by a hyphen.

For example, the ENABLE command activates session logging or a script. There is no default subcommand; in order to run this intrinsic command, you must use either

USING COMMAND MODE
TAE COMMAND LANGUAGE (TCL)

TAE> ENABLE-LOG

or

TAE>ENABLE-SCRIPT

You will find a complete description of the subcommands and variables associated with each intrinsic command in the help text for that particular topic.

5.3 MODE CHANGING

So far, this primer has presented the dual management features of Menu and Command Mode as if, once selected, one or the other must be used exclusively throughout a TAE session. This is not the case. The management modes can be switched at almost any time.

While it is true that Menu Mode provides guidance in managing the system for the new user and Command Mode provides efficiency for the experienced user, few users can be classified as exclusively novice or expert in handling a complex system. Even after much hands-on experience, you will know some parts of the system better than others, and your need for guidance in managing the system will vary not only from session to session, but within the same session.

TAE allows the user to switch modes at many points. Figure 5-1 illustrates the transitions you may make during a typical TAE session. At any given time, you are located in one of the three major phases. The lines and arrows indicate six transitions that can be made and the direction in which they can take place. The word or phrase located next to each of the six numbers on the diagram indicates the user request or system condition that causes the transition to happen. Table 5-2 gives a summary of the six traversals represented on the diagram.

