NASA TM-100575

NASA Technical Memorandum 100575

THE ENVIRONMENT FOR APPLICATION SOFTWARE INTEGRATION AND EXECUTION (EASIE) VERSION 1.0

VOLUME III

PROGRAM EXECUTION GUIDE

JAMES L. SCHWING
LAWRENCE F. ROWELL
RUSSELL E. CRISTE

April 1988

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

PREFACE

The Environment for Application Software Integration and Execution, EASIE, provides both a methodology and a set of software utility programs to ease the task of coordinating engineering design and analysis codes. The need for such techniques and tools has stemmed from the computer-aided design and engineering activities within Langley Research Center's Space Systems Division (SSD). In SSD, the Vehicle Analysis Branch (VAB), with emphasis on advanced transportation systems, and the Spacecraft Analysis Branch (SAB), with emphasis on advanced spacecraft, share a common need to integrate many stand-alone engineering analysis programs into coordinated, quick-turnaround, user-friendly design systems. In particular, the most needed capabilities include easy selection of application programs, quick review and modification of program input/output data, and logging of the actual steps that were executed during the study. Although the application programs used by VAB and SAB differ, the design methods used by their engineers are quite similar, and great efficiencies can be gained by providing a computer "environment" that yields the capabilities mentioned above.

EASIE is a user interface and a set of utility programs which supports rapid integration and execution of programs about a central relational database. In general, the EASIE system addresses the needs of four different classes of people who will

N88-25184#

be involved in the buildup of an engineering design system. Certain individuals may serve in more than one of these roles, but the following terms will help to clarify several distinct activities associated with the EASIE system.

The first classification represents the engineer/designer/analyst. This group conducts the design study through the execution of modeling and analysis programs and the generation of data required to evaluate the design against its objectives. EASIE documentation will refer to this group as "EASIE system users" or, more often, as "users." In general, these users are only interested in executing programs already installed into an EASIE design system.

A second group aided by EASIE will be referred to as "application programers." These programers/engineers are responsible for the development and improvement of modeling and analysis programs used in the engineering design process. They are the experts with respect to particular application programs and can define its input and output variables. This definition must be done before inclusion of that program with others in the integrated system.

The third group can be referred to as "program implementers," since their function is to provide an environment where all the software tools work together with a minimum of effort. These people will use information provided by the application programers and will install or modify the programs in an EASIE system by creating appropriate data constructs in the database and locating files where needed by the EASIE executive.

The fourth classification is that of "design team leader" or "design manager." This is the individual or group responsible for identifying parameters important to the design study and for configuration management of the data as it is produced by the design team. This design manager must have an overview of the total data requirements for the analysis process and must be concerned foremost with the integrity of the data.

With these terms defined, the four volumes of EASIE documentation can be associated with the groups most likely to use them. Each of the volumes addresses different aspects of the support tools, and each is intended to be usable independent of the others.

Volume I, Executive Overview, provides information about the functions, concepts, and historical development of EASIE and should be read by anyone trying to determine if EASIE would be beneficial to their work.

Volume II, Program Integration Guide, describes the portion of the EASIE tools supporting both the integration of application programs into a central database and the definition of the data dictionary used during data review and modification. This volume will be used primarily by the "program implementer" and the "design manager" in their responsibilities for the actual installation of appropriate programs into a fully-integrated design system. However, the "application programer" may also use tools described in this volume to assist in the documentation of input/output variables for the application program.

Volume III, Program Execution Guide, describes the portion of the EASIE tools supporting the selection and execution of application programs, building of menus, and editing of program data. This volume will be of foremost importance to the "users" who will perform design studies. In addition, the "program implementers" will find the sections concerning the construction of application-dependent procedures helpful. Finally, this document will also be used by the "design manager" for reviewing data and design activities.

Volume IV, System Installation and Maintenance Guide, describes the procedure of loading the EASIE system onto a computer. It also gives some insight into the hardware and software dependencies of the EASIE code. This, most likely, will be needed by the "program implementer" to familiarize himself with the directory structure and location of the various EASIE components. Although the design of EASIE is intended to reduce the system dependencies, this version nevertheless reflects in several ways the current implementation using the Relational Information Management (RIM*) database management system and the VAX/VMS$^+$ operating system.

* Trademark of Boeing Computer Services

$^+$ Trademark of the Digital Equipment Corporation

TABLE OF CONTENTS

Section Page

TABLE OF CONTENTS (continued)

LIST OF FIGURES

1.0 INTRODUCTION

This document describes the "executive" capabilities provided by the Environment for Application Software Integration and Execution, EASIE, and defines the command language and menus available under Version 1.0. References 1, 2, and 3 make up the three remaining volumes of the EASIE documentation set. EASIE provides users with two basic modes of execution. The first, Application-Derived Executive (ADE), is a menu-driven execution mode which provides users with sufficient guidance to review data, select menu action-items, and execute application programs. The other mode of execution, Complete Control Executive (CCE), provides an extended executive interface which allows in-depth control of the design process. For example, when using CCE, techniques are provided which allow the user to establish a design sequence and then automatically re-execute the sequence. This allows the engineer to refine input iteratively and review the results with minimum interaction. Users can switch between these modes as needed.

The sections have been organized and written so that a minimum of reading is needed to be able to begin using the EASIE system. Sections 2 and 3 provide an overview of terms and a sample session sufficient to get started. Sections 4 and 5 provide a more complete description of the system and a more complex sample session. Sections 6 and 7 describe the use of the EASIE procedures and the database structure (files, workspaces, data dictionary, etc.). Section 8 is a quick reference for the full EASIE command language.

2.0 CONCEPTS AND DEFINITIONS

The most predominant system design methodology uses the iterative technique. Here one progresses to a final solution through successive applications of analysis techniques to increasingly refined data. EASIE facilitates this process. With that in mind, the most important concepts for the EASIE user are the configuration data, the application programs, and the work environment itself. These three concepts will be described in the following subsections.

2.1 Configuration Data

Configuration data (so named because the data will describe a single point design or configuration) is stored in a system-managed database. In concept, the design manager or members of a design team will define, develop, and evolve the variables and constants required by their design objectives to describe a basic model. Once this engineering information is determined, it is used to construct a database along with other necessary descriptive information. For example, a typical data entry might include the variable name, a short description of what the variable represents, the type of units to be used when referencing the variable, a default value if applicable, and computer-specific information such as data type (real or integer, scalar or array, and parameter or attribute). These data are stored via a relational database manager. Several studies [4,5] have shown that this is more effective for the storage and retrieval of engineering data than typical hierachical or network databases. However, EASIE does not require that users unfamiliar with a relational database manager learn how to use one. An advantage of the EASIE user interface is that data held in the database are automatically communicated to either a user or an application program in an appropriate format.

Once the basic data definitions and values (serving as the starting point for design) have been made, a copy of this "master" database is placed in a controlled project directory. Access to this database is provided on a "read only" basis. That is, the users may display the configuration data for review, or they may make a copy of the database for their personal files. However, users are prevented from making changes in the master database. Updates to the master database can be entered only by the design manager (or configuration manager). As a side note, there may be several master databases representing different benchmark configurations or versions of design.

There are two methods of accessing a working copy of the configuration database. The first method requires familiarity with the database management software to call up information via the system relational database manager. The second method is to use the EASIE utility modules to perform the database management work.

The EASIE software interface provides a program called the "REVIEWER" which can access any data. An example would be to issue a command to view configuration data related to the inputs required for executing a particular application program. The REVIEWER, using information contained in the database, can then make the appropriate selections to retrieve the necessary input data and present that data at the terminal. As the data are presented, three options arise. First is to choose to move forward (or backward) through the data screens, second is to modify one or more of the entries, or third is to print the information on a nearby printer. These capabilities make the REVIEWER an effective tool for observing changes in data due to execution of an application program with new inputs. Use of the REVIEWER requires no specific knowledge of the database management software. To date, EASIE has been implemented in several versions using the ARIS [6], RIM [7], and PEARL [8] relational database managers.

2.2 The Application Program

Identification of a suite of application programs for use in analyzing a design problem is a necessary step in determining to use EASIE. When EASIE is selected for use, the data required by these suite of programs must be defined by an application programer and initialized in the configuration database [5]. Once the command to execute a given application program has been given, EASIE then takes over. Data are extracted from the database and put in a form appropriate for input to the application program. This is followed by the execution of the application program, and resulting answers are inserted into the configuration database. The software screen forms used to control the flow of data to and from the database are called data templates. A data template is basically a list of all data required for input (or supplied as output) by a given program along with their required data formats. Since data templates are generated by an EASIE utility program, new programs can be integrated easily into the application program suite. Finally, access to these data templates is used in conjunction with the REVIEWER to directly modify the variables in the database when presented during the Review process.

2.3 Working Environment

Experience has shown that an engineer rarely completes one version of a design at one terminal session. Additionally, engineers may also wish to keep several design versions active. This requires several different copies of the configuration database. For this reason the idea of a "workspace" is used with the EASIE system.

Fundamentally, a workspace is used to keep track of the current standing of a particular copy of the database (version of a design). The engineer will thus have different workspaces for each variant of a design.

During execution in the EASIE environment, the workspace automatically records not only the current version of the configuration database but also the current application program, data template, and procedure file in use. In addition, EASIE will automatically record all instructions given during the design activity associated with this workspace. This "command log" plays several important roles. First, it serves as an audit, so there is a record of the order commands were issued or how a particular result was derived. Secondly, the command log can be edited to create "procedures" from any designated portion of the file. Once a procedure has been created, execution of a series of activities can be initiated by invoking the single procedure.

3.0 SAMPLE SESSION USING THE APPLICATION-DERIVED EXECUTIVE (ADE)

Most users will likely access the EASIE system via the menu-driven mode known as ADE. Menus displayed during an ADE session are typically created (using EASIE utilities) by experienced engineers to guide new users through the proper sequence of steps to conduct some particular design activity. The purpose of this section is to describe, via a sample session, an interaction with EASIE for a given application. This example has been devised to illustrate capabilities of the EASIE system but it is not exhaustive. In addition, the menus displayed here have not been patterned to a specific model. An EASIE user or design manager can quickly develop a selection of menus to any desired format. A description of that process will follow in Section 6.0.

An example has been sketched to illustrate highlights of the EASIE system. It consists of four short programs that define and draw a box. Figure 1 represents the basic relationships among these programs and their data.

The BOX program extracts dimensional data from the database, calculates physical properties (volume), and stores it in the database. The MAKGEO program extracts the dimensional data from the database, creates a geometric boundary representation for the box, and stores that data on the database. The object of the DRAW program is to display the box geometry that exists in the database. Since the program DRAW exists only in binary format, the DRAW source code can not be modified to read the geometry data directly from the database. Thus to integrate program BOX an interface program must be coded that will reformat the geometry data.

The interface program DRAWIN is a preprocessor, construction of which is aided by the EASIE installation routines, that interacts with the database to obtain geometric information and provide it in a format compatible with DRAW. Details of the integration procedure can be found in the EASIE Volume II - Program Integration Guide. The combination of the DRAW and DRAWIN programs is referred to as DRAWIT. In the following discussion of the example, user input is indicated by underlined values and is assumed to be followed by a carriage return. Copies of the screens themselves are given in Appendix A and are referenced by the notation "Screen An," where n is the screen number.

The session commences with the user entering the following command sequence:

```
$ EASIE
ENTER TERMINAL TYPE:
 1 FOR TEKTRONIX 4014
 2 FOR TEKTRONIX 4105
 3 FOR TEKTRONIX 4107
 4 FOR TEKTRONIX 4109
 5 FOR TEKTRONIX 4115
 6 FOR TEKTRONIX 4129
 7 FOR AED767
OR <CR> FOR ALPHANUMERIC TERMINAL
```

Screen A1, the first screen presented, provides a menu of the commands available for basic interaction: selection, deletion, editing, and review of configuration data, program execution, and printed output. The first choice from this menu would be CS for the selection of a configuration database.

Screen A2 is the result of that selection. The first four lines displayed indicate the existence only of the master

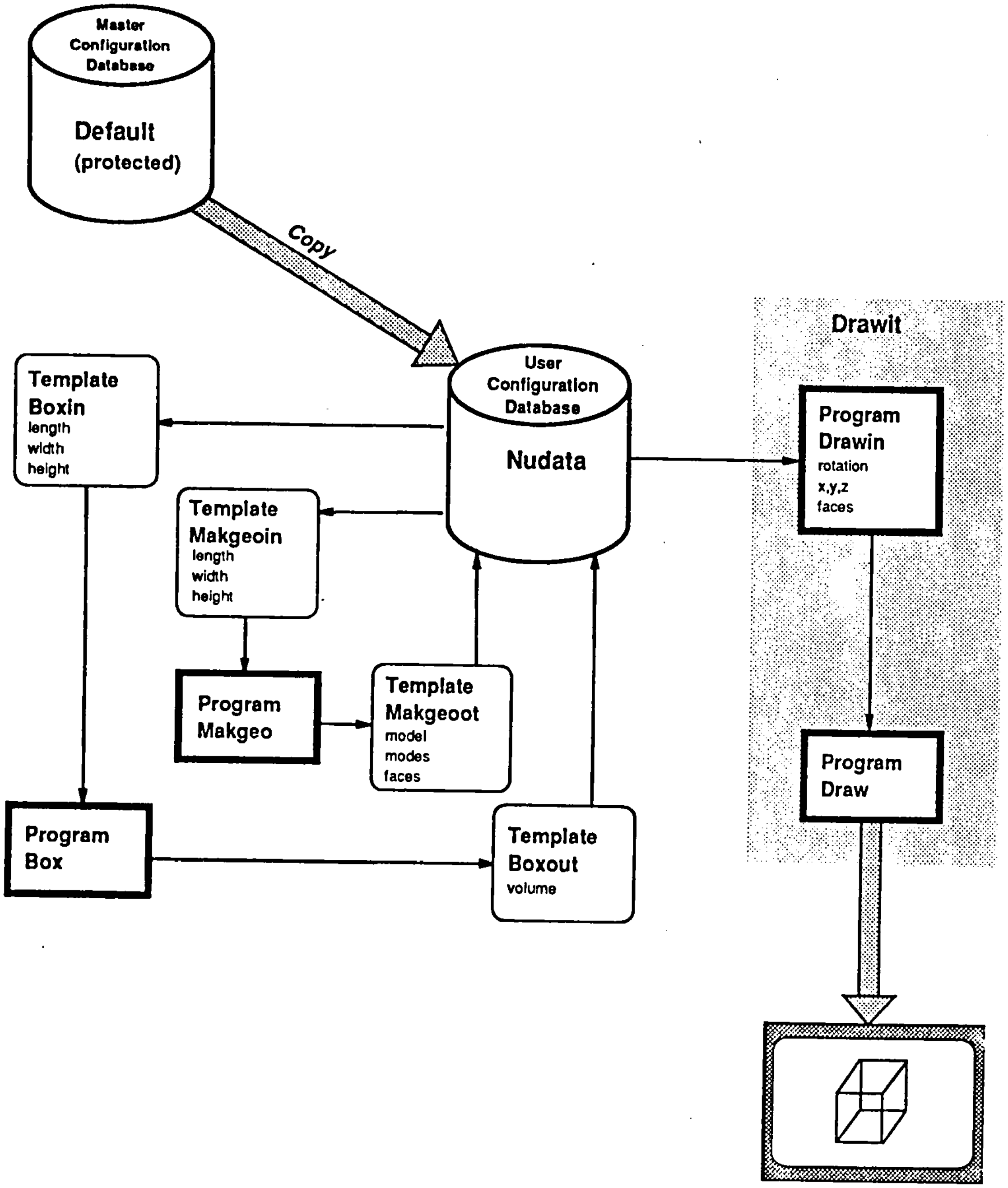


Figure 1. Flow Diagram For Sample Session Using EASIE

configuration database DEFAULT. Since EASIE users may change only personal databases, the only appropriate answers at this time to the "SHOULD A NEW CONFIGURATION BE CREATED (Y=yes):" query is Y. This is followed by specifying DEFAULT as the source configuration to be copied and by entering NUDATA as the name of the destination configuration copy.

Completion of the copy task returns program control to the main menu, screen A3. Here a choice of CD allows the modification of the configuration description file for NUDATA using text describing any special conditions worthy of remarks. Response to the editor prompt of * with EX will cause the program to exit from the editor, save the configuration description file, and return program control to the main menu.

A choice of R for reviewing data leads to screen A4. The menu presented in A4 allows selection of specific data in the database for review and modification. Here the example selects BI which specifies the input for the BOX program. Once this selection is made, EASIE begins execution to the REVIEWER (see Appendix C), and values of variables corresponding to inputs for the BOX program are extracted from the NUDATA database and displayed.

Screen A5 is the first screen presented by the REVIEWER. Recall that the DEFAULT database served as our model for NUDATA. Thus the REVIEWER has presented the default dimensions of 0, 0, 0.

The REVIEWER provides great flexibility to the EASIE system as it can access and modify any portion of the configuration database without the need to consider the type of database manager software in use. EASIE has already been implemented with three different database managers and had no presentation differences.

The commands for the REVIEWER are summarized below.

M	- modify the indicated cell value
C	- change categories of variables
N	- display the next page of variables
R	- redisplay this page
L	- define the length of the page
X	- expand the indicated entry (the descriptive text entries may first appear truncated)
E	- end and save modifications
Q	- quit without saving modifications
H	- help
CAT	- list categories of variables
SUB	- define a subset for later review
T	- toggle menu

Screen A5 shows modification of the box dimensions to 10, 20, and 30. Screen A6, a response to R, reflects those changes; whereas screen A7, a response of * to CAT, indicates that this is the only category of input variables for the BOX program. Exit from the REVIEWER with an E will save these changes and return program control to the review selection menu, screen A8. The choice of R will return the program control to the main menu.

Screen A9 shows the menu choices of the E command to execute a program. Screen A10 indicates the selection of B, the BOX program, and it commences execution immediately.

Upon completion of the BOX program, successive menu choices of R to return to the main menu, R to start the REVIEWER, and BO to show the BOX output data is shown on screen A11.

The exit from the REVIEWER is done this time with a Q, since no modifications were made. The input for program MAKGEO is the same dimensional information that was used in program BOX so the REVIEWER need not be invoked to proceed to execution of MAKGEO with menu choices R to return to the main menu, E to indicate execution mode, and M to choose execution of MAKGEO. The steps of this sequence appear in screen A12.

To check the input to the DRAWIT combination of programs and proceed to the REVIEWER, enter the sequence R to return to the main menu and R to start the REVIEWER. The D is entered to review the input data for DRAWIT, shown on screen A13. Once in the REVIEWER, use the CAT command and *, screen A13, to show that there are three categories of input to the DRAWIT programs. The R is entered to reprint the page, screen A14, which also defines the name and view angles for DRAWIT. Screen A15 is generated by the R (reprint page) command, entered as last line shown on screen A14, and displays those changes. Screens A16 and A17 show the geometry information generated by MAKGEO that will be used to define the box that is drawn by DRAWIT. Since no changes are made in these data, exit the REVIEWER a Q.

