SPERRY UNIVAC
COMPUTER SYSTEMS

ATTN: CHARLIE GIBBS

01101
CAV208M45541 UP 9106

SPERRY UNIVAC
1 - 1818 CORNWALL STREET
VANCOUVER B C UAS V6J 1C7

:#

PUBLICATIONS RELEASE

Operating System/3 (OS/3)

COBOL Editor (COBEDT)

User Guide/Programmer Reference

UP-9106

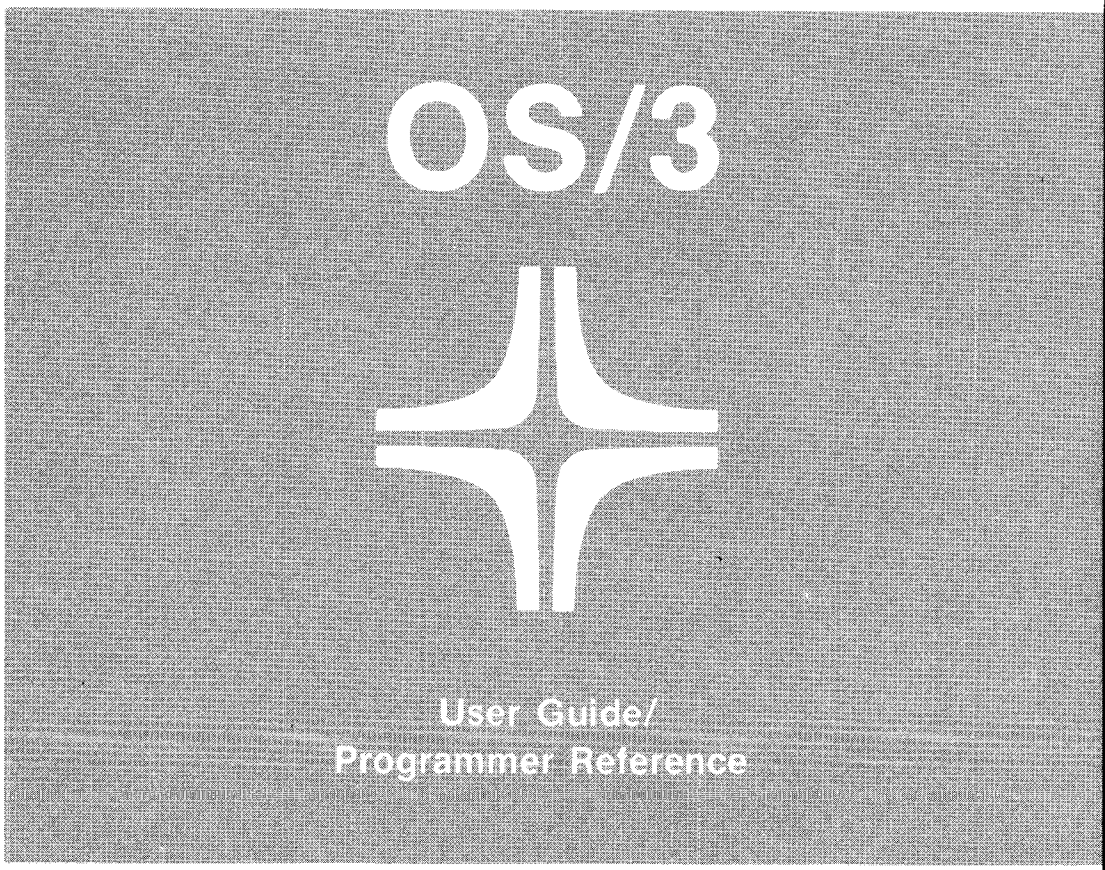
This Library Memo announces the release and availability of "SPERRY UNIVAC® Operating System/3 (OS/3) COBOL Editor (COBEDT) User Guide/Programmer Reference", UP-9106.

This manual describes the use of the COBOL editor to create and update COBOL source programs. Specifically, it explains and illustrates how to activate the COBOL editor, use it to create new source programs, use it to update existing COBOL source programs, and then terminate it. The manual also describes the editing commands available to the COBOL programmer and shows how the COBOL editor works with the OS/3 general editor (EDT).

Additional copies may be ordered by your local Sperry Univac representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, A31, B00, B31, 18, 18U, 19, 19U, 20, 20U, 21, 21U, 28U, 29U, 75, 75U, 76 and 76U (Cover and 168 pages)	Library Memo for UP-9106
		RELEASE DATE: October, 1982

COBOL Editor (COBEDT)



SPERRY UNIVAC

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-9106
RELEASE LEVEL: 8.0 Forward

Part/Section	Page Number	Update Level
Cover/Disclaimer		
PSS	1	
Preface	1 thru 3	
Contents	1 thru 5	
1	1 thru 5	
2	1 thru 8	
3	1 thru 7	
4	1 thru 52	
5	1 thru 17	
6	1 thru 19	
Appendix A	1 thru 20	
Appendix B	1 thru 11	
Appendix C	1 thru 11	
Index	1 thru 7	
User Comment Sheet		

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is one of a series designed to instruct and guide the user in the use of the SPERRY UNIVAC Operating System/3 (OS/3). It specifically describes the purpose and use of the COBOL editor (COBEDT), an interactive component of OS/3.

The intended audience is the novice user who has little knowledge of data processing and the COBOL language, or the more experienced user who may or may not be familiar with COBEDT.

This manual is divided as follows:

- Section 1. Introduction

  Explains what COBEDT is, what it's used for, and how it operates with the general editor (EDT).

- Section 2. Workstation Considerations

  Introduces features of the workstation and describes the structure and format conventions for COBEDT screens.

- Section 3. Using COBEDT

  Describes how to initiate and terminate the COBEDT session.

- Section 4. Creating COBOL Source Programs in Ordered Creation Mode

  Explains how to create COBOL source programs in the standard COBOL program order by using the COBEDT display screens. It also tells how to interactively compile COBOL programs and describes all the available ordered creation mode screens.

- Section 5. Creating COBOL Source Programs in Selective Creation Mode

  Explains how to selectively create portions of a COBOL source program by using the COBEDT display screens and describes all the available selective creation mode screens.

- Section 6. Editing COBOL Source Programs

  Explains through sample sessions how to edit COBOL source programs by using EDT commands in the COBEDT or the EDT environment.

- Appendix A. Procedure Division Verb Skeletons Screens

  Describes the purpose of the verb syntax skeleton screens and how to display them and shows all the available verb syntax skeleton screens.

- Appendix B. General Editor Command Summary

  Presents all the general editor commands, their formats, and a brief description of their functions.

- Appendix C. Workstation Command List

  Presents all the workstation commands and their formats.

The current versions of the following manuals are helpful to the COBEDT user in both the System 80 and Series 90 environments:

- General Editor (EDT) User Guide/Programmer Reference, UP-8828

  Describes the functions of the general editor commands and how to use them.

- Interactive Services Commands and Facilities User Guide/Programmer Reference, UP-8845

  Describes the commands and operating procedures for workstation terminals.

- Interactive Services Commands and Facilities Summary, UP-8938

  Summarizes the workstation commands.

- 1974 ANSI COBOL Programmer Reference, UP-8613

  Presents the rules for writing COBOL programs to be compiled by the 1974 ANSI COBOL compiler and executed under the control of OS/3.

- System Service Programs (SSP) User Guide, UP-8841 (System 80)
  System Service Programs (SSP) User Guide, UP-8062 (Series 90)

  Describe various system utilities (librarian, linkage editor, etc).

- System Messages Programmer/Operator Reference, UP-8076

  Describes the OS/3 system error messages.

- Spooling and Job Accounting Concepts and Facilities, UP-8869

  Describes basic spooling and job accounting concepts and options available to control spooling systems.

- Job Control User Guide, UP-8065

  Describes job control and its effective use.

Contents

3. USING COBEDT

4. CREATING COBOL SOURCE PROGRAMS IN ORDERED CREATION MODE

5. CREATING COBOL SOURCE PROGRAMS IN SELECTIVE CREATION MODE

6. EDITING COBOL SOURCE PROGRAMS

APPENDIXES

A. PROCEDURE DIVISION VERB SKELETON SCREENS

FIGURES

TABLES

1. Introduction

1.1. WHAT COBEDT IS USED FOR

Sperry Univac provides as part of Operating System/3 (OS/3) a COBOL editor, called COBEDT, to make COBOL source entry fast and easy.

As you may know, COBOL is a programming language resembling English. Just as in English, certain rules must be observed. Among the COBOL rules are that source statements must appear in a particular order and that each source statement must be entered in a particular column range or position.

In addition to these rules, COBOL has so many options that even experienced users have to refer to the COBOL reference manual frequently.

As a result, it is tedious to code large COBOL programs.

Well, COBEDT can make the work a lot simpler for you.

Creating/updating programs interactively

First, COBEDT is an interactive product. With it, you can create and update your COBOL source programs from a workstation. You can see your source entries displayed on the workstation screen as you key them in and therefore can check immediately for coding and typographical errors. Upon transmission of your entries, COBEDT also does a syntax check on them. Thus, there is yet another way of eliminating the traditional "first compilation" trivial errors before the program is compiled.

Immediate error and syntax checking

Screen formats

Second, COBEDT gives you screen formats to enter your source elements. The screen formats contain such things as required COBOL statements and directions for entering variable data. These formats may be displayed in sequence to prompt the inexperienced user through the creation of a COBOL source program, or the formats may be used on a selective basis for the more experienced user.

Now that you know what COBEDT can generally do for you, let's see exactly what COBEDT is and how it works.

1.2. WHAT COBEDT IS AND HOW IT WORKS

COBEDT is actually a subeditor of the general editor (EDT). It is activated from EDT and uses many of the facilities of EDT. Thus, you should be familiar with EDT before you start using COBEDT. For convenience, a brief overview of EDT is provided here. For a complete discussion of EDT, see the general editor (EDT) user guide/programmer reference.

Work-space file

Line number

EDT automatically creates a temporary disk file (known as the EDT work-space file) each time EDT is activated. This work-space file lasts for the duration of the EDT session, and it is here that all programs and files are created and updated. COBEDT uses this same work-space file when you create and update your COBOL programs. The work-space file has line numbers associated with it. You use these line numbers with COBEDT to both reference and manipulate your source code.

Creating programs

To create a new COBOL program via COBEDT, enter your source code into the EDT work-space file as shown in Figure 1-1. When all entries are made, you can then save a copy of the work-space file by writing it (via the EDT @WRITE command) to a permanent SAT file. Or, you can list it on the printer (via the EDT @LIST command) or even punch it on cards (via the EDT @PUNCH command).

Updating programs

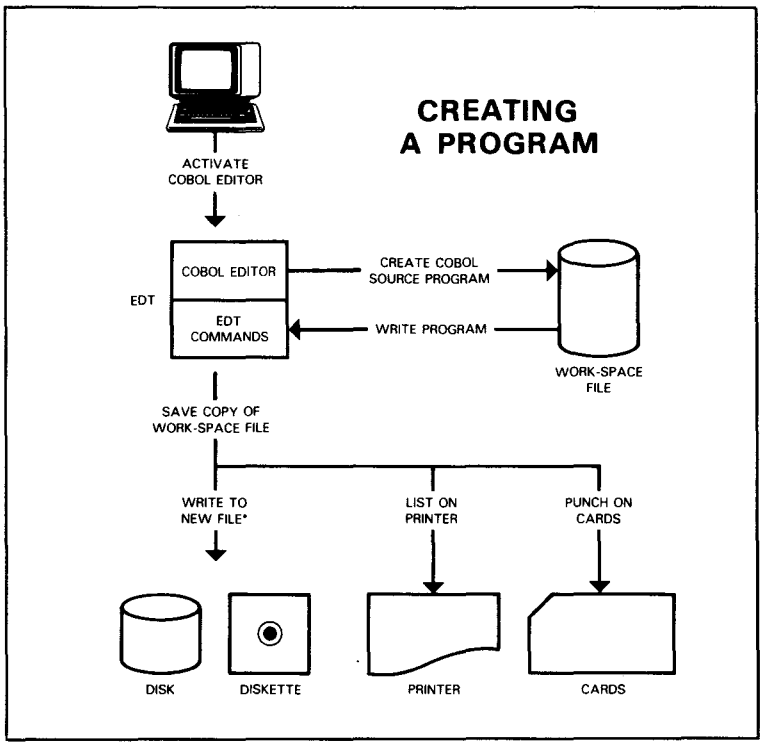
To update an existing COBOL program via COBEDT, first read a copy of the program into the work-space file via the EDT @READ command.The @READ command reads and then writes a copy of your program into the work-space file, and it also keeps the original version of your program in your library (permanent SAT file), thus providing you with a backup copy. All updating to your program is done in the work-space file. Once you finish updating your program, you can (as when creating a program) use EDT to save it, print it, or punch it.

NOTE:

Screen mode of EDT

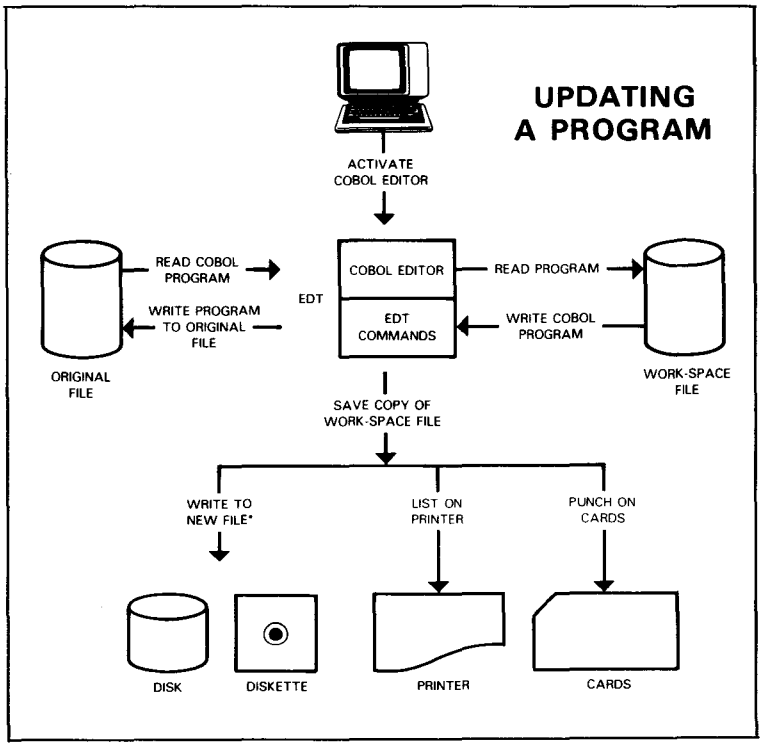
To make minor changes to your source program without syntax checking, use the screen mode of EDT. For detailed information, see the general editor (EDT) user guide/programmer reference.

Figure 1-1 shows how COBEDT operates with EDT to create a COBOL source program, while Figure 1-2 shows how COBEDT operates with EDT in updating a program.



*COBOL programs must be written to SAT files to be compiled.

Figure 1-1. Creating a COBOL Program

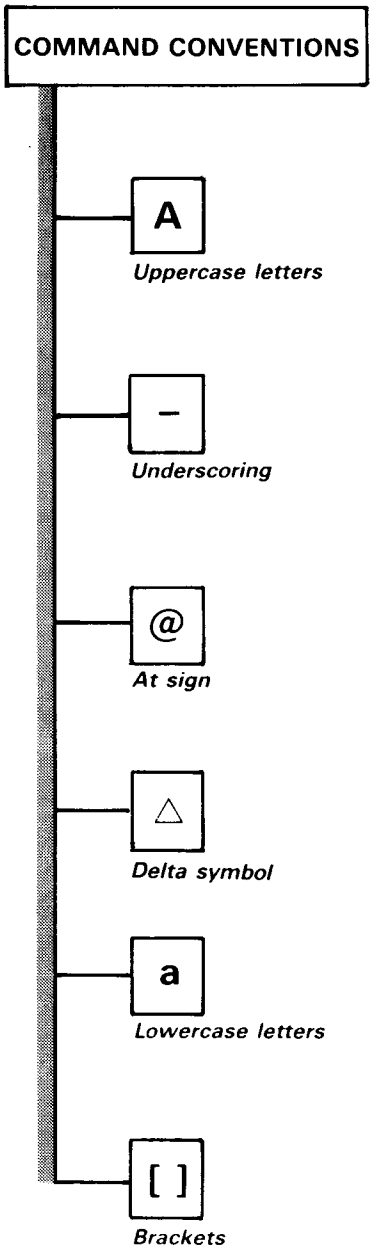


*COBOL programs must be written to SAT files to be compiled.

Figure 1-2. Updating a COBOL Program

1.3. COMMAND CONVENTIONS USED IN THIS MANUAL

The following command conventions are used in this manual:



Commands and parameters in uppercase (capital) letters must be keyed in literally. For example:

```
EDT
```

Underscoring (underlining) a portion of a command or parameter indicates its shortest abbreviation that must be keyed in. For example:

```
@COBOL
```

The at sign at the beginning of EDT commands must always be keyed in. For example:

```
@COB
```

The delta symbol indicates a space in command formats. For example:

```
EDT△@COBOL
```

Parameters or commands in lowercase letters designate user-defined variables. For example:

```
@PRINT line-range
```

Optional command parameters are enclosed in brackets. For example:

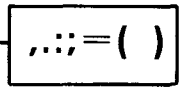
```
@PRINT [line-range]
```

COMMAND CONVENTIONS

Braces

Alternate choices for a command parameter are enclosed in braces. For example:

```
@WRITE FILENAME={filename  }
                {'filename'}
                {"filename"}
```



Special characters

Commas, periods, colons, semicoions, equal signs, and parentheses in command formats must be keyed in literally. For example:

```
@WRITE FILENAME=filename
```

Shading

Default values are shaded in command formats and in entries on COBEDT screens. For example:

```
Continuation Code (NRM)
```

Reverse print

User entries on COBEDT screens are in reverse print (white characters on black background). For example:

```
Continuation Code (CMD)
```

2. Workstation Considerations

2.1. LOGGING ON AND LOGGING OFF

Logging On

Why log on?

Since COBEDT is an interactive product, you access COBEDT via the workstation. If you're the first user, turn on the workstation. Allow the workstation a few seconds to warm up. When screen 2–1 appears, the workstation is ready and you may log on. Logging on identifies you as a legitimate user and gives you access to the system.

SCREEN 2-1

```
 000000      SSSS           /       333
00000000    SSSSSS         //      33333
00    00    SS   SS       ///     33   33
00    00    SS            ///         33
00    00     SS          ///          33
00    00      SS        ///           333
00    00       SS      ///            33
00    00    SS   SS   ///          33   33
00000000    SSSSSS    //           3333333
 000000      SSSS     /              3333

     SPERRY UNIVAC INTERACTIVE OPERATING SYSTEM
            DEPRESS TRANSMIT FOR LOGON
```

NOTE:

Before logging on, you must check with your system administrator to get a logon user-id and, if your system requires logon account numbers and passwords, a logon account number and password.

To log on, you may carry out either one of the following operations:

Logging on via the LOGON menu

- Press the XMIT key, fill in the LOGON menu displayed to you (screen 2-2), and then press the XMIT key again.

LOGON menu

SCREEN 2-2

```
                    OS/3 INTERACTIVE SERVICES
LOGON IDENTIFICATION:   USER-ID              >......<
                        ACCOUNT NUMBER       >....<
                        PASSWORD             >......<
OPTIONS:                EXECUTION PROFILE    >........<
                        BULLETIN             >YES<
                        LOG                  >YES<
```

Logging on via the LOGON command

- Enter SYSTEM mode, key in the LOGON command, and then press the XMIT key.

LOGON command

```
LOGON△user-id[,acct][,password][,exec-pro]
                [,BULLETIN={NO }][,LOG={YES}]
                           {YES}       {NO }
```

NOTE:

You may use the System 80 console workstation, the UNISCOPE 200 terminal with a 24 (lines) by 80 (columns) screen, or the UTS 400 terminal as the workstation (the System 80 workstation).

Before you can log on from a UNISCOPE or a UTS 400 terminal, you must first sign onto ICAM.

The logon procedure for the console workstation, the UNISCOPE terminal, and the UTS 400 terminal is the same as for the workstation, except the logon menu is obtained by entering WORKSTATION mode and then pressing the transmit key.

Logging Off

When you've finished using the workstation, take these steps to log off:

1. Enter SYSTEM mode if you are not already in it.
2. Key in the LOGOFF command: LOGOFF.
3. Press the XMIT key.

NOTE:

The logoff procedure for the console workstation, the UNISCOPE terminal, and the UTS 400 terminal is the same as for the workstation.

For more information on logging on and off, see the interactive services commands and facilities user guide/programmer reference.

2.2. MOVING THE CURSOR AND TRANSMITTING THE SCREEN

During a COBEDT session, the cursor can be repositioned on the workstation screen by pressing any of these keys:

- Tab keys (TAB FORWARD and TAB BACK)
- Cursor scan keys ← → ↑ ↓
- Cursor-to-home key
- Space bar
- RETURN key

When a COBEDT screen is displayed, the cursor is in the first unprotected field (user data field). To advance the cursor from field to field, press the TAB FORWARD key.

To fill in your data or overwrite the displayed default value, move the cursor to the desired user data field and then enter the desired data. To keep the default value, simply move the cursor to another data field.