USING COMMAND MODE
MODE CHANGING

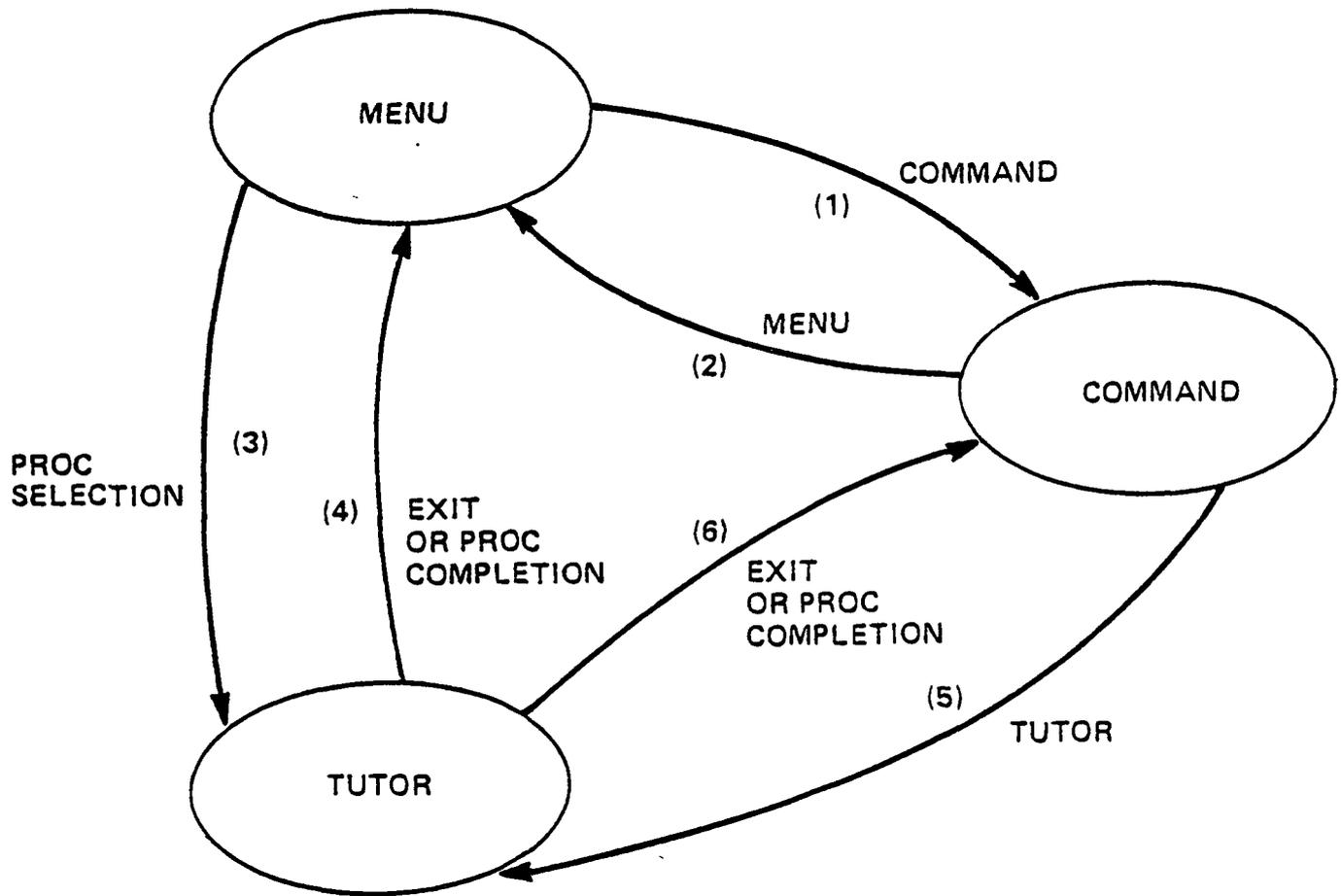


Figure 5-1: Mode Transition Diagram

USING COMMAND MODE
MODE CHANGING

Table 5-2: Summary of Mode Traversals

- (1) COMMAND is a menu display response that causes a user to become a TCL user.
- (2) MENU is a TCL request that allows a user to become a menu user.
- (3) If a proc entry on a menu is selected, TAE enters the Tutor phase for parameter specification. When directed by the user, Tutor phase activates the proc.
- (4) When a proc terminates or the user exits from the Tutor phase, the most recently displayed menu is re-activated.
- (5) For a TCL user, the TUTOR request causes Tutor phase to be entered for a proc.
- (6) When a proc terminates or the user exits from the Tutor phase, the TCL prompt re-appears.

Note that the Tutor phase "remembers" whether the user is a menu user or a TCL user to distinguish between traversal (4) and traversal (6). This means that you must return to your previous mode when exiting Tutor phase.

USING COMMAND MODE

QUICK REFERENCE SUMMARY

5.1 Transition from Menu to Command Mode

As a new user, you have learned the basics of the system--its resources, terminology, and configuration--from Menu Mode. Once you complete a few initial sessions, you can begin to address TAE through Command Mode. Since help is available at virtually every point in a Command session, you can experiment without fear of being unable to recover from mistakes.

5.2 TAE Command Language (TCL)

Command Mode allows you to call up a proc, modify it for a specific execution, and execute the run--on a single input line.

Communicating with TAE in its command language requires that you package your input in a specific format. To do this you must know the syntax of the TAE Command Language.

5.3 Mode Traversals

"User-friendly" is a relative term: its meaning depends on the point-of-view of the user. While the structure of Menu Mode is useful to the novice, its rigor may seem tedious to an expert.

To make matters even more complex, the terms "novice" and "expert" imply a dichotomy which seldom exists in the real world. Few of us can be classified as exclusively one or the other.

More than likely, even after experience with TAE, you will be an "expert" on some aspects and a "novice" on others. For example, most installations frequently add new procs and delete outdated ones, so the system may undergo periodic changes. Furthermore, some TAE systems are so large (or you may be called upon to use a certain feature so infrequently) that you will require aid in running some procs, but not others.

TAE accommodates this need for hybrid management by allowing you to traverse the three phases of any typical session. The system will allow three transitions: between Menu and Command; between Command and Tutor; and between Menu and Tutor. Each of the pairs is bi-directional, making the

USING COMMAND MODE

total number of possible transitions six.

CHAPTER 6

CREATING A PROC

Chapter Sections

- 6.1 Procs
- 6.2 Proc Constituents
- 6.3 Writing a Proc
 - 6.3.1 Create a Proc Definition File
 - 6.3.2 Create a Help File
 - 6.3.3 Create "Body" or Image
 - 6.3.4 Place All Constituents in a Proc Library
- 6.4 Practice Session

6.1 PROCS

In order to communicate with TAE, you must be able either to invoke or to create a program which the system can recognize. These programs are called "procs." A proc is either a process (an application program) or a procedure (a series of TCL commands) packaged into a file.

In Chapters 4 and 5, you learned how to invoke pre-existing procs. This chapter gives you the basics for creating your own TAE procs. However, this presentation cannot teach programming. This section assumes that you know how to develop software under the host operating system.

The examples in this chapter use VAX/VMS conventions.