Now execute DRAWIT using the sequence of R to return to the main menu, E to select execution mode, and D to select the DRAWIT programs. Screen A18 is the resulting display of the box as created on a Tektronix-type terminal.

The sample session described above, as with any ADE implementation, is designed by an experienced analyst to provide a preplanned set of options; for this example it is those available on the menus of screens A1, A4, and A10. EASIE provides the analyst with utilities to create a predefined procedure file with associated menu files. The procedure file executed has been linked to the particular USER-ID of choice and is automatically executed when EASIE is initiated. Details of procedure construction are given in section 6. For completeness, screens A19 and A20 are shown to illustrate the procedure and menu files linked to USER-ID: EXMENU. Screen A19 is a printout of the procedure file EXMENU.PROC. Screen A20 is a printout of the menu files EXMENU.PROC_1, EXMENU.PROC_2, and EXMENU.PROC_3.

4.0 THE COMPLETE CONTROL EXECUTIVE (CCE) ENVIRONMENT

The basic selections available in the ADE-mode procedures may not provide sufficient flexibility for some users. The Complete Control Executive, CCE-mode, is provided to meet this need. The CCE provides the flexibility of an operating system without the need of tracking a multitude of files, directories, or data. In CCE, commands can be issued via menu selection or directly typed. Various levels of menu, display, and help text are available. Version 1.0 of the EASIE system has seven different standard menus in addition to a "permanent" menu of commands. These menus will be presented below with a brief explanation, followed by an explanation of the EASIE command format which users may find easier to use as they become more familiar with the system.

The Permanent menu (figure 2) provides basic informational access and control. The Utility Selection (MAIN) menu (figure 3) acts as the gateway to the other six functionally-organized menus. To help familiarize users with the CCE command structure of EASIE, each menu provides, in its right-hand column, the command line necessary to obtain the same result as this menu choice. Thus users new to the CCE mode can sequence through menu choices to direct the design process, whereas users more familiar with the CCE mode can direct the design process immediately through keyboard commands. Composition of the menu in this fashion provides an instructive link allowing quick familiarization with the EASIE command structure. For example (figure 3), the command "ACT UTL WSC" activates the Workspace Control menu as would a menu choice of "1" from the Utility Selection Menu. The difference is that "ACT UTL WSC" can be given any time, not just when using the Utility Selection menu.

The Workspace Control menu (figure 4) defines the workspace environment. The Data Review/Modification menu (figure 5) provides the means for selecting a template or view of the database and for executing the "REVIEWER". The Application Execution menu (figure 6) provides control over program execution. The Procedure Execution menu (figure 7) provides control over procedure execution. A procedure is a user-defined collection of EASIE statements covering some portion of the design sequence such as those used when EASIE is set in the ADE mode. The Procedure Building menu (figure 8) would be used in defining a procedure. The Template Building menu (figure 9) provides the control in defining a new view into the configuration database.

All EASIE commands take on the following format:

```
<command verb> [<object type>] [<object name>] [other info as
required]
```

Here the characters within the less than, greater than graphics <> represent the name of a command or object, and characters within the left bracket, right bracket [] indicate optional information.

```
PERMANENT MENU:

H - HELP
D - DIRECTORY of CFG's, TPL's and WS's
I - INVENTORY of APPL's and PROC's
S - SYSTEM COMMANDS
C - Add a COMMENT to the command log
T - TOGGLE the MENU PRINT setting
R - RETURN to the PREVIOUS MENU
Q - QUIT this sequence of menus and
    RETURN to the MAIN MENU
L - LOGOUT

<CR> - Clear the screen and relist the menu
0 - (zero) cancel the command
```

Figure 2. Permanent Menu

```
UTILITY SELECTION (MAIN)


                                            COMMAND FORMAT
1 - WORKSPACE CONTROL UTILITY               ACT UTL  <WSC>
2 - DATA MODIFICATION UTILITY               ACT UTL  <DATA>
3 - APPLICATION EXECUTION UTILITY           ACT UTL  <APEX>
4 - PROCEDURE EXECUTION UTILITY             ACT UTL  <PREX>
5 - PROCEDURE BUILDING UTILITY              ACT UTL  <PBLD>
6 - TEMPLATE BUILDING UTILITY               ACT UTL  <TBLD>
```

Figure 3. Utility Selection Menu

```
WORKSPACE CONTROL

                                                      COMMAND FORMAT
 1 - READ DESCRIPTION   -  WORKSPACE           RD WS     <name>
 2 -                    -  CONFIGURATION       RD CFG    <name>
 3 -                    -  TEMPLATE            RD TPL    <name>
 4 -                    -  APPL. PROG.         RD APPL   <name>
 5 -                    -  PROCEDURE           RD PROC   <name>
 6 - CLEAR LOG OF OLD INFORMATION              CL
 7 - TYPE               -  COMMAND LOG         TY LOG    <name>
 8 -                    -  PROCEDURE           TY PROC   <name>
 9 - NEW                -  WORKSPACE           N WS
10 -                    -  CONFIGURATION       N CFG     <name>
11 - COPY               -  WORKSPACE           CP WS     <f,to>
12 -                    -  PROCEDURE           CP PROC   <f,to>
13 - ACTIVATE           -  WORKSPACE           ACT WS    <name>
14 -                    -  CONFIGURATION       ACT CFG   <name>
15 -                    -  TEMPLATE            ACT TPL   <name>
16 -                    -  APPL. PROG.         ACT APPL  <name>
17 -                    -  UTILITY             ACT UTL   <menu>
18 -                    -  INPUT TEMPL         ACT ITPL
19 -                    -  OUTPUT TEMPL        ACT OTPL
20 -                    -  PROCEDURE           ACT  UTL   <PBLD>
6 - TEMPLATE BUILDING UTILITY                   ACT UTL   <TBLD>
```

Figure 3. Utility Selection Menu

```
WORKSPACE CONTROL

                                                  COMMAND FORMAT
 1 - READ DESCRIPTION  -  WORKSPACE              RD WS     <name>
 2 -                   -  CONFIGURATION          RD CFG    <name>
 3 -                   -  TEMPLATE               RD TPL    <name>
 4 -                   -  APPL. PROG.            RD APPL   <name>
 5 -                   -  PROCEDURE              RD PROC   <name>
 6 - CLEAR LOG OF OLD INFORMATION                CL
 7 - TYPE              -  COMMAND LOG            TY LOG    <name>
 8 -                   -  PROCEDURE              TY PROC   <name>
 9 - NEW               -  WORKSPACE              N WS
10 -                   -  CONFIGURATION          N CFG     <name>
11 - COPY              -  WORKSPACE              CP WS     <f,to>
12 -                   -  PROCEDURE              CP PROC   <f,to>
13 - ACTIVATE          -  WORKSPACE              ACT WS    <name>
14 -                   -  CONFIGURATION          ACT CFG   <name>
15 -                   -  TEMPLATE               ACT TPL   <name>
16 -                   -  APPL. PROG.            ACT APPL  <name>
17 -                   -  UTILITY                ACT UTL   <menu>
18 -                   -  INPUT TEMPL            ACT ITPL
19 -                   -  OUTPUT TEMPL           ACT OTPL
20 -                   -  PROCEDURE              ACT PROC  <name>
21 -                   -  PROGRAM UFD            ACT PUFD  <path>
22 - SAVE TEMPORARY    -  WORKSPACE              SA WS     <name>
23 -                   -  PROCEDURE              SA PROC   <name>
24 - REMOVE FROM UFD   -  WORKSPACE              RM WS     <name>
25 -                   -  CONFIGURATION          RM CFG    <name>
26 -                   -  TEMPLATE               RM TPL    <name>
27 -                   -  PROCEDURE              RM PROC   <name>
28 - SET USER LOGIN CHARACTERISTICS              SLOG
```

Figure 4. Workspace Control Menu

```
DATA REVIEW/MODIFICATION

                                                    COMMAND FORMAT
 1 - READ DESCRIPTION   - TEMPLATE                  RD TPL   <name>
 2 -                    - CONFIGURATION             RD CFG   <name>
 3 - ACTIVATE           - CONFIGURATION             ACT CFG  <name>
 4 -           VIEW AS  - INPUT TMPL.               ACT ITPL
 5 -           VIEW AS  - OUTPUT TEMPL.             ACT OTPL
 6 -           VIEW AS  - ALT. TMPL.                ACT TPL  <name>
 7 - COPY               - CONFIGURATION             CP CFG   <f,to>
 8 -                    - TEMPLATE                  CP TPL   <f,to>
 9 - ACTIVIATE TEMPLATE BUILDING UTILITY            ACT UTL  <TBLD>
10 - REVIEW             - INPUT DB                  RVU IDB
11 -                    - OUTPUT DB                 RVU ODB
12 - REVIEW DB WITH ACTIVE TEMPLATE                 RVU
13 - PRINT              - INPUT DB                  PRVU IDB
14 -                    - OUTPUT DB                 PRVU ODB
15 - PRINT DB WITH ACTIVE TEMPLATE                  PRVU
16 - CHANGE CONFIGURATION DESCRIPTION               CD CFG
17 - REMOVE ALL DATA    - INPUT & OUTPUT            RM CFG
18 - TYPE CURRENT COMMANG LOG                       TY LOG   <-   >
```

Figure 5. Data Review/Modification Menu

```
APPLICATION EXECUTION

                                                    COMMAND FORMAT
 1 - READ DESCRIPTION       - APPL. PROG.           RD APPL  <name>
 2 -                        - CONFIGURATION         RD CFG   <name>
 3 -                        - TEMPLATE              RD TPL   <name>
 4 - NEW                    - CONFIGURATION         N CFG    <base>
 5 - ACTIVATE               - APPL. PROG.           ACT APPL <name>
 6 -                        - CONFIGURATION         ACT CFG  <name>
 7 - ACTIVATE DB VIEW AS    - INPUT TEMPL.          ACT ITPL
 8 -                        - OUTPUT TEMPL.         ACT OTPL
 9 -                        - ALT. TEMPL.           ACT TPL  <name>
10 - COPY                   - CONFIGURATION         CP CFG   <f,to>
11 - REVIEW DB WITH ACTIVE TEMPLATE                 RVU
12 - REVIEW                 - INPUT DB              RVU IDB
13 -                        - OUTPUT DB             RVU ODB
14 - PRINT DB WITH ACTIVE TEMPLATE                  PRVU
15 - PRINT                  - INPUT DB              PRVU IDB
16 -                        - OUTPUT DB             PRVU ODB
17 - EXECUTE CURRENT APPLICATION PROGRAM            EX APPL  <-   >
18 - CHANGE DESCRIPTION     - CONFIGURATION         CD CFG   <name>
19 - REMOVE ALL DATA        - INPUT & OUTPUT        RM CFG
20 - TYPE CURRENT COMMAND LOG                      TY LOG   <-   >
```

Figure 6. Application Execution Menu

```
PROCEDURE EXECUTION

                                                    COMMAND FORMAT
 1 - READ DESCRIPTION     - PROCEDURE        RD PROC    <name>
 2 -                      - CONFIGURATION    RD CFG     <name>
 3 -                      - APPL. PROG.      RD APPL    <name>
 4 -                      - TEMPLATE         RD TPL     <name>
 5 - TYPE A GIVEN COMMAND LOG                TY LOG     <name>
 6 - ACTIVATE             - PROCEDURE        ACT PROC   <name>
 7 -                      - CONFIGURATION    ACT CFG    <name>
 8 - ACTIVATE DB VIEW AS  - INPUT TEMPL.     ACT ITPL
 9 -                      - OUTPUT TEMPL.    ACT OTPL
10 -                      - ALT. TEMPL.      ACT TPL    <name>
11 - COPY                 - PROCEDURE        CP PROC    <f,to>
12 -                      - CONFIGURATION    CP CFG     <f,to>
13 - NEW                  - CONFIGURATION    N CFG      <base>
14 - PROCEDURE BUILDING UTILITY              ACT UTL    <PBLD>
15 - REVIEW THE DB WITH THE ACTIVE TMPL.     RVU
16 - REVIEW                                  RVU IDB
17 -                      - OUTPUT DB        RVU ODB
18 - PRINT THE DB WITH THE ACTIVE TEMPLAT    PRVU
19 - PRINT                - INPUT DB         PRVU IDB
20 -                      - OUTPUT DB        PRVU ODB
21 - EXECUTE THE CURRENT COMMAND FILE        EX PROC    <-   >
22 - CHANGE DESCRIPTION   - PROCEDURE        CD PROC    <name>
23 -                      - CONFIGURATION    CD CFG     <name>
24 - REMOVE ALL DATA      - INPUT & OUTPUT   RM CFG
25 - TYPE THE CURRENT COMMAND LOG            TY LOG     <-   >
```

Figure 7. Procedure Execution Menu

```
PROCEDURE BUILDING

                                                    COMMAND FORMAT
 1 - READ DESCRIPTION       - PROCEDURE      RD PROC    <name>
 2 - TYPE                   - COMMAND LOG    TY LOG     <name>
 3 -                        - PROCEDURE      TY PROC    <name>
 4 - CLEAR LOG OF OLD INFORMATION            CL
 5 - ACTIVATE               - PROCEDURE      ACT PROC   <name>
 6 - COPY                   - PROCEDURE      CP PROC    <f,to>
 7 - DEFINE A NEW PROCEDURE                  N PROC
 8 - EDIT AN EXISTING PROCEDURE              ED PROC    <name>
 9 - EDIT A LOG TO BUILD PROCEDURE           ED LOG     <name>
10 - SAVE A TEMPORARY PROCEDURE              SA PROC    <new>
11 - CHANGE THE PROCEDURE DESCRIPTION        CD PROC
12 - REMOVE THE PROCEDURE                    RM PROC
13 - TYPE THE CURRENT COMMAND LOG            TY LOG     <-  >
```

Figure 8. Procedure Building Menu

```
TEMPLATE BUILDING

                                              COMMAND FORMAT
 1 - READ DESCRIPTION OF TEMPLATE             RD TPL   <name>
 2 - ACTIVATE             - INPUT TEMPLATE    ACT ITPL
 3 -                      - OUTPUT TEMPLATE   ACT OTPL
 4 -                      - ALT TEMPLATE      ACT TPL  <name>
 5 - COPY                 - TEMPLATE          CP TPL   <f,to>
 6 - DEFINE A NEW TEMPLATE                    N TPL
 7 - EDIT THE ACTIVE TEMPLATE                 ED TPL   <-  >
 8 - CHANGE THE TEMPLATE DESCRIPTION          CD TPL
 9 - REMOVE TEMPLATE FROM UFD                 RM TP    <name>
10 - TYPE THE CURRENT COMMAND LOG             TY LOG   <-  >
```

Figure 9. Template Building Menu

The sections which follow are organized in the following fashion. Section 4.1 defines the permanent menu commands, so called since they may be executed at any time by typing the single letter that appears in the Permanent Menu, (figure 2). Sections 4.2, 4.3, and 4.4 collectively represent the "core" of the CCE environment, that portion of EASIE which is required to achieve basic manipulation of the design environment. Sections 4.5 and 4.6 describe additional features of the CCE environment which provide users with greater flexibility in design manipulation.

Beginning users of the CCE mode are recommended to concentrate their readings initially on Sections 4.1 to 4.4. The sample CCE session developed in Chapter 5 involves only commands and objects defined in these "core" sections.

4.1 Permanent Menu Commands

For CCE users, all permanent menu commands can be given by a single letter abbreviation.

H - Help

This command will provide information on EASIE commands and structures. It also provides access to program and data description files that can help explain the use of these objects.

D - Directory

This command will list all configuration databases, data templates and saved workspaces accessible to the user. "Master" files are those existing in the central project directory and are available on a "read only" basis. "User" files represent information previously saved by the users.

I - Inventory

This command provides a list of all accessible application programs and procedures. Again "Master" implies "read only" access and "User" implies user-created.

S - System Command

The system is currently implemented on Prime and VAX computers. This command indicates that what follows should be interpreted as a PRIMOS or VAX/VMS operating system command.

Example: In order to invoke the VMS command to "show current users" through VMS, one needs to issue the command:

```
    SHOW USERS.
Through EASIE this becomes
    S  SHOW USERS.
```

C - Add a Comment to the Command Log

This allows notes to be inserted in the log for later reference and clarity.

T - Toggle the Amount of Menu Information Displayed

This allows the selection of how much or how little information to display. There are four possible settings:

1. Full information.
2. Status variables and utility menus only.
3. Status variables only.
4. Prompt only.

Each time the command T is issued, the next state in the circular sequence is chosen.

R - Return to the Previous Menu

As users move through the development of a design, different menus will be accessed in some sequence. This command is used to return to the previous menu.

Q - Quit this Sequence of Menus

This command will dispose the current sequence of menus and return immediately to the Utility Selection menu.

L - Logout

This command gives an orderly closeout of the EASIE system and returns the user to the computer's operating system.

<CR> - Carriage Return

When issued alone, this will clear the screen for the user or reprint any menus or status information designated by the current display setting. Recall that the display format is set via the "T" command above.

0 - Zero

Will cancel a command when given as a response during a query process for more information.

Example:

```
ENTER COMMAND:  ACT APPL
ENTER FILE TO BE ACTIVATED:  0
** COMMAND CANCELLED **
```

<BREAK> or **< P>** - Send a Break Character

This will cause an application program to halt execution and ask the user for instructions concerning how to proceed.

4.2 Core Object Types

The following terms describe the object types necessary for execution of CCE commands at the "core" level. Illustrations of object types will be cited from the sample sessions contained in Section 5.0. Additional object types are described in Section 4.5.

APPL - Application Program

This includes any application program that can be executed from EASIE. Screen B7 illustrates an inventory listing which includes the application programs associated with the BOX example.

CFG - Configuration Database

A configuration database is a collection of all constants, parameters, and variables pertinent to a given design system or group of analysis programs. Screen B7 illustrates a directory listing including the "master" configuration database DEFAULT for the BOX example.

LOG - Command Log

This is a collection of all commands executed up to the current moment. Screens B17, B19, and B24 illustrate various stages in the development of a command log.

PROC - Procedure Command File

A procedure command file is a collection of EASIE system commands that can be set up for processing by the EASIE command interpreter (Screen 19). Screen B18 illustrates one method of deriving a procedure file from the command log.

UTL - Utility Menu

A utility menu groups together a set of EASIE commands typically used to accomplish a specific task, in a menu format for selection by the user. Figures 2 through 9 of Section 4.0 illustrate the eight "standard" menus.

TPL - Data Template

This includes a list of variables (and their location in the configuration database) created for use by the REVIEWER program. A directory listing as on Screen B6 illustrates the template available to the user.

ITPL - Input Data Template

This represents the input data template for an application program.

OTPL - Output Data Template

This represents the output data template for an application program.