Once you make your selections and enter all the desired data, move the cursor to the bottom of the screen (past the last user data field on the screen) and then press the XMIT key to transmit the screen.

If you want to keep all the default values on the screen, simply press the TAB BACK key once to move the cursor to the bottom of the screen and then press the XMIT key to transmit.

2.3. GENERAL STRUCTURE OF COBEDT SCREENS

All COBEDT screens have similar structures. The following is a line-by-line description of the general structure of these screens.

Line 1

Line 1 contains one of the three header information messages:

```
OS/3 EDT/COBOL                                    COBOL EDITOR(V8.0/1)
```

or

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
```

or

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
```

Line 2

Line 2 contains a string of asterisks (*).

```
************************************************************
```

Lines 3 to 19

User data fields

Lines 3 to 19 contain descriptive information and user data fields that vary from screen to screen. (You enter variable data in user data fields.)

Lines 20 and 21

Lines 20 and 21 are for you to enter one or more EDT commands:

```
EDT Command:@
```

The maximum length of this field is 128 characters. It's your responsibility to stay within this limit.

Line 22

Line 22, like line 2, contains a string of asterisks (*). Lines 2 and 22 separate user lines from system lines.

Lines 23 and 24

Lines 23 and 24 are left blank for the system to enter EDT and COBEDT error messages. Here's a sample COBEDT error message:

```
CED020 NUMERIC DATA REQUIRED.
CORRECT DATA AND TRANSMIT.
```

2.4. FORMAT NOTATIONS FOR COBEDT SCREENS

General Screen Format Notations

Here are the general screen format notations other than those for COBOL source elements.

General Screen Format Notations

( ) — *Enclosing user entries*

The user entries (data fields) are enclosed in parentheses. For example:

```
Continuation Code ( NRM )
```

You may overwrite the displayed default values in those fields with other acceptable values.

[ ] — *Enclosing next screen name*

The brackets next to the continuation code field enclose the name of the next screen to be displayed if the normal continuation code NRM is selected. (On ordered creation mode screens, this description is applicable only in ordered creation mode processing.) For example:

```
Continuation Code ( NRM ) [Next Screen is Environment Division]
```

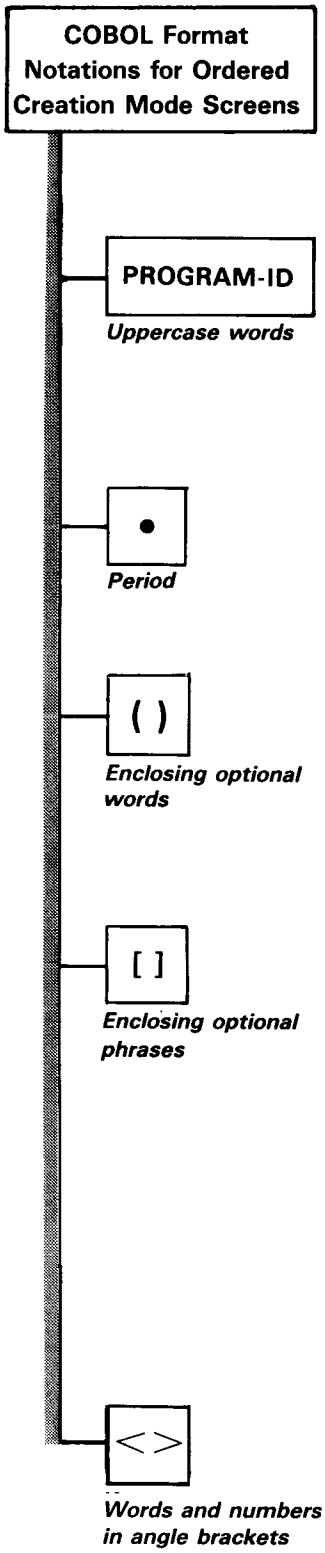
nnnn.nnnn — *Current work-space line number*

The current work-space line number appears in the form: nnnn.nnnn. It occurs to the right of the name of every COBOL source line entry type screen. The number is in the range of .0001 to 9999.9999. For example:

```
                Identification Division                 Line    1.
A   B
IDENTIFICATION DIVISION.
PROGRAM-ID.
```

COBOL Format Notations

Here are the COBOL format notations for ordered creation mode screens.



Uppercase words not enclosed in parentheses are required words. They're generated by COBEDT when the screen is transmitted. For example:

```
IDENTIFICATION DIVISION.
PROGRAM-ID.
```

Periods are required characters. They're generated by COBEDT when the screen is transmitted.

Uppercase words enclosed in parentheses aren't required by the COBOL syntax. However, they're generated by COBEDT when the screen is transmitted. For example:

```
ACCESS (MODE IS) SEQUENTIAL
```

Phrases enclosed in brackets aren't required in a COBOL program.

If a phrase itself is a complete one (that is, no user input is required), it's generated by COBEDT when the screen is transmitted with the required data entered. For example:

```
[ACCESS (MODE IS) SEQUENTIAL]
```

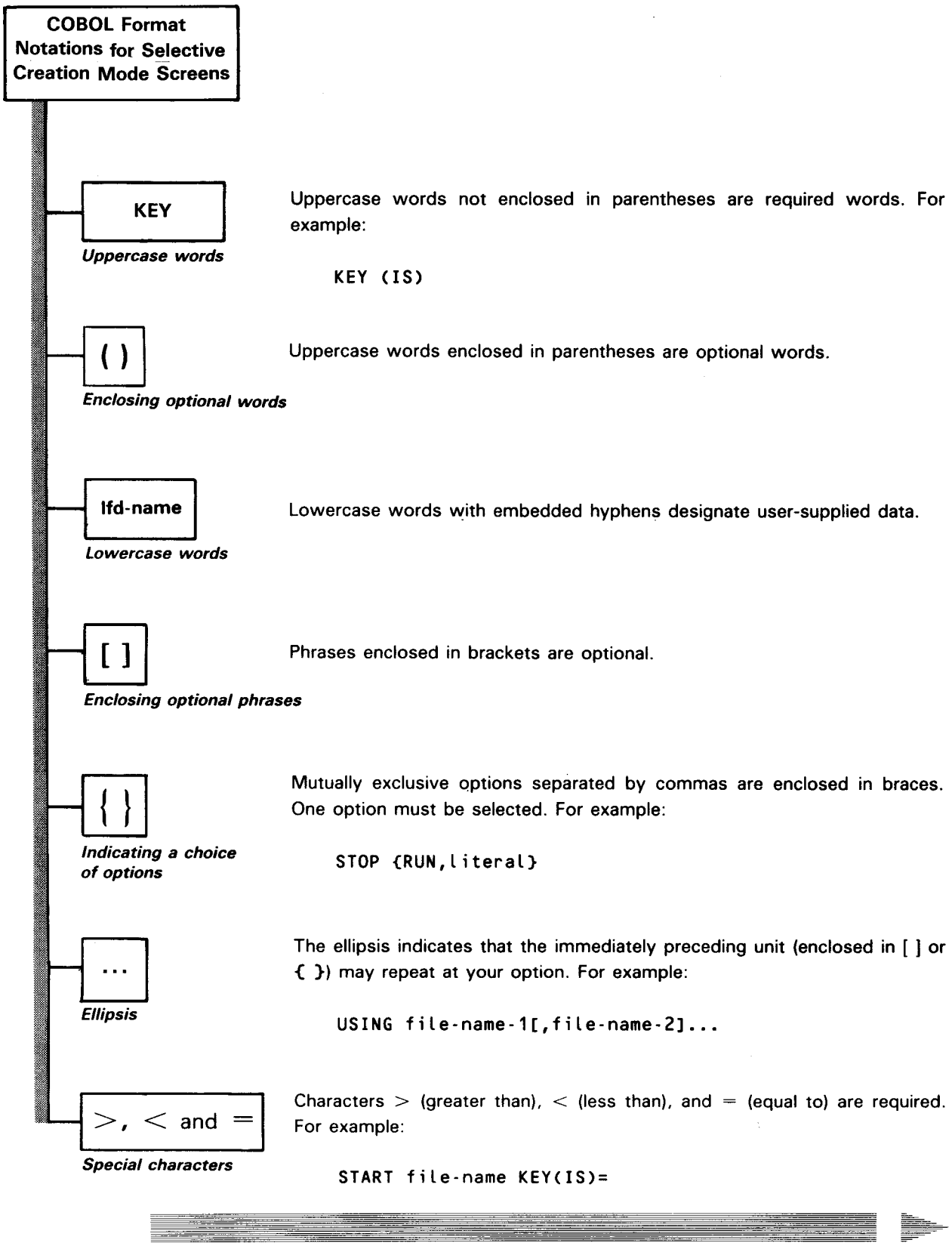
If a phrase needs your data input, it's generated only when the required data is entered and the screen is transmitted. For example:

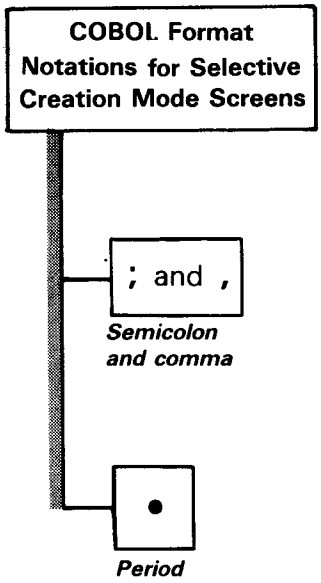
```
[RESERVE <1,2>AREA(S)]
```

A word or a series of words or numbers separated by commas enclosed in angle brackets < > describes the data or choices of data to be entered in the immediately preceding field. Numbers and words in uppercase must be entered literally. Lowercase designates user-supplied data. For example:

```
RERUN ON    <DISC,DISK,TAPE,lfdname>-     <lfdname>- <1,2>
```

Here are the COBOL format notations for selective creation mode screens.





Semicolons and commas (except those separating options) are optional characters. In source programs, semicolons and commas are interchangeable. For example:

```
[; INVALID KEY imperative-statement]
```

Periods must be used to terminate division and section headers, paragraph names, paragraphs in the identification and environment divisions, entries in the data division, and sentences in the procedure division. For example:

```
PROCEDURE DIVISION.
```

3. Using COBEDT

3.1. ACTIVATING COBEDT

Using COBEDT with EDT

Because COBEDT is a specialized language editor (subeditor) of EDT, COBEDT can be used only while EDT is activated.

Activating both EDT and COBEDT

If EDT is not already activated, activate both EDT and COBEDT by keying in:

```
EDT△@COBOL
```

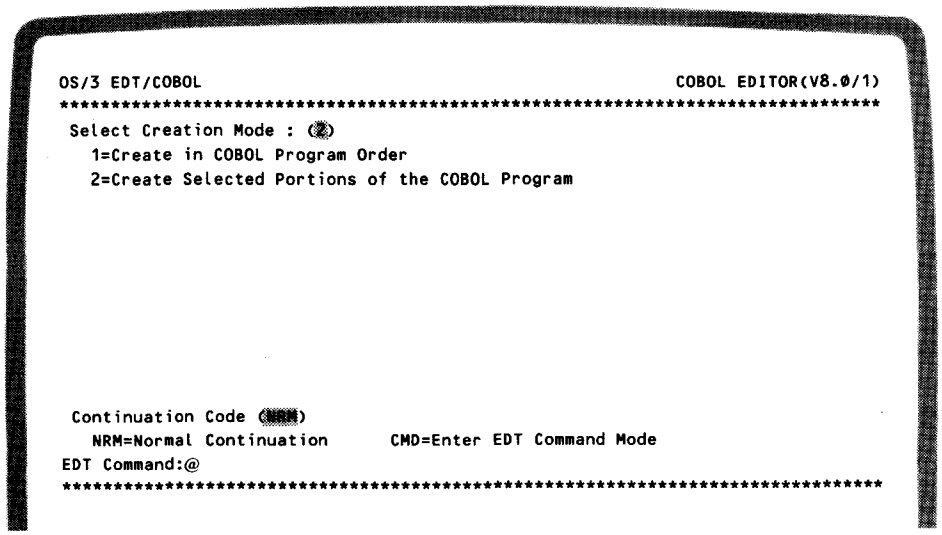
Activating COBEDT only

If you are already in an EDT session, key in:

```
@COBOL
```

3.2. INITIAL DISPLAY – THE OPTION SELECT SCREEN

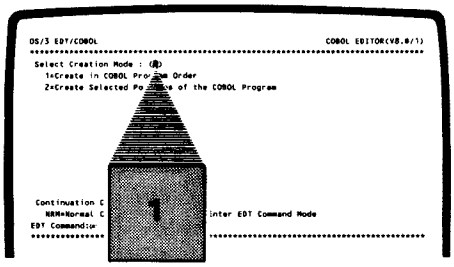
Once COBEDT is activated, the first workstation screen you see is the option select screen.



Use this screen to choose one of the two COBEDT operating modes, ordered and selective, to create your COBOL program. You also can use this screen to issue the EDT commands directly in COBEDT or to temporarily return to the EDT command mode. The following describes the options provided on the screen.

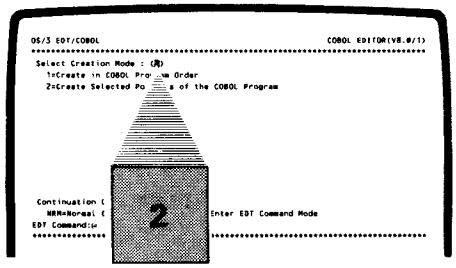
Select Creation Mode

Select option 1



when you want to create your COBOL program in the ordered creation mode. To select option 1, key in 1 over the displayed 2.

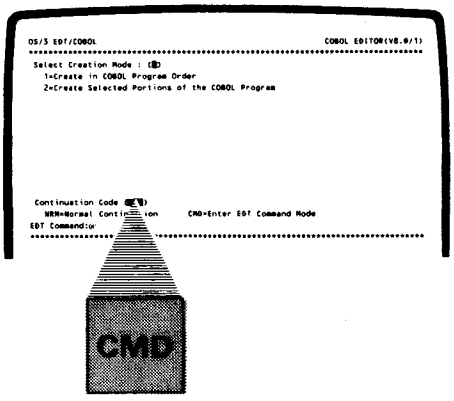
Select option 2



when you want to create your COBOL program in the selective creation mode. To select option 2, simply press the TAB FORWARD key to move the cursor to the next user field.

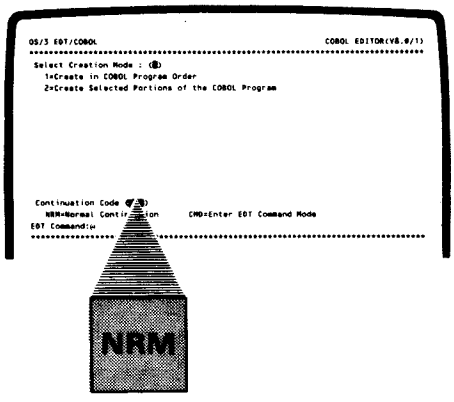
Continuation Code

Select CMD



when you want to temporarily return to EDT command mode and issue the EDT commands there. To select the CMD code, key in CMD over the displayed NRM.

Select NRM



when you want COBEDT to continue its normal processing and display the next required screen. The screen that appears next depends on the option selected in the select creation mode field. The next display is the

identification division screen if option 1 was selected

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Identification Division              Line 1.
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID.
 [AUTHOR.                                                                  .]
 [INSTALLATION.                                                            .]
 [DATE-WRITTEN.                                                            .]
 [DATE-COMPILED.                                                           .]
 [SECURITY.                                                                .]



 Continuation Code (NRM)  [Next Screen is Environment Division]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   CON = Display Control Division Screen
EDT Command:@
*******************************************************************************
```

or the standard COBOL coding form screen if option 2 was selected.

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form           Line 1.
 CIA   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (NRM)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

EDT Command:@ Field

You may enter one or more EDT commands here without leaving COBEDT. To enter the command, place the cursor to the right of the @ sign and then key in the command.

Transmitting the Option Select Screen

Once you make all your selections on the option select screen, move the cursor to the bottom of the screen (past the last selection on the screen) and then press the XMIT key to transmit your selections to COBEDT. If you want to keep all the default options, simply press the TAB BACK key to move the cursor to the bottom of the screen and then transmit.

Invalid options

If you make mistakes in selecting the options, upon transmission, COBEDT will redisplay the screen with an error message. You then correct the errors and retransmit the screen.

3.3. REENTERING COBEDT FROM EDT

To reenter COBEDT after you've temporarily returned to EDT through the CMD continuation code during a COBEDT session, key in the command:

```
@FORMAT
```

The @FORMAT command returns you to the same COBEDT creation mode as you exited and COBEDT resumes its normal processing.

NOTE:

If you use the @BLOCK screen command, be sure to terminate it before returning to COBEDT with @FORMAT. To terminate @BLOCK, use function key F14 or the @RESTORE screen command. Otherwise, you'll get errors.

Now, let's see what might result.

Example 1

SCREEN 1

COBEDT enters the EDT command mode when this option select screen is transmitted.

```
OS/3 EDT/COBOL                                              COBOL EDITOR(V8.0/1)
*******************************************************************************
 Select Creation Mode : (1)
   1=Create in COBOL Program Order
   2=Create Selected Portions of the COBOL Program




 Continuation Code (CMD)
   NRM=Normal Continuation      CMD=Enter EDT Command Mode
EDT Command:@
*******************************************************************************
```

SCREEN 2

When you return to COBEDT by keying in @FORMAT, the first screen displayed is the identification division screen.

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Identification Division              Line 1
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID.
 [AUTHOR.                                                                    .]
 [INSTALLATION.                                                              .]
 [DATE-WRITTEN.                                                              .]
 [DATE-COMPILED.                                                             .]
 [SECURITY.                                                                  .]



 Continuation Code (NRM)  [Next Screen is Environment Division]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   CON = Display Control Division Screen
EDT Command:@
*******************************************************************************
```

Example 2

SCREEN 1

COBEDT enters the EDT command mode when this standard COBOL coding form screen is transmitted.

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 000
 C|A   B
 END-IT.

     CLOSE TRANFIL.
     STOP RUN.
 |
 |
 |
 |
 |
 |
 Continuation Code (CMD)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

SCREEN 2

When you return to COBEDT by keying in @FORMAT, another standard COBOL coding form screen with the current work-space line number is displayed.

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 37
 C|A   B
 |
 |
 |
 |
 |
 |
 |
 |
 |
 |
 Continuation Code (CMD)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

3.4. TERMINATING COBEDT

To terminate a COBEDT session, key in either of the following commands:

@COBOL△END

The @COBOL END command terminates COBEDT but not EDT.

or

@HALT

The @HALT command terminates both COBEDT and EDT.

4. Creating COBOL Source Programs in Ordered Creation Mode

4.1. GENERAL FEATURES OF THE ORDERED CREATION MODE

Two creation modes
Ordered
Selective

COBEDT has two operating modes, the ordered creation mode and the selective creation mode. The ordered creation mode is designed to assist the inexperienced users in developing a COBOL source program, whereas the selective creation mode is mainly for the experienced COBOL users. (See Section 5 for the selective creation mode.)

Screens displayed before reaching the working-storage section

In the ordered creation mode, COBEDT displays a sequence of creation screens. These screens provide the information to create the COBOL source statements in the standard COBOL program order up to but not including the working-storage section of the data division. The following two screens exemplify this sequence.

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Identification Division              Line nnnn.nnnn
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID.
 [AUTHOR.                                                                  .]
 [INSTALLATION.                                                            .]
 [DATE-WRITTEN.                                                            .]
 [DATE-COMPILED.                                                           .]
 [SECURITY.                                                                .]



 Continuation Code (NRM)  [Next Screen is Environment Division]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   CON = Display Control Division Screen
EDT Command:@
*******************************************************************************
```

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Environment Division                  Line nnnn.nnnn
 A   B
 ENVIRONMENT DIVISION.   <Enter X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].   <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                <alphabet-name>]
     [SEGMENT-LIMIT IS   <segment-number>].
 Continuation Code (NRM)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen         SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen   SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@
*******************************************************************************
```

Each screen serves as a template that provides spaces for you to fill in the required data to complete that portion of the program described.

The data division coding form screen

When the working-storage section is reached, COBEDT displays the data division coding form screen, which provides spaces to enter your data and allows you to request other creation screens to aid in your coding. The screen is redisplayed until either COBEDT is terminated or the procedure division coding form screen is requested.

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Data Division Coding Form               Line nnnn.nnnn
  Level#  Data Description




 Continuation Code (NRM)   [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation        SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode     PDC = Display Procedure Division Coding Form
   ICD = Display Input CD Screen    DAT = Display Data Item Description Screen
   OCD = Display Output CD Screen   CND = Display Condition Name Screen
                                    REN = Display RENAMES Screen
EDT Command:@
*******************************************************************************
```

The procedure division coding form screen

The procedure division coding form screen provides spaces for coding the procedure division. During the process, you may also request the display of COBOL verb skeleton screens to assist you in coding the verbs. The procedure division coding form screen is redisplayed until COBEDT is terminated.

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Procedure Divsion Coding Form          Line nnnn.nnnn
  CIA   B
   |
   |
   |
   |
   |
   |
   |
   |
   |
   |
 Continuation Code (NRM)   [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
*******************************************************************************
```

NOTE:

Appendix A describes the COBOL verb skeleton screens.