CREATING A PROC PROC CONSTITUENTS

6.2 PROC CONSTITUENTS

If the proc is an application program, the parameters are defined in a Proc Definition File (PDF) and the program exists as an executable image (EXE). If the proc is a procedure, the PDF contains the parameter definitions, followed by the sequence of TCL commands. For both processes and procedures, the PDF also contains or points to the help information for the proc.

6.3 WRITING A PROC

Writing a proc normally involves four steps:

- STEP 1: Write a Proc Definition File (PDF).
- STEP 2: Write a help file (HLP) or append the help information to the end of the PDF.
- STEP 3: If the proc is a procedure, add the "body"--that is, the list of TCL commands.
If it is a process, write the code in a programming language and create the executable image.
- STEP 4: Place the files from steps 1, 2, and 3 in the same proc library.

The next four sections give instructions and examples for each step.

6.3.1 Step One: Proc Definition File (PDF)

A PDF is a text file created using one of the host system's text editors. PDFs define the parameters for a proc, and if the proc is a procedure, the PDF defines the commands to be executed for the procedure. PDFs are required for all procs. A sample shows how each is constructed.

CREATING A PROC
WRITING A PROC

Illustration: Process PDF

The format of a process PDF is:

```
PROCESS HELP=help-location  
PARM    name1    description1  
PARM    name2    description2  
      .  
      .  
      .  
END-PROC
```

Illustration: Procedure PDF

A TAE procedure allows you to package a group of TCL commands and to selectively control the execution of the commands.

The format of a procedure PDF is:

```
PROCEDURE HELP=help-location  
PARM    name1    description1  
PARM    name2    description2  
      .  
      .  
      .  
BODY  
      Procedure body of executable TCL commands  
END-PROC
```

Both processes and procedures may have parameters, and the command that defines the parameter, the PARM command, is the same for all procs. The following keywords are used to describe each parameter:

```
TYPE =    INTEGER  
          REAL  
          (STRING, size-of-string)  
          FILE
```

The type of the parameter.

CREATING A PROC
WRITING A PROC

COUNT = m:M
The minimum and maximum number of values
expected for the parameter.

DEFAULT = value
The value to be assumed if the proc is invoked
with no user-supplied value for the parameter.

VALID = (a,b:e,f)
The allowable set of values assumed by a
parameter.

ACCESS = IN (existing file)
OUT (valid name syntax)
INOUT (existing file)
NOCHECK (no checks are made)
Kind of access for a file parameter

The order in which the parameters are specified in the PDF
is the order in which they appear on the tutor screen. This
sequence also determines the order in which their values
must be entered if using positional parameters when running
the proc from command mode.

The following is an example of a PDF for a process:

PROCESS

PARM	device	type=string	default="plotter"	
PARM	filnam	type=file	access=in	count=(1:35)
PARM	gridsize	type=integer	default=16	valid=(4:2500)
PARM	area	type=real	count=2	

END-PROC

The following is an example of a PDF for a procedure:

```
PROCEDURE HELP=*  
PARM BAND TYPE=STRING  
REFGBL DATA  
LOCAL X TYPE=STRING  
BODY  
LET X=DATA//BAND  
DISPLAY X  
PROCNM FILE=&X  
END-PROC
```

CREATING A PROC
WRITING A PROC

```
.TITLE
  Text to be used for the header of the tutor
  and help displays
  .
  .
  .
.END
```

6.3.2 Step Two: Help Files

Help information on TAE procs can be written in one of two ways: as a separate file with the same name as the proc and Help type (.HLP in VAX/VMS), or as text appended to the PDF.

In the first case, a text file (on VAX: xxxx.HLP), is created using the host text editor. In the second case, the PDF must include, after the PROCEDURE or PROCESS command, the text "HELP=#." This tells TAE that the help text is appended. In either case, the format of the help information is the same:

```
.TITLE          Title to appear on each tutor and help
                 screen of this proc.

.HELP           Help information on the proc itself.

.LEVEL1        Header to tell TAE that first level
                 help on the proc's variables follows.

.VAR name1     Help information on name1 to be
                 displayed on the tutor screen.

.VAR name2     ...

.LEVEL2        Header to tell TAE that second level
                 or detailed help on the variables
                 follows.

.VAR name1     Detailed information on name1. This
                 text is displayed on the HELP PARM
                 tutor command or on the TCL command
                 "HELP-PARM parm=name1 proc=procname".

.END           End of help text.
```

An example of a .HLP file follows:

```
.TITLE
  Text to be used for the header of the tutor
```