WS - Workspace

This represents a collection of information relevant to the current status of some design. The directory command (illustrated on Screen B21) provides a sample listing of the currently available workspaces. Screen B22 illustrates the activation of a previously saved workspace for further analysis.

IDB - Input Database

This represents that portion of the configuration database associated with the input of some application program, that is, that portion defined by the input template ITPL.

ODB - Output Database

This represents that portion of the configuration database associated with the output of some application program, that is, that portion defined by the output template OTPL.

Note: At times a CCE user will be creating a fresh copy of an object, such an object will be assigned the temporary name T$. For example, at login the user is working with a workspace T$. Any such temporary objects may be saved as permanent at anytime during an EASIE session. If a CCE user attempts to logout of a session with temporary objects, EASIE will solicit whether the user wants to save or dispose of those objects.

4.3 Workspace Variables

One of the purposes of providing a workspace to the CCE user is to track a set of state variables. Once defined, these variables serve as default object names for all commands unless otherwise specified. As previously mentioned, design activities in the CCE environment are conducted and tracked with the help of program and database files. The names of these files are the basic workspace variables. A full list is provided in Section 7. CCE users are presented with complete and flexible control of design activities through definition of the workspace variables. The commands providing this control are described in Sections 4.4 and 4.5.

A brief description of the types of variables associated with the workspace, including how they are used and what values are allowable, is found below.

Configuration Database

Any file of type CFG is a configuration database in the user's file directory. "Master" configurations cannot be associated with the user's workspace. Once the CFG is specified, all programs will refer to this database for data. A list of available databases may be obtained with the D command. This workspace variable is defined by the commands ACTIVATE, NEW, and EDITVAR.

Application Program

Any file of the type APPL is an application program. Once the APPL is defined, any reference to execution will default to this program. Input and output databases and templates will refer specifically to those variables required for input or output for this program. A list of available application programs can be obtained with the I command. This workspace variable is defined by the commands ACTIVATE and EDITVAR.

Data Template

A file of type TPL is a data template that contains a subset of the variables associated with this database. This subset may be user-selected to reflect a particular interest in the design. Once associated with the workspace, it becomes the default selection mechanism for variables in the database to be displayed to the designer. A list of available templates is obtained via the D command. This workspace variable is defined by the ACTIVATE and EDITVAR commands.

Procedure Command File

A file of type PROC is a procedure command file which contains a sequence of EASIE commands. This file allows designers to prepackage sequences of design steps and execute them as a group without further user intervention. Once defined, any reference to procedure execution will default to this file. A list of procedures can be obtained via the I command. This workspace variable is defined by the ACTIVATE, NEW, and EDITVAR commands.

Active Workspace

A file of type WS is an active workspace which defines all associated objects and states of execution. It is possible to SAVE, ATTACH, or DELETE any given workspace in the user file directory. Saving a workspace records the status of that particular project. Attaching a workspace restores the previously saved status. A list of saved workspaces is available through the D command. This workspace variable is defined by the ACTIVATE, NEW, and EDITVAR commands. The workspace T$ is a temporary workspace.

Program File Directory

This is the directory where the project application programs, configuration databases, and input/output templates are found. Definition of this workspace variable is via the ACTIVATE and EDITVAR commands.

Command Autodefault

A CCE user can decide whether a partially specified command will assume default values or have the system prompt the missing information. The value of this variable determines how it will be completed. If the value is true, the command is completed by using the value of the current reference in the workspace for that object. For example EX APPL would execute the current application program. If the value is false, the system will prompt the user for the rest of the command. For example, EX APPL would cause the response ENTER NAME OF PROGRAM:. This workspace variable can be defined by the ACTIVATE and EDITVAR commands.

User Knowledge Level

This variable indicates to the system how much help should be provided to the user. Allowable values are integers from 1 to 3, where 1 represents a novice user and 3 represents an expert user. Definition of this workspace variable is through EDITVAR.

Output Print Level

This variable indicates to the system how much menu information should be routinely provided. The allowable values are integers from 1 to 4, where 1 prints status and all menus, 2 prints status and utility menu, 3 prints status, and 4 prints a prompt only. Definition of this workspace variable is through the T and EDITVAR commands.

Description Files

Each of the basic file types associated with the EASIE system has a description file associated with it. This includes application program files, configuration databases, data templates, command procedures, and workspaces. The intention of the description files is to provide a vehicle by which users of the system can quickly determine basic descriptive information related to a specified file. For example, a user who wishes to know something about a certain configuration database should read the associated description file.

The person or group creating a given file provides the textual description of that file. EASIE automatically creates the description file and enters the user's name and time of creation. The user must then enter the text describing the file. The description can be updated later, using the Change Description command, CD, found in Section 4.5.

4.4 CCE Core Commands

This section contains a list of commands available to the CCE user. The full command name will be followed by its abbreviation if one is provided.

ACTIVATE, ACT

Form: ACT <type> [<filename>]

This command activates any type of object for the user's workspace. This command will define the object indicated as the current reference object of that type.

Example: ACT CFG NUDATA

This command will cause the configuration database, NUDATA to become the current reference database. Not only is this important for later analysis, but it allows the use of defaults. That is, subsequent to the above activation, any command which refers to a configuration database will refer specifically to NUDATA.

Allowable Object Types:

APPL, CFG, ITPL, OTPL, PROC, TPL, WS, UTL, BUFD, PUFD, and DFLT. (An explanation of BUFD, PUFD and DFLT is available in Section 4.5, Additional Object Types.)

Notes:

1. APPL, CFG, PROC,TPL

   If the indicated file does not exist or is not given, a prompt to supply an existing file of the proper type will be displayed.

2. ITPL, OTPL

   No filename is required and will be ignored if supplied. This command defines the value of the current reference template to be the same as the input or output template of the current reference application program. If no program is currently referenced, information reporting this condition will be displayed.

3. UTL

   The allowable filenames for this object type are: MAIN, WSC, DATA, APEX, PREX, PBLD, and TBLD (the standard utility menus). This command displays the indicated utility menu.

4. WS

   This command will restore the characteristics and configuration of a previously defined workspace. If no workspace name is referenced, the user will be asked to provide one.

Caution:

Before using this command, save the currently active workspace, if it has not been saved previously (i.e., if it is listed as temporary).

COPY, CP

Form: CP <type>[<fromfilename> <tofilename>]

Command used to make a copy of an object.

Example: CP CFG DEFAULT NUDATA

This command will make a copy of the configuration database DEFAULT, naming the result NUDATA.

Allowable Object Types:

APPL, CFG, PROC, TPL, WS, FILE

Notes:

1. If the filenames are not supplied, EASIE will solicit them. If the from-filename does not exist or if the to-filename does exist, further quering with concern for proper naming of these files will be displayed.

2. FILE

   This object type allows copying non-EASIE object files without returning to the operating system. Files must be named with appropriate paths and extensions, which is not required for EASIE files.

EDIT, ED

Form: ED <type> [<filename>]

This command invokes the system editor for certain operations.

Example: ED PROC NEWPROC

This command allows the engineer to modify the procedure NEWPROC using the system editor.

Allowable Object Types:

LOG, PROC, TPL

Notes:

1. LOG

   Makes a copy of a log file so that it can be turned into a procedure. All system comments and prompts are deleted before the copy is made. If no log file is specified, the log attached to the current workspace is used.

2. PROC

   This invokes the system editor to modify the indicated procedure file. If no file is given, the reference procedure is used.

3. TPL

   This invokes an EASIE editor designed for defining a user subtemplate.

The template edited will be altered. Thus it should generally begin as a copy of some other desired template (see the COPY command). Template editor commands follow:

```
Sn - Save the variable with reference number
     "n" for the subtemplate.
Fn - Free previously marked variable from the
     subtemplate.
Nn - move "n" pages through the template, note
     n maybe "+" or "-".
T  - move to the top page of variables.
B  - move to the bottom page of variables.
R  - reprint the current page.
E  - end edit and save results.
Q  - quit edit, do not save results, restore
     original template.
```

EXECUTE, EX

Form: EX <type> [<filename>]

This command executes a given application program or procedure command file.

Example: EX APPL BOX

This command will execute the application program BOX.

Allowable Object Types:

APPL, PROC

Notes:

If no filename is given, the current application program or reference procedure will be executed.

PRINT, PR

Form: PR <type> [<filename>]

This command is used to print the indicated file at a local hard copy printer.

Example: PR LOG T$

This command will cause the log associated with the temporary workspace, T$, to be printed.

Allowable Object Types:

LOG, PROC, FILE, BAT

Notes:

1. LOG, PROC

   If no filename is given, then the currently active log or reference procedure command file will be printed.

2. FILE

   This allows the user to print non-EASIE object files without going back to the operating system.

3. BAT

   This prints the system runtime log for the indicated batch job.

PRINTVU, PRVU

Form: PRVU [<type>]

This command prints a template or "view" of a configuration database.

Example: PRVU IDB

This command will obtain the values of the input variables for the current application program from the current configuration database and print them with their database description.

Allowable Object Types:

IDB, ODB, TPL

Notes:

1. If no object type is listed, the system will print variables indicated by the current template and the current configuration database.

2. If there is no active configuration database, the command will terminate execution, and an informative message will be displayed.

READ DESCRIPTION, RD

Form: RD <type> [<filename>]

This command reads the description file associated with any workspace, program procedure, template, or database.

Example: RD APPL BOX

This command will send the description file of the application program BOX for display on the terminal.

Allowable Object Types:

APPL, CFG, ITPL, OTPL, PROC, TPL, WS

Notes:

1. APPL, CFG, PROC, TPL, WS

   If no filename is given, then the description for the current value in the workspace will be presented.

2. ITPL, OTPL

   The current input or output template description will be presented, thus no filename is necessary.

REMOVE, RM

Form: RM <type> [<filename>]

Command to remove a file from the user's file directory.

Example: RM CFG NUDATA

This command will remove the configuration database NUDATA from the user's file directory.

Allowable Object Types:

APPL, CFG, PROC, TPL, WS, FILE

Notes:

1. To prevent accidental deletion, the engineer will need to verify the REMOVE command. Only files from a personal file directory may be removed.

2. APPL, CFG, PROC, TPL, WS.
   If the object file to be removed is one currently active in the workspace, a warning will be displayed on the terminal.

3. FILE

   This type will allow the engineer to remove any personal file without leaving the EASIE system.

REVIEW, RVU

Form: RVU [<type>]

Command to invoke the interactive "REVIEWER" program. This command will display for possible modification a "view" of a configuration database. A view of a database is defined as the collection of variables defined by a data template.

Example: RVU IDB

This command will obtain values of the input variables for the current application program from the current database. Interactive commands may then be issued to modify these data.

Allowable Object Types:

IDB, ODB, TPL

Notes:

1. If no object type is given, the REVIEWER will be executed using the current reference data template and the current configuration database.

2. If there is no active configuration database, the command will terminate execution, and an informative message will be displayed.

SAVE, SA

Form: SA <type> [<filename>]

Command to save a current temporary object for later work.

Example: SA PROC NEWPROC

This command will save the current temporary procedure as NEWPROC.

Allowable Object Types:

PROC, WS

Notes:

1. In general it is assumed that the current reference procedure or work space is temporary, that is, has the name T$.

2. WS

Save may also be used to copy the current status of the workspace into a new workspace.

TYPE, TY

Form: TY <type> [<filename>]

This command displays a file at the terminal. Please note that lines are displayed in groups for the engineer's convenience.

Example: TY PROC NEWPROC

This command will display the procedure command file NEWPROC.

Allowable Object Types:

LOG, PROC, FILE, BAT

Notes:

1. LOG, PROC

   If no filename is specified then, the currently active log or procedure command file will be displayed.

2. FILE

   This allows the display of non-EASIE object files without going back to the operating system.

3. BAT

   This allows the typing of a BATCH JOB event file e.g., TY BAT BAT3.

4.5 Additional Object Types

The following list completes the collection of allowable symbols that can be used as the object of a CCE command. Recall the "core" object types are found in Section 4.2.

BAT - Batch Files

These files relate to the execution of EASIE programs by the batch processor. This includes both a command file to be run by the batch processor and a log file generated during a batch execution.

DFLT - Default Indicator Switch

The value of the default indicator determines if a partial CCE command is completed with a currently active object (the usual setting) or if the prompt to complete the command is displayed. A switch is accomplished via the command, ACT DFLT.

D,T - Log File Indicators

These objects are used exclusively with the CLEARLOG command. CL D clears the log prior to some date while CL T erases the entire log file.

FILE - Generic System File.

At times an engineer may need to refer to a general system file, one not generated by EASIE. This object allows access to such a file through CCE. The complete filename must be specified. For example to type out a general ASCII file named MY.TEXT Enter: TYPE FILE MY.TEXT.

JMPC - Procedure Counter Jump Control

This core object type is used exclusively by the GET command to help control procedure flow. Refer to Section 6.3 for a detailed discussion.

JMPL - Procedure Counter Jump Control (See JMPC).

BUFD - User File Directories

This allows CCE users to search other base (BUFD) or project (PUFD) directories for additional procedures or data. For example, the command ACT PUFD BILL will activate the user file directory BILL. File searches will now automatically include this directory.

PUFD - User File Directories (See BUFD).

QRYC - Procedure Counter Control Via Query

This object is used exclusively by the GET command to control procedure flow in response to a users interactive query. Refer to Section 6.3 for a detailed discussion.

QRYL - Procedure Counter Control Via Query (See QRYC).

MENU - Menu File

This object is used exclusively by the GET command to present an interactive menu. Refer to Section 6.4 for a detailed discussion.

LACT - Log Activation Time

This core type object is used exclusively by the GET command to enter the current time into a log file. This is usually used at the start of a command procedure file to time stamp the associated command log. The format is GET LACT.

MODE - Set Mode

This core object type is used exclusively by the GET command to ensure proper transition from the procedure driven ADE mode to the interactive CCE mode. The format is GET MODE.

4.6 Additional CCE Commands

The following list completes the collection of CCE commands. Recall that "core" commands are given in Section 4.4. The full command name will be followed by its abbreviation if one is provided.

Change Description, CD

Form: CD <type> [<filename>]

This command allows expansion, modification, or updating of description file using the system editor.

Example: CD WS EXAMPLEWS

This sample command allows the expansion or updating of the description file for the workspace EXAMPLEWS.

Allowable Object Types:

APPL,CFG,ITPL,OTPL,PROC,TPL,WS

Notes:

1. APPL, CFG, PROC, TPL, WS

   If no filename is given, then the description for the current value in the workspace will be available for change.

2. ITPL, OTPL

No filename is necessary. The description of the current input or output template will be made available for change.

CLEARLOG, CL

Form: CL [D|T]

This command allows removal of old information cluttering the log file of the attached workspace.

Example: CL T

This command would clear the log completely.

Allowable Types:
D - clear information prior to a given date
T - clear the log totally

Notes:

1. D

A menu of all logged dates and times is displayed, and all log information prior to the date selected will be deleted.

2. T

All information in the log will be deleted.

CNAME, CN (Change Name)

Form: CN <type> [<oldname> <newname>]

This command allows the changing of the name of any user files, workspaces, configurations, templates, or procedures.

Example: CN TPL OLDTPL NUTPL

This command will change the name of the template from OLDTPL to NUTPL.

Allowable Object Types:

APPL, CFG, PROC, TPL, WS, FILE

Notes:

1. APPL, CFG, PROC, TPL, WS

If the old and the new file are not supplied, they will be solicted.

2. FILE

The name of any file in the personal directory can be changed without returning to the operating system.

EDITVAR, VAR

Form: VAR

Changes in the active variables of a workspace can be interactively edited using this command.

Example: VAR

This causes the menu shown below to appear.

```
Allowable Object Types:
none required

NUMBER      LOGIN VARIABLE                        VALUE
------------------------------------------------------------------
   1     USER KNOWLEDGE LEVEL                       2
   2        OUTPUT PRINT LEVEL                      4
   3        COMMAND AUTODEFAULT                     T
   4     CONFIGURATION DATABASE                   NUDATA
   5       USER DATA TEMPLATE                     BOXIN
   6     PROCEDURE COMMAND FILE
   7       APPLICATION PROGRAM                     BOX
   8       ATTACHED WORKSPACE                      T$
   9     PROGRAM FILE DIRECTORY              TOEASIE:[PROG]
------------------------------------------------------------------
n - modify "n"th variable              "0" - to cancel
R - reprint the choices
>
```

GET

Form: GET <type> [<options>][<messages>]

This command is used by programers as they build a procedure. It is used to interactively request input data during the execution of that procedure. See Section 6.2 for an in-depth discussion of the GET command.

Example: GET CFG

This command presents a listing of both master and user database configurations and displays a request to choose one. The chosen configuration is then activated in the current workspace.

Allowable Object Types:

APPL, CFG, JMPC, JMPL, LACT, MENU, MODE, PROC,
QRYC, QRYL, TPL, WS

Notes:

1. APPL, CFG, PROC, TPL, WS
An appropriate list of master and user objects are displayed on the terminal. The choice will be automatically activated in the current workspace. If a "master" CFG is chosen it will, in addition, be copied into the user's file directory.

2. JMPC, JMPL
These jump statements cause an unconditional change in the value of the procedure counter. JMPC causes a counted, relative change in the procedure counter. The jump may be either positive or negative.

```
e.g. GET JMPC 7     (pc = pc +7)
     GET JMPC -2    (pc = pc -2)
```

JMPL causes the procedure counter to be set to the statement following the indicated label. Aside label statements must have the form

```
C LABEL: <label_id>
```

```
e.g. GET JMPL A5 (pc is set to statement
                  after C LABEL:A5)
```

3. QRYC, QRYL
These conditional jump statements can cause the procedure counter to change based upon the answer to a query. QRYC causes a counted, relative change in the procedure counter. The jump may be either positive or negative. The jump occurs only if the response to the query is "Y".

```
e.g. GET QRYC -11
     RERUN THE APPLICATION  (Y/N)?
```

When the answer is "Y", pc = pc-11 and QRYL causes a jump to the indicated label.

```
e.g. GET QRYL
     LAST EXIT THE PROCEDURE  (Y/N)?
```

When the respond is "Y", the pc would be set to the statement after C LABEL:LAST

4. LACT
This causes the current date and time to be entered into the log. Each procedure should begin with a GET LACT so that the activation date will be placed in the log.

5. MENU
This causes the indicated menu to be presented to the terminal. When a choice is made, the pc will be reset to execute the portion of the procedure corresponding to that choice. An in-depth discussion of this statement appears in section 6.4.

```
e.g. GET MENU 2
```

6. MODE
This statement is used to insure that EASIE recognizes terminal input and output modes after procedure execution.

```
e.g. GET MODE
```

NEW, N

Form: N <type> [<name>]

This command is used to get a fresh object.

Example: N WS

This command will cause the replacement of the current workspace with a new, temporary workspace.