4.2. OPERATING IN ORDERED CREATION MODE

Entering the Ordered Creation Mode

To create your COBOL source program via the ordered creation mode, you must enter option 1 in the select creation mode field of the option select screen. When this screen is processed, the identification division screen, which is the first screen in the ordered creation mode, is displayed.

```
OS/3 EDT/COBOL
**********************************
 Select Creation Mode : 1
   1=Create in COBOL Program Order
   2=Create Selected Portions of the COBOL Pro
```

Keying in Source Data on the Screens

When an ordered creation mode screen is displayed, the cursor is in the line field of the screen. To key in your source data, advance the cursor to the desired user fields and then key in your data. Pressing the TAB FORWARD key advances the cursor from field to field.

COBOL Syntax Error Detection and Handling

When an ordered creation screen is transmitted, COBEDT verifies the validity of the data entered on the screen before creating the source lines. For a template display screen (such as the environment division screen), COBEDT checks whether all required data for that screen is entered. For a data division coding form screen, COBEDT checks whether the level numbers entered are valid. (Note that 01 through 49, 66, 77, and 88 are valid level numbers.) For a procedure division coding form screen, COBEDT checks whether the section headers and the paragraph names are started in column 8 and checks whether every procedure division sentence entered contains a valid COBOL verb keyword. However, COBEDT does not check the syntax of the verb.

Valid level numbers

An error found

If an error is found, an error message appears on the last two lines of the current screen and the location of the error is indicated by a blinking field. For example:

An erroneous entry – 0E

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*************************************************************************
                        Environment Division              Line 5.

 A  B
 ENVIRONMENT DIVISION.  <ENTER X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
    [(WITH) DEBUGGING MODE].  <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
    [(PROGRAM COLLATING) SEQUENCE (IS)              <alphabet-name>]
    [SEGMENT-LIMIT IS 0E<segment-number>].
 Continuation Code (NRM)  [Next Screen is File Control]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen       SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen

EDT Command:@
*************************************************************************
CED020 NUMERIC DATA REQUIRED.
CORRECT DATA AND TRANSMIT.
```

How to correct an error

To correct the error, reposition the cursor to the beginning of the blinking field (if it does not automatically reposition itself to the error field) and key in the correct data. If you choose not to correct it, simply retransmit the screen. The source lines are then created and written to the work-space file as they are, and no additional action is taken. Multiple errors on a screen are handled by COBEDT one at a time.

Multiple errors

The Work-Space Line Numbers

In ordered creation mode processing, the current work-space line number displayed on an ordered creation mode screen can't be altered. To insert new lines into those lines already created, enter the selective creation mode and code these new lines on the standard COBOL coding form screen. However, in this case, you must make certain that these new lines are placed properly in the work-space file to avoid the overwriting of the lines. (See the first example in Section 6.)

Entering Selective Creation Mode or EDT Command Mode

During ordered creation mode processing, you may temporarily enter the selective creation mode or the EDT command mode by keying in the required continuation code on the current ordered creation screen.

Why enter the selective creation mode?

Enter the selective creation mode to display a standard COBOL coding form, a COBOL verb skeleton, the COBOL program skeleton, or any ordered creation mode screen that cannot be requested directly from the current screen.

Why enter the EDT command mode?

Enter the EDT command mode to edit your source lines with the various EDT commands in the EDT environment.

When the screen is transmitted, COBEDT first processes source lines entered and then enters either the EDT command mode or the selective creation mode. Upon return, COBEDT resumes ordered creation mode processing.

4.3. SAMPLE CREATION SESSION

This sample session shows how to create a COBOL source program in the ordered creation mode. The example starts with choosing the creation mode on the option select screen. We then code the source statements in the standard COBOL program order via a sequence of creation screens. The example ends with writing the program to a permanent SAT file.

SCREEN 1 OPTION SELECT SCREEN

```
OS/3 EDT/COBOL                                              COBOL EDITOR(V8.0/1)
*******************************************************************************
 Select Creation Mode : (1)
   1=Create in COBOL Program Order
   2=Create SELECTED Portions of the COBOL Program




 Continuation Code (   )
   NRM=Normal Continuation      CMD=Enter EDT Command Mode
EDT Command:@
*******************************************************************************
```

On this screen, we enter 1 in the select creation mode field and keep NRM in the continuation code field. Upon transmission, the identification division screen is displayed, with current work-space line number 1.

SCREEN 2 **IDENTIFICATION DIVISION SCREEN**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Identification Division              Line 1
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID. PROG1
 [AUTHOR. J.JONES                                                          .]
 [INSTALLATION. UNIVAC                                                     .]
 [DATE-WRITTEN.                                                            .]
 [DATE-COMPILED.                                                           .]
 [SECURITY.                                                                .]



 Continuation Code (NRM)     [Next Screen is Environment Division]
   NRM = Normal Continuation        SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode     CON = Display Control Division Screen
EDT Command:@
*******************************************************************************
```

Here, we fill in our source data and select the NRM continuation code. Upon transmission, the environment division screen is displayed after the source lines are created and placed in the work-space file.

SOURCE LINES CREATED

```
1.0000        IDENTIFICATION DIVISION.
2.0000        PROGRAM-ID. PROG1.
3.0000        AUTHOR. J.JONES.
4.0000        INSTALLATION. UNIVAC.
```

SCREEN 3 **ENVIRONMENT DIVISION SCREEN**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                            Environment Division                Line
 A   B
 ENVIRONMENT DIVISION.   <ENTER X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].   <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                <alphabet-name>]
     [SEGMENT-LIMIT IS   <segment-number>].
 Continuation Code      [Next Screen is File Control]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen           SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen     SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@
*******************************************************************************
```

On this screen, the lines not enclosed in brackets are created by default. Thus, for our program, we simply press the TAB BACK key to move the cursor to the bottom of the screen and then transmit. The file control screen is displayed, with current work-space line number 9, after the source lines are created and placed in the work-space file.

SOURCE LINES CREATED

```
5.0000        ENVIRONMENT DIVISION.
6.0000        CONFIGURATION SECTION.
7.0000        SOURCE-COMPUTER. UNIVAC-OS3.
8.0000        OBJECT-COMPUTER. UNIVAC-OS3.
```

SCREEN 4 **FILE CONTROL SCREEN**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                                  File Control                    Line
 A   B
 INPUT-OUTPUT SECTION.   <Enter X if Line is not to be created in Select Mode>
 FILE-CONTROL.
 Continuation Code (INX)  [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         REL = Relative File Select
   CDR = Cardreader File Select         INX = Indexed File Select
   CDP = Cardpunch File Select          TAP = Tape File Select
   PRT = Printer File Select            DIS = Sequential Disk File Select
   IOC = Display I-O Control Screen
EDT Command:@
*******************************************************************************
```

On this screen, we overwrite the default continuation code NRM with INX to request the indexed select screen for coding the next part of our program, the INDEXED SELECT clause. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then displays the requested screen.

SOURCE LINES CREATED

```
 9.0000        INPUT-OUTPUT SECTION.
10.0000        FILE-CONTROL.
```

SCREEN 5 **INDEXED SELECT SCREEN**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Indexed Select                        Line 11.

 A   B
     SELECT
         TRANFIL                                <file-name>
         ASSIGN (TO) DISC-TRANFIL <lfdname>-V<F,V>
         [RESERVE   <1,2> AREA(S')]
         ORGANIZATION (IS) INDEXED
         [ACCESS (MODE IS) SEQUENTIAL<SEQUENTIAL,RANDOM,DYNAMIC>]
         RECORD (KEY IS) TRAN-KEY                          <data-name>
         [(FILE) STATUS (IS)                                  <data-name>].

 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         FST = Display File Select Type Codes
   fff = Next File Select Code          ALK = Alternate Record Keys Required
EDT Command:@
*******************************************************************************
```

On this screen, we enter our source data and select the NRM continuation code. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then displays a data division FD screen with file name TRANFIL in its FD file-name field.

SOURCE LINES CREATED

```
11.0000          SELECT TRANFIL
12.0000              ASSIGN TO DISC-TRANFIL-V
13.0000              ORGANIZATION IS INDEXED
14.0000              ACCESS MODE IS SEQUENTIAL
15.0000              RECORD KEY IS TRAN-KEY.
```

SCREEN 6 **DATA DIVISION – FD SCREEN**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                    Data Division - File Section (FD)        Line [redacted]

 A   B
 FD      TRANFIL                          <file-name>
         [BLOCK CONTAINS          <integer> [TO          <integer>]
                    <RECORDS,CHARACTERS>]
         [RECORD (CONTAINS)     <integer> [TO      <integer>] (CHARACTERS)]
         LABEL RECORD(S) (ARE) STANDARD        <STANDARD,OMITTED,data-name>
         [VALUE OF [FILE-ID (IS)                              <data-name>]
             [PASSWORD (IS)                                 <data-name>]
         [CODE-SET (IS)                                   <alphabet-name>].

 Continuation Code ([redacted])  [Next Screen is FD Data Record Form]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           LIN = Linage Screen Required
   SDD = Current File is a Sort File-Display SD Screen

EDT Command:@
*******************************************************************************
```

We enter our source data and select the NRM continuation code. (Note that file name TRANFIL is automatically displayed by COBEDT.) Upon transmission, COBEDT first creates the DATA DIVISION and FILE SECTION source lines since this FD is the first entry in this division and section. Then it creates the source lines from the screen. After placing the source lines in the work-space file, COBEDT displays the FD data record form screen.

SOURCE LINES CREATED

```
16.0000       DATA DIVISION.
17.0000       FILE SECTION.
18.0000       FD  TRANFIL
19.0000           LABEL RECORDS ARE STANDARD.
```

SCREEN 7 **FD DATA RECORD FORM SCREEN**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              FD Data Record Form                 Line 20
 C|A   B                          41
 |01  TRAN-OUT.                     <data-record-name>
 |    03  TRAN-KEY.
 |      05  TRAN-PUB          PIC XX.
 |      05  TRAN-REPORT       PIC 99.
 |      05  TRAN-SORT         PIC X(23).
 |      05  TRAN-SEQUENCE     PIC 9(4).
 |    03  TRAN-VARIABLE         PIC X
 |        OCCURS 1 TO 100 TIMES
 |        DEPENDING ON SYS-REC-SIZE.
 Continuation Code (NRM)  [Next Screen varies]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

Again, we enter our source statements and select the NRM continuation code. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then displays the data division coding form screen.

SOURCE LINES CREATED

```
20.0000       01  TRAN-OUT.
21.0000           03  TRAN-KEY.
22.0000             05  TRAN-PUB          PIC XX.
23.0000             05  TRAN-REPORT       PIC 99.
24.0000             05  TRAN-SORT         PIC X(23).
25.0000             05  TRAN-SEQUENCE     PIC 9(4).
26.0000           03  TRAN-VARIABLE         PIC X
27.0000               OCCURS 1 TO 100 TIMES
28.0000               DEPENDING ON SYS-REC-SIZE.
```

SCREEN 8 **DATA DIVISION CODING FORM SCREEN**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Data Division Coding Form              Line 29
 Level#  Data Description
 01      DUMMY-OUT.
 03      DUMMY-KEY.
 05      DUMMY-PUB           PIC XX.
 05      DUMMY-REPORT        PIC 99.
 05      DUMMY-SORT          PIC X(23).
 05      DUMMY-SEQUENCE      PIC 9(4).
 03      DUMMY-VARIABLE      PIC X(100).
 77      SYS-REC-SIZE     PIC 9(4) VALUE 0.
Continuation Code (PDC)  [Next Screen is Data Division Coding Form]
  NRM = Normal Continuation       SEL = Enter Selective Creation Mode
  CMD = Enter EDT Command Mode    PDC = Display Procedure Division Coding Form
  ICD = Display Input CD Screen   DAT = Display Data Item Description Screen
  OCD = Display Output CD Screen  CND = Display Condition Name Screen
                                  REN = Display RENAMES Screen

EDT Command:@
*******************************************************************************
```

Here, we enter our source statements for the working-storage section and overwrite the continuation code NRM with PDC to request the procedure division coding form screen be displayed next. Upon transmission, COBEDT first creates the WORKING-STORAGE SECTION source line and then the source lines entered, since the latter lines are the first entries in this section. As you can see, COBEDT also checks the level numbers and automatically indents data description entries reflecting the hierarchy of the data. After placing these source lines in the work-space file, COBEDT displays the procedure division coding form screen.

SOURCE LINES CREATED

```
29.0000       WORKING-STORAGE SECTION.
30.0000       01  DUMMY-OUT.
31.0000           03  DUMMY-KEY.
32.0000             05  DUMMY-PUB           PIC XX.
33.0000             05  DUMMY-REPORT        PIC 99.
34.0000             05  DUMMY-SORT          PIC X(23).
35.0000             05  DUMMY-SEQUENCE      PIC 9(4).
36.0000           03  DUMMY-VARIABLE     PIC X(100).
37.0000       77  SYS-REC-SIZE     PIC 9(4) VALUE 0.
```

SCREEN 9 **PROCEDURE DIVISION CODING FORM SCREEN**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Procedure Division Coding Form          Line 38.

  CIA   B
  PROCEDURE DIVISION.
  PRIMARY-PARAGRAPH.
  |     OPEN OUTPUT TRANFIL.
  |     MOVE "AA" TO DUMMY-PUB.
  |     MOVE 99 TO DUMMY-REPORT.
  |     MOVE ALL "B" TO DUMMY-SORT.
  |     MOVE 1 TO DUMMY-SEQUENCE.
  |     MOVE ALL "C" TO DUMMY-VARIABLE.

 Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton

EDT Command:@
*******************************************************************************
```

On this screen, we enter our source statements and select the NRM continuation code. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then displays another procedure division coding form screen.

SOURCE LINES CREATED

```
38.0000         PROCEDURE DIVISION.
39.0000         PRIMARY-PARAGRAPH.
40.0000             OPEN OUTPUT TRANFIL.
41.0000             MOVE "AA" TO DUMMY-PUB.
42.0000             MOVE 99 TO DUMMY-REPORT.
43.0000             MOVE ALL "B" TO DUMMY-SORT.
44.0000             MOVE 1 TO DUMMY-SEQUENCE.
45.0000             MOVE ALL "C" TO DUMMY-VARIABLE.
```

SCREEN 10 **PROCEDURE DIVISION CODING FORM SCREEN**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Procedure Division Coding Form          Line 46.

  CIA   B
  LOOP-HERE.
  |     ADD 1 TO SYS-REC-SIZE.
  |     MOVE DUMMY-OUT TO TRAN-OUT.
  |     WRITE TRAN-OUT INVALID KEY
  |
  |
  |
  |
  |
 Continuation Code (DISPLAY)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation       SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode    vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
*******************************************************************************
```

Here, we enter more procedure division source statements. Because we need assistance in coding the next verb DISPLAY, we overwrite the default code NRM with DISPLAY to request the DISPLAY verb skeleton. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then displays the requested screen.

SOURCE LINES CREATED

```
46.0000         LOOP-HERE.
47.0000             ADD 1 TO SYS-REC-SIZE.
48.0000             MOVE DUMMY-OUT TO TRAN-OUT.
49.0000             WRITE TRAN-OUT INVALID KEY
```

SCREEN 11 **DISPLAY STATEMENT SCREEN**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                              Display Statement

 DISPLAY {identifier-1,literal-1} [,{identifier-2,literal-2}] ...
     [UPON mnemonic-name [USING {identifier-3,literal-3}]]
*******************************************************************************
                          Procedure Division Coding Form          Line 50

 C|A   B
  |     DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
  |    IF SYS-REC-SIZE < 100
  |        ADD 1 TO DUMMY-SEQUENCE
  |        GO TO LOOP-HERE.

 Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton

EDT Command:@
*******************************************************************************
```

On this screen, we code the DISPLAY verb and several additional procedure division source statements. Upon transmission, COBEDT creates the source lines, places them in the work-space file, and then returns to a procedure division coding form screen.

SOURCE LINES CREATED

```
50.0000              DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
51.0000            IF SYS-REC-SIZE < 100
52.0000                ADD 1 TO DUMMY-SEQUENCE
53.0000                GO TO LOOP-HERE.
```

SCREEN 12 **PROCEDURE DIVISION CODING FORM SCREEN**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                        Procedure Division Coding Form          Line 56
  CIA   B
  END-IT.
       CLOSE TRANFIL.
       STOP RUN.
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (CMD)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
********************************************************************************
```

Now we enter the last paragraph for our program and the CMD continuation code. Upon transmission, COBEDT creates the source lines. After placing the source lines in the work-space file, COBEDT enters the EDT command mode, and the next work-space line number (57.0000) appears on the workstation screen by itself.

SOURCE LINES CREATED

```
54.0000        END-IT.
55.0000             CLOSE TRANFIL.
56.0000             STOP RUN.
```

At this point, the creation of our program is complete. Let's see what our entire program looks like in the work-space file by using the @PRINT command before we terminate COBEDT.

```
57.0000▷@PRINT
 1.0000       IDENTIFICATION DIVISION.
 2.0000       PROGRAM-ID. PROG1.
 3.0000       AUTHOR. J.JONES.
 4.0000       INSTALLATION. UNIVAC.
 5.0000       ENVIRONMENT DIVISION.
 6.0000       CONFIGURATION SECTION.
 7.0000       SOURCE-COMPUTER. UNIVAC-OS3.
 8.0000       OBJECT-COMPUTER. UNIVAC-OS3.
 9.0000       INPUT-OUTPUT SECTION.
10.0000       FILE-CONTROL.
11.0000           SELECT TRANFIL
12.0000               ASSIGN TO DISC-TRANFIL-V
13.0000               ORGANIZATION IS INDEXED
14.0000               ACCESS MODE IS SEQUENTIAL
15.0000               RECORD KEY IS TRAN-KEY.
16.0000       DATA DIVISION.
17.0000       FILE SECTION.
18.0000       FD  TRANFIL
19.0000           LABEL RECORDS ARE STANDARD.
20.0000       01  TRAN-OUT.
21.0000           03  TRAN-KEY.
22.0000             05  TRAN-PUB          PIC XX.
23.0000             05  TRAN-REPORT       PIC 99.
24.0000             05  TRAN-SORT         PIC X(23).
25.0000             05  TRAN-SEQUENCE     PIC 9(4).
26.0000           03  TRAN-VARIABLE         PIC X
27.0000               OCCURS 1 TO 100 TIMES
28.0000               DEPENDING ON SYS-REC-SIZE.
29.0000       WORKING-STORAGE SECTION.
30.0000       01  DUMMY-OUT.
31.0000           03  DUMMY-KEY.
32.0000             05  DUMMY-PUB           PIC XX.
33.0000             05  DUMMY-REPORT        PIC 99.
34.0000             05  DUMMY-SORT          PIC X(23).
35.0000             05  DUMMY-SEQUENCE      PIC 9(4).
36.0000           03  DUMMY-VARIABLE      PIC X(100).
37.0000       77  SYS-REC-SIZE    PIC 9(4) VALUE 0.
```

SCREEN 2

SCREEN 3

SCREEN 4

SCREEN 5

SCREEN 6

SCREEN 7

SCREEN 8

SCREEN 9

```
38.0000        PROCEDURE DIVISION.
39.0000        PRIMARY-PARAGRAPH.
40.0000            OPEN OUTPUT TRANFIL.
41.0000            MOVE "AA" TO DUMMY-PUB.
42.0000            MOVE 99 TO DUMMY-REPORT.
43.0000            MOVE ALL "B" TO DUMMY-SORT.
44.0000            MOVE 1 TO DUMMY-SEQUENCE.
45.0000            MOVE ALL "C" TO DUMMY-VARIABLE.
```

SCREEN 10

```
46.0000        LOOP-HERE.
47.0000            ADD 1 TO SYS-REC-SIZE.
48.0000            MOVE DUMMY-OUT TO TRAN-OUT.
49.0000            WRITE TRAN-OUT INVALID KEY
```

SCREEN 11

```
50.0000                DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
51.0000            IF SYS-REC-SIZE < 100
52.0000                ADD 1 TO DUMMY-SEQUENCE
53.0000                GO TO LOOP-HERE.
```

SCREEN 12

```
54.0000        END-IT.
55.0000            CLOSE TRANFIL.
56.0000            STOP RUN.
57.0000▷@COBOL END
57.0000▷@WRITE MO=PROG1,FILE=TESTFIL,VSN=A00011,SAT=YES,SIZE=1
57.0000▷
```

Now, we key in the @COBOL END command to end this COBEDT creation session. We then write this program (PROG1) to a permanent SAT file (TESTFIL) on disk volume A00011. Again, line number 57.0000 is displayed, ready to accept more entries.

NOTE:

You must allocate a permanent file, if not already existing, before you can write a copy of the work-space file to it. For our example, we include the SIZE parameter in the @WRITE command line to allocate the new file, TESTFIL.

Compiling the Sample COBOL Program

After creating and saving sample program PROG1, we want to compile it to see whether there are any compilation errors. We must, therefore, write and run a job control stream to invoke the COBOL compiler, as shown in the following sample workstation entries:

```
1.   57.0000▷@DELETE
2.    1.0000▷// JOB JOBNAM
      2.0000▷//PROG1 COBL74 IN=(A00011,TESTFIL),OBJ=(A00011,FILE.OBJ)
      3.0000▷       ,ERRFIL=(MYERR,A00011,ERR.LST)
      4.0000▷/&
3.    5.0000▷@WRITE MO=JOBNAM,FIL=FILE.JCS,VSN=A00011,SAT=YES,SIZE=1
4.    6.0000▷@HALT
```

First, delete the copy of program PROG1 in the work-space file with the @DELETE command.

Second, create a job control stream to compile the program. Also, define an error file for compile-time diagnostics (ERRFIL parameter), if any.

Third, save the job control stream in SAT file FILE.JCS on disk volume A00011.

Fourth, end EDT with the @HALT command.

Now, enter SYSTEM mode.

Finally, key in the RV workstation command to run the job control stream:

```
RV JOBNAM:(FILE.JCS,A00011)
```

NOTE:

Assume SAT file FILE.OBJ and MIRAM file ERR.LST have already been allocated through the ALLOCATE workstation command. (See Appendix C for the command format.) Also, include the SIZE parameter in the @WRITE command line to allocate the new file, FILE.JCS.

After the job is executed, a message indicating successful or unsuccessful compilation is displayed at the workstation.

If the compilation is successful, we can proceed to link and execute it. For information, see the 1974 ANSI COBOL programmer reference and the job control user guide.

If there are compilation errors, instead of waiting for the compiler to print the error listing, we can immediately activate the EDT error file processor (EFP) to display both the errors and their corresponding source lines. We can then correct and recompile the program. For more information on EFP, see the general editor (EDT) user guide/programmer reference. For editing COBOL programs, see Section 6 in this manual.

4.4. ORDERED CREATION MODE SCREENS

This section describes all the available ordered creation mode screens.

Control Division Screen

SCREEN 4-1 CONTROL DIVISION

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                            Control Division                    Line nnnn.nnnn
 A   B
 CONTROL DIVISION.   <Enter X if CONTROL DIVISION and ALPHABET SECTION
 ALPHABET SECTION.    are not to be created in Select Mode>
     [SOURCE-ALPHABET (CHARACTERS ARE)
                                        <literal> [THRU       <literal>]
                                        <literal> [THRU       <literal>]
                                        <literal> [THRU       <literal>]
                                        <literal> [THRU       <literal>]
     [MESSAGES (ARE)                          ].
 Continuation Code (NRM)  [Next Screen is Identification Division]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         RDP = Redisplay This Screen
EDT Command:@
*******************************************************************************
```

CON code

In a COBOL source program, the control division is optional. If included, it must be the first entry in the program. The control division screen can be displayed by entering the CON continuation code on the identification division screen (screen 4–2). Entering the RDP continuation code on the current control division screen redisplays the control division screen.