CREATING A PROC
WRITING A PROC

and help displays.

```
.HELP
  General help text to be displayed for the
  help command.

.LEVEL1
.VAR variable-1
  Level-1 help text for variable-1.
.Var variable-2
  Level-1 help text for variable-2.
.
.
.SUBCMD subcommand-1
  Level-1 help text for the named subcommand.

.LEVEL2
.VAR variable-1
  Level-2 help text for variable-1
.VAR variable-2
  Level-2 help text for variable-2
.
.
.SUBCMD subcommand-1
  Level-2 help text for the named subcommand.
.END

.PAGE:      May be used in the .HELP text or in the
             .LEVEL2 .VAR and .SUBCMD text.
.INCLUDE:   References text contained in another file.
             The referenced file may not contain help
             Directives other than .PAGE.
```

6.3.3 Step Three: Create "Body" Or Image

Step three varies, depending on whether you are creating a procedure or a process.

Procedure

If the proc is a procedure, add the "body"--that is, the executable TCL "program." Table 6-1 gives a partial list of commands which can be used within a procedure body.

CREATING A PROC
WRITING A PROC

Table 6-1: Procedure-Oriented TCL Commands

FOR/END-FOR	Loop with a variable assignment
IF/ELSE	Conditional execution
LET	Variable assignment
WRITE	Write to the Standard Output Device

An example of a procedure PDF using some of the above commands follows:

```
PROCEDURE

REFGBL OK
SUBCMD INTS
    PARM P2 INTEGER
    PARM P3 INTEGER
END-SUB
SUBCMD STRINGS
    PARM P2 (STRING, 30)
    PARM P3 (STRING, 30)
END-SUB

LOCAL C INTEGER
LOCAL D (STRING,30)

BODY

LET _ONFAIL="CONTINUE"
LET C = 100
LET D = "100"
IF (OK="TRUE")
    IF (_SUBCMD="INTS")
        MYWRITE-INT (&P2, &P3, &C)
    ELSE-IF (_SUBCMD="STRINGS")
        MYDIS (&P, &P3, &D)
    END-IF
ELSE
    WRITE "OK SWITCH NOT TRUE."
END-IF
END-PROC
```

Process

This section tells how to implement a program (that is, a "process") to run under TAE. As described previously, a program running under TAE becomes known to TAE through a PDF. The PDF defines the program's parameters and indicates where the help information for the program can be found.

TAE provides a number of library subroutines for use by the

CREATING A PROC
WRITING A PROC

application programs. The purpose of the library subroutines is to:

- o Provide an interface between the application program and TAE for parameter passing, termination, logging, and dynamic parameters.
- o Standardize access to host services, enhancing the portability of application software.

The TAE subroutines are grouped by function as follows:

- o XI - Image file access
- o XM - Message processing
- o XR/XQ - Parameter interface
- o XT - Terminal I/O
- o XZ - Miscellaneous services

The TAE Applications Programmer's Reference Manual contains information on these packages. The XR/XQ subroutines are the most frequently used, therefore those are the ones that will be discussed in this primer.

The XR and XQ packages have entry points for managing the parameter interface between the application program and TAE. For example, in a sample program TEST.FOR, the TAE routines to read in, change, and save parameters would be as follows:

```
program test
.
.
.
call xrinim          !Initializes XR package
call xrintg         !Retrieves an integer parameter
call xrstr          !Retrieves a string parameter
.
.
.
call xqintg         !Updates integer variable
call xqwrtd        !Writes block with new values to disk
.                  !for later retrieval by this or another
.                  !program
.
end
```

6.3.4 Step Four: Proc Library

The files that you have created in steps 1, 2, and 3 must reside in the same proc library. A proc library is defined as a collection of PDFs, HLPs, and EXEs grouped under a library name. When TAE encounters a proc reference--as in a proc invocation--the PDF is located using a hierarchy of several libraries:

- o The "user library." This is generally the library you are logged on to or defaulted to.
- o If the PDF is not found in the user library, the "application libraries" are searched. These libraries are defined with the TCL "SETLIB" command.
- o If the PDF is not found in steps 1 or 2, the "system library" is searched.

This hierarchy search allows you to have a "personalized" version of a proc. For example, if a proc has identical names in the system library and in your library your version will be selected.

When you are in command mode, the TCL "SHOW" command will list the names of the active libraries from the above 3 steps.