Allowable Object Types:

CFG, PROC, TPL, UTL, WS

Notes:

1. CFG, TPL
Since new configuration databases and templates must use some existing configuration or template as a base, the name of the base configuration or template will be requested if it has not been provided. The name given will be attached to the current work space.

2. PROC
A new (empty) procedure file will be attached to the current workspace.

3. UTL
A new utility menu will be attached to the current workspace. This is equivalent to the ACT UTL.

4. WS
   A new (empty) temporary workspace will be provided after the current workspace is saved or deleted.

SETLOGIN, SLOG

Form: SLOG

This Command allows the engineer to create or modify variables specifying characteristics of the login environment. For example a designer may always wish to begin a session with a particular workspace and configuration database active. This command allows those objects to be specified. Upon issuing the SLOG command, the designer is placed in an interactive mode for defining elements of the login file.

Example: SLOG

This causes the menu shown below to appear.

```
Allowable Object types:
None required.

NUMBER      LOGIN VARIABLE         VALUE
-------------------------------------------------
  1        USER KNOWLEDGE LEVEL      2
  2         OUTPUT PRINT LEVEL       2
  3         COMMAND AUTODEFAULT      T
  4     CONFIGURATION DATABASE
  5         USER DATA TEMPLATE
  6     PROCEDURE COMMAND FILE
  7         APPLICATION PROGRAM
  8         ATTACHED WORKSPACE     T$
  9     PROGRAM FILE DIRECTORY     TOEASIE:[PROG]
 10         MENU PRESENTED         UTILITY SELECTION
 11     BEGIN EXECUTING PROCEDURE          F
 12       PROCEDURE TO EXECUTE
------------------------------------------------
n - modify "n"th variable  S - save changes
R - reprint choices       Q - quit, ignore changes
>
```

STATUS, STAT

Form: STAT

This command returns the status of all jobs which have been submitted to the batch processor. Logs for jobs that have completed execution appear in the "BATCH JOB event files". These files can be read with the TYPE command. Active BATCH JOBs are listed in the Active BATCH Queue.

Example: STAT
This will cause the following response.

```
BATCH JOB event files

BAT3
BAT1
BAT2

Active BATCH Queue

    Batch queue  JRC$BATCH

    Batch queue  SYS$BATCH

 Jobname         Username      Entry    Status
 BAT2            VAB             191    Executing
```

Allowable Object Types:

None are required.

SUBMIT, SUB

Form: SUB <type> [<filename>]

This command sends a given application program to the BATCH processor for execution in the background.

Example: SUB APPL BOX

This command will submit the application program BOX to the BATCH processor for execution.

Allowable Object Types:

APPL

Notes:

1. There must be a configuration database active before this command is issued so that the BATCH processor will know where to find necessary data.

2. The user will be notified when processing is completed for a given submission.

3. All submitted jobs take on a name like BAT<n> for some integer n.

5.0 SAMPLE SESSION USING THE COMPLETE CONTROL EXECUTIVE (CCE)

The following EASIE session is included as a sample for the CCE user to follow. The screens given in Appendix B were recorded during the session. In the listing below, each command line must be followed with a carriage return. References to screen in Appendix B will be denoted by Screen Bn where n represents the screen number. User inputs will be underlined. This sample session has been put together to highlight the capabilities of the EASIE system, using the CCE environment. The programs used as the design system for this simple example are the same ones used in the ADE sample session in section 3.0. In summary they are:

```
BOX    - generates volume based on dimensional data.
MAKGEO - generates coordinate geometry based on
         dimensional data.
DRAWIN - a preprocessor for the DRAW program.
DRAW   - plot routine for the box.
```

Figure 1 in section 3.0 expresses the relationships among these programs.

The session commences with the user entering the following command sequence:

```
$ EASIE
  ENTER TERMINAL TYPE:
  1 FOR TEKTRONIX 4014
  2 FOR TEKTRONIX 4105
  3 FOR TEKTRONIX 4107
  4 FOR TEKTRONIX 4109
  5 FOR TEKTRONIX 4115
  6 FOR TEKTRONIX 4129
  7 FOR AED767
 OR <CR> FOR ALPHANUMERIC TERMINAL

ENTER USER ID :
EXAMPLE
```

Screen B1 is the general login screen presented to users who log in using default login characteristics. Notice that it includes a summary listing (STATUS) of important workspace variables. Next it displays the permanent menu also seen in figure 2. Finally it displays the utility selection menu through which other command menus may be accessed. The utility selection menu is also shown in figure 3. Figures 4 through 9 display the other utility menus.

Screen B2 shows the selection of menu choice 1 which results in screen B3. Notice that menu choice 1 activated the workspace control utility and that the corresponding menu is now presented in screen B3.

Users that become familiar with any of the aspects of EASIE have the option of turning off the additional printing associated with menu selection. In this example familiarity with the permanent menu commands can eliminate its display. The command T appearing on screen B3 directs EASIE to move to the next screen display mode (known as menu "toggling"). The result screen B4 no longer displays the permanent menu.

In general there are four such screen display modes:

1. Display STATUS, PERMANENT MENU and UTILITY MENU.
2. Display STATUS and UTILITY MENU.
3. Display STATUS.
4. Only prompt the user for the next command.

The T command sequences through this list in a circular fashion.

Screen B5 shows the entering of the permanent menu command D. Note that any valid EASIE command can be issued at any time whether or not the associate menu is present. The D command asks for a directory of configuration databases, templates, and workspaces resulting in screen B6. Recall here that the term "Master" refers to those files set up for everyone working on the project. They may be copied or executed in the case of programs but may not be modified by project members other than the design team leader. For screen B6 the user has no currently defined configuration databases or data templates. The user workspace T$RTA4 is the temporary workspace the user is currently using. Temporary workspaces are assigned by default during user login.

Next the permanent menu command of I was issued to obtain an inventory of application programs and procedures available to the user. Screen B7 is that inventory.

Generally, design operations are conducted in the EASIE system by identifying a database for gathering configuration values and running application programs that interact with that database. Since the user in this example has no "USER" databases, one must be created by copying from the "Master" list. As screen B6 indicates, DEFAULT is the only choice. Screen B8 shows how the copy command is issued. Here the user first chooses menu item 10 asking for a new configuration database and is then prompted for the names of the source for this database and the destination or new name. The response is DEFAULT and NUDATA, respectively. As a result, screen B9 shows that NUDATA is now the reference configuration database. This implies that anytime an application program calls for data, EASIE will reference NUDATA.

Screen B9 also illustrates the choosing of the application program to run. First enter the menu choice of 16 asking to activate a program and then respond with the name of that program i.e. BOX. Screen B10 illustrates that BOX has become the application program, thus any reference to an APPL object will correspond to BOX.

To start the execution of the REVIEWER, issue the command RVU IDB as shown on screen B10. Since the reference configuration database is NUDATA, the REVIEWER will select data from that database. The object type IDB refers to the input data for the active program. Since BOX is the active program, the REVIEWER will select the BOX input data for display.

Screen B11 shows the REVIEWER'S first screen. Modification of length, width, and height proceeds precisely as in the ADE example of section 3. Screen B12 appears when the REVIEWER is directed to list all categories for the current input via the command CAT. Screen B13 is presented by the REVIEWER after an R command to reprint the screen. Notice the new dimensional values are appropriately recorded. At this point the data values are correct and exiting from the REVIEWER with the E command will save the changes introduced.

EASIE automatically returns to command mode ready to execute the application program. To assist with this process, the application execution utility is initiated with the following sequence of commands. Q is the permanent menu command to quit the current utility menu and return to utility selection. That is followed by a choice of 3 for application execution resulting in screen B14. Execution of BOX is initiated with a command of

17. Note that using the CCE mode command EX APPL would bypass the activation of the application execution utility.

Upon completing execution, the REVIEWER is called to look at the output data. As a side comment, if BOX had presented any interactive input or output, it would have occurred prior to EASIE soliciting the next command. Screen B14 shows the REVIEWER being invoked with RVU ODB for review of the output data from BOX. At this point, entering the command 13 would have obtained the same result. Screen B15 presents the screen of output for BOX. Since there is no desire to change output values by hand, the quit without saving mods command Q is issued.

Upon return to EASIE, a review of exactly what has been done during this session is obtained through choice 20 in the application execution utility resulting in screen B16. Notice that all EASIE commands that have been issued since the start of this session are recorded in the log file following the > character. These commands are captured in a completed, fully expanded manner. For example, the commands for creating a new database on screen B8 are stored here as

```
CP CFG DEFAULT NUDATA
```

The log includes system explanations on lines starting with *.

At this point, it may be determined that this log contains a sequence of commands that will be reused frequently. All or part of a log can be turned into a procedure which allows execution of several steps via a single command. To activate the procedure building utility, use the command sequence Q followed by 5 in the utility selection menu as shown in screen B17. The menu pick 9 is selected in order to edit the log and define it as a procedure.

At this point the system editor is invoked, and a modified copy of the log file is opened for editing as shown in screen B18. Notice two things. First, all system comment lines, those preceded by *, are removed and second, all > symbols are removed from the front of command lines. Now edit the commands as desired. In this example, this results in saving the three command lines.

```
RVU IDB
EX APPL BOX
RVU ODB
```

Exit the editor when satisfied with the collected commands. Now issue the command TY LOG, and Screen B19 is displayed. Note that now appended to the log file is the information concerning activating the procedure building utility and editing the log as a procedure.

To save the procedure just created, issue the command 10 and respond to the name request with NEWPROC. To exit the program, log out with the L command. At this point the EASIE system displays the message it is in a temporary workspace and queries about saving it. Respond with a Y and the name SAVDWS to save the work completed at this session.

When it is desired to resume working with the design, log in as at the start of Section 5.0 as shown on screen B20. Also, screen B20 shows the workspace as a temporary workspace, T$, and the inventory list in response to the Permanent Menu command I. Screen B21 is the response to the D command requesting directory and database information. Screen B20 reflects the newly saved "USER" procedure NEWPROC while screen B21 displays the newly created database, NUDATA, and workspace, SAVDWS.

Screen B22 shows the selection of menu pick 13 to activate the workspace saved earlier with the name SAVDWS. The system automatically responds with screen B23 restoring the condition existing just before the logout. Compare the last status on screen B19 with the status on screen B23. The command TY LOG shows on screen B24 what has happened upon requesting workspace activation.

To execute the procedure created earlier using the Procedure Building Utility, enter EX PROC as shown on screen B25. Here several assumptions are made by EASIE. Since no PROC name was specified, EASIE uses the current reference procedure NEWPROC. Also, any need for data will refer to the active database NUDATA, and any unspecified APPL references will interact with the current application program BOX.

Recall the first line of NEWPROC is RVU IDB. Screen B26 reflects this application of the REVIEWER. The user changes the width of the box to 10 and exits, saving the change with E. EASIE continues by immediately proceeding to the second statement in NEWPROC, namely EX APPL BOX. The last entries of screen B26 reflect that execution. Finally, the last line of NEWPROC is executed, RVU ODB. Screen B27 is the result of that REVIEWER. Exiting the REVIEWER returns the user to the EASIE command level.

The final illustration is to enter the TY LOG command. Screen B28 is the result. Notice the log has tracked the session through the execution of NEWPROC.

6.0 PROCEDURES

Given the ease with which an introductory user can learn to manipulate data and execute programs in the Application Derived Executive mode, ADE, this mode will probably become the predominant choice of system users. As mentioned in section 3.0, control of the system during ADE mode is governed by command procedure. Thus the construction of such procedures should be considered in order to fit the needs of users who will use the ADE mode.

6.1 Basic Procedures

The example below illustrates that one can build and execute procedures which do not specify the names of configuration, application programs, and others. Let the following procedure be called EX1.

```
RVU IDB
EX APPL
RVU ODB
```

Naturally, for this procedure to make any sense, several conditions must be satisfied. First, a workspace variable called AUTODEFAULT must be set to TRUE, so that partial commands like these can be completed with defaults from the workspace STATUS variables. The EDITVAR and SETLOGIN commands described in Section 4.6 provide one method for making this definition. Secondly, those workspace variables must already be defined. Specifically, the reference database must be selected along with an application program for execution.

To complete this discussion, let us assume that the AUTODEFAULT mode has been set to TRUE and that workspace variables assume the values indicated below and relate to sample user sessions described in Sections 3 and 5.

Configuration Database: NUDATA
Application Program: BOX
Input Template: BOXIN
Output Template: BOXOUT

Under these assumptions, let us consider the execution of procedure EX1.

The statement RVU IDB causes the REVIEWER to be executed for the input data to BOX. This means that the values of variables listed in the template BOXIN found in the configuration database NUDATA will be presented to the user for potential modification.

The statement EX APPL causes the application program BOX to execute. The process begins by automatically extracting the values of input data, those just reviewed, from NUDATA. Execution proceeds as usual, and the final results, if any, are automatically stored back in NUDATA.

The final command statement RVU ODB executes the REVIEWER again. This time the values displayed correspond to those listed in the output data template BOXOUT.

Not requiring the names of application programs, the configuration database, and the data templates, to be specified in the procedure adds significant flexibility to the utility of the procedure. By simply changing the values of variables associated with the workspace, a totally different design execution would be performed by the execution of procedure EX1.

6.2 Get Command: The Basics

The procedure EX1 in the previous example executes only when the AUTODEFAULT mode is set to TRUE and certain workspace variables have been defined. For a CCE user, EASIE commands such as ACTIVATE allow for the set up of the workspace variables prior to execution of procedure EX1. However, an ADE user is not expected to access the EASIE CCE commands such as those described in Section 4.0. This section describes the command formats that will allow a procedure to obtain this information for itself, that is, commands which remove the requirement for predefining workspace variables.

Certainly the procedure writer could add a command such as ACT APPL BOX. This would cause workspace variables to take on the values associated with the application program BOX. However, the flexibility of not naming workspace variables mentioned for procedure EX1 in the last section is completely lost. That is, suppose it is desired to run the MAKGEO program instead.

The GET command addresses this situation directly. When included in a procedure file, it will GET information concerning workspace STATUS variables. For example, the addition of the statement GET APPL to procedure EX1 would initiate the following actions. First, a list of all project-associated application programs is presented, and a choice is solicited. Once made, that choice is automatically activated as the current application program. Below is a collection of GET statements that cause a similar response.

```
GET APPL          - for application programs
GET CFG           - for configuration databases
GET PROC          - for new procedures
GET TPL           - for a database review template
GET WS            - for a new workspace
```

Using the GET statements, modify procedure EX1 to generate the procedure EX2.

```
GET CFG
GET APPL
RVU IDB
EX APPL
RVU ODB
```

The addition of the first two lines guarantees that the appropriate workspace variables have been defined by the time they are needed in lines 3, 4, and 5.

6.3 Get Command: Procedure Flow Control

Since design is generally iterative in its nature, the procedures controlling the EASIE sessions for ADE users should have the ability to jump and loop when needed over EASIE commands within the procedures. During the execution of a procedure, EASIE will keep track of its position within the procedure via a procedure counter (pc). The GET command provides several techniques for modifying the value of the procedure counter and thus allowing for jumps and loops.

Basically, the procedure counter may be changed in two ways. First, the procedure counter can be changed by the addition of a numerical offset. The offset may be either positive or negative. Alternately, the procedure counter may be reset by jumping to a labeled statement within the procedure. In either case the jump can be (but need not be) made conditional upon the response entered. Examples of each type of procedure modification follow. Refer back to Section 4.6 for the formal definition of the GET command sequence.

6.3.1 Unconditional Jump Via Numerical Offset

The form for an unconditional jump via a numerical offset is GET JMPC <n> where <n> represents the amount of the offset. That is pc = pc + n.

Example 1.

```
     GET JMPC 3
     .....
     .....
     .....  <-- next procedure statement executed
```

Example 2.

```
     .....
     .....  <-- next procedure statement executed
     .....
     .....
     GET JMPC  -4
```

6.3.2 Unconditional Jump to a Label

Labels are placed in a procedure with a comment statement of the form:

```
     C LABEL:<label_id>
```

where <label_id> is any sequence of alphanumeric information. The statement GET JMPL <label_id> will search for the appropriate <label_id> and set the pc (either positive or negative) to the statement following that <label_id>.

Example 3.

```
     GET JMPL THERE
     .....
     .....
     .....
     C LABEL:THERE
     ..... <-- next procedure statement executed.
```

6.3.3 Conditional Jump Via Numeric Offset

Conditional jumps depend on the response to a procedure-generated query. The format for a conditional jump via numeric offset is GET QRYC <n> <query_text>. Here <n> is again the offset, and <query_text> is the text to be used by the procedure to query the ADE user. EASIE assumes the jump should be performed if a Y is the response entered and assumes no jump for any other response.

Example 4.

```
     GET QRYC 5  NEW PROGRAM (Y=yes):
     ..... <-- next procedure statement for any response but Y.
     .....
     .....
     .....
     ..... <-- next procedure statement for a response of Y.
```

6.3.4 Condition Jump to a Label

Combining the ideas of 6.3.2 and 6.3.3, the format is:

```
     GET QRYL <label_id> <query_text>.
```

Example 5.

```
     GET QRYL  NEWPLACE   NEWPROGRAM  (Y=yes):
     .....<-- next procedure statement for any response but Y.
     .....
     .....
     C LABEL:  NEWPLACE
     ..... <-- next procedure statement for a response of Y.
```

6.3.5 An Example

Now extend procedure EX2 of section 6.2 to create procedure EX3.

```
     GET CFG
     GET APPL
     C LABEL:  EXECUTE
     RVU IDB
     EX APPL
     RVU ODB
     GET QRYC 4        NEW PROGRAM  (Y=yes):
     GET QRYC 5        NEW DATABASE (Y=yes):
     GET QRYL EXECUTE  RERUN PROGRAM SEQUENCE (Y=yes):
     GET JMPL   DONE
```

```
GET APPL
GET JMPC    -4
GET CFG
GET JMPC    -5
C LABEL:    DONE
L
```

Note that there are two label statements: EXECUTE and DONE. The seventh statement, GET QRYC 4 NEW PROGRAM (Y=yes):, causes "NEW PROGRAM (Y=yes):" to be displayed and the procedure to wait for user input. If the first character of the input is Y, then pc = pc + 4, and the procedure will execute statement GET APPL next. Any other response causes pc = pc + 1. The ninth statement, GET QRYL EXECUTE RERUN PROGRAM SEQUENCE (Y=yes):, works in a similar fashion, except that a Y response causes the pc to reset so that RVU IDB is the next statement executed. The GET JMPC and GET JMPL statements work in a similar manner. Finally, the L command causes the user to logout.

6.4 Get Command: Menu Manipulation

Procedure EX3 is limited in the sense that it always forces a key entry in the series of three queries. This can become quite tedious, and it is generally more user friendly to present a list of choices and execute the portion of the procedure that corresponds to the selected choice.