RDP code

All user data is optional.

All user data on this screen is optional. COBEDT doesn't create any source lines if no data is entered. In ordered creation mode processing, although the control division screen can be redisplayed, COBEDT generates the CONTROL DIVISION and ALPHABET SECTION source lines only once.

Return to ID division screen

After you've completed your control division entries, return to the identification division screen with the NRM continuation code.

Identification Division Screen

SCREEN 4–2 IDENTIFICATION DIVISION

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Identification Division              Line nnnn.nnnn
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID.
 [AUTHOR.                                                                    .]
 [INSTALLATION.                                                              .]
 [DATE-WRITTEN.                                                              .]
 [DATE-COMPILED.                                                             .]
 [SECURITY.                                                                  .]

 Continuation Code (NRM)  [Next Screen is Environment Division]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       CON = Display Control Division Screen
EDT Command:@
*******************************************************************************
```

The identification division screen assists you in creating the identification division, an essential part of a COBOL source program. It's the first screen displayed after entering the ordered creation mode. You must fill in your PROGRAM-ID if you select the NRM continuation code.

PROGRAM-ID

Control division

If you want to include the control division in your program, you must not enter the program ID or other source data but overwrite the continuation code NRM with CON to display the control division screen. The control division must be coded before the identification division. After you complete your control division, transmitting the control division screen with the NRM continuation code returns you to the identification division screen.

Environment Division Screens

SCREEN 4-3 ENVIRONMENT DIVISION

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                            Environment Division                 Line nnnn.nnnn
 A   B
 ENVIRONMENT DIVISION.   <Enter X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].   <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                 <alphabet-name>]
     [SEGMENT-LIMIT IS   <segment-number>].

 Continuation Code (    )  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen         SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen   SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@
*******************************************************************************
```

The environment division screen provides the information necessary to create the basic portion of the environment division and to display any one of the six special-names screens (screens 4–5 through 4–10). This screen is displayed when COBEDT completes processing the identification division screen. If no user data is entered, COBEDT still creates the source lines for those not enclosed in brackets. You may, however, inform COBEDT not to create this portion of the division by entering an X in the ENVIRONMENT DIVISION line. If you enter an X in the ENVIRONMENT DIVISION line and a code CMD or SEL in the continuation code field, upon returning to ordered creation mode processing, the environment division screen is redisplayed. On the other hand, if you have not entered an X in that line, upon return, the special-names select screen (screen 4–4) is displayed, allowing you to request a special-names screen.

The X option for environment division

SCREEN 4–4 **SPECIAL-NAMES SELECT**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Special-Names Select Screen

   SN1 = Display Special Names 1 Screen      SSW = Display SYSSWCH Screen
   SN2 = Display Special Names 2 Screen      ALP = Display Alphabet-Name Screen
   SCH = Display SYSCHAN Screen              CLN = Display Class-Name Screen


 Continuation Code (    )   [Next Screen is File Control]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           sss = Next Special-Names Screen
EDT Command:@
*******************************************************************************
```

The special-names select screen lists the available special-names screens and their display codes.

This screen is displayed each time you reenter the ordered creation mode after temporarily entering either the EDT command mode or the selective creation mode from one of the following screens:

- The environment division screen without an X entered in its ENVIRONMENT DIVISION line, but with a code CMD or SEL in its continuation code field.
- Any one of the special-names screens with source data entered and a code CMD or SEL in its continuation code field. (See screens 4–5 through 4–10 for special-names screens.)

Special-names screens

The following special-names screens provide all the information to create the special-names section of the environment division. There are six special-names screens. The SYSCHAN, alphabet-name, SYSSWCH, and class-name screens are for coding the portions of the special-names section indicated by their names. Special-names screen 1 and 2 are for coding the rest of the special-names section. Each special-names screen also provides the information to display any one of the other five special-names screens.

All user data on each special-names screen is optional. COBEDT doesn't create any source lines if no data is entered. If you temporarily enter either the EDT command mode or the selective creation mode from a special-names screen without filling in any source data on the screen, upon returning to ordered creation mode processing, the same screen is redisplayed. On the other hand, if you have filled in the source data, upon return, the special-names select screen is displayed, allowing you to request another special-names screen.

In ordered creation mode processing, even though several special-names screens can be used, COBEDT generates the SPECIAL-NAMES header only once. COBEDT also places a period at the end of the special-names section.

SCREEN 4-5 **SPECIAL-NAMES SCREEN 1**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                      Special-Names Screen (1 of 6)          Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
     [SYSSCOPE IS                                  <mnemonic-name>]
     [SYSWORK IS                                  <mnemonic-name>
          ASSIGN (TO)          <lfdname-1>]
     [SYSFORMAT IS                                  <mnemonic-name>
          ASSIGN (TO)          <lfdname-2>]
     [CURRENCY (SIGN) IS  <literal>]
     [DECIMAL-POINT IS COMMA].   <Enter X if Line is to be created>
 Continuation Code (____)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet-Name Screen   SCH = Display SYSSCHAN Screen
   CLN = Display Class-Name Screen      SSW = Display SYSSWCH Screen
EDT Command:@
*******************************************************************************
```

SN1 code

Special-names screen 1 is displayed when requested by the SN1 continuation code from the environment division screen, the special-names select screen, or other special-names screens.

After the screen is transmitted with the desired source data entered, COBEDT first creates the SPECIAL-NAMES source line if not already created and then other source lines from the screen. Keeping the code NRM in the continuation code field indicates that you have finished all your entries for the special-names section and the period displayed on the screen is generated. Otherwise, the period is generated at the end of the section. Also, if there are more entries for the section, COBEDT moves the source lines created for the CURRENCY SIGN IS and DECIMAL-POINT IS COMMA items to the end of the section.

SCREEN 4-6

SPECIAL-NAMES SCREEN 2

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                        Special-Names Screen (2 of 6)          Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
     [SYSIN IS                                  <mnemonic-name>]
     [SYSCONSOLE IS                                 <mnemonic-name>]
     [SYSLST IS                                 <mnemonic-name>]
     [SYSLOG IS                                 <mnemonic-name>]
     [SYSCOM IS                                 <mnemonic-name>]
     [SYSTERMINAL IS                                <mnemonic-name>]
     [SYSOUT IS                                 <mnemonic-name>]

 Continuation Code (SN2)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN1 = Display Special Names 1 Screen
   ALP = Display Alphabet-Name Screen   SCH = Display SYSCHAN Screen
   SSW = Display SYSSWCH Screen         CLN = Display Class-Name Screen

EDT Command:@
********************************************************************************
```

SN2 code

Special-names screen 2 is displayed when requested by the SN2 continuation code from the environment division screen, the special-names select screen, or other special-names screens.

SCREEN 4-7

SPECIAL-NAMES SYSCHAN

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                        Special-Names Screen (3 of 6)          Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]
     [SYSCHAN-   IS                             <mnemonic-name>]

Continuation Code (SCH)  [Next Screen is File Control]
  NRM = Normal Continuation          SEL = Enter Selective Creation Mode
  CMD = Enter EDT Command Mode       RDP = Redisplay This Screen
  ALP = Display Alphabet-Name Screen CLN = Display Class-Name Screen
  SSW = Display SYSSWCH Screen       SNn = Display Special-Names Screen n(n=1 or 2)
EDT Command:@
********************************************************************************
```

The SYSCHAN screen provides the information to create the SYSCHAN portion of the special-names. This screen is displayed when requested by the SCH continuation code from the environment division screen, the special-names select screen, or other special-names screens.

SCH code

RDP code

To code more SYSCHAN source lines, enter the code RDP in the continuation code field to redisplay the screen.

SCREEN 4-8 **SPECIAL-NAMES ALPHABET-NAME**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Special-Names Screen (4 of 6)          Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
                                    <alphabet-name> (IS)
         [           <STANDARD-1, NATIVE, STANDARD-0>]. or
         [[    <literal> [THRU     <literal>]]
         [    <literal> [THRU     <literal>]]].
         [    <literal> [ALSO     <literal>] [ALSO     <literal>]
         [ALSO     <literal>] [ALSO     <literal>] [ALSO     <literal>]
         [ALSO     <literal>] [ALSO     <literal>] [ALSO     <literal>]
         [ALSO     <literal>] [ALSO     <literal>]].
 Continuation Code (   ) [Next Screen is File Control]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   RDP = Redisplay This Screen
   SSW = Display SYSSWCH Screen   CLN = Display Class-Name Screen
   SCH = Display SYSCHAN Screen   SNn = Display Special-Names Screen n(n=1 or 2)
EDT Command:@
*******************************************************************************
```

The alphabet-name screen provides the information to create the alphabet-name portion of the special-names. This screen is displayed when requested by the ALP continuation code from the environment division screen, the special-names select screen, or other special-names screens.

ALP code

To code more alphabet-name source lines, enter the code RDP in the continuation code field to redisplay the screen.

RDP code

SCREEN 4-9 **SPECIAL-NAMES SYSSWCH**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                      Special-Names Screen (5 of 6)          Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
     SYSSWCH-  <0,1,2,3,4,5,6,7>
       [IS                                  <mnemonic-name>]
       [ON (STATUS) IS                             <condition-name>]
       [OFF (STATUS) IS                             <condition-name>]
     SYSSWCH-  <0,1,2,3,4,5,6,7>
       [IS                                  <mnemonic-name>]
       [ON (STATUS) IS                             <condition-name>]
       [OFF (STATUS) IS                             <condition-name>]
 Continuation Code (SSW)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         RDP = Redisplay This Screen
   ALP = Display Alphabet-Name Screen   CLN = Display Class-Name Screen
   SCH = Display SYSCHAN Screen         SN1 = Display Special-Names 1 Screen
   SN2 = Display Special-Names 2 Screen
EDT Command:@
*******************************************************************************
```

SSW code

The SYSSWCH screen provides the information to create the SYSSWCH portion of the special-names. This screen is displayed when requested by the SSW continuation code from the environment division screen, the special-names select screen, or other special-names screens.

Required data

When required data is missing

When you code a SYSSWCH source line, you must include either the ON STATUS IS or the OFF STATUS IS field along with other required data in the line. If neither field is coded, an error message is displayed and you are given an opportunity to add the missing data. If you don't add the missing data, the source line is created as is and no further action is taken by COBEDT.

RDP code

To code more SYSSWCH source lines, enter the code RDP in the continuation code field to redisplay the screen.

SCREEN 4-10 **SPECIAL-NAMES CLASS-NAME**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                      Special-Names Screen (6 of 6)            Line nnnn.nnnn
 A   B
 SPECIAL-NAMES.   <Enter X if Header is not to be created in Select Mode>
     CLASS-NAME (IS)                                    <mnemonic-name>
         [VALUE (IS)                     <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]
                                         <literal> [THRU       <literal>]]
 Continuation Code (   )  [Next Screen is File Control]
   NRM = Normal Continuation             SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode          RDP = Redisplay This Screen
   ALP = Display Alphabet-Name Screen    SCH = Display SYSCHAN Screen
   SSW = Display SYSSWCH Screen          SN1 = Display Special-Names 1 Screen
   SN2 = Display Special-Names 2 Screen
EDT Command:@
*******************************************************************************
```

The class-name screen provides the information to create the class-name portion of the special-names. This screen is displayed when requested by the CLN continuation code from the environment division screen, the special-names select screen, or other special-names screens.

CLN code

To code more class-names or values, enter the code RDP in the continuation code field to redisplay the screen.

RDP code

SCREEN 4-11 **INPUT-OUTPUT SECTION FILE CONTROL**

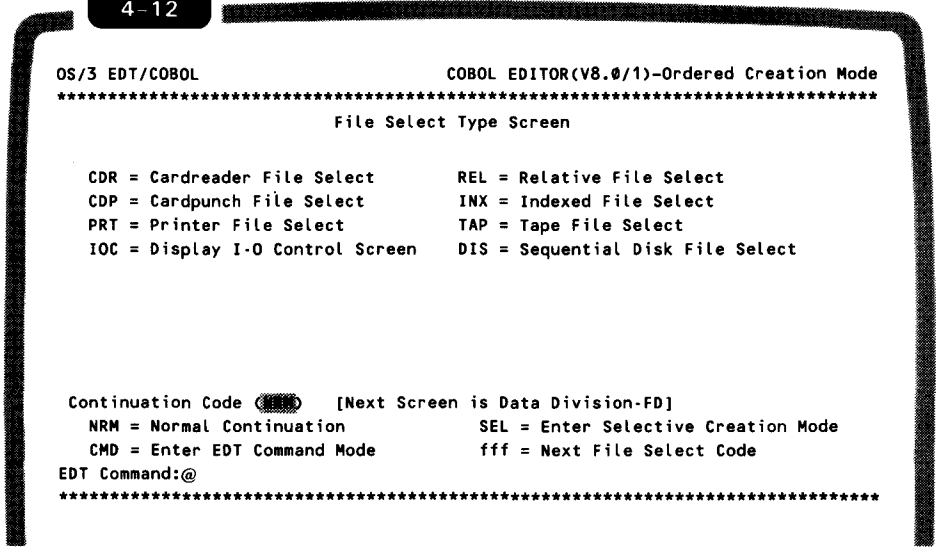
```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              File Control                  Line nnnn.nnnn
 A   B
 INPUT-OUTPUT SECTION.   <Enter X if Section is not to be created>
 FILE CONTROL.
 Continuation Code (   )  [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        REL = Relative File Select
   CDR = Cardreader File Select        INX = Indexed File Select
   CDP = Cardpunch File Select         TAP = Tape File Select
   PRT = Printer File Select           DIS = Sequential Disk File Select
   IOC = Display I-O Control Screen
EDT Command:@
*******************************************************************************
```

The file control screen provides the information to initialize the file-control portion of the input-output section and to display either a desired file select screen or the I-O control screen.

This screen is displayed if you selected the NRM continuation code on the environment division screen, the special-names select screen, or any of the special-names screens.

Keeping the code NRM in the continuation code field indicates that you have finished coding the environment division and the data division coding form screen (screen 4–36) is to be displayed next for coding the working-storage section of the data division.

SCREEN 4–12 **FILE SELECT TYPE**



The file select type screen lists the available file select and the I-O control screens and their display codes.

FST code

This screen is displayed when requested by the FST continuation code from any of the file select screens or from the indexed select alternate record keys screen. (See screens 4–13 through 4–20.)

This screen is also displayed automatically each time you reenter the ordered creation mode after temporarily entering either the EDT command mode or the selective creation mode from a file select screen or the alternate record keys screen with source data entered and a code CMD or SEL in its continuation code field.

Seven select screens

There are seven file select screens for seven COBOL file types, namely cardreader, cardpunch, printer, tape, sequential disk, relative, and indexed. Each screen provides the information to create the file select portion of the input-output section as indicated by its name.

Since the select clause is optional in a COBOL program, COBEDT doesn't create any source lines if no user data is entered on a file select screen. If you do code a select clause, COBEDT adds the file name specified to a temporary internal table to be used in the creation of the FD source lines. If you want to display another file select screen or the I-O control screen, enter the proper file select code on the current select screen.

If you temporarily enter either the EDT command mode or the selective creation mode from a file select screen without filling in any source data on the screen, upon returning to the ordered creation mode, the same screen is redisplayed. On the other hand, if you have filled in the source data, upon return, the file select type screen is displayed, allowing you to request another file select screen or the I-O control screen.

SCREEN 4-13 **CARDREADER SELECT**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Cardreader Select                 Line nnnn.nnnn
 A   B
     SELECT          <OPTIONAL, spaces>
                                        <file-name>
         ASSIGN (TO) CARDREADER-        <lfdname>-F
         [RESERVE   <1,2> AREA(S)]
         [ORGANIZATION (IS) SEQUENTIAL]
         [ACCESS (MODE IS) SEQUENTIAL]
         [(FILE) STATUS (IS)                                  <data-name>].
 Continuation Code (nnn)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

CDR code

The cardreader select screen is displayed when requested by the CDR select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

SCREEN 4-14 **CARDPUNCH SELECT**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Cardpunch Select                    Line nnnn.nnnn
 A   B
     SELECT          <OPTIONAL, spaces>
                                        <file-name>
         ASSIGN (TO) CARDPUNCH-        <lfdname>-  <F,U,V>
         [RESERVE   <1,2> AREA(S)]
         [ORGANIZATION (IS) SEQUENTIAL]
         [ACCESS (MODE IS) SEQUENTIAL]
         [(FILE) STATUS (IS)                                    <data-name>].
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

CDP code

The cardpunch select screen is displayed when requested by the CDP select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

SCREEN 4-15 **PRINTER SELECT**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Printer Select                      Line nnnn.nnnn
 A   B
     SELECT          <OPTIONAL, spaces>
                                        <file-name>
         ASSIGN (TO) PRINTER-          <lfdname>-  <FC,UC,VC>
         [RESERVE   <1,2> AREA(S)]
         [ORGANIZATION (IS) SEQUENTIAL]
         [ACCESS (MODE IS) SEQUENTIAL]
         [(FILE) STATUS (IS)                                    <data-name>].
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

PRT code

The printer select screen is displayed when requested by the PRT select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

SCREEN 4-16 **TAPE SELECT**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                                Tape Select                      Line nnnn.nnnn
 A   B
     SELECT          <OPTIONAL, spaces>
                                        <file-name>
         ASSIGN (TO) TAPE-        <lfdname>-  <F,U,V,FC,UC,VC>
         [RESERVE   <1,2> AREA(S)]
         [ORGANIZATION (IS) SEQUENTIAL]
         [ACCESS (MODE IS) SEQUENTIAL]
         [(FILE) STATUS (IS)                                  <data-name>].
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

TAP code

The tape select screen is displayed when requested by the TAP select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

SCREEN 4-17 **SEQUENTIAL DISK SELECT**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                           Sequential Disk Select                Line nnnn.nnnn
 A   B
     SELECT          <OPTIONAL, spaces>
                                        <file-name>
         ASSIGN (TO) DISC-        <lfdname>-  <F,V,FC,VC>
         [RESERVE   <1,2> AREA(S)]
         [ORGANIZATION (IS) SEQUENTIAL]
         [ACCESS (MODE IS) SEQUENTIAL]
         [(FILE) STATUS (IS)                                  <data-name>].
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

DIS code

The sequential disk select screen is displayed when requested by the DIS select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

SCREEN 4-18 **RELATIVE SELECT**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                            Relative Select                    Line nnnn.nnnn
 A   B
     SELECT
                                      <file-name>
         ASSIGN (TO) DISC-         <lfdname>- <F,V>
         [RESERVE   <1,2> AREA(S)]
         ORGANIZATION (IS) RELATIVE
         [ACCESS (MODE IS)           <SEQUENTIAL,RANDOM,DYNAMIC>]
         [RELATIVE (KEY IS)                                <data-name>]
         [(FILE) STATUS (IS)                               <data-name>].
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        FST = Display File Select Type Codes
   fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

REL code

The relative select screen is displayed when requested by the REL select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

Required data

If you indicate ACCESS MODE IS RANDOM or DYNAMIC, the RELATIVE KEY IS field is then required. If this field is not coded, an error message is displayed.

SCREEN 4-19 **INDEXED SELECT**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                            Indexed Select                     Line nnnn.nnnn
 A   B
     SELECT
                                      <file-name>
         ASSIGN (TO) DISC-         <lfdname>- <F,V>
         [RESERVE   <1,2> AREA(S)]
         ORGANIZATION (IS) INDEXED
         [ACCESS (MODE IS)           <SEQUENTIAL,RANDOM,DYNAMIC>]
         RECORD (KEY IS)                              <data-name>
         [(FILE) STATUS (IS)                             <data-name>].
 Continuation Code (INX)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        FST = Display File Select Type Codes
   fff = Next File Select Code         ALK = Alternate Record Keys Required
EDT Command:@
*******************************************************************************
```

INX code

The indexed select screen is displayed when requested by the INX select code from the file control screen, the file select type screen, other file select screens, or the alternate record keys screen.