6.4 PRACTICE SESSION

Writing an application program will give you some hands-on experience and will allow you to review and apply the previous instructions. As a practice exercise, compile, link, and execute a proc that will perform this sequence of events:

- o Average a series (consisting of between 2 and 20 places) of real numbers;
- o Display the average;
- o End the proc.

Use the four steps explained in the above segments to guide you:

CREATING A PROC
PRACTICE SESSION

- Step One: Write a Proc Definition File (.PDF).
- Step Two: Append the help information to the end of the .PDF.
- Step Three: Create the executable image (.EXE).
- Step Four: Place the files in the same proc library.

Step One and Step Two

First, decide how to enter the numbers into the program. It can be done in two ways: as several individual parameters, or as one multi-valued parameter. Since the latter requires only one call from the program and one parameter input from the user, it is the more efficient of the two options.

Having made this choice, create the PDF and the Help text:

```
process help=*  
parm numbers type=real count=(2:20)  
end-proc
```

```
.title  
Compute the average of real numbers
```

```
.help
```

```
      This program averages from two to twenty  
      real numbers, as specified by the user, and  
      displays the average.
```

```
.level1  
.var numbers  
Numbers to be averaged  
(maximum of 20)
```

```
.level2  
.var numbers  
      The numbers to be averaged by this program  
      can be any integer or real number, with a  
      minimum of two and a maximum of twenty numbers.  
.end
```

CREATING A PROC
PRACTICE SESSION

Step Three

Next, create the executable image (.EXE).

Write AVERAGE.FOR:

```
      program average

      include 'tae$inc:pgminc.fin'

      real
      . numbers(20),      !variable to receive the parameter
      . sum,              !sum of input numbers
      . ave

      integer
      . block(xprdim),   !V-block with parameters
      . icount,          !Number of parameters received
      . istat

      call xrinim(block,xprdim,xabort,istat)
      call xrreal(block,'numbers',20,numbers,icount,istat)

      sum=0

      do i=1,icount
      sum=numbers(i)+sum
      end do
      ave=sum/icount

      call xtinit(itype,lines,icol)
      call xtout(20, 10, 'AVERAGE=')
      write(5,10) ave
10    format(f15.5)
      end
```

After coding your program, you are ready to compile the source code and link with the TAE libraries. For this example, use the VAX/VMS commands.

```
$FOR AVERAGE
$LINK AVERAGE,TAE$OLB:LIB/LIBRARY, -
      TAE$OLB:COTS/LIBRARY
```

CREATING A PROC
PRACTICE SESSION

Step Four

Make sure that the files created in the above three steps are all located in the same library.

To run the program AVERAGE, you log onto TAE. If you are in command mode:

To execute AVERAGE:

```
TAE>AVERAGE N=(234,73)
      or
```

To display the tutor screen for AVERAGE:

```
TAE>TUTOR AVERAGE
```

If you are in menu mode and AVERAGE has been incorporated in a menu, you can access AVERAGE by entering the appropriate number in the menu that references it. This will also display AVERAGE's tutor screen.

By adapting the four steps, you can write a procedure to average reals as follows;

```
procedure      !the help information is separate
parm numbers type=integer count=(2:20)
local sum integer initial=0
local i integer initial=2
local average type=real initial=0
body

let sum=numbers(1) + numbers(2)

loop
  let I=I+1
  if ($count(numbers) < I) break
      !$count is a TAE integer function
  let sum = numbers (I) + sum
end-loop

let average=$float(sum)/$float($count(numbers))
      !$float is a real function

display average

end-proc
```

CREATING A PROC

QUICK REFERENCE SUMMARY

Procs are the operational components of TAE. A proc is either a process (that is, an application program) or a procedure (a series of TCL commands) packaged into a file.

Creating a proc normally involves four steps:

1. Write a proc definition file (PDF).
2. Write a help file (.HLP) or append the help information to the end of the PDF.
3. If the proc is a procedure, add the "body" -- that is, the executable TCL program. If it is a process, write the code and create the executable image.
4. Place the files from steps 1, 2, and 3 in the same proc library.

CHAPTER 7

GLOSSARY

attention sequence

A host-dependent sequence of characters typed by a user to interrupt the execution of a proc.

command

A string of text interpreted by TCL. A command is either a proc invocation command or an intrinsic command.

command mode

The TAE mode under which the user types TCL commands.

command procedure

Equivalent to "procedure".

command string

A series of command lines that constitutes one TAE command. Lines beyond the first are "continued" lines, i.e., the plus character is the last non-comment character of the preceding line.