Menus can be presented to the ADE user via the "GET MENU" command. The format for this command is:

```
GET MENU <n>      where <n> represents an associated menu
                  number.
```

The combination of this feature, along with the ability to jump and loop within a procedure as described above in Section

6.3, provides EASIE with the flexibility to make the ADE interface work.

In order to fully utilize the menu feature, a more complete understanding of the files associated with a procedure is necessary. For reference see screens A19 and A20 of Appendix A.

Recall that Appendix A represented the screens displayed during a typical session for an ADE user. In this session the user automatically begins executing the procedure EXMENU at login to the EASIE system. As can be seen in screen A19 the system command

"TYPE EXMENU.PROC"

shows the procedure commands for EXMENU that are contained in the file EXMENU.PROC.

A review of the commands in this procedure reveals the use of the GET MENU command three times--namely, commands 2, 4, and 6. Since each of these has a different number, it refers to each of the menus listed on screen A20. For example, GET MENU 2 refers to the menu contained in file EXMENU.PROC_2. In general, the use of the statement GET MENU <n> in a procedure with the name <proc_id> requires the existence of a menu file with the name <proc_id>.PROC_<n>.

In addition the menu file will be in the following format:

<jump_label> <choice_id> <menu_text>.

Here <jump_label> is 8 alphanumeric characters and corresponds to a label back in the procedure with the form

C LABEL: <jump_label>.

When this menu is called, the next two fields are presented to the user as follows:

<choice_id> - <menu_text>

The user must then select one of the <choice_id>'s presented by typing it. The procedure then continues by resetting the procedure counter to the statement following the corresponding <jump_label>.

The following procedure, EX4, is a menu-modified version of the procedure EX3 that appeared in Section 6.3.

```
GET CFG
GET APPL
C LABEL:  MENU
GET MENU 3
GET JMPL MENU
C LABEL:  DATAB
GET CFG
GET JMPL MENU
C LABEL:  PROG
GET APPL
GET JMPL MENU
C LABEL:  INPUT
RVU IDB
GET JMPL MENU
C LABEL:  OUTPUT
RVU ODB
GET JMPL MENU
C LABEL:  EXE
EX APPL
GET JMPL MENU
C LABEL:  DONE
L
```

These commands would appear in a file EX4.PROC. In addition to the procedure file EX4.PROC, the menu file EX4.PROC_3 should be built to contain:

```
DATAB        1 CHANGE THE CURRENT CONFIGURATION DATABASE
PROG         2 CHANGE THE CURRENT APPLICATION PROGRAM
INPUT        3 REVIEW THE INPUT DATA
OUTPUT       4 REVIEW THE OUTPUT DATA
EXE          5 EXECUTE THE APPLICATION PROGRAM
DONE         6 EXIT THE PROCEDURE
```

Execution of the statement GET MENU 3 causes the following menu to be presented.

```
1 - CHANGE THE CURRENT CONFIGURATION DATABASE
2 - CHANGE THE CURRENT APPLICATION PROGRAM
3 - REVIEW THE INPUT DATA
4 - REVIEW THE OUTPUT DATA
5 - EXECUTE THE APPLICATION PROGRAM
6 - EXIT THE PROCEDURE
```

A choice of 4 would cause RVU ODB, the command following C LABEL: OUTPUT, to be the next statement executed.

6.5 Get Command: Extras

There are a few other small points to be made concerning the GET statement and the development of procedures.

The command GET LACT puts an activation time into the workspace log. It is recommended that procedures written for use in the automatic execution mode, ADE, begin with this command, as this will simplify the interpretation of command sequencing in the log.

The command GET MODE is used to insure that EASIE expects interactive input from the procedure user. In general, when a procedure is active, commands are being sent to the EASIE command processor from the procedure, and thus are not expecting feedback from a user. Therefore, with the exception of the GET command itself, other EASIE commands expecting input from a user should be preceded by a GET MODE Command.

Finally, as noted earlier in this section, the AUTODEFAULT should be set to true. For ADE users this should be preset in their EASIE login variable file. CCE users may set AUTODEFAULT with the ACTIVATE, EDITVAR, or SETLOGIN commands.

7.0 OF FILES, DATABASES, AND OTHER THINGS

An EASIE user file directory will contain a large variety of files. Though an explanation of each of these files follows, there would generally be little reason for a general user to become involved with any of the internal details or naming conventions used in these files. Such details can and generally should be left for the EASIE system to monitor, freeing the user to concentrate upon the design process itself. A summary of these files appears in figure 10 at the end of this section.

7.1 The Login Files

The login file is used as a tool to identify the appropriate EASIE system characteristics for a given user and analysis system. For instance, in the first sample session discussed in Section 3, the login identified the user as an ADE user and immediately began execution of procedure EXMENU. In the second sample session discussed in Section 5, the login identified the user as a CCE user and entered the command mode.

EASIE determines the login characteristics from the login ID. In the ADE example, the user entered an ID of EXMENU. EASIE takes this as a search key. An extension of ".VAR" is appended to the ID, here the result would be EXMENU.VAR. EASIE then searches through the user's file directory and its own file directory for that file. The user's file directory is searched first, since any such file would contain modifications specific for that user. If the file is not found in the user's file directory, EASIE's directory is searched for a generic copy of the login variable file. The reader is reminded that a CCE user can customize login variables via the SETLOGIN command. At that point, an appropriately modified copy of the login variable file is placed in the user's file directory.

7.1.1 The <user_id>.VAR File

<user_id>.VAR - holds values defined for the following list of variables. An "*" indicates the variables must be defined at login.

Variable		**Definition**
*USRID	-	The user's login identification.
USRLVL	-	The user's level of familarity with the EASIE system, range: 1..3 where 3=> experienced.
MENUCMD	-	Boolean, indicates if the command last issued was selected from a menu.
REFCFG	-	The name of the reference configuration database.
APPLTMPL	-	The name of a special data template for a given application program.
REFTMPL	-	The name of a user selected data template.
APPLTMPLIN APPLTMPLOUT	-	The names of input and output templates associated with the given application program.
REFPCF	-	The name of the reference procedure command file.
CURPROG	-	The name of the reference application program.
*WKSP	-	The name of the workspace. If no specific workspace is used a temporary workspace should be listed with the name T$.
*CURMENU	-	The name of the menu to be displayed, usually "UTILITY SELECTION (MAIN)" at login.
*CURMENUIDX	-	The index of the login menu, usually "1".
*INUNIT	-	The logical unit number from which input should be received, usually "5" for the VAX/VMS system.
*OUTUNIT	-	The logical unit number to which output would be directed, usually "6" for the VAX/VMS system.
*PRINTLEVL	-	Indicates the amount of extra help a user desires from EASIE, range: 1-4 where 1=> full menus, help, etc. and 4=> prompt only.

*HOMEUFD
*BASEUFD
*PROGUFD - The file directories which contain information pertinent to this project. HOME=> user, BASE=> EASIE specific information and PROG=> related programs and data. (Note: If required, the BASEUFD can be changed with the command ACT BUFD<filename>. Similarly ACT PUFD<filename> defines a new program directory).

*MENULIST - An encoded stack listing the values of the menus used, usually intialized to "!".

*SEQEXEC - Boolean, used to indicate when a sequence of programs is executing, usually initialized to "F".

*SEQNUM - The number of the program being executed, usually initialized to "0".

*PROCSP - Procedure stack pointer. Since procedures may call other procedures, a stack for tracking the executing procedures has been implemented, usually initialized to "0".

*PROCPC - Procedure counter, the point of current execution in a given procedure, usually initialized to "0".

*PROCEX - Boolean, used to indicate when a procedure is in execution, usually initialized to "F".

*EMPTYPROCSTACK - Boolean, used to indicate the existence of stacked procedures, usually initialized to "T".

*AUTODEFAULT - Boolean, used to indicate that EASIE commands given without specifying object names should default to values defined by the current workspace, usually initialized to "T". (Note: The autodefault mode can be set with the VAR command or by the command ACT DFLT.)

EXPROCFILE - The name of the currently executing procedure file.

7.1.2 Sample Login File for an ADE User

For this sample, the <user_id> is EXMENU and represents login file of the type required for the automatic execution mode, ADE.

```
USRID                     EXMENU
USRLVL                    1
MENUCMD                   F
REFCFG
APPLTMPL
REFTMPL
APPLTMPLIN
APPLTMPLOUT
REFPCF                    EXMENU
CURPROG
WKSP                      SAVDWS
CURMENU                   UTILITY SELECTION (MAIN)
CURMENUIDX                1
INUNIT                    5
OUTUNIT                   6
PRINTLEVL                 4
HOMEUFD                   VAB
BASEUFD                   TOAIDE:[EXMENU.CFG]
PROGUFD                   TOAIDE:[EXMENU.PROG]
MENULIST                  !
SEQEXEC                   F
SEQNUM                    0
PROCSP                    0
PROCPC                    1
PROCEX                    T
EMPTYPROCSTACK            T
AUTODEFAULT               T
EXPROCFILE                EXMENU.PROC
```

7.1.3 Sample Login File for a CCE User

For this sample, the <user_id> is EXAMPLE and represents the login file of the type required for complete control execute mode, CCE.

```
USRID                   EXAMPLE
USRLVL                  2
MENUCMD                 F
REFCFG                  NUDATA
APPLTMPL
REFTMPL
APPLTMPLIN              TOAIDE:[EXAMPLE.PROG]BOXIN
APPLTMPLOUT             TOAIDE:[EXAMPLE.PROG]BOXOUT
REFPCF
CURPROG
WKSP                    SAVDWS
CURMENU                 UTILITY SELECTION (MAIN)
CURMENUIDX              1
INUNIT                  5
OUTUNIT                 6
PRINTLEVL               2
HOMEUFD                 VAB
BASEUFD                 TOAIDE:[EXAMPLE.CFG]
PROGUFD                 TOAIDE:[EXAMPLE.PROG]
MENULIST                !
SEQEXEC                 F
SEQNUM                  0
PROCSP                  0
PROCPC                  0
PROCEX                  F
EMPTYPROCSTACK          T
AUTODEFAULT             T
EXPROCFILE
```

7.2 Workspace Files

An EASIE user would refer to a workspace only with its identifying name. Let <WS_name> represent that name, an example might be SAVDWS. The following types of files would always appear in the user's file directory for each workspace.

<WS_name>.WS - A file which has a structure identical with that of the <user_id>.VAR file previously discussed. This information is sufficient to represent the current state of a configuration development.

<WS_name>.WSD - A description file to which the user may add information concerning the specifics of project development found in this workspace. (See the RD and CD commands.)

<WS_name>.LOG - A file containing the sequence of commands issued by the user along with potential system and user comments.

Therefore, if a user had a workspace called SAVDWS, the files SAVDWS.WS, SAVDWS.WSD, and SAVDWS.LOG would appear in the user's file directory.

7.3 Configuration Database Files

An EASIE user would refer to a configuration database only by its identifying name. Let <cfg_name> represent that name, an example being NUDATA. The following types of files would appear in the user's file directory and are associated with a user configuration database. These types of files would appear in the project file directory when they are associated with master configuration databases. Master databases may be copied, but may not be modified or deleted. User databases may be copied, modified, or deleted, by the user or application programmer.

<cfg_name>.CFG - This represents a subdirectory. This subdirectory contains all database information, schema, and data for the given configuration.

<cfg_name>.CFGD - This is a description file to which the user may add information concerning the project associated data.

Therefore, if a user had a database called NUDATA, the files NUDATA.CFG and NUDATA.CFGD would appear in the user's file directory.

7.4 Procedure Files

An EASIE user would refer to a procedure only by its identifying name. Let <proc_name> represent that name, an example might be NEWPROC. The following types of files may appear in the user's file directory and are associated with user procedures. These types of files would also appear in the project file directory when they are associated with master procedures.

<proc_name>.PROC - A file which contains a collection of EASIE commands for processing by the EASIE command interpretor.

<proc_name>.PROCD - A description file that may contain a description of uses and parameters for the associated procedure.

<proc_name>.PROC_<n> - A menu file associated with this procedure. See Section 6.4 for a complete explanation.

Therefore, if a user had a procedure called NEWPROC, the files NEWPROC.PROC and NEWPROC.PROCD would appear in the user's file directory. Further, if any menus were associated with this procedure, they would appear in the user's file directory with names such as NEWPROC.PROC_1.

7.5 Template Files

An EASIE user would refer to a template only with its identifying name. Let <tpl_name> represent that name; an example might be BOXIN. The following types of files would appear in the user's file directory and are associated with a user template. These types of files would also appear in the project file directory when associated with master templates.

<tpl_name>.REV - This file contains configuration data locations in the database in a format appropriate for the REVIEWER program.

<tpl_name>.TPLD - This is a description file that the user may use to describe the type of data identified by this template.

Therefore, if a user had a template called BOXIN, the files BOXIN.REV and BOXIN.TPLD would appear in the user's file directory.

7.6 Application Program Files

An EASIE user would refer to an application program only by its identifying name. Let <appl_name> represent that name; an example might be BOX. The following files would generally appear in the main project file directory.

<appl_name>.APPLD - This is a description file detailing the use of the associated program.

<appl_name>.EXE
<appl_name>.COM - One or more of these will comprise the executable elements of the application program.

Therefore, if a user had access to a program called BOX, the file BOX.APPLD and either of the files BOX.EXE or BOX.COM would appear in the main project file directory.

7.7 Batch Files

Batch Files are created when time consuming application programs are submitted to the system's batch processors for execution, freeing the EASIE processor for further work. An EASIE user should refer to a batch file only by its identifying name. If BAT<n> represents that name, where <n> is some integer, an example might be BAT2. The following types of files may appear in the user's file directory and are associated with the batch job.

BAT<n>.COM - This file contains instructions for the batch processor.

BAT<n>.SAV - This is a log file containing information regarding the results of the batch execution.

Therefore, if a user had executed an application program in a batch mode which received the name BAT2, the files BAT2.COM and BAT2.SAV would appear in the users file directory. Note, the log file may be accessed through EASIE by the command TY BAT BAT2.

7.8 Temporary Files

The following files may appear during the processing of an EASIE session. They will be deleted at logout and should not be referred to or deleted by an EASIE user, as they keep track of valuable intermediate data.

File		Definition
T$.VAR	-	file which contains the current values of all status variables throughout an EASIE session.
T$.LOG	-	the current command log, if the user has not assigned or recalled a permanent workspace.
T$.PROC	-	file used to store a new procedure until it is saved and named by the user.
T$.APPL	-	file used to keep track of an execution sequence for a given application.
T$.PROCSTACK	-	file used to store the stack; formed as a procedure to call other procedures.
T$.TPL,T$,S$	-	files used during the interactive editing of a data template.

FILE TYPE	GENERIC NAME	FILENAME	USER FILE	MASTER FIL
LOGIN FILE	<USER_ID>	<USER_ID>.VAR	HOMEUFD	
WORKSPACE FILES	<WS_NAME>	<WS_NAME>.WS	HOMEUFD	
		<WS_NAME>.WSD	HOMEUFD	
	<WS_NAME>.LOG	HOMEUFD		
CONFIGURATION DATABASES	<CFG_NAME>	<CFG_NAME>.CFG	HOMEUFD	PROGUFD
		<CFG_NAME>.CFGD	HOMEUFD	PROGUFD
PROCEDURE FILES	<PROC_NAME>	<PROC_NAME>.PROC	HOMEUFD	PROGUFD
		<PROC_NAME>.PROCD	HOMEUFD	PROGUFD
		<PROC_NAME>.PROC_<n>	HOMEUFD	PROGUFD
TEMPLATE FILES	<TPL_NAME>	<TPL_NAME>.REV	HOMEUFD	
		<TPL_NAME>.TPLD	HOMEUFD	PROGUFD
APPLICATION PROGRAMS	<APPL_NAME>	<APPL_NAME>.EXE..		PROGUFD
		<APPL_NAME>.COM..		PROGUFD

Figure 10 - Table of EASIE Associated Files

8.0 COMMAND SUMMARY

The following sections summarize and collect the EASIE command information described previously in Section 4.0.

8.1 Permanent Menu Commands

The Permanent Menu commands provide the user with the basic information access and control to perform a design study.

H - Help

D - Directory: gives a listing of currently accessible files for configuration databases, data templates, and saved workspaces.

I - Inventory: gives a listing of currently accessible application programs and procedures.

S - System Command: Used to pass a command to the operating system.

C - Comment: used to place a comment in the command log.

T - Toggle the display mode rotating among:

1. Full information
2. Status variables and current utility menu
3. Status variables only
4. Prompt only.

R - Return to Previous Menu.

Q - Quit this Sequence of Menus and return to the utility selection menu.

L - Logout.

<CR> - Carriage Return: clears the screen and reprints menu information if active.

0 - Zero: cancels a command sequence.

<Break> - halts an execution sequence.

8.2 CCE Commands

The CCE mode of EASIE provides the flexibility for a user to perform a design study without the need of tracking a multitude of files, directories, and data.

Short Form	Full Form	Definition
ACT	ACTIVATE	Associate the indicated object with the user's workspace. Example: ACT CFG NUDATA Allowable Object Types: APPL, CFG, ITPL, OTPL, PROC, TPL, WS, UTL, BUFD, PUFD, and DFLT.
CD	CHANGE DESCRIPTION	Change description of the indicated object. Example: CD TPL USERINFO Allowable Objects Types: APPL, CFG, ITPL, OTPL, PROC, TPL, WS.
CL	CLEARLOG	Removes prior information from a cluttered command log. Example: CL D Allowable Object Types: D - prior to a given date T - total, a new log started
CN	CNAME	Change the name of a file as indicated. Example: CN TPL OLDTPL NEWTPL Allowable Object Types: APPL, CFG, PROC, TPL, WS, FILE.
CP	COPY	Copy one file to another. Example: CP CFG DEFAULT NUDATA Allowable Object Types: APPL, CFG, PROC, TPL, WS, FILE.
ED	EDIT	Invoke an editor. Example: ED PROC NEWPROC Allowable Object Types: LOG, PROC, TPL.
EX	EXECUTE	Executes an indicated program or procedure. Example: EX APPL BOX Allowable Object Types: APPL, PROC.

Short Form	Full Form	Definition
GET	GET	Allows interactive input during procedure execution. Example: GET PROC Allowable Object Types: APPL, CFG, JMPC, JMPL, LACT, MENU, MODE, PROC, QRYC, QRYL, TPL, WS.
N	NEW	Create a new object. Example: NEW WS Allowable Object Types: CFG, PROC, TPL, UTL, WS.
PR	PRINT	Print a file. Example: PR LOG T$ Allowable Object Types: LOG, PROC, FILE, BAT.
PRVU	PRINTVU	Print a view of the database. Example: PRVU IDB Allowable Object Types: IDB, ODB, TPL.
RD	READ DESCRIP-TION	Read a file description. Example: RD APPL BOX Allowable Object Types: APPL, CFG, ITPL, OTPL, PROC, TPL, WS.
RM	REMOVE	Remove a file from your directory. Example: RM CFG NUDATA Allowable Object Types: APPL, CFG, PROC, TPL, WS, FILE.
RVU	REVIEW	Review data from the configuration database. Example: RVU IDB Allowable Object Types: IDB, ODB, TPL.
SA	SAVE	Save the indicated object. Example: SA PROC NEWPROC Allowable Object Types: PROC, WS.
SLOG	SETLOGIN	Define your login characteristics. Example: SLOG No object type used in this command.
STAT	STATUS	Get the status of batch jobs. Example: STAT No object type used in this command.