ALK code

If you want to specify alternate record keys for the indexed file, enter code ALK in the continuation code field to display the alternate record keys screen for coding.

SCREEN 4-20

INDEXED SELECT ALTERNATE RECORD KEYS

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                    Indexed Select Alternate Record Keys       Line nnnn.nnnn
 A   B
         [ALTERNATE RECORD (KEY IS)                                <data-name>
             (WITH)            <DUPLICATES,spaces>]
         [ALTERNATE RECORD (KEY IS)                                <data-name>
             (WITH)            <DUPLICATES,spaces>]
         [ALTERNATE RECORD (KEY IS)                                <data-name>
             (WITH)            <DUPLICATES,spaces>]
         [ALTERNATE RECORD (KEY IS)                                <data-name>
             (WITH)            <DUPLICATES,spaces>]
 Continuation Code (NRM)  [Next Screen is Data Division-FD]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       FST = Display File Select Type Codes
   RDP = Display This Screen Again    fff = Next File Select Code
EDT Command:@
*******************************************************************************
```

The alternate record keys screen provides the information to create the alternate key source lines for an indexed select clause. This screen is displayed when requested by code ALK from the indexed select screen.

ALK code

Keys are optional.

The alternate record keys are optional. If no user data is entered on the screen, no source lines are created. If you want to code more alternate keys, enter RDP in the continuation code field to redisplay the screen. If you want to display another file select screen or the I-O control screen, enter the proper file select code in the continuation code field.

Coding more keys

SCREEN 4-21

I-O CONTROL

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              I-O Control                  Line nnnn.nnnn
 A   B
 I-O-CONTROL.   <Enter X if Line is not to be created in Select Mode>
     [RERUN ON [        <DISC,DISK,TAPE>-]        <lfdname>[- <1,2>]
        EVERY          <integer> RECORDS (OF)
                                  <filename>]
     [  <Enter X if filenames to be created without SAME AREA line>
      SAME           <RECORD,SORT,SORT-MERGE> (AREA FOR)
                                  <filename>
                                  <filename>]
     [MULTIPLE FILE (TAPE CONTAINS)   <Enter X if Line is to be created>
                                  <filename>
        POSITION          <integer>]
 Continuation Code (NRM)  [Next Screen is I-O Control Applys]
   NRM = Normal Continuation     SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode  RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

The I-O control screen provides the information to create part of the I-O control paragraph.

IOC code

This screen is displayed when requested by the IOC continuation code from the file control screen, the file select type screen, the file select screens, or the alternate record keys screen. Note that you request the display of the I-O control screen only after you finish coding all the desired select clauses.

RDP code

If no user data is entered on this screen, no source lines are created. If you want to include more files in the SAME AREA or MULTIPLE FILE clauses, enter the code RDP in the continuation code field to redisplay this screen.

SCREEN 4-22 **I-O CONTROL APPLYS**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              I-O Control Applys                 Line nnnn.nnnn
 A   B
     [APPLY BLOCK-COUNT ON   <Enter X if Line is to be created>
                                        <filename,TAPES>]
         [                               <filename>]]
     [APPLY VERIFY ON   <Enter X if Line is to be created>
                                        <filename>]
         [                               <filename>]]
     [APPLY INDEX-AREA OF   <Enter X if Line is to be created>
                  <integer> CHARACTERS ON
                                        <filename>
         [                               <filename>]].
 Continuation Code (RDP)   [Next Screen is Data Division - FD]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

The apply screen provides the information to create APPLYs in the I-O control paragraph. This screen is displayed when COBEDT completes processing the I-O control screen.

RDP code
Multiple file names

If no user data is entered, no source lines are created. If you want to include more file names in any of the APPLY clauses, enter code RDP in the continuation code field to redisplay this screen. If multiple file names are used, each APPLY clause must be completed with all of its file names entered before the next APPLY clause can be coded.

Data Division Screens

SCREEN 4-23 DATA DIVISION – FD

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                     Data Division - File Section (FD)          Line nnnn.nnnn
 A   B
 FD                                 <file-name>
     [BLOCK (CONTAINS)          <integer> [TO          <integer>]
                 <RECORDS,CHARACTERS>]
     [RECORD (CONTAINS)     <integer> [TO      <integer>] (CHARACTERS)]
     LABEL RECORD(S) (ARE)                  <STANDARD,OMITTED,data-name>
     [VALUE OF [FILE-ID (IS)                                <data-name>]
         [PASSWORD (IS)                                  <data-name>]
     [CODE-SET (IS)                                 <alphabet name>].
 Continuation code (NRM) [Next Screen is FD Data Record Form]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       LIN = Linage Screen Required
   SDD = Current File is a Sort File-Display SD Screen
EDT Command:@
********************************************************************************
```

After the source lines for the desired SELECT clauses and the I-O CONTROL paragraph are created, COBEDT checks the temporary internal file name table created during the processing of the SELECT screens and then displays the FD screen for each of the file names listed with the file name placed in its FD file-name field.

SDD code

If the file is a sort file, you must not enter source data here but enter the SDD continuation code to request the SD screen (screen 4–26) for coding the SD source lines. If you want to code the LINAGE clause with the current FD, enter the LIN continuation code to display the linage screen (screen 4–24) for coding. In this case, the period displayed on the current FD screen is not generated.

LIN code

Period on the FD screen

First FD entry

In the ordered creation mode processing, if the current FD is the first entry in the file section, COBEDT first creates the DATA DIVISION and FILE SECTION source lines and then the source lines from the current FD screen.

If you temporarily enter either the EDT command mode or the selective creation mode from an FD screen without filling in any source data on the screen, upon returning to the ordered creation mode, the same screen is redisplayed. On the other hand, if you have filled in the source data, upon return, the FD data record form screen (screen 4–25) is displayed.

NOTE:

The internal table for the file names specified in SELECT clauses is created only in ordered creation mode processing and lasts for the duration of the current COBEDT session. If you request an FD screen in the selective creation mode, you must place the file name specified in a SELECT clause in the FD file-name field.

SCREEN 4-24 **LINAGE CLAUSE**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                        File Section - Linage                    Line nnnn.nnnn
 A   B
     [LINAGE (IS)                                  <data-name,integer> LINES
        [(WITH) FOOTING (AT)                          <data-name,integer>]
        [(LINES AT) TOP                               <data-name,integer>]
        [(LINES AT) BOTTOM                            <data-name,integer>]].




 Continuation Code (NRM)  [Next Screen is FD Data Record Form]
   NRM = Normal Continuation     SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode
EDT Command:@
********************************************************************************
```

LIN code

The linage clause screen provides the information to create the linage clause source lines. This screen is displayed when requested by the LIN continuation code from the FD screen.

All user data on the screen is optional. If no data is entered, no source lines are created.

SCREEN 4-25 **FD DATA RECORD FORM**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              FD Data Record Form                  Line nnnn.nnnn
 CIA   B                           41
  01                                 <data-record-name>
  |
  |
  |
  |
  |
  |
  |
  |
  Continuation Code (NRM)  [Next Screen varies]
   NRM = Normal Continuation          SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode       RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

The FD data record form screen provides you space to create a data record associated with an FD, SD, or CD. This screen is displayed after an FD, SD, or CD is coded and processed and can be redisplayed as many times as necessary to complete the record definition.

Data record name

The data record name must be coded the first time the screen is displayed. On subsequent displays, if the data record name field is left blank, the data on the screen is considered a continuation of the current 01 level.

When the normal continuation is indicated, the next screen to be displayed is either the FD screen (screen 4-23) when there are more files to be defined or the data division coding form screen (screen 4-36).

SCREEN 4-26

DATA DIVISION – SD

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                      Data Division - File Section (SD)          Line nnnn.nnnn
 A   B
 SD                                  <file-name>
     [RECORD (CONTAINS)            <integer>
                  [TO              <integer>] (CHARACTERS)].


 Continuation Code (   )  [Next Screen is FD Data Record Form]
   NRM = Normal Continuation       SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode    FDD = Display FD Screen
EDT Command:@
*******************************************************************************
```

SDD code

The SD screen provides the information to create the SD entry of a sort/merge file. This screen is displayed when requested by the SDD continuation code from the FD screen. In ordered creation mode processing, the name of the file is automatically displayed in the SD file-name field by COBEDT.

FDD code

If the file is not a sort file and you have requested the SD screen by mistake, you may return to the FD screen by specifying the FDD continuation code on the SD screen.

If you temporarily enter either the EDT command mode or the selective creation mode from an SD screen without filling in any source data on the screen, upon returning to the ordered creation mode, the same screen is redisplayed. On the other hand, if you have filled in the source data, upon return, the FD data record form screen is displayed.

SCREEN 4-27 **COMMUNICATION SECTION INPUT CD 1**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                  Communication Section Input CD (1 of 2)      Line nnnn.nnnn
 A   B
 CD                                <cd-name>
     (FOR) [       <INITIAL,spaces>] (INPUT)
      [(SYMBOLIC) QUEUE (IS)                                    <data-name>]
      [(SYMBOLIC) SUB-QUEUE-1 (IS)                              <data-name>]
      [(SYMBOLIC) SUB-QUEUE-2 (IS)                              <data-name>]
      [(SYMBOLIC) SUB-QUEUE-3 (IS)                              <data-name>]
      [MESSAGE DATE (IS)                                     <data-name>]
 Continuation Code (   )  [Next Screen is FD Data Record Form]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        NCD = Display Second Part of CD
   ICD = Display Input CD Screen       OCD = Display Output CD Screen
EDT Command:@
*******************************************************************************
```

ICD code

The input CD 1 screen provides the information to create the initial part of an input CD. This screen is displayed when requested by the ICD continuation code from the data division coding form screen (screen 4-36), the input CD 2 screen, or the output CD screen.

CD name missing

The user data on this screen is optional. If no data is entered, COBEDT doesn't create any source lines. If data is entered but the CD name is not included, COBEDT displays an error message.

First CD entry

If the current CD is the first entry in the communication section, COBEDT first creates the COMMUNICATION SECTION source line and then the source lines from the current screen.

SCREEN 4-28 **COMMUNICATION SECTION INPUT CD 2**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                  Communication Section Input CD (2 of 2)      Line nnnn.nnnn
 A   B
      [MESSAGE TIME (IS)                                        <data-name>]
      [(SYMBOLIC) SOURCE (IS)                                      <data-name>]
      [TEXT LENGTH (IS)                                      <data-name>]
      [END KEY (IS)                                       <data-name>]
      [STATUS KEY (IS)                                       <data-name>]
      [(MESSAGE) COUNT (IS)                                     <data-name>]
 Continuation Code (   ) [Next Screen is FD Data Record Form]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        ICD = Display Input CD Screen
   OCD = Display Output CD Screen
EDT Command:@
*******************************************************************************
```

NCD code

The input CD 2 screen provides the information to create the remainder of the input CD. This screen is displayed when requested by the NCD continuation code from the input CD 1 screen.

CD name missing

The user data on this screen is optional. If no data is entered, COBEDT doesn't create any source lines. If no data was entered on the input CD 1 screen but data is entered on this second one, COBEDT responds with an error message and redisplays the first input CD screen for you to add the CD name. If you don't add the name, the input CD source lines are created without the name.

SCREEN 4–29 **COMMUNICATION SECTION OUTPUT CD**

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                      Communication Section Output CD          Line nnnn.nnnn
 A   B
 CD                              <cd-name> (FOR) OUTPUT
     [DESTINATION COUNT (IS)                                  <data-name>]
     [TEXT LENGTH (IS)                                 <data-name>]
     [STATUS KEY (IS)                                  <data-name>]
     [DESTINATION TABLE OCCURS          <integer> TIMES
         [INDEXED BY                                   <index-name>
                                                       <index-name>]]
     [ERROR KEY (IS)                                   <data-name>]
     [SYMBOLIC DESTINATION (IS)                              <data-name>].
 Continuation Code (NRM) [Next Screen is FD Data Record Form]
   NRM = Normal Continuation           SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode        OCD = Display Output CD Screen
   ICD = Display Input CD Screen
EDT Command:@
*******************************************************************************
```

OCD code

The output CD screen is displayed when requested by the OCD continuation code from the input CD screens or the data division coding form screen (screen 4–36).

CD name missing

The user data on this screen is optional. If no data is entered, COBEDT doesn't create any source lines. If data is entered without the CD name, an error message is displayed.

If you want more than two index-names, enter the code OCD in the continuation code field to redisplay this screen.

SCREEN 4–30 **DATA ITEM DESCRIPTION**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Data Item Description                 Line nnnn.nnnn
  <level-number>                                    <data-name,FILLER>
   [REDEFINES                                     <data-name>]
   [PIC (IS)                                         <character string>]
   [USAGE (IS)          <COMP,COMP-n,DISPLAY,INDEX>]
   [[SIGN (IS)]           <LEADING,TRAILING>
      [SEPARATE (CHARACTER)]]   <Enter X for SEPARATE CHARACTER>
   [SYNCHRONIZED        <LEFT,RIGHT>]
   [JUSTIFIED RIGHT]    <Enter X if line is to be created>
   [BLANK (WHEN) ZERO]  <Enter X if line is to be created>
   [VALUE (IS)                                                  <literal>]
 Continuation Code (NRM)   [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation            RDP = Display This Screen Again
   CMD = Enter EDT Command Mode         SEL = Enter Selective Creation Mode
   OCC = Display OCCURS Screen
 EDT Command:@
*******************************************************************************
```

The data item description screen provides the information to create a data item in the data division. This screen is displayed when requested by the DAT continuation code from the OCCURS screen (screen 4–31), the KEY IS screen (screen 4–32), the INDEXED BY screen (screen 4–33), or the data division coding form screen (screen 4–36).

DAT code

All the data on this screen is optional. If no data is entered, no source lines are created.

SCREEN 4–31 **OCCURS CLAUSE**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Data Item OCCURS Clause               Line nnnn.nnnn
 [OCCURS       <integer> [TO      <integer>] (TIMES)
    [DEPENDING (ON)                                 <data-name>]]









 Continuation Code (NRM)   [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         KEY = Display KEY IS Screen
   DAT = Display Data Description       IDX = Display INDEXED BY Screen
EDT Command:@
*******************************************************************************
```

The OCCURS screen provides the information to create an OCCURS clause. This screen is displayed when requested by the OCC continuation code from the data item description screen.

OCC code

All the data on this screen is optional. If no data is entered, no source lines are created.

SCREEN 4-32 KEY IS CLAUSE

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Data Item KEY IS Clause                Line nnnn.nnnn
 [          <ASCENDING,DESCENDING> (KEY IS)
                                    <data-name>
                                    <data-name>
                                    <data-name>
                                    <data-name>
                                    <data-name>
                                    <data-name>
                                    <data-name>]
 Continuation Code (NRM)   [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           RDP = Display This Screen Again
   IDX = Display INDEXED BY Screen        DAT = Display Data Description
EDT Command:@
*******************************************************************************
```

KEY code

The KEY IS screen provides the information to create a KEY IS clause. This screen is displayed when requested by the KEY continuation code from the OCCURS screen.

All data on this screen is optional. If no data is entered, no source lines are created.

SCREEN 4-33 INDEXED BY CLAUSE

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Data Item INDEXED BY Clause              Line nnnn.nnnn
 [INDEXED (BY)   <Enter X if line is to be created>
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]
     [                            <index-name>]]
 Continuation Code (NRM)   [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           RDP = Display This Screen Again
   DAT = Display Data Description
EDT Command:@
*******************************************************************************
```

IDX code

The INDEXED BY screen provides the information to create an INDEXED BY clause. This screen is displayed when requested by the IDX continuation code from the OCCURS screen or the KEY IS screen.

All data on this screen is optional. If no data is entered, no source lines are created.

SCREEN 4-34 **RENAMES CLAUSE**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              RENAMES Data Item                  Line nnnn.nnnn
 A   B
   66                                  <data-name>
       RENAMES                                <data-name>
       [THRU                                 <data-name>]




 Continuation Code (    )    [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation             SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode          RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

REN code

The RENAMES screen provides the information to create a RENAMES clause. This screen is displayed when requested by the REN continuation code from the data division coding form screen (screen 4–36).

The clause starts in column 10

All user data on this screen is optional. If no data is entered, no source lines are created. If data is entered, COBEDT creates the source line with level number 66 starting in column 10 to make the RENAMES clause stand out from other level numbers.

SCREEN 4-35 **CONDITION NAME DATA ITEM**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                           Condition Name Data Item              Line nnnn.nnnn
 A   B
 88                                <condition-name>
     VALUE IS                                              <literal>
         [THRU                                              <literal>]
       [                                               <literal>
         [THRU                                               <literal>]]
       [                                               <literal>
         [THRU                                               <literal>]]
       [                                               <literal>
         [THRU                                               <literal>]]
 Continuation Code (    )    [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation             SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode          RDP = Display This Screen Again
EDT Command:@
*******************************************************************************
```

CND code

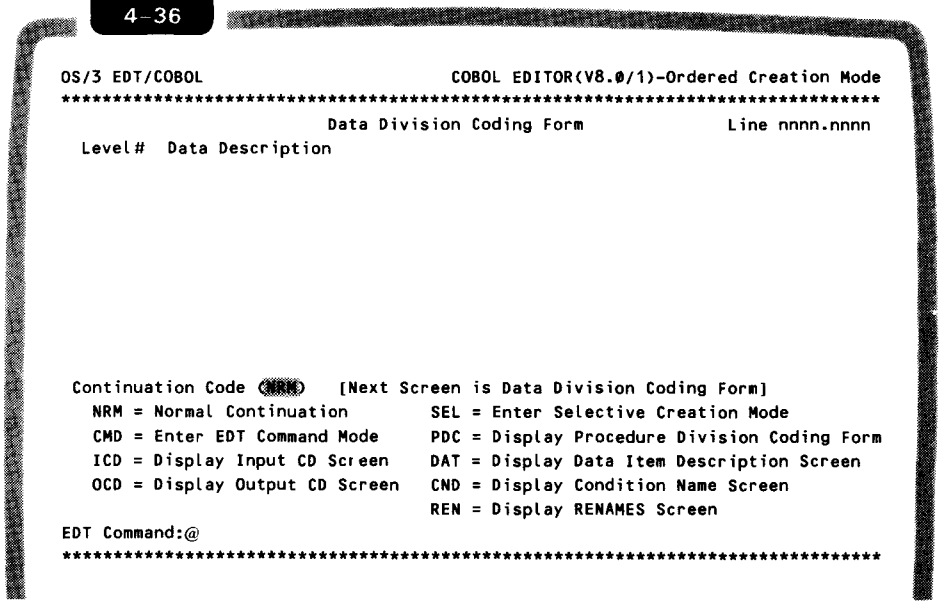
The condition name screen is displayed when requested by the CND continuation code from the data division coding form screen (screen 4–36).

Indentation of lines

All user data on this screen is optional. If no data is entered, no source lines are created. If data is entered, the source lines with level number 88 are created and indented two columns (character positions) from the beginning of the associated data description source line. For example:

```
Ø2  PAYROLL-PERIOD     PIC 9.
  88  WEEKLY           VALUE Ø.
  88  MONTHLY          VALUE 1.
```

SCREEN 4-36 **DATA DIVISION CODING FORM**



The data division coding form screen lets you create the source lines for the working-storage section by entering the level numbers and their associated data descriptions. This screen is displayed if you selected the NRM continuation code on any one of the following screens: file-control, FD data record form (if all the FD/SD files have been defined), data item description, OCCURS, KEY IS, INDEXED BY, RENAMES, or condition name. This screen is redisplayed until either COBEDT is terminated or a procedure division coding form screen is requested.

Level numbers 01 and 77
Level number 66

Other level numbers

When the screen is transmitted, COBEDT checks the level number of each source line. If it is 01 or 77, the source line is started at column 8. If it is 66, the source line is started at column 10. COBEDT indents source lines with level numbers other than 01, 77, or 66. Source lines at first level below level 01 are created with the level numbers starting at column 12. Subsequent source lines with new and greater level numbers are indented an additional two columns.

NOTE:

When you request the data division coding form screen in the selective creation mode, all source lines with level numbers other than 01, 66, or 77 begin in column 12.

In the ordered creation mode, after the data division coding form screen is first displayed and transmitted, COBEDT creates the DATA DIVISION source line (if it has not already been created), the WORKING-STORAGE SECTION source line, and then the source lines from this first screen.