GLOSSARY

directive

A delimiter for a section of text in a help file or menu definition file. Directives begin with a period.

file

A related collection of data resident on disk.

global variable

A value known across all proc invocations.

help file

A text file describing: a proc and the proc's parameters; a menu; or a TAE mode.

host

The computer system upon which TAE is executing. A host consists of the computer type and the operating system.

interactive level

The "primary level" for sessions that are initiated interactively.

intrinsic command

A primitive TCL command that supports the framework of the language.

library

A named collection of files. The syntax of a library name is host-dependent.

GLOSSARY

mandatory parameter

A proc parameter that has no default value.

menu definition file (MDF)

A text disk file defining the entries for one menu.

menu mode

The TAE mode under which the user selects entries from a menu to direct TAE activity.

menu tree

A hierarchy of menus. The root menu references several menus, these menus each reference several menus, and so on.

message key

An identifier displayed with each TAE message. The message key is enclosed in brackets preceding the message text.

multi-valued parameter

A parameter whose value is a vector of values.

parameter

The name of a value that is passed to a proc.

named parameter

A parameter declared as "TYPE=NAME". Such a parameter is passed to a proc "by reference".

GLOSSARY

positional parameter

A parameter associated with a value according to the position of the value on the command line.

proc

A function that may be activated by TAE. A proc is either a process or a procedure.

proc definition file (PDF)

A text disk file defining the parameters for a proc. For a procedure, the PDF also defines the commands to be executed for the procedure.

proc invocation

A TCL command that requests activation of a proc.

procedure

A collection of TAE commands that may be executed as one function.

process

A task that is activated by TAE to perform a function.

root menu

The highest-level (initial) menu in the menu tree.

session

The period from logging onto TAE until logging out.

GLOSSARY

statement

An intrinsic TAE command.

subcommand

A proc "modifier" that selects a subfunction of the proc.

substitution

The replacement of a variable marked by an ampersand by the string representing the value of the variable.

system manager

The individual assigned to manage the software configuration of the system (user authorization, library contents, etc.).

TAE Monitor

The task that manages a TAE session. There is one per interactive user.

task

The collection of executable code and associated data areas that competes with other tasks for processor resources (memory, instruction execution, etc.).

TCL

The TAE Command Language.

text file

A sequential disk file, each record of which contains no more than 132 printable characters.

GLOSSARY

tutor mode

A mode of TAE in which the user interactively supplies parameter values in preparation for proc activation.

tutor session

The period from entry to tutor mode until (a) the proc is executed or (b) tutor mode is terminated with the EXIT command.

tutor subcommand display

A tutor display in which the user selects a subcommand from among a displayed set of subcommands.

variable

A named value in TCL. A variable may contain a character string, an integer value, or a real value. There are three classes of variables: proc parameters, local variables, and global variables.

white space

A series of tabs and/or spaces.

APPENDIX A

TAE COMMAND LANGUAGE (TCL)

Communicating with TAE in its command language requires that you package your input data in a command string. The simplest form of a TCL command line is

```
TAE>  command name    variables list
```

Entering an item for the first element--the command name--invokes the proc. The second element--the variables list--allows you to enter the values for the proc's parameters.

These values may be entered on the command line in one of two formats: (1) keyword or (2) positional.

Command names and keywords can be truncated, even down to a single letter, so long as the abbreviation is unique. The examples in this section will use the full name form to ensure clarity.

In the keyword format, each value is appended to the parameter name (the keyword) by an equals sign. For example, the proc COPY has two parameters: "FROM" and "TO." Using the keyword format, you instruct the computer to activate the proc and copy File 1 to File 2 by entering the following command string.

```
TAE>COPY  FROM=FILE1  TO=FILE2
```

The separator between parameters may be a comma, and/or any number of spaces or tabs. A comma may be preceded or followed by spaces and tabs, as in these examples:

TAE COMMAND LANGUAGE (TCL)

```
TAE>COPY FROM=FILE1    TO=FILE2
```

```
TAE>COPY FROM=FILE1,   TO=FILE2
```

```
TAE>COPY FROM=FILE1    , TO=FILE2
```

In the second format (positional) the parameter names may be omitted, provided the proc-specific order is maintained.