Short Form	Full Form	Definition
SUB	SUBMIT	Submit a job for batch processing. Example: SUB APPL BOX Allowable Object Types: APPL.
TY	TYPE	Type the indicated file. Example: TY PROC NEWPROC Allowable Object Types: LOG, PROC, FILE, BAT.
VAR	EDITVAR	Change values of the current status variables in the workspace. Example: VAR No object types used in this command.

Appendix A - SCREENS FOR A SAMPLE ADE SESSION

```
      CS - SELECT A CONFIGURATION
      DC - DELETE A USER CONFIGURATION
      CD - EDIT A CONFIGURATION DESCRIPTION FILE
       R - REVIEW PROGRAM INPUT
       E - EXECUTE A PROGRAM
       P - PRINT OUTPUT FILES
       X - EXIT

Input: label - menu choice, <CR> - reprint menu
CS
```

SCREEN A1

```
     MASTER CONFIGURATIONS
DEFAULT

     USER CONFIGURATIONS

SHOULD A NEW CONFIGURATION BE CREATED (Y = yes):
Y

COPY SOURCE CONFIGURATION (FOR DEFAULT VALUES)
TO THE DESTINATION CONFIGURATION

ENTER SOURCE CONFIGURATION ("1" TO LIST):
DEFAULT

ENTER DESTINATION CONFIGURATION ("1" TO LIST):
NUDATA

COMMENCING VMS FILE COPY
COMMENCING DATABASE COPY
```

SCREEN A2

```
        CS - SELECT A CONFIGURATION
        DC - DELETE A USER CONFIGURATION
        CD - EDIT A CONFIGURATION DESCRIPTION FILE
         R - REVIEW PROGRAM INPUT
         E - EXECUTE A PROGRAM
         P - PRINT OUTPUT FILES
         X - EXIT

Input: label - menu choice, <CR> - reprint menu
CD

EDIT THE DESCRIPTION OF NUDATA
"EX" TO SAVE EDIT, "QUIT" TO QUIT EDIT - NO CHANGE
    1      DEFAULT CONFIGURATION FOR EXAMPLES IN EASIE DOCUMENTATION
*EX

DUB2:[EASIE]NUDATA.CFGD;2 2 lines


        CS - SELECT A CONFIGURATION
        DC - DELETE A USER CONFIGURATION
        CD - EDIT A CONFIGURATION DESCRIPTION FILE
         R - REVIEW PROGRAM INPUT
         E - EXECUTE A PROGRAM
         P - PRINT OUTPUT FILES
         X - EXIT

Input: label - menu choice, <CR> - reprint menu
R
```

SCREEN A3

```
        BI - REVIEW INPUT FOR BOX
        BO - REVIEW OUTPUT FROM BOX
         M - REVIEW INPUT FOR MAKGEO
         D - REVIEW INPUT FOR DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
BI
```

SCREEN A4

```
                        CATEGORY       1:    DIMEN
BOX DIMENSIONS

-----------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
-----------------------------------------------------------------------
  1 !      0.0000000 ! LENGTH    !              ! BOX LENGTH   ! M
  2 !      0.0000000 ! WIDTH     !              ! BOX WIDTH    ! M
  3 !      0.0000000 ! HEIGHT    !              ! BOX HEIGHT   ! M
-----------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
M 1

ENTER NEW VALUE:
10

>
M 2

ENTER NEW VALUE:
20

>
M 3

ENTER NEW VALUE:
30

>
R
```

SCREEN A5

```
                        CATEGORY      1:    DIMEN
BOX DIMENSIONS

------------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
------------------------------------------------------------------------
  1 !     10.0000000 ! LENGTH    !              ! BOX LENGTH   ! M
  2 !     20.0000000 ! WIDTH     !              ! BOX WIDTH    ! M
  3 !     30.0000000 ! HEIGHT    !              ! BOX HEIGHT   ! M
------------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
CAT
```

SCREEN A6

```
DESIGNATE THE CATEGORIES OF INTEREST
"*" INDICATES ALL CATEGORIES
"," IS DELIMITER BETWEEN CATEGORIES
":" INDICATES A RANGE OF CATEGORIES
*

     ID  CATEGORY  CATEGORY DESCRIPTION

      1  DIMEN     BOX DIMENSIONS

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
E
```

SCREEN A7

```
        BI - REVIEW INPUT FOR BOX
        BO - REVIEW OUTPUT FROM BOX
         M - REVIEW INPUT FOR MAKGEO
         D - REVIEW INPUT FOR DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
R
```

SCREEN A8

```
        CS - SELECT A CONFIGURATION
        DC - DELETE A USER CONFIGURATION
        CD - EDIT A CONFIGURATION DESCRIPTION FILE
         R - REVIEW PROGRAM INPUT
         E - EXECUTE A PROGRAM
         P - PRINT OUTPUT FILES
         X - EXIT

Input: label - menu choice, <CR> - reprint menu
E
```

SCREEN A9

```
         B - EXECUTE BOX
         M - EXECUTE MAKGEO
         D - EXECUTE DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
B

* STARTING EXECUTION OF BOX
 EXECUTING BOX
```

SCREEN A10

```
       B - EXECUTE BOX
       M - EXECUTE MAKGEO
       D - EXECUTE DRAWIT
       R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
R

      CS - SELECT A CONFIGURATION
      DC - DELETE A USER CONFIGURATION
      CD - EDIT A CONFIGURATION DESCRIPTION FILE
       R - REVIEW PROGRAM INPUT
       E - EXECUTE A PROGRAM
       P - PRINT OUTPUT FILES
       X - EXIT

Input: label - menu choice, <CR> - reprint menu
R

      BI - REVIEW INPUT FOR BOX
      BO - REVIEW OUTPUT FROM BOX
       M - REVIEW INPUT FOR MAKGEO
       D - REVIEW INPUT FOR DRAWIT
       R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
BO

                            CATEGORY      1:    DIMEN
BOX DIMENSIONS

-----------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME   !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
-----------------------------------------------------------------------
  1 !   6000.0000000 ! VOLUME   !              ! BOX VOLUME   ! M
-----------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
Q
```

SCREEN A11

```
      BI - REVIEW INPUT FOR BOX
      BO - REVIEW OUTPUT FROM BOX
       M - REVIEW INPUT FOR MAKGEO
       D - REVIEW INPUT FOR DRAWIT
       R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
R


      CS - SELECT A CONFIGURATION
      DC - DELETE A USER CONFIGURATION
      CD - EDIT A CONFIGURATION DESCRIPTION FILE
       R - REVIEW PROGRAM INPUT
       E - EXECUTE A PROGRAM
       P - PRINT OUTPUT FILES
       X - EXIT

Input: label - menu choice, <CR> - reprint menu
E


       B - EXECUTE BOX
       M - EXECUTE MAKGEO
       D - EXECUTE DRAWIT
       R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
M

* STARTING EXECUTION OF MAKGEO
 EXECUTING MAKGEO
```

SCREEN A12

```
        BO - REVIEW OUTPUT FROM BOX
         M - REVIEW INPUT FOR MAKGEO
         D - REVIEW INPUT FOR DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
D

                              CATEGORY       1:    MODEL
MODEL INFORMATION

-----------------------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME   ! SUBSCRIPT  ! DESCRIPTION       ! UNITS
-----------------------------------------------------------------------------------
  1 !                ! NAME     !            ! MODEL NAME        !
  2 !      0.0000000 ! ROTATION !      1     ! MODEL X,Y,Z ROTA  ! DEGREES
  3 !      0.0000000 !          !      2     !                   !
  4 !      0.0000000 !          !      3     !                   !
-----------------------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
CAT

DESIGNATE THE CATEGORIES OF INTEREST
"*" INDICATES ALL CATEGORIES
"," IS DELIMITER BETWEEN CATEGORIES
":" INDICATES A RANGE OF CATEGORIES
*

     ID  CATEGORY  CATEGORY DESCRIPTION

      1  MODEL     MODEL INFORMATION
      2  NODES     NODE POINT COORDINATES
      3  FACES     CONNECTIVITY OF NODES TO FORM FACES

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
R
```

SCREEN A13

```
                                CATEGORY       1:    MODEL
MODEL INFORMATION

-------------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    ! SUBSCRIPT  ! DESCRIPTION       ! UNITS
-------------------------------------------------------------------------
  1 !                ! NAME      !            ! MODEL NAME        !
  2 !      0.0000000 ! ROTATION  !          1 ! MODEL X,Y,Z ROTA  ! DEGREES
  3 !      0.0000000 !           !          2 !                   !
  4 !      0.0000000 !           !          3 !                   !
--------------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
M 1

ENTER NEW VALUE:
NUBOX

>
M 2

ENTER NEW VALUE:
25

>
M 3

ENTER NEW VALUE:
25

>
M 4

ENTER NEW VALUE:
25

>
R
```

SCREEN A14

```
                              CATEGORY       1:    MODEL
MODEL INFORMATION
------------------------------------------------------------------------------
 L  !  PRESENT VALUE  !   NAME    ! SUBSCRIPT ! DESCRIPTION       ! UNITS
------------------------------------------------------------------------------
  1 ! NUBOX           ! NAME      !           ! MODEL NAME        !
  2 !      25.0000000 ! ROTATION  !         1 ! MODEL X,Y,Z ROTA  ! DEGREES
  3 !      25.0000000 !           !         2 !                   !
  4 !      25.0000000 !           !         3 !                   !
------------------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
C 2

OK TO REPLACE CHANGES? (Y) >
Y
```

SCREEN A15

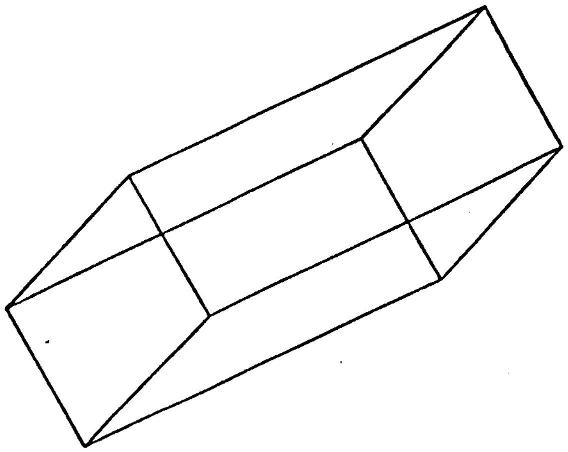
```
                              CATEGORY       2:    NODES
NODE POINT COORDINATES
-------------------------------------------
NAME   !  X         !  Y         !  Z
-------------------------------------------
COL    !          1 !          2 !          3
-------------------------------------------
ROW :
     1!   -5.0000 !  -15.0000 !  -10.0000
     2!    5.0000 !  -15.0000 !  -10.0000
     3!    5.0000 !   15.0000 !  -10.0000
     4!   -5.0000 !   15.0000 !  -10.0000
     5!   -5.0000 !  -15.0000 !   10.0000
     6!    5.0000 !  -15.0000 !   10.0000
     7!    5.0000 !   15.0000 !   10.0000
     8!   -5.0000 !   15.0000 !   10.0000
-------------------------------------------

M r c : modify value (r = row# or range; c = column#,name(subscript), or range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n rows per page, X r c : expand row# n, column# c
S : set columns to be displayed
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
C 3
```

SCREEN A16

```
                             CATEGORY        3:    FACES
CONNECTIVITY OF NODES TO FORM FACES
---------------------------------------------------------
NAME    !  FACE      !  FACE      !  FACE      !  FACE
INDEX   !          1 !          2 !          3 !          4
---------------------------------------------------------
COL     !        1   !        2   !        3   !        4
---------------------------------------------------------
ROW :
      1!           1 !          2 !          3 !          4
      2!           2 !          6 !          7 !          3
      3!           6 !          5 !          8 !          7
      4!           5 !          1 !          4 !          8
      5!           4 !          3 !          7 !          8
      6!           5 !          6 !          2 !          1
---------------------------------------------------------

M r c : modify value (r = row# or range; c = column#,name(subscript), or range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n rows per page, X r c : expand row# n, column# c
S : set columns to be displayed
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
Q
```

SCREEN A17

```
        BI - REVIEW INPUT FOR BOX
        BO - REVIEW OUTPUT FROM BOX
         M - REVIEW INPUT FOR MAKGEO
         D - REVIEW INPUT FOR DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
R

        CS - SELECT A CONFIGURATION
        DC - DELETE A USER CONFIGURATION
        CD - EDIT A CONFIGURATION DESCRIPTION FILE
         R - REVIEW PROGRAM INPUT
         E - EXECUTE A PROGRAM
         P - PRINT OUTPUT FILES
         X - EXIT

Input: label - menu choice, <CR> - reprint menu
E

         B - EXECUTE BOX
         M - EXECUTE MAKGEO
         D - EXECUTE DRAWIT
         R - RETURN TO MAIN MENU

Input: label - menu choice, <CR> - reprint menu
D

* STARTING EXECUTION OF DRAWIT
 EXECUTING DRAWIN
 EXECUTING DRAW
```



SCREEN A18

```
Input: label - menu choice, <CR> - reprint menu
X

SAVE CURRENT WORKSPACE STATUS (Y=Yes):
N

* WORKSPACE STATUS DELETED AT LOGOUT
* LOGOUT - BYE *


$ TYPE EXMENU.PROC
C LABEL: MM
GET MENU 1
C LABEL: R
GET MENU 2
C LABEL: E
GET MENU 3
C LABEL: CS
GET CFG
GET JMPL MM
C LABEL: DC
RM CFG
GET JMPL MM
C LABEL: CD
CD CFG -
GET JMPL MM
C LABEL: BIR
RVU BOXIN
GET JMPL R
C LABEL: BOR
RVU BOXOUT
GET JMPL R
C LABEL: MR
RVU MAKGEOIN
GET JMPL R
C LABEL: DR
RVU DRAWIN
GET JMPL R
C LABEL: BX
EX APPL BOX
GET JMPL E
C LABEL: MX
EX APPL MAKGEO
GET JMPL E
C LABEL: DX
EX APPL DRAWIT
GET JMPL E
C LABEL: X
L
N
$
```

SCREEN A19

```
$ TYPE EXMENU.PROC 1
CS       CS       SELECT A CONFIGURATION
DC       DC       DELETE A USER CONFIGURATION
CD       CD       EDIT A CONFIGURATION DESCRIPTION FILE
R        R        REVIEW PROGRAM INPUT
E        E        EXECUTE A PROGRAM
P        P        PRINT OUTPUT FILES
X        X        EXIT
$


$ TYPE EXMENU.PROC 2
BIR      BI       REVIEW INPUT FOR BOX
BOR      BO       REVIEW OUTPUT FROM BOX
MR       M        REVIEW INPUT FOR MAKGEO
DR       D        REVIEW INPUT FOR DRAWIT
MM       R        RETURN TO MAIN MENU
$


$ TYPE EXMENU.PROC 3
BX       B        EXECUTE BOX
MX       M        EXECUTE MAKGEO
DX       D        EXECUTE DRAWIT
MM       R        RETURN TO MAIN MENU
$ LO
```

SCREEN A20

Appendix B - SCREENS FOR A SAMPLE CCE SESSION

```
        STATUS:
                                      WORKSPACE:    T$            REF_CONFIG:
UTIL_IN_USE:  UTILITY SELECTI
                                      APPLIC_PROG:                REF_TEMPLATE:
REF_PROCFILE:

          PERMANENT MENU:

          H - HELP
          D - DIRECTORY of CFG's, TPL's and WS's
          I - INVENTORY of APPL's and PROC's
          S - SYSTEM COMMANDS for PRIMOS
          C - Add a COMMENT to the command log
          T - TOGGLE the MENU PRINT setting
          R - RETURN to the PREVIOUS MENU
          Q - QUIT this sequence of menus and
              RETURN to the MAIN MENU
          L - LOGOUT

          <CR> - Clear the screen and relist the menu

          UTILITY SELECTION (MAIN)

                                           COMMAND FORMAT
          1 - WORKSPACE CONTROL UTILITY            ACT UTL  <WSC >
          2 - DATA MODIFICATION UTILITY            ACT UTL  <DATA>
          3 - APPLICATION EXECUTION UTILITY        ACT UTL  <APEX>
          4 - PROCEDURE EXECUTION UTILITY          ACT UTL  <PREX>
          5 - PROCEDURE BUILDING UTILITY           ACT UTL  <PBLD>
          6 - TEMPLATE BUILDING UTILITY            ACT UTL  <TBLD>

        ENTER COMMAND:
```

SCREEN B1

```
          UTILITY SELECTION (MAIN)

                                           COMMAND FORMAT
          1 - WORKSPACE CONTROL UTILITY            ACT UTL  <WSC >
          2 - DATA MODIFICATION UTILITY            ACT UTL  <DATA>
          3 - APPLICATION EXECUTION UTILITY        ACT UTL  <APEX>
          4 - PROCEDURE EXECUTION UTILITY          ACT UTL  <PREX>
          5 - PROCEDURE BUILDING UTILITY           ACT UTL  <PBLD>
          6 - TEMPLATE BUILDING UTILITY            ACT UTL  <TBLD>

        ENTER COMMAND:
        1
```