From a data division coding form screen, you may request the data item description screen, the condition name screen, or the RENAMES screen to assist you in coding the working-storage section.

Coding the linkage section

To code the linkage section, you must enter the selective creation mode and key in your data on the standard COBOL coding form screen (screen 5–1). When you return to the ordered creation mode, another data division coding form screen is displayed.

Coding the communication section

To code the communication section, you must use the appropriate continuation code to display the desired CD screen for coding. After the communication section has been created, a data division coding form screen is redisplayed.

Calling a procedure division coding form screen

To display a procedure division coding form screen for coding the procedure division, enter code PDC in the continuation code field of the current data division coding form screen.

Procedure Division Screen

SCREEN 4-37 **PROCEDURE DIVISION CODING FORM**

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                          Procedure Divsion Coding Form          Line nnnn.nnnn
  C|A   B
   |
   |
   |
   |
   |
   |
   |
   |
   |
   |
 Continuation Code (NRM)   [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
*******************************************************************************
```

PDC code

This screen is initially displayed when requested by the PDC continuation code from the data division coding form screen. The first three lines of the form let you create margin A source lines. These source lines include the PROCEDURE DIVISION header, section headers, paragraph names, and the DECLARATIVES and END DECLARATIVES statements. You may use the entire form to create margin B source lines, which are the procedure division verb statements.

The PROCEDURE DIVISION source line and a paragraph name source line must be coded the first time the procedure division coding form screen is displayed.

NOTE:

Appendix A presents all the verb skeleton screens and verb skeleton display codes.

4.5. SUMMARY OF ORDERED CREATION MODE SCREENS AND THEIR DISPLAY CODES

Table 4–1 lists all the available ordered creation mode screens and their display codes. The screen numbers of the screens in Table 4–1 correspond to the screen numbers labeled in 4.4.

NOTE:

For display code NRM, the resulting screens listed in Table 4–1 are applicable only in ordered creation mode processing. In selective creation mode processing, display code NRM always causes a standard COBOL coding form screen to be displayed. See Section 5 for more information.

Table 4–1. Ordered Creation Mode Screen Summary (Part 1 of 3)

Screen No.	Screen Name	Display Code	Calling Screen
4–1	Control division	CON	Identification division
4–2	Identification division	NRM	First display in the ordered mode control division
4–3	Environment division	NRM	Identification division
4–5	Special-names screen 1	SN1	Environment division Special-names select* Other special-names screens
4–6	Special-names screen 2	SN2	Environment division Special-names select* Other special-names screens
4–7	Special-names SYSCHAN	SCH	Environment division Special-names select* Other special-names screens
4–8	Special-names alphabet-name	ALP	Environment division Special-names select* Other special-names screens
4–9	Special-names SYSSWCH	SSW	Environment division Special-names select* Other special-names screens
4–10	Special-names class-name	CLN	Environment division Special-names select* Other special-names screens
4–11	File control	NRM	Environment division Special-names select* Special-names screens

Table 4-1. Ordered Creation Mode Screen Summary (Part 2 of 3)

Screen No.	Screen Name	Display Code	Calling Screen
4-12	File select type	FST	Cardreader select Cardpunch select Printer select Tape select Sequential disk select Relative select Indexed select Alternate record keys
4-13	Cardreader select	CDR	File control File select type Other select screens Alternate record keys
4-14	Cardpunch select	CDP	File control File select type Other select screens Alternate record keys
4-15	Printer select	PRT	File control File select type Other select screens Alternate record keys
4-16	Tape select	TAP	File control File select type Other select screens Alternate record keys
4-17	Sequential disk select	DIS	File control File select type Other select screens Alternate record keys
4-18	Relative select	REL	File control File select type Other select screens Alternate record keys
4-19	Indexed select	INX	File control File select type Other select screens Alternate record keys
4-20	Alternate record keys	ALK	Indexed select
4-21	I-O control	IOC	File control File select type Select screens Alternate record keys
4-22	I-O control apply	NRM	I-O control

Table 4–1. Ordered Creation Mode Screen Summary (Part 3 of 3)

Screen No.	Screen Name	Display Code	Calling Screen
4–23	FD	NRM	File select type Select screens Alternate record keys I-O control apply
4–24	LINAGE	LIN	FD
4–25	FD data record form	NRM	FD LINAGE SD Input CD Output CD
4–26	SD	SDD	FD
4–27	Input CD 1	ICD	Data division coding form Input CD 2 Output CD
4–28	Input CD 2	NCD	Input CD 1
4–29	Output CD	OCD	Input CD 1 Input CD 2 Data division coding form
4–30	Data item description	DAT	OCCURS KEY IS INDEXED BY Data division coding form
4–31	OCCURS	OCC	Data item description
4–32	KEY IS	KEY	OCCURS
4–33	INDEXED BY	IDX	OCCURS KEY IS
4–34	RENAMES	REN	Data division coding form
4–35	Condition name	CND	Data division coding form
4–36	Data division coding form	NRM	File control FD data record form** Data item description OCCURS KEY IS INDEXED BY RENAMES Condition name Data division coding form
4–37	Procedure division coding form	PDC NRM	Data division coding form Procedure division coding form

* COBEDT automatically display the special-names select screen under certain conditions. See screen 4–4 in 4.4 for an explanation.

** Transmission of the FD data record form screen with NRM code causes the display of a data division coding form screen only if all the FD/SD files have been defined. See screens 4–25 and 4–36 in 4.4 for more information.

5. Creating COBOL Source Programs in Selective Creation Mode

5.1. OPERATING IN SELECTIVE CREATION MODE

When to use the selective creation mode

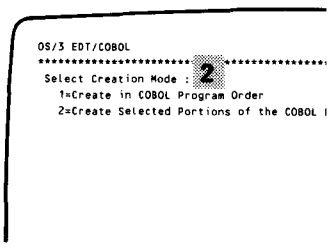
If you're an experienced COBOL user, you may wish to use the selective creation mode, instead of the ordered creation mode, to assist in coding your COBOL source program. In the selective creation mode, COBEDT lets you create different portions of your program in any order you choose. COBEDT normally displays the standard COBOL coding form screens for you to enter source data.

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line nnnn.nnnn
 CIA   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (NRM)   [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation        TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode     sss = Display Creation Screen sss
   RET = Return to Ordered Mode     vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

All other creation screens (for example, the ordered creation mode screens and the COBOL verb skeletons) are available upon request.

Entering the selective creation mode

To create your COBOL program via the selective creation mode, you must keep option 2 in the select creation mode field of the option select screen. Once this screen is processed, a standard COBOL coding form screen appears, and you may enter your source data.



No syntax checking

COBEDT does not check the syntax of the data entered on a standard COBOL coding form screen.

Work-space line numbers

If you create your program in other than the standard COBOL program order, you must make certain that the current position in the work-space file reflected by the line number displayed on the screen is where you want COBEDT to place the source lines

Changing line numbers

created. Do this either by directly changing the line number displayed on the current screen or by getting into the EDT command mode and then using the @ command to reset the line number. COBEDT doesn't check whether any existing lines are overwritten by the lines being created.

Using the ordered creation mode screens

You may at any time request that one of the ordered creation mode screens be displayed. You may then use it to create that portion of the program and change its line number if desired.

Checking the syntax

COBEDT checks to ensure that all required data for that screen is entered. If required data is missing, upon transmission, an error message is displayed and the location of the error is indicated by a blinking field. To correct the error, reposition the cursor to the beginning of the blinking field (if it does not automatically reposition itself to the error field) and key in the required data. If you choose not to correct it, simply retransmit the screen. The source lines are then written to the work-space file as they are, and no additional action is taken. (For more information on the ordered creation mode, see Section 4.)

Adopting other selective creation mode screens

You may also request that other selective creation mode screens be displayed to aid in your coding. These screens include COBOL verb skeletons, the COBOL program skeleton, and the creation mode screen list (5.2).

NOTE:

In selective creation mode processing, when you transmit an ordered or a selective creation mode screen with NRM in its continuation code field, the next screen displayed is always a standard COBOL coding form screen.

Entering EDT command mode

Further, you may enter EDT command mode at any time by entering code CMD in the continuation code field of the current screen. Upon return, COBEDT resumes selective creation mode processing and displays a standard COBOL coding form screen.

5.2. SELECTIVE CREATION MODE SCREENS

The selective creation mode screens include the standard COBOL coding form screen, the COBOL program skeleton screen, the creation mode display screen list screen, and the procedure division verb skeleton screens. Appendix A describes the verb skeleton screens. The rest of the screens are presented here.

Standard COBOL Coding Form Screen

SCREEN 5-1 STANDARD COBOL CODING FORM

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line nnnn.nnnn
 CIA   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (NRM)   [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation        TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode     sss = Display Creation Screen sss
   RET = Return to Ordered Mode     vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

The standard COBOL coding form screen is the first screen displayed under the selective creation mode. You may specify any one of the continuation codes indicated on the screen and/or enter EDT commands without entering any source lines. You may also change the line number displayed at any time.

RET code

If the screen is displayed through temporarily entering the selective creation mode from the ordered creation mode, overwriting default continuation code NRM with RET returns you to ordered mode processing.

Creation Mode Display Screen List Screen

SCREEN 5.2 **CREATION MODE DISPLAY SCREEN LIST**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                      Creation Mode Display Screen List

 CON=Control Division           CLN=Class Names              IOC=I-O Control
 IDE=Identification Div.        FIL=File Control             APP=Applys
 ENV=Environment Division       CDR=Cardreader Select        FDD=FD Display
 SN1=Special-Names 1            CDP=Cardpunch Select         LIN=Linage
 SN2=Special-Names 2            PRT=Printer Select           SDD=SD Display
 SCH=SYSCHAN                    TAP=Tape Select              ICD=Input CD
 ALP=Alphabet Name              DIS=Sequential Disk Select   OCD=Output CD
 SSW=SYSSWCH                    REL=Relative Select          FDR=FD Data Rcd
 DAT=Data Item Description      INX=Indexed Select           REN=Renames Desc.
 OCC=Occurs                     CND=Condition-Name Desc.     SKE=Program Skel.
 DDC=Data Division Coding Form  IDX=INDEXED BY               KEY=KEY IS
 PDC=Procedure Division Coding Form
 Display Screen NRM
   enter screen code to display the desired screen
   default NRM indicates no screen display and return to normal processing
EDT Command:@
*******************************************************************************
```

Choosing your screen

This screen lists all the available creation mode screens and their display codes. To display one of these screens, overwrite NRM in the display screen field with the corresponding code. Otherwise, COBEDT takes default value NRM and, upon transmission, displays a standard COBOL coding form screen.

COBOL Program Skeleton Screens

The COBOL program skeleton describes the divisions and sections that make up a COBOL program. The skeleton is divided into two screens. The first screen includes those portions of a program from the control division to the environment divison (screen 5-3), whereas the second one includes the remainder of the program from the data division to the procedure division (screen 5-4).

SCREEN 5-3

CONTROL, IDENTIFICATION, AND ENVIRONMENT DIVISION SKELETON

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Selective Creation Mode
********************************************************************************
       COBOL Program Skeleton CONTROL to ENVIRONMENT DIVISIONS (1 of 2)
 A   B
 [CONTROL DIVISION.
  ALPHABET SECTION.]
 IDENTIFICATION DIVISION.
 PROGRAM-ID. program-name.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3.
 OBJECT-COMPUTER. UNIVAC-OS3.
 [SPECIAL-NAMES. special-names entry.]
 [INPUT-OUTPUT SECTION.
  FILE-CONTROL.
  [I-O-CONTROL.]]
 Continuation Code (***) [Next Screen is Part 2 of Program Skeleton]
EDT Command:@
********************************************************************************
```

SCREEN 5-4

DATA AND PROCEDURE DIVISION SKELETON

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Selective Creation Mode
********************************************************************************
       COBOL Program Skeleton DATA to PROCEDURE DIVISIONS (2 of 2)
 A   B
 DATA DIVISION.
 [FILE SECTION.]
 [WORKING-STORAGE SECTION.]
 [LINKAGE SECTION.]
 [COMMUNICATION SECTION.]
 PROCEDURE DIVISION.
 [DECLARATIVES.
  {section-name SECTION [segment-number]. declaratives-sentence
   [paragraph-name. [sentence]...]...}
  END DECLARATIVES.
 {section-name SECTION [segment-number].
  [paragraph-name. [sentence]...]...}
 Continuation Code (***) [Next Screen is Standard COBOL Coding Form]
EDT Command:@
********************************************************************************
```

The next screen displayed is a standard COBOL coding form screen.

5.3. SAMPLE CREATION SESSION

This sample session shows how to create a COBOL source program in the standard COBOL program order via the selective creation mode. The example starts with choosing the creation mode on the option select screen. We then code the source statements with the standard COBOL coding form screens and several other creation screens. The example ends with writing the program to a permanent SAT file. The program created in this example is identical to the one in Section 4.

SCREEN 1 **OPTION SELECT**

```
OS/3 EDT/COBOL                                              COBOL EDITOR(V8.0/1)
*******************************************************************************
 Select Creation Mode : (2)

   1=Create in COBOL Program Order
   2=Create Selected Portions of the COBOL Program




 Continuation Code (NRM)
   NRM = Normal Continuation      CMD = Enter EDT Command Mode
EDT Command:@
*******************************************************************************
```

To use the standard COBOL coding screens, we select 2 for the creation mode and NRM for normal continuation on the option select screen. Upon transmission, a standard COBOL coding form screen is displayed with current work-space line number 1.

SCREEN 2 **STANDARD COBOL CODING FORM**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 1
 CIA   B
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PROG1.
 AUTHOR. J.JONES.
 INSTALLATION. UNIVAC.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3.
 OBJECT-COMPUTER. UNIVAC-OS3.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 Continuation Code (INX)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

On this screen, we key in our COBOL source statements. Because we need assistance in coding the next part of the program, the INDEXED SELECT clause, we key in creation screen request code INX in the continuation code field to display the indexed select screen. Upon transmission, the source lines are created and placed in the work-space file, and then the requested screen is displayed.

SOURCE LINES CREATED

```
1.0000      IDENTIFICATION DIVISION.
2.0000      PROGRAM-ID. PROG1.
3.0000      AUTHOR. J.JONES.
4.0000      INSTALLATION. UNIVAC.
5.0000      ENVIRONMENT DIVISION.
6.0000      CONFIGURATION SECTION.
7.0000      SOURCE-COMPUTER. UNIVAC-OS3.
8.0000      OBJECT-COMPUTER. UNIVAC-OS3.
9.0000      INPUT-OUTPUT SECTION.
10.0000     FILE-CONTROL.
```

SCREEN 3 **INDEXED SELECT**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Indexed Select                      Line 11

 A   B
     SELECT
         TRANFIL                          <file-name>
         ASSIGN (TO) DISC-TRANFIL <lfdname>-V<F,V>
         [RESERVE   <1,2> AREA(S)]
         ORGANIZATION (IS) INDEXED
         [ACCESS (MODE IS) SEQUENTIAL<SEQUENTIAL,RANDOM,DYNAMIC>]
         RECORD (KEY IS) TRAN-KEY                    <data-name>
         [(FILE) STATUS (IS)                            <data-name>].

  Continuation Code (NRM)  [Next Screen is Data Division-FD]
    NRM = Normal Continuation          SEL = Enter Selective Creation Mode
    CMD = Enter EDT Command Mode       FST = Display File Select Type Codes
    fff = Next File Select Code        ALK = Alternate Record Keys Required
EDT Command:@
*******************************************************************************
```

Here, we fill in our data and select default continuation code NRM. Upon transmission, a standard COBOL coding form screen is displayed, with current work-space line number 16, after the source lines are created and placed in the work-space file.

SOURCE LINES CREATED

```
11.0000          SELECT TRANFIL
12.0000              ASSIGN TO DISC-TRANFIL-V
13.0000              ORGANIZATION IS INDEXED
14.0000              ACCESS MODE IS SEQUENTIAL
15.0000              RECORD KEY IS TRAN-KEY.
```

On the following four standard COBOL coding form screens, we enter our source statements and select default continuation code NRM. Upon transmission, COBEDT creates the source lines as shown after each screen.

SCREEN 4 **STANDARD COBOL CODING FORM 1**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 16
 C|A   B
  |
  | DATA DIVISION.
  | FILE SECTION.
  | FD  TRANFIL
  |     LABEL RECORDS ARE STANDARD.
  |
  |
  |
  |
  |
  |
 Continuation Code (NRM)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

SOURCE LINES CREATED

```
16.0000       DATA DIVISION.
17.0000       FILE SECTION.
18.0000       FD  TRANFIL
19.0000           LABEL RECORDS ARE STANDARD.
```

SCREEN 5 STANDARD COBOL CODING FORM 2

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 20.

 CIA   B
 01  TRAN-OUT.
     03  TRAN-KEY.
       05  TRAN-PUB          PIC XX.
       05  TRAN-REPORT       PIC 99.
       05  TRAN-SORT         PIC X(23).
       05  TRAN-SEQUENCE     PIC 9(4).
     03  TRAN-VARIABLE       PIC X
         OCCURS 1 TO 100 TIMES
         DEPENDING ON SYS-REC-SIZE.

 Continuation Code (NRM)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

SOURCE LINES CREATED

```
20.0000        01  TRAN-OUT.
21.0000            03  TRAN-KEY.
22.0000              05  TRAN-PUB          PIC XX.
23.0000              05  TRAN-REPORT       PIC 99.
24.0000              05  TRAN-SORT         PIC X(23).
25.0000              05  TRAN-SEQUENCE     PIC 9(4).
26.0000            03  TRAN-VARIABLE       PIC X
27.0000                OCCURS 1 TO 100 TIMES
28.0000                DEPENDING ON SYS-REC-SIZE.
```

SCREEN 6 **STANDARD COBOL CODING FORM 3**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line [    ]

 ClA   B
 WORKING-STORAGE SECTION.
 01  DUMMY-OUT.
     03  DUMMY-KEY.
       05  DUMMY-PUB           PIC XX.
       05  DUMMY-REPORT        PIC 99.
       05  DUMMY-SORT          PIC X(23).
       05  DUMMY-SEQUENCE      PIC 9(4).
     03  DUMMY-VARIABLE     PIC X(100).
 77  SYS-REC-SIZE    PIC 9(4) VALUE 0.

 Continuation Code [    ]  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv

EDT Command:@
*******************************************************************************
```

SOURCE LINES CREATED

```
29.0000       WORKING-STORAGE SECTION.
30.0000       01  DUMMY-OUT.
31.0000           03  DUMMY-KEY.
32.0000             05  DUMMY-PUB           PIC XX.
33.0000             05  DUMMY-REPORT        PIC 99.
34.0000             05  DUMMY-SORT          PIC X(23).
35.0000             05  DUMMY-SEQUENCE      PIC 9(4).
36.0000           03  DUMMY-VARIABLE     PIC X(100).
37.0000       77  SYS-REC-SIZE    PIC 9(4) VALUE 0.
```

SCREEN 7 **STANDARD COBOL CODING FORM 4**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 38
 CA   B
 PROCEDURE DIVISION.
 PRIMARY-PARAGRAPH.
     OPEN OUTPUT TRANFIL.
     MOVE "AA" TO DUMMY-PUB.
     MOVE 99 TO DUMMY-REPORT.
     MOVE ALL "B" TO DUMMY-SORT.
     MOVE 1 TO DUMMY-SEQUENCE.
     MOVE ALL "C" TO DUMMY-VARIABLE.
 |
 |
 Continuation Code (NRM)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

SOURCE LINES CREATED

```
38.0000       PROCEDURE DIVISION.
39.0000       PRIMARY-PARAGRAPH.
40.0000           OPEN OUTPUT TRANFIL.
41.0000           MOVE "AA" TO DUMMY-PUB.
42.0000           MOVE 99 TO DUMMY-REPORT.
43.0000           MOVE ALL "B" TO DUMMY-SORT.
44.0000           MOVE 1 TO DUMMY-SEQUENCE.
45.0000           MOVE ALL "C" TO DUMMY-VARIABLE.
```

After COBEDT has placed these source lines in the work-space file, it displays another standard COBOL coding form screen with current work-space line number 46.

SCREEN 8 STANDARD COBOL CODING FORM

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
********************************************************************************
                          Standard COBOL Coding Form          Line 46
 CIA   B
 |
 LOOP-HERE.
     ADD 1 TO SYS-REC-SIZE.
     MOVE DUMMY-OUT TO TRAN-OUT.
     WRITE TRAN-OUT INVALID KEY
 |
 |
 |
 |
 |
 Continuation Code (DISPLAY)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
********************************************************************************
```

SOURCE LINES CREATED

Here, we enter more source statements. Because we need assistance in coding the next verb DISPLAY, we overwrite default code NRM with DISPLAY to request the DISPLAY verb skeleton. Upon transmission, the skeleton screen appears, with current work-space line number 50, after the source lines are created and placed in the work-space file.