In the COPY illustration, the parameters can be entered positionally as follows:

```
TAE>COPY    FILE1    FILE2
```

Whether you use keyword or positional entries depends on the particular situation and how well you remember the names and order of the proc's parameters.

APPENDIX B
ULOGON AND ULOGOFF

The ULOGON procedure is a way to customize the initialization of a TAE session. This procedure is usually created for each user by the system manager. ULOGON can be used to define your own TCL commands, the directories where your application software resides, or to configure your TAE session.

Example ULOGON.PDF

```
procedure
body
defcmd command=a*dd string=setlib-add
defcmd t tutor

!   Add [myaplib] to the list of
!   application libraries set by
!   SLOGON.PDF +++

setlib-add (*,[myaplib])

!   Restore the environment

++ taegbl|restore=gbl|
enable-log
.
.
.
end-proc
```

The ULOGOFF procedure is a way to customize the exiting from a TAE session. This procedure can be used to delete unnecessary files in your directories.

Example ULOGOFF.PDF

```
procédure
```

ULOGON AND ULOGOFF

```
body
dcl delete *.tmp;*,last.par;*
dcl purge session.tsl
```

```
!      Save session environment
```

```
++   taegbl|save=gbl|
     end-proc
```

++TAEGBL is a TAE proc. Additional information is available through the help text and the tutor display.

+++SLOGON.PDF is the system initialization proc. Your ULOGON.PDF should be compatible with SLOGON. If you have questions, see your system manager.

APPENDIX C

NOSCREEN TUTOR

NOSCREEN is a special form of tutor mode that does not require a CRT terminal. NOSCREEN tutor exists to support these situations:

- o Hard copy terminals.
- o Unsupported CRTs, that is, CRTs for which TAE can only operate in the scrolling mode.
- o Low-speed CRTs, where the time for screen update is prohibitive.
- o Personal choice: some users prefer the NOSCREEN tutor over normal tutor sessions.

Operation of NOSCREEN tutor is similar to screen tutor; the differences are as follows:

- o When NOSCREEN tutor is initially entered, a list of parameter names is automatically displayed.
- o The user is prompted for commands with "TAE-TUTOR>."
- o The paging commands ("PAGE" and "Carriage Return") are not available.
- o When a parameter is given a value, there is no automatic echo of the value.
- o The LIST command (unique to NOSCREEN tutor) may be used to display parameter names and current values. The LIST command may contain

NOSCREEN TUTOR

a list of parameters to be displayed. If no list is present, all parameters for the proc are displayed.

- o In a manner similar to the LIST command, the DISPLAY command displays parameter names and values and, in addition, descriptive text for each parameter.

Example of a NOSCREEN Tutor Session:

```
TAE>TUTOR XPTST  
Parameters: WINDOW, FILE, FORMAT, and THRESHOLD.  
TAE-TUTOR>LIST  
WINDOW = no value  
FILE = "IMAGE15"  
FORMAT = "BSQ"  
THRESHOLD = 2.45983  
TAE-TUTOR>WINDOW = (0, 511, 0, 511)  
TAE-TUTOR>LIST WINDOW  
WINDOW = (0, 511, 0, 511)  
TAE-TUTOR>RUN
```

BIBLIOGRAPHIC DATA SHEET

1. Report No. NASA TM-86062	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle PRIMER FOR THE TRANSPORTABLE APPLICATIONS EXECUTIVE		5. Report Date February 84	
		6. Performing Organization Code	
7. Author(s) Patricia Ann Carlson; et al Rose-Hulman Institute of Technolgoy		8. Performing Organization Report No.	
9. Performing Organization Name and Address NASA/Goddard Space Flight Center Advanced Systems Development Branch Greenbelt, MD 20771		10. Work Unit No.	
		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code	
15. Supplementary Notes			
<p>16. Abstract</p> <p>This document is an introduction to the Transportable Applications Executive (TAE), developed by the NASA/GSFC Information Extraction Division. TAE is an interactive multipurpose executive that provides commonly required functions for scientific analysis systems.</p> <p>The TAE Primer discusses the concept of an executive and presents in detail the various components of TAE. These include on-line help information, the use of menus or commands to access analysis programs, and TAE command procedures.</p>			
17. Key Words (Selected by Author(s)) Transportable Applications Executive		18. Distribution Statement Unclassified-Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 81	22. Price*