SCREEN B2

```
STATUS:
WORKSPACE:    T$        REF_CONFIG:         UTIL_IN_USE:   WORKSPACE CONTR
APPLIC_PROG:            REF_TEMPLATE:       REF_PROCFILE:

 PERMANENT MENU:
 H - HELP
 D - DIRECTORY of CFG's, TPL's and WS's
 I - INVENTORY of APPL's and PROC's
 S - SYSTEM COMMANDS for PRIMOS
 C - Add a COMMENT to the command log
 T - TOGGLE the MENU PRINT setting
 R - RETURN to the PREVIOUS MENU
 Q - QUIT this sequence of menus and
     RETURN to the MAIN MENU
 L - LOGOUT

 <CR> - Clear the screen and relist the menu

 WORKSPACE CONTROL
                                     COMMAND FORMAT
 1 - READ DESCRIPTION   - WORKSPACE     RD WS     <name>
 2 -                    - CONFIGURATION RD CFG    <name>
 3 -                    - TEMPLATE      RD TPL    <name>
 4 -                    - APPL. PROG.   RD APPL   <name>
 5 -                    - PROCEDURE     RD PROC   <name>
 6 - CLEAR LOG OF OLD INFORMATION       CL
 7 - TYPE               - COMMAND LOG   TY LOG    <name>
 8 -                    - PROCEDURE     TY PROC   <name>
 9 - NEW                - WORKSPACE     N WS
10 -                    - CONFIGURATION N CFG     <base>
11 - COPY               - WORKSPACE     CP WS     <f,to>
12 -                    - PROCEDURE     CP PROC   <f,to>
13 - ACTIVATE           - WORKSPACE     ACT WS    <name>
14 -                    - CONFIGURATION ACT CFG   <name>
15 -                    - TEMPLATE      ACT TPL   <name>
16 -                    - APPL. PROG.   ACT APPL  <name>
17 -                    - UTILITY       ACT UTL   <menu>
18 -                    - INPUT TEMPL   ACT ITPL
19 -                    - OUTPUT TEMPL  ACT OTPL
20 -                    - PROCEDURE     ACT PROC  <name>
21 -                    - PROGRAM UFD   ACT PUFD  <path>
22 - SAVE TEMPORARY     - WORKSPACE     SA WS     <name>
23 -                    - PROCEDURE     SA PROC   <name>
24 - REMOVE FROM UFD    - WORKSPACE     RM WS     <name>
25 -                    - CONFIGURATION RM CFG    <name>
26 -                    - TEMPLATE      RM TPL    <name>
27 -                    - PROCEDURE     RM PROC   <name>
28 - SET USER LOGIN CHARACTERISTICS     SLOG

ENTER COMMAND:
T

 NEW PRINT MODE: CURRENT MENU ONLY
```

SCREEN B3

```
STATUS:

WORKSPACE:    T$                REF_CONFIG:
UTIL_IN_USE: WORKSPACE CONTR APPLIC_PROG:
REF_TEMPLATE:                   REF_PROCFILE:

  WORKSPACE CONTROL

                                          COMMAND FORMAT
  1 - READ DESCRIPTION    - WORKSPACE     RD WS    <name>
  2 -                     - CONFIGURATION RD CFG   <name>
  3 -                     - TEMPLATE      RD TPL   <name>
  4 -                     - APPL. PROG.   RD APPL  <name>
  5 -                     - PROCEDURE     RD PROC  <name>
  6 - CLEAR LOG OF OLD INFORMATION        CL
  7 - TYPE                - COMMAND LOG   TY LOG   <name>
  8 -                     - PROCEDURE     TY PROC  <name>
  9 - NEW                 - WORKSPACE     N WS
 10 -                     - CONFIGURATION N CFG    <base>
 11 - COPY                - WORKSPACE     CP WS    <f,to>
 12 -                     - PROCEDURE     CP PROC  <f,to>
 13 - ACTIVATE            - WORKSPACE     ACT WS   <name>
 14 -                     - CONFIGURATION ACT CFG  <name>
 15 -                     - TEMPLATE      ACT TPL  <name>
 16 -                     - APPL. PROG.   ACT APPL <name>
 17 -                     - UTILITY       ACT UTL  <menu>
 18 -                     - INPUT TEMPL   ACT ITPL
 19 -                     - OUTPUT TEMPL  ACT OTPL
 20 -                     - PROCEDURE     ACT PROC <name>
 21 -                     - PROGRAM UFD   ACT PUFD <path>
 22 - SAVE TEMPORARY      - WORKSPACE     SA WS    <name>
 23 -                     - PROCEDURE     SA PROC  <name>
 24 - REMOVE FROM UFD     - WORKSPACE     RM WS    <name>
 25 -                     - CONFIGURATION RM CFG   <name>
 26 -                     - TEMPLATE      RM TPL   <name>
 27 -                     - PROCEDURE     RM PROC  <name>
 28 - SET USER LOGIN CHARACTERISTICS      SLOG

ENTER COMMAND:
```

SCREEN B4

```
ENTER COMMAND:
D
```

SCREEN B5

```
    MASTER CONFIGURATIONS
DEFAULT

     MASTER TEMPLATES
BOXIN
BOXOUT
DRAWIN
MAKGEOIN

     USER CONFIGURATIONS
     USER WORKSPACES
T$RTA4
     USER TEMPLATES
```

SCREEN B6

```
ENTER COMMAND:
I

    MASTER APPLICATION PROGRAMS
BOX
DRAWIT
MAKGEO

     USER APPLICATION PROGRAMS
     USER PROCEDURES


ENTER COMMAND:
D
```

SCREEN B7

```
 STATUS:

WORKSPACE:   T$              REF_CONFIG:
UTIL_IN_USE: WORKSPACE CONTR APPLIC_PROG:
REF_TEMPLATE:                REF_PROCFILE:

  WORKSPACE CONTROL

                                   COMMAND FORMAT
  1 - READ DESCRIPTION   - WORKSPACE      RD WS    <name>
  2 -                    - CONFIGURATION  RD CFG   <name>
  3 -                    - TEMPLATE       RD TPL   <name>
  4 -                    - APPL. PROG.    RD APPL  <name>
  5 -                    - PROCEDURE      RD PROC  <name>
  6 - CLEAR LOG OF OLD INFORMATION        CL
  7 - TYPE               - COMMAND LOG    TY LOG   <name>
  8 -                    - PROCEDURE      TY PROC  <name>
  9 - NEW                - WORKSPACE      N WS
 10 -                    - CONFIGURATION  N CFG    <base>
 11 - COPY               - WORKSPACE      CP WS    <f,to>
 12 -                    - PROCEDURE      CP PROC  <f,to>
 13 - ACTIVATE           - WORKSPACE      ACT WS   <name>
 14 -                    - CONFIGURATION  ACT CFG  <name>
 15 -                    - TEMPLATE       ACT TPL  <name>
 16 -                    - APPL. PROG.    ACT APPL <name>
 17 -                    - UTILITY        ACT UTL  <menu>
 18 -                    - INPUT TEMPL    ACT ITPL
 19 -                    - OUTPUT TEMPL   ACT OTPL
 20 -                    - PROCEDURE      ACT PROC <name>
 21 -                    - PROGRAM UFD    ACT PUFD <path>
 22 - SAVE TEMPORARY     - WORKSPACE      SA WS    <name>
 23 -                    - PROCEDURE      SA PROC  <name>
 24 - REMOVE FROM UFD    - WORKSPACE      RM WS    <name>
 25 -                    - CONFIGURATION  RM CFG   <name>
 26 -                    - TEMPLATE       RM TPL   <name>
 27 -                    - PROCEDURE      RM PROC  <name>
 28 - SET USER LOGIN CHARACTERISTICS      SLOG

ENTER COMMAND:
10

 COPY SOURCE CONFIGURATION (FOR DEFAULT VALUES)
TO THE DESTINATION CONFIGURATION

ENTER SOURCE CONFIGURATION ("1" TO LIST):
DEFAULT

 ENTER DESTINATION CONFIGURATION ("1" TO LIST):
NUDATA

COMMENCING VMS FILE COPY
COMMENCING DATABASE COPY
```

SCREEN B8

```
 STATUS:


WORKSPACE:   T$                REF_CONFIG:   NUDATA
UTIL_IN_USE: WORKSPACE CONTR APPLIC_PROG:
REF_TEMPLATE:                REF_PROCFILE:



  WORKSPACE CONTROL


                                     COMMAND FORMAT
  1 - READ DESCRIPTION    - WORKSPACE      RD WS     <name>
  2 -                     - CONFIGURATION  RD CFG    <name>
  3 -                     - TEMPLATE       RD TPL    <name>
  4 -                     - APPL. PROG.    RD APPL   <name>
  5 -                     - PROCEDURE      RD PROC   <name>
  6 - CLEAR LOG OF OLD INFORMATION         CL
  7 - TYPE                - COMMAND LOG    TY LOG    <name>
  8 -                     - PROCEDURE      TY PROC   <name>
  9 - NEW                 - WORKSPACE      N WS
 10 -                     - CONFIGURATION  N CFG     <base>
 11 - COPY                - WORKSPACE      CP WS     <f,to>
 12 -                     - PROCEDURE      CP PROC   <f,to>
 13 - ACTIVATE            - WORKSPACE      ACT WS    <name>
 14 -                     - CONFIGURATION  ACT CFG   <name>
 15 -                     - TEMPLATE       ACT TPL   <name>
 16 -                     - APPL. PROG.    ACT APPL  <name>
 17 -                     - UTILITY        ACT UTL   <menu>
 18 -                     - INPUT TEMPL    ACT ITPL
 19 -                     - OUTPUT TEMPL   ACT OTPL
 20 -                     - PROCEDURE      ACT PROC  <name>
 21 -                     - PROGRAM UFD    ACT PUFD  <path>
 22 - SAVE TEMPORARY      - WORKSPACE      SA WS     <name>
 23 -                     - PROCEDURE      SA PROC   <name>
 24 - REMOVE FROM UFD     - WORKSPACE      RM WS     <name>
 25 -                     - CONFIGURATION  RM CFG    <name>
 26 -                     - TEMPLATE       RM TPL    <name>
 27 -                     - PROCEDURE      RM PROC   <name>
 28 - SET USER LOGIN CHARACTERISTICS       SLOG


ENTER COMMAND:
16

* NO DEFAULT ALLOWED FOR ACTIVATE APPLICATION **
ENTER APPLICATION PROGRAM NAME ("1" TO LIST)
BOX
```

SCREEN B9

```
STATUS

WORKSPACE:    T$                  REF_CONFIG:   NUDATA
UTIL_IN_USE: WORKSPACE CONTR APPLIC_PROG:  BOX
REF_TEMPLATE:                     REF_PROCFILE:


  WORKSPACE CONTROL


                                          COMMAND FORMAT
  1 - READ DESCRIPTION    - WORKSPACE       RD WS     <name>
  2 -                     - CONFIGURATION   RD CFG    <name>
  3 -                     - TEMPLATE        RD TPL    <name>
  4 -                     - APPL. PROG.     RD APPL   <name>
  5 -                     - PROCEDURE       RD PROC   <name>
  6 - CLEAR LOG OF OLD INFORMATION          CL
  7 - TYPE                - COMMAND LOG     TY LOG    <name>
  8 -                     - PROCEDURE       TY PROC   <name>
  9 - NEW                 - WORKSPACE       N WS
 10 -                     - CONFIGURATION   N CFG     <base>
 11 - COPY                - WORKSPACE       CP WS     <f,to>
 12 -                     - PROCEDURE       CP PROC   <f,to>
 13 - ACTIVATE            - WORKSPACE       ACT WS    <name>
 14 -                     - CONFIGURATION   ACT CFG   <name>
 15 -                     - TEMPLATE        ACT TPL   <name>
 16 -                     - APPL. PROG.     ACT APPL  <name>
 17 -                     - UTILITY         ACT UTL   <menu>
 18 -                     - INPUT TEMPL     ACT ITPL
 19 -                     - OUTPUT TEMPL    ACT OTPL
 20 -                     - PROCEDURE       ACT PROC  <name>
 21 -                     - PROGRAM UFD     ACT PUFD  <path>
 22 - SAVE TEMPORARY      - WORKSPACE       SA WS     <name>
 23 -                     - PROCEDURE       SA PROC   <name>
 24 - REMOVE FROM UFD     - WORKSPACE       RM WS     <name>
 25 -                     - CONFIGURATION   RM CFG    <name>
 26 -                     - TEMPLATE        RM TPL    <name>
 27 -                     - PROCEDURE       RM PROC   <name>
 28 - SET USER LOGIN CHARACTERISTICS        SLOG

 ENTER COMMAND:
RVU IDB
```

SCREEN B10

```
EDIT:
>
CAT

DESIGNATE THE CATEGORIES OF INTEREST
"*" INDICATES ALL CATEGORIES
"," IS DELIMITER BETWEEN CATEGORIES
":" INDICATES A RANGE OF CATEGORIES
*

    ID  CATEGORY  CATEGORY DESCRIPTION

       1  DIMEN     BOX DIMENSIONS

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>R
```

SCREEN B11

```
                              CATEGORY       1:    DIMEN
BOX DIMENSIONS
---------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT   ! DESCRIPTION ! UNITS
---------------------------------------------------------------------
  1 !      0.0000000 ! LENGTH    !              ! BOX LENGTH  ! M
  2 !      0.0000000 ! WIDTH     !              ! BOX WIDTH   ! M
  3 !      0.0000000 ! HEIGHT    !              ! BOX HEIGHT  ! M
---------------------------------------------------------------------
M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>M 1

ENTER NEW VALUE:
10

>M 2

ENTER NEW VALUE:
20

>M 3

ENTER NEW VALUE:
30
>R
```

SCREEN B12

```
                          CATEGORY        1:     DIMEN
BOX DIMENSIONS

---------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT    ! DESCRIPTION  ! UNITS
---------------------------------------------------------------------
  1 !      10.0000000 ! LENGTH    !               ! BOX LENGTH   ! M
  2 !      20.0000000 ! WIDTH     !               ! BOX WIDTH    ! M
  3 !      30.0000000 ! HEIGHT    !               ! BOX HEIGHT   ! M
---------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
E


ENTER COMMAND:
Q


 STATUS:


WORKSPACE:    T$                REF_CONFIG:    NUDATA
UTIL_IN_USE: UTILITY SELECTI APPLIC_PROG:   BOX
REF_TEMPLATE:                   REF_PROCFILE:

  UTILITY SELECTION (MAIN)

                                         COMMAND FORMAT
  1 - WORKSPACE CONTROL UTILITY              ACT UTL   <WSC >
  2 - DATA MODIFICATION UTILITY              ACT UTL   <DATA>
  3 - APPLICATION EXECUTION UTILITY          ACT UTL   <APEX>
  4 - PROCEDURE EXECUTION UTILITY            ACT UTL   <PREX>
  5 - PROCEDURE BUILDING UTILITY             ACT UTL   <PBLD>
  6 - TEMPLATE BUILDING UTILITY                ACT UTL   <TBLD>

ENTER COMMAND:
3
```

SCREEN B13

```
 STATUS:

WORKSPACE:   T$              REF_CONFIG:   NUDATA
UTIL_IN_USE: APPLICATION EXE APPLIC_PROG:  BOX
REF_TEMPLATE:                REF_PROCFILE:

  APPLICATION EXECUTION

                                   COMMAND FORMAT
  1 - READ DISCRIPTION    - APPL. PROG.     RD APPL  <name>
  2 -                     - CONFIGURATION   RD CFG   <name>
  3 -                      - TEMPLATE       RD TPL   <name>
  4 - NEW                 - CONFIGURATION   N CFG    <base>
  5 - ACTIVATE            - APPL. PROG.     ACT APPL <name>
  6 -                     - CONFIGURATION   ACT CFG  <name>
  7 - ACTIVATE DB VIEW AS- INPUT TEMPL.     ACT ITPL
  8 -                     - OUTPUT TEMPL.   ACT OTPL
  9 -                     - ALT. TEMPL.     ACT TPL  <name>
 10 - COPY                - CONFIGURATION   CP CFG   <f,to>
 11 - REVIEW DB WITH ACTIVE TEMPLATE        RVU
 12 - REVIEW              - INPUT DB        RVU IDB
 13 -                     - OUTPUT DB       RVU ODB
 14 - PRINT DB WITH ACTIVE TEMPLATE         PRVU
 15 - PRINT               - INPUT DB        PRVU IDB
 16 -                     - OUTPTU DB       PRVU ODB
 17 - EXECUTE CURRENT APPLICATION PROGRAM   EX APPL  <-   >
 18 - CHANGE DESCRIPTION - CONFIGURATION    CD CFG   <name>
 19 - REMOVE ALL DATA     - INPUT & OUTPUT  RM CFG
 20 - TYPE CURRENT COMMAND LOG              TY LOG   <-   >

ENTER COMMAND:
17

 STARTING EXECUTION OF BOX
 EXECUTING BOX

ENTER COMMAND:
RVU ODB
```

SCREEN B14

```
                             CATEGORY        1:    DIMEN
BOX DIMENSIONS
-------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
-------------------------------------------------------------------
  1 !   6000.0000000 ! VOLUME    !              ! BOX VOLUME   ! M
-------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
Q

 STATUS:

WORKSPACE:   T$                  REF_CONFIG:   NUDATA
UTIL_IN_USE: APPLICATION EXE APPLIC_PROG:  BOX
REF_TEMPLATE:                    REF_PROCFILE:

  APPLICATION EXECUTION

                                     COMMAND FORMAT
  1 - READ DISCRIPTION   - APPL. PROG.      RD APPL   <name>
  2 -                    - CONFIGURATION    RD CFG    <name>
  3 -                     - TEMPLATE        RD TPL    <name>
  4 - NEW                - CONFIGURATION    N CFG     <base>
  5 - ACTIVATE           - APPL. PROG.      ACT APPL  <name>
  6 -                    - CONFIGURATION    ACT CFG   <name>
  7 - ACTIVATE DB VIEW AS- INPUT TEMPL.     ACT ITPL
  8 -                    - OUTPUT TEMPL.    ACT OTPL
  9 -                    - ALT. TEMPL.      ACT TPL   <name>
 10 - COPY               - CONFIGURATION    CP CFG    <f,to>
 11 - REVIEW DB WITH ACTIVE TEMPLATE        RVU
 12 - REVIEW             - INPUT DB         RVU IDB
 13 -                    - OUTPUT DB        RVU ODB
 14 - PRINT DB WITH ACTIVE TEMPLATE         PRVU
 15 - PRINT              - INPUT DB         PRVU IDB
 16 -                    - OUTPTU DB        PRVU ODB
 17 - EXECUTE CURRENT APPLICATION PROGRAM   EX APPL   <-   >
 18 - CHANGE DESCRIPTION - CONFIGURATION    CD CFG    <name>
 19 - REMOVE ALL DATA    - INPUT & OUTPUT   RM CFG
 20 - TYPE CURRENT COMMAND LOG              TY LOG    <-   >

ENTER COMMAND:
20
```

SCREEN B15

```
 WS ACTIVATED ON :  4-DEC-87
>ACT UTL  WSC
>T
* TOGGLE TTY PRINT MODE
>D
* DIRECTORY
>I
* INVENTORY
>CP CFG DEFAULT NUDATA
* COPY FROM:TOAIDE:[EXAMPLE.CFG.DEFAULT]          TO:NUDATA
* CARRIED OUT COMMAND: N CFG
>ACT APPL BOX
>RVU IDB
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  APEX
>EX APPL BOX
*    EXECUTING BOX
>RVU ODB
>TY LOG   T$RTA4

ENTER COMMAND:
Q

 STATUS:


WORKSPACE:   T$              REF_CONFIG:   NUDATA
UTIL_IN_USE: UTILITY SELECTI APPLIC_PROG:  BOX
REF_TEMPLATE:                REF_PROCFILE:



  UTILITY SELECTION (MAIN)


                                     COMMAND FORMAT
  1 - WORKSPACE CONTROL UTILITY              ACT UTL  <WSC >
  2 - DATA MODIFICATION UTILITY              ACT UTL  <DATA>
  3 - APPLICATION EXECUTION UTILITY          ACT UTL  <APEX>
  4 - PROCEDURE EXECUTION UTILITY            ACT UTL  <PREX>
  5 - PROCEDURE BUILDING UTILITY             ACT UTL  <PBLD>
  6 - TEMPLATE BUILDING UTILITY              ACT UTL  <TBLD>

ENTER COMMAND:
5
```