```
46.0000       LOOP-HERE.
47.0000           ADD 1 TO SYS-REC-SIZE.
48.0000           MOVE DUMMY-OUT TO TRAN-OUT.
49.0000           WRITE TRAN-OUT INVALID KEY
```

SCREEN 9 **DISPLAY STATEMENT**

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                              DISPLAY Statement
 DISPLAY {identifier-1,literal-1} [,{identifier-2,literal-2}] ...
     [UPON mnemonic-name [USING {identifier-3,literal-3}]]
*******************************************************************************
                          Procedure Division Coding Form          Line 50.
 CIA   B
  |     DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
  |   IF SYS-REC-SIZE < 100
  |     ADD 1 TO DUMMY-SEQUENCE
  |     GO TO LOOP-HERE.
  |
 Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
*******************************************************************************
```

SOURCE LINES CREATED

On this screen, we code the DISPLAY verb and more procedure division source statements and select default continuation code NRM. Upon transmission, a standard COBOL coding form screen appears, with current work-space line number 54, after the source lines are created and placed in the work-space file.

```
50.0000              DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
51.0000            IF SYS-REC-SIZE < 100
52.0000              ADD 1 TO DUMMY-SEQUENCE
53.0000              GO TO LOOP-HERE.
```

SCREEN 10 STANDARD COBOL CODING FORM

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 54
 CIA   B
 END-IT.
     CLOSE TRANFIL.
     STOP RUN.
 |
 |
 |
 |
 |
 |
 |
 Continuation Code (CMD)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

Now we enter the last paragraph for our program and overwrite default continuation code NRM with CMD. Upon transmission, COBEDT creates the source lines and places them in the work-space file. COBEDT then enters the EDT command mode, and the next work-space line number (57.0000) appears on the workstation screen by itself.

SOURCE LINES CREATED

```
54.0000        END-IT.
55.0000            CLOSE TRANFIL.
56.0000            STOP RUN.
```

At this point, the creation of the program is complete. We key in the @PRINT command to see what our entire program looks like in the work-space file before terminating COBEDT.

```
57.0000▷@PRINT
 1.0000       IDENTIFICATION DIVISION.
 2.0000       PROGRAM-ID. PROG1.
 3.0000       AUTHOR. J.JONES.
 4.0000       INSTALLATION. UNIVAC.
 5.0000       ENVIRONMENT DIVISION.
 6.0000       CONFIGURATION SECTION.
 7.0000       SOURCE-COMPUTER. UNIVAC-OS3.
 8.0000       OBJECT-COMPUTER. UNIVAC-OS3.
 9.0000       INPUT-OUTPUT SECTION.
10.0000       FILE-CONTROL.
11.0000           SELECT TRANFIL
12.0000               ASSIGN TO DISC-TRANFIL-V
13.0000               ORGANIZATION IS INDEXED
14.0000               ACCESS MODE IS SEQUENTIAL
15.0000               RECORD KEY IS TRAN-KEY.
16.0000       DATA DIVISION.
17.0000       FILE SECTION.
18.0000       FD  TRANFIL
19.0000           LABEL RECORDS ARE STANDARD.
20.0000       01  TRAN-OUT.
21.0000           03  TRAN-KEY.
22.0000             05  TRAN-PUB          PIC XX.
23.0000             05  TRAN-REPORT       PIC 99.
24.0000             05  TRAN-SORT         PIC X(23).
25.0000             05  TRAN-SEQUENCE     PIC 9(4).
26.0000           03  TRAN-VARIABLE       PIC X
27.0000               OCCURS 1 TO 100 TIMES
28.0000               DEPENDING ON SYS-REC-SIZE.
29.0000       WORKING-STORAGE SECTION.
30.0000       01  DUMMY-OUT.
31.0000           03  DUMMY-KEY.
32.0000             05  DUMMY-PUB           PIC XX.
33.0000             05  DUMMY-REPORT        PIC 99.
34.0000             05  DUMMY-SORT          PIC X(23).
35.0000             05  DUMMY-SEQUENCE      PIC 9(4).
36.0000           03  DUMMY-VARIABLE      PIC X(100).
37.0000       77  SYS-REC-SIZE    PIC 9(4) VALUE 0.
```

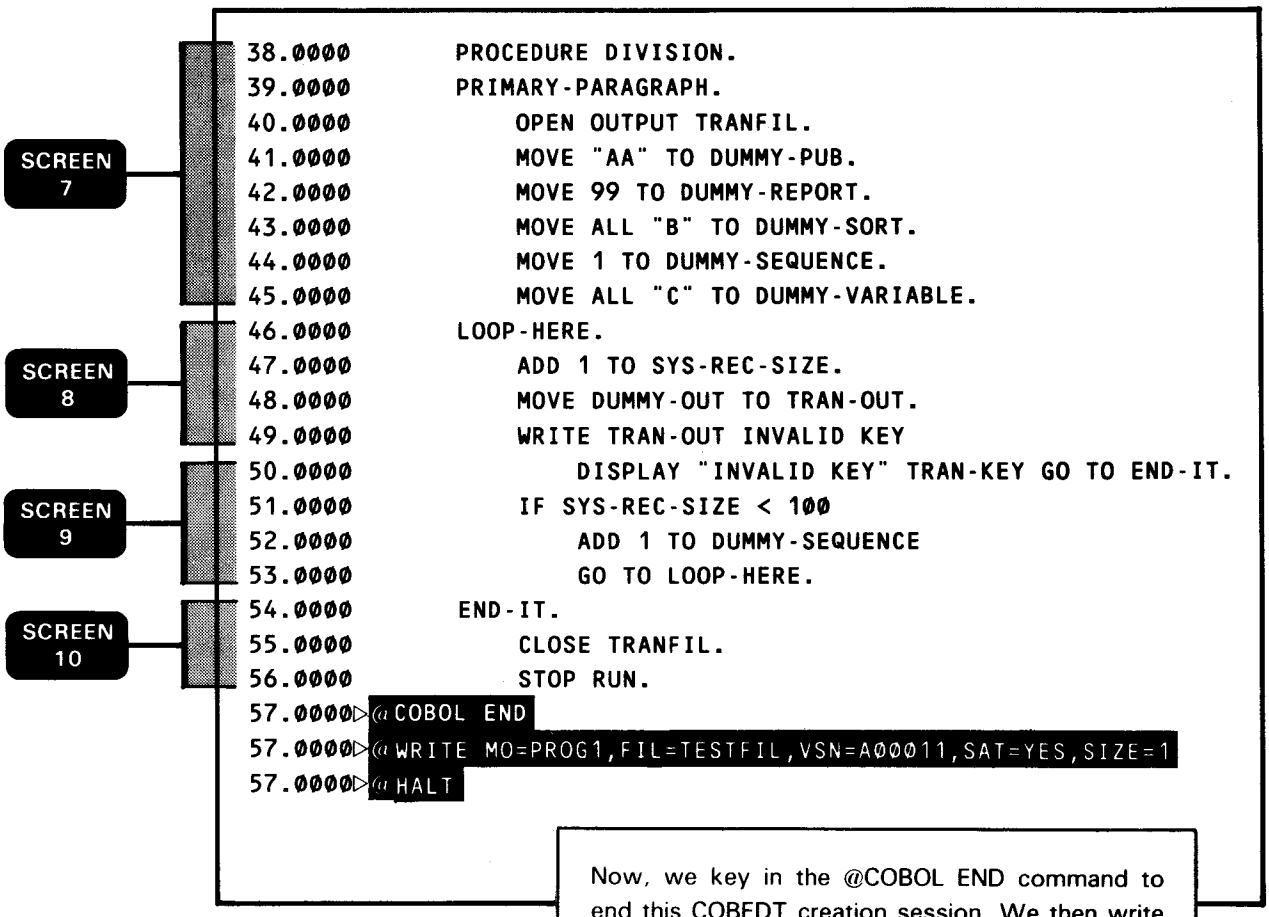
SCREEN 2

SCREEN 3

SCREEN 4

SCREEN 5

SCREEN 6



Now, we key in the @COBOL END command to end this COBEDT creation session. We then write our program (PROG1) to a permanent SAT file (TESTFIL) on disk volume A00011. Finally, we terminate this EDT session with the @HALT command.

NOTE:

For information on program compilation, see 4.3.

6. Editing COBOL Source Programs

Know your EDT commands

You may either edit a COBOL source program while creating it or edit an existing COBOL program. The commands of general editor EDT play important roles in the program editing process. Therefore, you should understand the function of these commands and familiarize yourself with their use to successfully modify your programs. For your reference, Appendix B summarizes the EDT commands. If you need a detailed explanation of these commands, see the general editor (EDT) user guide/programmer reference, UP-8828 (current version).

Editing an existing program

@READ command

To edit an existing COBOL source program, you may use either COBEDT or general editor EDT. Since EDT does not perform syntax checking or provide assistance through program creation screens as COBEDT, it is usually used for making minor changes. In both cases, however, you must first read a copy of the program into the work-space file by entering the EDT @READ command immediately after calling the editor. Thereafter, the methods of editing an existing program under COBEDT are essentially identical to editing a program during its creation.

Editing a program during creation

During the creation of your COBOL program in the ordered or selective mode, to edit your source lines already entered, you may:

- key in the appropriate EDT commands directly in the EDT command field of the current screen; or
- specify the CMD continuation code on that screen to enter the EDT command mode and then key in EDT commands there.

You may also request the display of creation screens to assist in editing your program.

6.1. SAMPLE SESSION FOR EDITING A PROGRAM DURING ITS CREATION

Purpose of this session

This example shows how to edit your COBOL program while you're still in the process of creating it in the ordered creation mode of COBEDT. The program used here is based on the one created in Section 4.

Let's begin this example from the identification division screen.

SCREEN 1

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                          Identification Division                Line 1
 A   B
 IDENTIFICATION DIVISION.   <Enter X if Line not to be created in Select Mode>
 PROGRAM-ID. PROG1
 [AUTHOR. J.JONES                                                          .]
 [INSTALLATION. UNIVAC                                                     .]
 [DATE-WRITTEN.                                                            .]
 [DATE-COMPILED.                                                           .]
 [SECURITY.                                                                .]



 Continuation Code (NRM)  [Next Screen is Environment Division]
   NRM = Normal Continuation       SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode    CON = Display Control Division Screen
EDT Command:@PRINT
********************************************************************************
```

Here, we enter three source statements for the identification division and then the @PRINT command, which requests the display of the work-space file. We then press the XMIT key and the source lines are displayed.

SOURCE LINES CREATED

```
1.0000      IDENTIFICATION DIVISION.
2.0000      PROGRAM-ID. PROG1.
3.0000      AUTHOR. J.JONES.
4.0000      INSTALLATION. UNIVAC.
```

SCREEN 2

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Environment Division                Line 5
 A   B
 ENVIRONMENT DIVISION. X <Enter X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].   <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                <alphabet-name>]
     [SEGMENT-LIMIT IS   <segment-number>].
 Continuation Code (SEL)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen         SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen   SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@
*******************************************************************************
```

After we press the XMIT key again to continue COBEDT processing, the environment division screen appears, with current work-space line number 5. At this point, however, we decide to add more source statements to the identification division. Therefore, we enter X in the ENVIRONMENT DIVISION line and overwrite continuation code NRM with SEL.

When this environment division screen is transmitted, COBEDT enters the selective creation mode and displays a standard COBOL coding form screen with current work-space line number 5.

SCREEN 3

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 4.1
 CIA   B
  |
  | DATE-WRITTEN.   DECEMBER 6, 1981.
  | DATE-COMPILED.  DECEMBER 6, 1981.
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (RET)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

On this screen, we enter two additional identification division source lines, overwrite line number 5 with 4.1 to avoid these two lines being overwritten, and overwrite continuation code NRM with RET. After we transmit this screen, COBEDT returns to the ordered creation mode and redisplays the environment division screen. The current work-space line number displayed on the screen is still 5 as we exited, since in ordered creation mode processing the line numbers displayed on the ordered mode screens can't be changed.

SCREEN 4

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Environment Division                  Line
 A   B
 ENVIRONMENT DIVISION.   <Enter X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].   <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                  <alphabet-name>]
     [SEGMENT-LIMIT IS   <segment-number>].
 Continuation Code (NRM)  [Next Screen is File Control]
   NRM = Normal Continuation            SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode         SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen         SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen   SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@PRINT
*******************************************************************************
```

SOURCE LINES CREATED UP TO THIS TIME FROM SCREEN 1

Here, we simply enter the @PRINT command in the EDT command field and then press the XMIT key. The source lines are then displayed. We press the XMIT key again to continue creating our source program.

```
1.0000       IDENTIFICATION DIVISION.
2.0000       PROGRAM-ID. PROG1.
3.0000       AUTHOR. J.JONES.
4.0000       INSTALLATION. UNIVAC.
4.1000       DATE-WRITTEN.   DECEMBER 6, 1981.
4.2000       DATE-COMPILED.  DECEMBER 6, 1981.
5.0000       ENVIRONMENT DIVISION.
6.0000       CONFIGURATION SECTION.
7.0000       SOURCE-COMPUTER. UNIVAC-OS3.
8.0000       OBJECT-COMPUTER. UNIVAC-OS3.
```

Assume we've reached the end of the program and entered the last paragraph for it.

SCREEN 5

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Procedure Division Coding Form              Line 56.
  CÍA   B
  END-IT.

      CLOSE TRANFIL.
      STOP RUN.

 Continuation Code (CMD>  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation       SEL = Enter Selective Creation Mode
   CMD = ENTER EDT COMMAND MODE    vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
*******************************************************************************
```

EDT COMMAND MODE ENTERED

We overwrite continuation code NRM with CMD to enter the EDT command mode. After the screen is processed, the next work-space line number (56.0000) by itself is displayed, ready to accept the EDT commands.

```
56.0000▷@PRINT 38:52
38.0000        PROCEDURE DIVISION.
39.0000        PRIMARY-PARAGRAPH.
40.0000            OPEN OUTPUT TRANFIL.
41.0000            MOVE "AA" TO DUMMY-PUB.
42.0000            MOVE 99 TO DUMMY-REPORT.
43.0000            MOVE ALL "B" TO DUMMY-SORT.
44.0000            MOVE 1 TO DUMMY-SEQUENCE.
45.0000            MOVE ALL "C" TO DUMMY-VARIABLE.
46.0000        LOOP-HERE.
47.0000            MOVE DUMMY-OUT TO TRAN-OUT.
48.0000            WRITE TRAN-OUT INVALID KEY
49.0000                ADD 1 TO DUMMY-SEQUENCE
50.0000                DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
51.0000            IF SYS-REC-SIZE < 100
52.0000                GO TO LOOP-HERE.
56.0000▷@FORMAT
```

To examine the source lines 38 through 52, we enter the @PRINT command. After these lines are displayed and checked, we enter the @FORMAT command to return to COBEDT at a procedure division coding form screen with line number 56. We will make further changes to the program via this screen.

SCREEN 6

```
OS/3 EDT/COBOL                        COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                        Procedure Division Coding Form            Line 46.1
  CA   B
   |
   |
   |
   |
   |    ADD 1 TO SYS-REC-SIZE.
   |
   |
   |
   |
   |
 Continuation Code (CMD)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@MOVE 49 TO 51.1
*******************************************************************************
```

EDT COMMAND MODE ENTERED

Because we decide to insert a new source line between lines 46 and 47 and to move line 49 to 51.1, we overwrite line number 56 with 46.1 on the screen and enter the new source line and the @MOVE command. We also overwrite continuation code NRM with CMD. Upon transmission, the next work-space line number (46.2000) by itself is displayed.

```
46.2000▷@PRINT 45:55
45.0000           MOVE ALL "C" TO DUMMY-VARIABLE.
46.0000       LOOP-HERE.
46.1000           ADD 1 TO SYS-REC-SIZE.
47.0000           MOVE DUMMY-OUT TO TRAN-OUT.
48.0000           WRITE TRAN-OUT INVALID KEY
50.0000               DISPLAY "INVALID KEY" TRAN-KEY GO TO END-IT.
51.0000           IF SYS-REC-SIZE < 100
51.1000               ADD 1 TO DUMMY-SEQUENCE
52.0000               GO TO LOOP-HERE.
53.0000       END-IT.
54.0000           CLOSE TRANFIL.
55.0000           STOP RUN.
46.2000▷@WRITE MO=PROG1,FIL=TESTFIL,VSN=A00011,SAT=YES,SIZE=1
46.2000▷@COBOL END
```

Here, we first enter the @PRINT command to see the source lines from 45 through 55. Since we've completed creating our program and made the desired changes, we then enter the @WRITE command to write our program to a permanent SAT file and the @COBOL END command to terminate the current COBEDT session.

6.2. SAMPLE SESSION FOR EDITING A PROGRAM VIA THE ORDERED CREATION MODE OF COBEDT

Purpose of this session

This example shows how to edit an existing COBOL source program via the ordered creation mode of COBEDT.

Let's start with a program named PROG2 that contains only the procedure division source lines. We want to add to the program the source lines for other divisions in the COBOL program order to make it a complete program.

After you call both EDT and COBEDT, the option select screen is displayed.

SCREEN 1

```
OS/3 EDT/COBOL                                              COBOL EDITOR(V8.0/1)
*******************************************************************************
 Select Creation Mode : (1)
   1=Create in COBOL Program Order
   2=Create Selected Portions of the COBOL Program




 Continuation Code (CMD)
   NRM=Normal Continuation      CMD=Enter EDT Command Mode
EDT Command:@
*******************************************************************************
```

EDT COMMAND MODE ENTERED

For this sample session, we enter 1 in the select creation mode field and CMD in the continuation code field. The next screen displayed contains only the current work-space line number. Since this is a new COBEDT session, the line number is 1.0000.

```
  1.0000▷@READ MO=PROG2,FIL=TEST2,VSN=A00022,SAT=YES
101.0000▷@MOVE  1:100 TO 1000
101.0000▷@1
  1.0000▷@FORMAT
```

@READ command

@MOVE command

@ command

@FORMAT command

Here, we enter the @READ command to transfer a copy of our program (PROG2) from the permanent SAT file (TEST2) into the work-space file. The next work-space line number displayed is 101.0000, since there are 100 lines in PROG2. To ensure that these lines aren't overwritten by future entries, we use the @MOVE command to move them to a new line location, leaving enough space for the new lines. We then enter the @ command to set the current work-space line number back to 1.

Entering the @FORMAT command returns us to COBEDT and the identification division screen with work-space line number 1. From here on, we create the remaining sections of our program in the usual manner.

Assume we've completed editing our program and 400 additional source lines have been created and placed in the work-space file. A data division coding form screen is displayed, with next work-space line number 401.

SCREEN 2

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
********************************************************************************
                          Data Division Coding Form              Line 401
  Level#  Data Description

 Continuation Code (CMD)  [Next Screen is Data Division Coding Form]
   NRM = Normal Continuation       SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode    PDC = Display Procedure Division Coding Form
   ICD = Display Input CD Screen   DAT = Display Data Item Description Screen
   OCD = Display Output CD Screen  CND = Display Condition Name Screen
                                   REN = Display RENAMES Screen
EDT Command:@
********************************************************************************
```

EDT COMMAND MODE ENTERED

We overwrite continuation code NRM with CMD to get into the EDT command mode. Upon transmission, the current work-space line number (401.0000) by itself is displayed, ready to accept the EDT commands.

```
401.0000▷@COBOL END
401.0000▷@WRITE MO=PROG2,FIL=TEST2,VSN=A00022,SAT=YES
ED031  OVERWRITE? (YES or NO)▷Y
401.0000▷@HALT
```

@COBOL END command

@WRITE command

EDT message

@HALT command

We enter the @COBOL END command to terminate the COBEDT session. Then we enter the @WRITE command to write our edited program (PROG2) to the permanent SAT file (TEST2). Because a copy of our program already existed on the file, we're asked if we want to overwrite the previous version. We respond yes (key in Y). Finally, we terminate the EDT session by entering @HALT.

6.3. SAMPLE SESSION FOR EDITING A PROGRAM VIA THE SELECTIVE CREATION MODE OF COBEDT

Purpose of this session

This example shows how to edit an existing COBOL source program via the selective creation mode of COBEDT.

Assume we have a program named PROG3 containing many syntax errors. We decide to activate COBEDT to assist in correcting these errors. We choose to operate in the selective creation mode because we only wish to use the creation screens selectively.

Let's begin this example from the option select screen.

SCREEN 1

```
OS/3 EDT/COBOL                                          COBOL EDITOR(V8.0/1)
********************************************************************************
 Select Creation Mode : 2

   1=Create in COBOL Program Order
   2=Create Selected Portions of the COBOL Program




 Continuation Code (NRM)
   NRM=Normal Continuation      CMD=Enter EDT Command Mode
EDT Command:@READ MO=PROG3,FIL=TEST3,VSN=A00033,SAT=YES
********************************************************************************
```

We select the default options and enter the @READ command to transfer a copy of our program (PROG3) into the work-space file. Upon transmission, a standard COBOL coding form screen is displayed, with current work-space line number 101. (Assume PROG3 contains 100 source lines.)

SCREEN 2

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 101.
 CIA   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (NRM)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@PRINT 70:71
*******************************************************************************
```

SOURCE LINES 70 AND 71 DISPLAYED

We enter the @PRINT command in the EDT command field to display the work-space lines 70 and 71, which contain a syntax error, and then transmit.

```
70.0000          INSPECT NAME TALLYING FOR ALL CHARACTERS
71.0000               AFTER INITAL 'C'.
```

Although we realize that the coding for the INSPECT verb is wrong, we are not certain about its correct format. Therefore, we request COBEDT to display the INSPECT verb skeleton for us. Another standard COBOL coding form screen is displayed, with the current work-space line number (101), after we press the XMIT key.