SCREEN B16

```
 STATUS:

WORKSPACE:   T$                REF_CONFIG:   NUDATA
UTIL_IN_USE: PROCEDURE BUILD APPLIC_PROG:  BOX
REF_TEMPLATE:                  REF_PROCFILE:

  PROCEDURE BUILDING

                                      COMMAND FORMAT
  1 - READ DESCRIPTION    - PROCEDURE       RD PROC   <name>
  2 - TYPE                - COMMAND LOG     TY LOG    <name>
  3 -                     - PROCEDURE       TY PROC   <name>
  4 - CLEAR LOG OF OLD INFORMATION          CL
  5 - ACTIVATE            - PROCEDURE       ACT PROC <name>
  6 - COPY                - PROCEDURE       CP PROC   <f,to>
  7 - DEFINE A NEW PROCEDURE                N PROC
  8 - EDIT AN EXISTING PROCEDURE            ED PROC   <name>
  9 - EDIT A LOG TO BUILD A PROCEDURE       ED LOG    <name>
 10 - SAVE A TEMPORARY PROCEDURE            SA PROC   <new >
 11 - CHANGE THE PROCEDURE DESCRIPTION      CD PROC
 12 - REMOVE THE PROCEDURE                  RM PROC
 13 - TYPE THE CURRENT COMMAND LOG          TY LOG    <-   >

ENTER COMMAND:
9

EDIT T$RTA4.PROC
Input file does not have standard text file format
```

SCREEN B17

```
*1:55

    1       ACT UTL  WSC
    2       T
    3       D
    4       I
    5       CP CFG DEFAULT NUDATA
    6       ACT APPL BOX
    7       RVU IDB
    8       Q
    9       ACT UTL  APEX
   10       EX APPL BOX
   11       RVU ODB
   12       TY LOG   T$RTA4
   13       Q
   14       ACT UTL  PBLD
   15       ED LOG T$RTA4

*DEL 1:6

 lines deleted
    7       RVU IDB

*DEL 8:9

 lines deleted
   10       EX APPL BOX

*DEL 12:15

 lines deleted
[EOB]
*1:15

    7       RVU IDB
   10       EX APPL BOX
   11       RVU ODB

*RES
 lines resequenced
*1:15

    1       RVU IDB
    2       EX APPL BOX
    3       RVU ODB

*EXIT

DUB2:[EASIE]T$RTA4.PROC;2 3 lines
```

SCREEN B18

```
ENTER COMMAND:
TY LOG

 WS ACTIVATED ON :  4-DEC-87
>ACT UTL  WSC
>T
* TOGGLE TTY PRINT MODE
>D
* DIRECTORY
>I
* INVENTORY
>CP CFG DEFAULT NUDATA
* COPY FROM:TOAIDE:[EXAMPLE.CFG.DEFAULT]          TO:NUDATA
* CARRIED OUT COMMAND: N CFG
>ACT APPL BOX
>RVU IDB
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  APEX
>EX APPL BOX
*    EXECUTING BOX
>RVU ODB
>TY LOG   T$RTA4
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  PBLD
>ED LOG T$RTA4
>TY LOG T$RTA4

             [See Screen B17 - PROCEDURE BUILDING MENU]

ENTER COMMAND:
10

ENTER PROCEDURE NAME:
NEWPROC

ENTER COMMAND:
L

SAVE CURRENT WORKSPACE STATUS (Y=Yes):
Y

WORKSPACE HAS ONLY TEMP NAME
INPUT WORKSPACE NAME:
SAVDWS

 STATUS:

WORKSPACE:    SAVDWS            REF_CONFIG:   NUDATA
UTIL_IN_USE: PROCEDURE BUILD APPLIC_PROG:  BOX
REF_TEMPLATE:                   REF_PROCFILE: NEWPROC

 LOGOUT - BYE *
```

SCREEN B19

```
$ EASIE
ENTER TERMINAL TYPE:
 1 FOR TEKTRONIX 4014
 2 FOR TEKTRONIX 4105
 3 FOR TEKTRONIX 4107
 4 FOR TEKTRONIX 4109
 5 FOR TEKTRONIX 4115
 6 FOR TEKTRONIX 4129
 7 FOR AED767
OR <CR> FOR ALPHANUMERIC TERMINAL

 ENTER USER ID :
EXAMPLE
```

[See Screen B1 - PERMANENT MENU]

```
ENTER COMMAND
I

    MASTER APPLICATION PROGRAMS
BOX
DRAWIT
MAKGEO

    USER APPLICATION PROGRAMS
    USER PROCEDURES
NEWPROC
```

SCREEN B20

```
ENTER COMMAND:
D

    MASTER CONFIGURATIONS
DEFAULT

    MASTER TEMPLATES
BOXIN
BOXOUT
DRAWIN
MAKGEOIN

    USER CONFIGURATIONS
NUDATA
    USER WORKSPACES
SAVDWS
T$RTA4
    USER TEMPLATES
```

SCREEN B21

```
 STATUS:

WORKSPACE:    T$                REF_CONFIG:
UTIL_IN_USE: WORKSPACE CONTR APPLIC_PROG:
REF_TEMPLATE:                   REF_PROCFILE:

  WORKSPACE CONTROL

                                         COMMAND FORMAT
  1 - READ DESCRIPTION    - WORKSPACE        RD WS     <name>
  2 -                     - CONFIGURATION    RD CFG    <name>
  3 -                     - TEMPLATE         RD TPL    <name>
  4 -                     - APPL. PROG.      RD APPL   <name>
  5 -                     - PROCEDURE        RD PROC   <name>
  6 - CLEAR LOG OF OLD INFORMATION           CL
  7 - TYPE                - COMMAND LOG      TY LOG    <name>
  8 -                     - PROCEDURE        TY PROC   <name>
  9 - NEW                 - WORKSPACE        N WS
 10 -                     - CONFIGURATION    N CFG     <base>
 11 - COPY                - WORKSPACE        CP WS     <f,to>
 12 -                     - PROCEDURE        CP PROC   <f,to>
 13 - ACTIVATE            - WORKSPACE        ACT WS    <name>
 14 -                     - CONFIGURATION    ACT CFG   <name>
 15 -                     - TEMPLATE         ACT TPL   <name>
 16 -                     - APPL. PROG.      ACT APPL  <name>
 17 -                     - UTILITY          ACT UTL   <menu>
 18 -                     - INPUT TEMPL      ACT ITPL
 19 -                     - OUTPUT TEMPL     ACT OTPL
 20 -                     - PROCEDURE        ACT PROC  <name>
 21 -                     - PROGRAM UFD      ACT PUFD  <path>
 22 - SAVE TEMPORARY      - WORKSPACE        SA WS     <name>
 23 -                     - PROCEDURE        SA PROC   <name>
 24 - REMOVE FROM UFD     - WORKSPACE        RM WS     <name>
 25 -                     - CONFIGURATION    RM CFG    <name>
 26 -                     - TEMPLATE         RM TPL    <name>
 27 -                     - PROCEDURE        RM PROC   <name>
 28 - SET USER LOGIN CHARACTERISTICS         SLOG

ENTER COMMAND:
13

ENTER FILE TO BE ACTIVATED ("1" TO LIST):
SAVDWS
```

SCREEN B22

```
 STATUS:


WORKSPACE:   SAVDWS           REF_CONFIG:   NUDATA
UTIL_IN_USE: PROCEDURE BUILD  APPLIC_PROG:  BOX
REF_TEMPLATE:                 REF_PROCFILE: NEWPROC


  PROCEDURE BUILDING

                                     COMMAND FORMAT
  1 - READ DESCRIPTION   - PROCEDURE        RD PROC  <name>
  2 - TYPE               - COMMAND LOG      TY LOG   <name>
  3 -                    - PROCEDURE        TY PROC  <name>
  4 - CLEAR LOG OF OLD INFORMATION          CL
  5 - ACTIVATE           - PROCEDURE        ACT PROC <name>
  6 - COPY               - PROCEDURE        CP PROC  <f,to>
  7 - DEFINE A NEW PROCEDURE                N PROC
  8 - EDIT AN EXISTING PROCEDURE            ED PROC  <name>
  9 - EDIT A LOG TO BUILD A PROCEDURE       ED LOG   <name>
 10 - SAVE A TEMPORARY PROCEDURE            SA PROC  <new >
 11 - CHANGE THE PROCEDURE DESCRIPTION      CD PROC
 12 - REMOVE THE PROCEDURE                  RM PROC
 13 - TYPE THE CURRENT COMMAND LOG          TY LOG   <-   >
```

SCREEN B23

```
ENTER COMMAND:
TY LOG

 WS ACTIVATED ON :  4-DEC-87
>ACT UTL  WSC
>T
* TOGGLE TTY PRINT MODE
>D
* DIRECTORY
>I
* INVENTORY
>CP CFG DEFAULT NUDATA
* COPY FROM:TOAIDE:[EXAMPLE.CFG.DEFAULT]          TO:NUDATA
* CARRIED OUT COMMAND: N CFG
>ACT APPL BOX
>RVU IDB
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  APEX
>EX APPL BOX
*    EXECUTING BOX
>RVU ODB
>TY LOG   T$RTA4
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  PBLD
>ED LOG T$RTA4
>TY LOG T$RTA4
>SA PROC NEWPROC
>L
* LOGOUT
>SA PROC NEWPROC
>SA WS SAVDWS
* WS ACTIVATED ON :  4-DEC-87
>ACT APPL BOX
>ACT CFG NUDATA
* ACTIVATING CONFIGURATION DATABASE *
>TY LOG SAVDWS
```

SCREEN B24

```
 STATUS:


WORKSPACE:   SAVDWS           REF_CONFIG:   NUDATA
UTIL_IN_USE: PROCEDURF BUILD  APPLIC_PROG:  BOX
REF_TEMPLATE:                 REF_PROCFILE: NEWPROC


  PROCEDURE BUILDING


                                      COMMAND FORMAT
  1 - READ DESCRIPTION    - PROCEDURE       RD PROC  <name>
  2 - TYPE                - COMMAND LOG     TY LOG   <name>
  3 -                     - PROCEDURE       TY PROC  <name>
  4 - CLEAR LOG OF OLD INFORMATION          CL
  5 - ACTIVATE            - PROCEDURE       ACT PROC <name>
  6 - COPY                - PROCEDURE       CP PROC  <f,to>
  7 - DEFINE A NEW PROCEDURE                N PROC
  8 - EDIT AN EXISTING PROCEDURE            ED PROC  <name>
  9 - EDIT A LOG TO BUILD A PROCEDURE       ED LOG   <name>
 10 - SAVE A TEMPORARY PROCEDURE            SA PROC  <new >
 11 - CHANGE THE PROCEDURE DESCRIPTION      CD PROC
 12 - REMOVE THE PROCEDURE                  RM PROC
 13 - TYPE THE CURRENT COMMAND LOG          TY LOG   <-   >

ENTER COMMAND:
EX PROC

 BEGIN PROCEDURE EXECUTION ON NEWPROC
```

SCREEN B25

```
                              CATEGORY       1:    DIMEN
BOX DIMENSIONS

----------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME   !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
----------------------------------------------------------------------
  1 !      10.0000000 ! LENGTH   !               ! BOX LENGTH   ! M
  2 !      20.0000000 ! WIDTH    !               ! BOX WIDTH    ! M
  3 !      30.0000000 ! HEIGHT   !               ! BOX HEIGHT   ! M
----------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
M 2

ENTER NEW VALUE:
10

>
E

 STARTING EXECUTION OF BOX
 EXECUTING BOX
```

SCREEN B26

```
                          CATEGORY       1:    DIMEN
BOX DIMENSIONS

-------------------------------------------------------------------------
 L  ! PRESENT VALUE  !   NAME    !  SUBSCRIPT   ! DESCRIPTION  ! UNITS
-------------------------------------------------------------------------
  1 !   3000.0000000 ! VOLUME    !              ! BOX VOLUME   ! M
-------------------------------------------------------------------------

M n : modify value (n = line#,name(subscript),or line range)
C n : change category (n = id or name)
N n : next page (n = + or - pages)
R : reprint page, L n : n line#'s per page, X n : expand line# n
E : end and save mods, Q : quit without saving mods, H : help
CAT : list categories, SUB : define review subset, T : toggle menu
EDIT:
>
Q

 EXITING PROCEDURE NEWPROC.PROC
* SAVED CURRENT STATUS OF ATTACHED WORKSPACE


  STATUS:


WORKSPACE:    SAVDWS            REF_CONFIG:    NUDATA
UTIL_IN_USE: PROCEDURE BUILD APPLIC_PROG:   BOX
REF_TEMPLATE:                   REF_PROCFILE: NEWPROC


  PROCEDURE BUILDING


                                   COMMAND FORMAT
  1 - READ DESCRIPTION    - PROCEDURE      RD PROC   <name>
  2 - TYPE                - COMMAND LOG    TY LOG    <name>
  3 -                     - PROCEDURE      TY PROC   <name>
  4 - CLEAR LOG OF OLD INFORMATION         CL
  5 - ACTIVATE            - PROCEDURE      ACT PROC  <name>
  6 - COPY                - PROCEDURE      CP PROC   <f,to>
  7 - DEFINE A NEW PROCEDURE               N PROC
  8 - EDIT AN EXISTING PROCEDURE           ED PROC   <name>
  9 - EDIT A LOG TO BUILD A PROCEDURE      ED LOG    <name>
 10 - SAVE A TEMPORARY PROCEDURE           SA PROC   <new >
 11 - CHANGE THE PROCEDURE DESCRIPTION     CD PROC
 12 - REMOVE THE PROCEDURE                 RM PROC
 13 - TYPE THE CURRENT COMMAND LOG         TY LOG    <-   >
```

SCREEN B27

```
ENTER COMMAND:
TY LOG

 WS ACTIVATED ON :  4-DEC-87
>ACT UTL  WSC
>T
* TOGGLE TTY PRINT MODE
>D
* DIRECTORY
>I
* INVENTORY
>CP CFG DEFAULT NUDATA
* COPY FROM:TOAIDE:[EXAMPLE.CFG.DEFAULT]          TO:NUDATA
* CARRIED OUT COMMAND: N CFG
>ACT APPL BOX
>RVU IDB
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  APEX
>EX APPL BOX
*    EXECUTING BOX
>RVU ODB
>TY LOG   T$RTA4
>Q
* QUIT THIS MENU, RETURN TO MAIN
>ACT UTL  PBLD
>ED LOG T$RTA4
>TY LOG T$RTA4
>SA PROC NEWPROC
>L
* LOGOUT
>SA PROC NEWPROC
>SA WS SAVDWS
* WS ACTIVATED ON :  4-DEC-87
>ACT APPL BOX
>ACT CFG NUDATA
* ACTIVATING CONFIGURATION DATABASE *
>TY LOG SAVDWS
>EX PROC NEWPROC
* BEGIN PROCEDURE EXECUTION ON NEWPROC
>RVU IDB
>EX APPL BOX
*    EXECUTING BOX
>RVU ODB
* EXITING PROCEDURE NEWPROC.PROC
* SAVED CURRENT STATUS OF ATTACHED WORKSPACE
>SA WS SAVDWS
>TY LOG SAVDWS

ENTER COMMAND:
L

SAVE CURRENT WORKSPACE STATUS (Y=Yes):
Y

* LOGOUT - BYE *
```

SCREEN B28

References

1. Rowell, Lawrence F.; and Davis, John S.: The Environment For Application Software Integration and Execution (EASIE) Version 1.0. VOLUME I - EXECUTIVE OVERVIEW. NASA TM-100573 April 1988.

2. Jones, Kennie H.; Randall, Donald P.; Stallcup, Scott S.; and Rowell, Lawrence F.: The Environment For Application Software Integration and Execution (EASIE) Version 1.0. VOLUME II - PROGRAM INTEGRATION GUIDE. NASA TM-100574, April 1988.

3. Randall, Donald P.; Jones, Kennie H.; and Rowell, Lawrence F.: The Environment For Application Software Integration and Execution (EASIE) Version 1.0. VOLUME IV - SYSTEM INSTALLATION AND MAINTENANCE GUIDE. NASA TM-100576, April 1988.

4. Jacky, J. P. and KaPet, I. J., "A General Purpose Data Entry Program", CACM, V. 26, No. 6, pp. 409-417, June 1983.

5. Dube, R. P. and Smith, M. R., "Managing Geometric Information With a Database Management System", IEEE Computer Graphics and Applications, V. 3, No. 7, pp. 57-62, October 1983.

6. Wilhite, A. W. and Crisp, V., A Relational Information System, ARIS User Manual, NASA/Langley Research Center, 1983.

7. BCS RIM - Relational Information Management System Version 6.0 User Guide, Boeing Computer Services Company, May 1983.

8. I-DEAS^{TM} USER'S GUIDE, Structural Dynamics Research Corporation, 1986.

Report Documentation Page

1. Report No.

NASA TM-100575

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

The Environment for Application Software Integration and Execution (EASIE) Version 1.0, Volume III - Program Execution Guide

5. Report Date

April 1988

6. Performing Organization Code

7. Author(s)

James L. Schwing, Lawrence F. Rowell, and Russell E. Criste

8. Performing Organization Report No.

10. Work Unit No.

506-49-31-01

9. Performing Organization Name and Address

NASA Langley Research Center
Hampton, VA 23665-5225

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, DC 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

James L. Schwing, Old Dominion University, Norfolk, Virginia.
Lawrence F. Rowell, Langley Research Center, Hampton, Virginia.
Russell E. Criste, Computer Sciences Corporation, Hampton, Virginia.

16. Abstract

The Environment for Application Software Integration and Execution, EASIE, provides a methodology and a set of software utility programs to ease the task of coordinating engineering design and analysis codes. EASIE was designed to meet the needs of conceptual design engineers that face the task of integrating the results of many stand-alone engineering analysis programs. EASIE provides access to these programs via a quick, uniform, user-friendly interface. In addition, EASIE provides utilities which aid in the execution of the following tasks: selection of applicatio programs, modification and review of program data, automatic definition and coordination of data files during program execution and a logging of steps executed throughou a design study. Volume III, the Program Execution Guide, describes the "executive" capabilities provided by EASIE and defines the command language and menus available under Version 1.0. EASIE provides users with two basic modes of operation. One is the Application-Derived Executive (ADE) which provides users with sufficient guidance to quickly review data, select menu action items, and execute application programs. The second is the Complete Control Executive (CCE) which provides a full executive interface allowing users in-depth control of the design process.

17. Key Words (Suggested by Author(s))

EASIE
Executive Software
Program Integration

18. Distribution Statement

Unclassified - Unlimited
Subject Category - 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	123	A06