SCREEN 3

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                                Standard COBOL Coding Form          Line 101
 CIA   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (INSPECT)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

Request the INSPECT verb skeleton

On this screen, we enter verb display code INSPECT in the continuation code field.

Upon transmission, the INSPECT statement screen is displayed, with the current work-space line number (101).

SCREEN 4

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                              INSPECT Statement (1 of 3)
 INSPECT identifier-1 TALLYING {identifier-2 FOR
     {{{ALL,LEADING {identifier-3,literal-1}},CHARACTERS},
     [{BEFORE,AFTER} (INITIAL) {identifier-4,literal-2}]}...}...
*******************************************************************************
                          Procedure Division Coding Form          Line 70.
 CIA   B
  |    INSPECT NAME TALLYING COUNTR FOR CHARACTERS
  |
  |
  |


 Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@PRINT 1:4
*******************************************************************************
```

SOURCE LINES 1 THROUGH 4 DISPLAYED

On this screen, we change line number 101 to 70 and enter a correct source line to overwrite the old one. We also enter the @PRINT command in the EDT command field to see the first four lines of our program. After we press the XMIT key, the source lines are displayed.

```
1.0000       IDENTIFICATION DIVISION.
2.0000       PROGRAM-ID. PROG3.
3.0000       ENVIRONMENT DIVISION.
4.0000       SPECIAL-NAMES.
```

We notice that the basic portion of the environment division is missing and the necessary source lines must be added. After we press the XMIT key again to continue COBEDT normal processing, a standard COBOL coding form screen appears, with current work-space line number 71.

SCREEN 5

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line 3.1
 C|A   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (ENV)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

Here, we change line number 71 to 3.1 to insert new source lines after the ENVIRONMENT DIVISION line and to avoid overwriting any existing lines. We then enter code ENV in the continuation code field to display the environment division screen to assist in our coding. Upon transmission, the following screen is displayed, with work-space line number 3.1.

SCREEN 6

```
OS/3 EDT/COBOL                          COBOL EDITOR(V8.0/1)-Ordered Creation Mode
*******************************************************************************
                              Environment Division                Line 3.1
 A   B
 ENVIRONMENT DIVISION.    <Enter X if Division is not to be created>
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. UNIVAC-OS3
     [(WITH) DEBUGGING MODE].    <Enter X if Line is to be created>
 OBJECT-COMPUTER. UNIVAC-OS3
     [(PROGRAM COLLATING) SEQUENCE (IS)                 <alphabet-name>]
     [SEGMENT-LIMIT IS    <segment-number>].
 Continuation Code (NRM)  [Next Screen is File Control]
   NRM = Normal Continuation              SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode           SN1 = Display Special-Names 1 Screen
   SCH = Display SYSCHAN Screen           SN2 = Display Special-Names 2 Screen
   ALP = Display Alphabet Name Screen     SSW = Display SYSSWCH Screen
   CLN = Display Class Name Screen
EDT Command:@
*******************************************************************************
```

We simply press the TAB BACK key to move the cursor to the bottom of the screen and transmit. The four source lines not enclosed in brackets are created by default. Then, a standard COBOL coding form screen is displayed, with current work-space line number 3.5.

SCREEN 7

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                          Standard COBOL Coding Form          Line
 C|A   B
  |
  |
  |
  |
  |
  |
  |
  |
  |
  |
 Continuation Code (CMD)  [Next Screen is the Standard COBOL Coding Form]
   NRM = Normal Continuation      TMP = Display Creation Screen List
   CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
   RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*******************************************************************************
```

EDT COMMAND MODE ENTERED

On this screen, we only enter code CMD in the continuation code field. Upon transmission, the current work-space line number (3.5000) by itself is displayed.

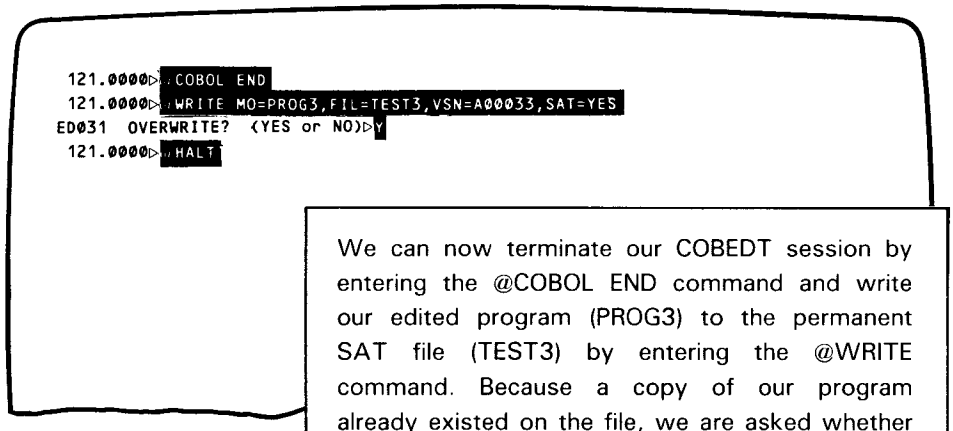
```
3.5000▷@PRINT 1:4
1.0000       IDENTIFICATION DIVISION.
2.0000       PROGRAM-ID.  PROG3.
3.0000       ENVIRONMENT DIVISON.
3.1000       ENVIRONMENT DIVISON.
3.2000       CONFIGURATION SECTION.
3.3000       SOURCE-COMPUTER.  UNIVAC-OS3.
3.4000       OBJECT-COMPUTER.  UNIVAC-OS3.
4.0000       SPECIAL-NAMES.
3.5000▷@DELETE 3.1
3.5000▷@FORMAT
```

@PRINT command
@DELETE command

@FORMAT command

We enter the @PRINT command to display source lines 1 through 4, the @DELETE command to delete a duplicated ENVIRONMENT DIVISION line (line 3.1000), and the @FORMAT command to return to COBEDT and transmit. A standard COBOL coding form screen is then displayed, with current work-space line number 3.5. We continue editing our program in COBEDT.

Assume we have finished editing our program PROG3, which now contains 120 lines, and that we are in the EDT command mode and the next work-space line number (121.0000) by itself is displayed.



@COBOL END command

@WRITE command

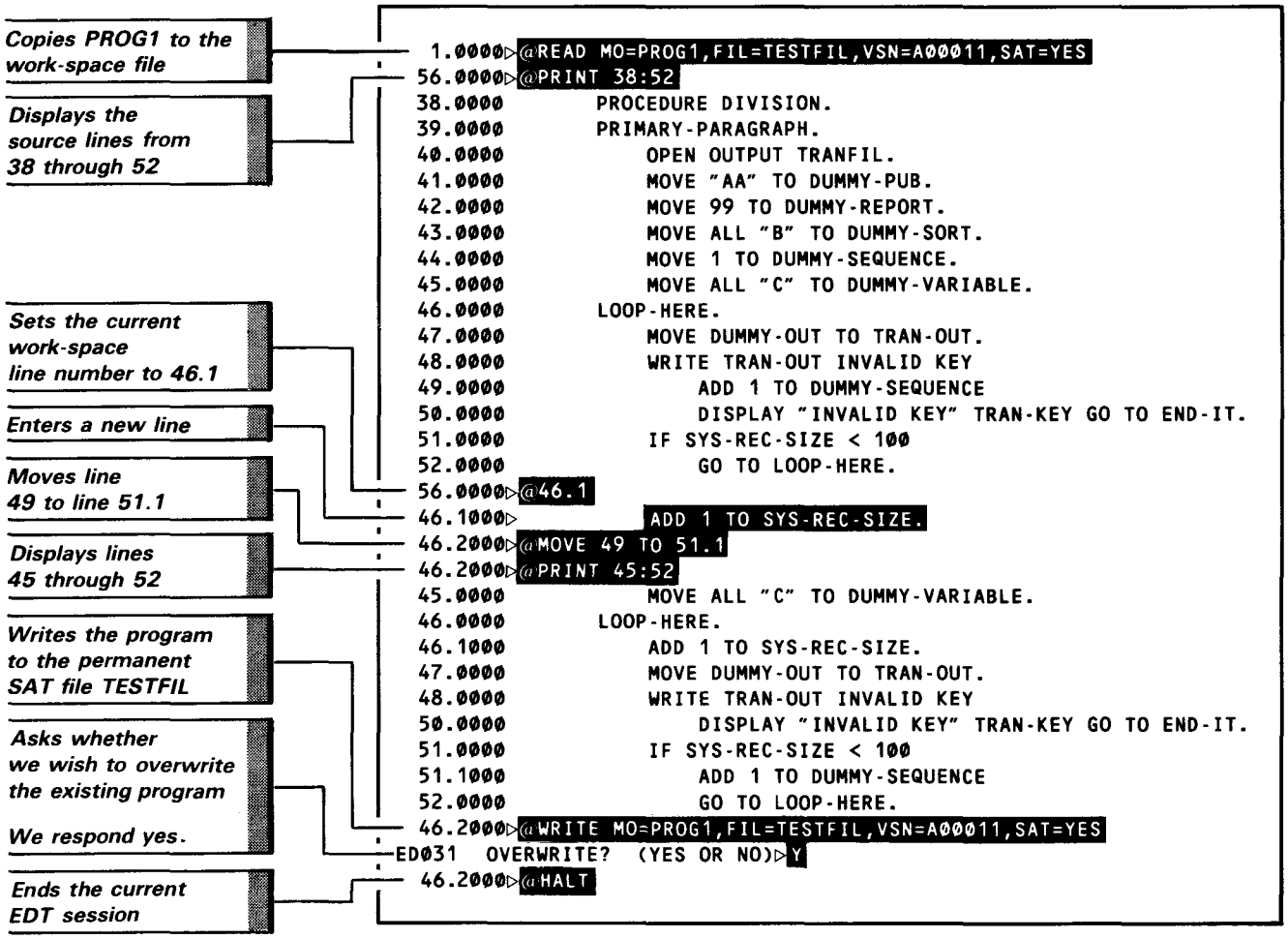
EDT message

@HALT command

We can now terminate our COBEDT session by entering the @COBOL END command and write our edited program (PROG3) to the permanent SAT file (TEST3) by entering the @WRITE command. Because a copy of our program already existed on the file, we are asked whether we want to overwrite the previous version. We respond yes (key in Y). Finally, we terminate our EDT session by entering the @HALT command.

6.4. SAMPLE SESSION FOR EDITING A PROGRAM VIA EDT

If an existing COBOL program needs only minor modification, you may make the changes directly in the EDT environment as shown in this sample session. The program used here is based on the one created in Section 4.



NOTE:

This sample is done under EDT line mode. You may also use EDT screen mode and/or the EDT error file processor (EFP) to edit your program. For more information, see the general editor (EDT) user guide/programmer reference.

Appendix A. Procedure Division Verb Skeleton Screens

A.1. DISPLAYING THE VERB SKELETON SCREENS

Purpose of verb screens

Request a verb screen

The procedure division verb skeleton screens provide you all the formats of the COBOL verbs (statements) supported by Sperry Univac. You can display any one of these screens by entering a proper verb display code in the continuation code field of a procedure division coding form screen, a standard COBOL coding form screen, or another verb skeleton screen.

Use verb name as display code

If the verb skeleton to be displayed is the only or the first format of the verb, you can simply use the verb name as the verb display code, such as CALL for the CALL verb. If the name is longer than eight characters, only the first eight characters are entered in the continuation code field, such as TRANSFOR for the TRANSFORM verb.

Choosing from multiple formats of a verb

Use verb name or description for displaying format 1

If there is more than one format for a verb, you must specifically identify the one to be displayed. For the first format, enter the first eight characters of either the verb name or the verb description without space in the continuation code field. For example, the first format of the ADD verb is ADD TO; both ADD and ADDTO are legitimate display codes.

Use verb description for displaying format 2 and above

Use other methods to identify formats

For displaying the formats of a verb other than the first one, you may enter the first eight characters of the verb description without space in the continuation code field. If the verb format cannot be uniquely identified by the first eight characters, such as SUBTRACT GIVING and SUBTRACT CORRESPONDING, you may enter either a special display code unique for the verb format (A.3) or the basic verb name appended with a number from 2 to n, where n is the highest format number of a verb. If the verb name has eight or more characters, only the first seven characters are used when appended with the format number. For example, the SUBTRACT GIVING format can be displayed by entering either the special code SUBTRCTG or SUBTRAC2.

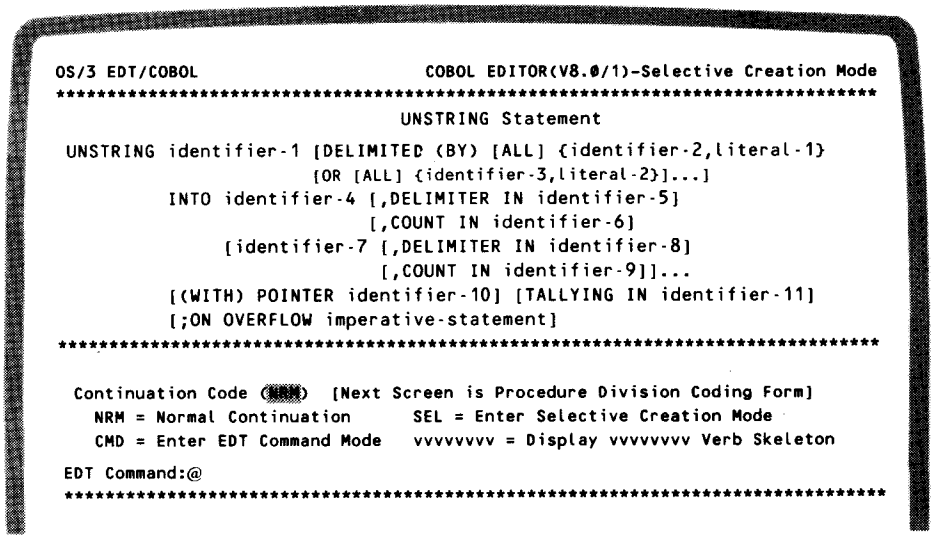
A.2. CODING THE VERBS

If a shortened version of the procedure division coding form is displayed below the verb skeleton on the screen, for example, the ADD statement screen:

```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
********************************************************************************
                        ADD Statement (1 of 3)
  ADD {identifier-1,literal-1} [{identifier-2,literal-2}]...
      TO identifier-m [ROUNDED] [identifier-n [ROUNDED]]...
      [;(ON) SIZE ERROR imperative-statement]
********************************************************************************
                        Procedure Division Coding Form          Line nnnn.nnnn
 CIA   B
  |
  |
  |
  |

 Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
   NRM = Normal Continuation      SEL = Enter Selective Creation Mode
   CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton
EDT Command:@
********************************************************************************
```

you may code the verb and/or any other desired verbs here.

If there is no coding space available below the verb skeleton on the screen, for example, the UNSTRING statement screen:

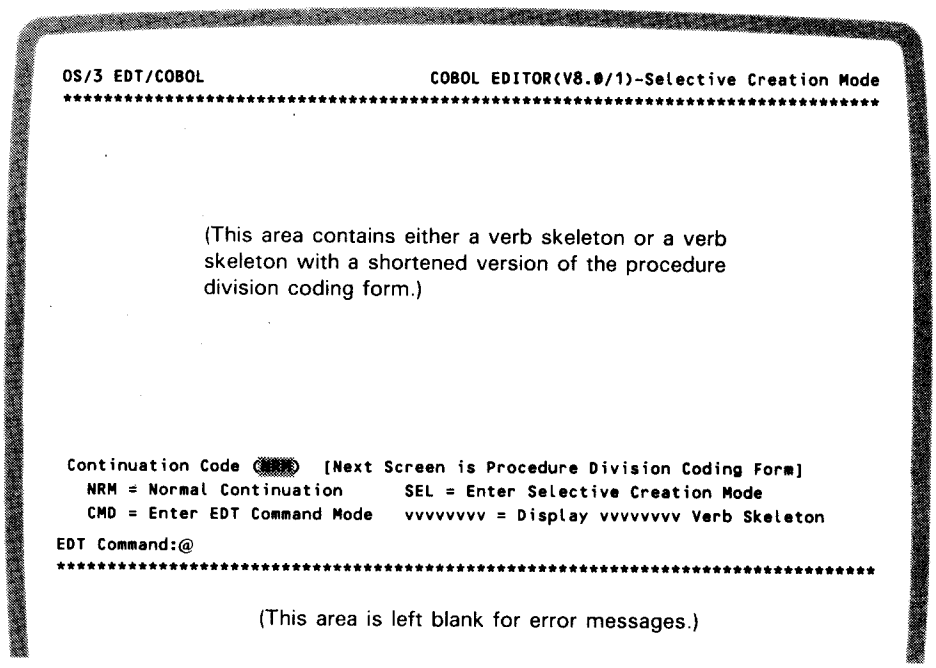


```
OS/3 EDT/COBOL                    COBOL EDITOR(V8.0/1)-Selective Creation Mode
*******************************************************************************
                              UNSTRING Statement
 UNSTRING identifier-1 [DELIMITED (BY) [ALL] {identifier-2,literal-1}
                       [OR [ALL] {identifier-3,literal-2}]...]
          INTO identifier-4 [,DELIMITER IN identifier-5]
                            [,COUNT IN identifier-6]
               [identifier-7 [,DELIMITER IN identifier-8]
                             [,COUNT IN identifier-9]]...
          [(WITH) POINTER identifier-10] [TALLYING IN identifier-11]
          [;ON OVERFLOW imperative-statement]
*******************************************************************************

  Continuation Code (NRM)  [Next Screen is Procedure Division Coding Form]
    NRM = Normal Continuation      SEL = Enter Selective Creation Mode
    CMD = Enter EDT Command Mode   vvvvvvvv = Display vvvvvvvv Verb Skeleton

 EDT Command:@
*******************************************************************************
```

you should transmit the screen with NRM in the continuation code field to display a coding form screen. If you are in ordered creation mode processing, a full procedure division coding form screen appears. If you are in selective creation mode processing, a standard COBOL coding form screen appears.

A.3. VERB SKELETON SCREENS

All the procedure division verb skeleton screens have similar structures and provide identical choices of continuation codes:



To avoid redundancy in this appendix, we present only the verb skeleton portion of the screens.

The pointers for the verb skeletons are the actual verb display codes and can be used directly to display the screens.

Making Low-Volume Data Available to the Specified Data Item (ACCEPT Statement)

ACCEPT or ACCEPTFR

SCREEN ACCEPT Statement

```
                    ACCEPT Statement (1 of 4)
ACCEPT identifier [FROM user-name]
```

ACCEPT2, ACCEPTDT, ACCEPTDY, or ACCEPTTI

SCREEN ACCEPT Statement

```
                    ACCEPT Statement (2 of 4)
ACCEPT identifier [FROM {DATE,DAY,TIME}]
```

ACCEPT3 or ACCEPTME

SCREEN ACCEPT Statement

```
                    ACCEPT Statement (3 of 4)
ACCEPT cd-name (MESSAGE) COUNT
```

ACCEPT4 or ACCEPTUS

SCREEN ACCEPT Statement

```
                    ACCEPT Statement (4 of 4)
ACCEPT identifier-1 [,identifier-2]...FROM mnemonic-name
    USING {identifier-3,literal}
```

Summing Numeric Data Items and Storing the Result (ADD Statement)

ADD or ADDTO

SCREEN ADD Statement

```
                    ADD Statement (1 of 3)
ADD {identifier-1,literal-1} [{identifier-2,literal-2}]...
    TO identifier-m [ROUNDED] [identifier-n [ROUNDED]]...
    [;(ON) SIZE ERROR imperative-statement]
```

ADD2 or ADDGIVIN

SCREEN ADD Statement

```
                    ADD Statement (2 of 3)
ADD {identifier-1,literal-1} [{identifier-2,literal-2}]...
    GIVING identifier-m [ROUNDED] [identifier-n [ROUNDED]]...
    [;(ON) SIZE ERROR imperative-statement]
```

ADD3 or ADDCORRE

SCREEN ADD Statement

```
                    ADD Statement (3 of 3)
ADD {CORRESPONDING,CORR} identifier-1 TO identifier-2 [ROUNDED]
    [;(ON) SIZE ERROR imperative-statement]
```

Changing the Sequence of Operations (ALTER Statement)

ALTER

SCREEN ALTER Statement

```
                        ALTER Statement
ALTER procedure-name-1 TO [PROCEED TO] procedure-name-2
    [,procedure-name-3 TO [PROCEED TO] procedure-name-4]...
```

Transferring Control from One Object Program to Another (CALL Statement)

CALL

SCREEN CALL Statement

```
                        CALL Statement
CALL {identifier-1,literal-1}
    [USING {data-name-1,cd-name-1,identifier-2,file-name-1}
    [,{data-name-2,cd-name-2,identifier-3,file-name-2}]...]
    [;(ON) OVERFLOW imperative-statement]
```

Releasing the Main Storage Areas Occupied by the Named Program (CANCEL Statement)

CANCEL

SCREEN CANCEL Statement

```
                        CANCEL Statement
CANCEL {identifier-1,literal-1} [{,identifier-2,literal-2}]...
```

Terminating the Processing of Files (CLOSE Statement)

CLOSE or CLOSESEQ for CLOSE SEQUENTIAL

SCREEN CLOSE Statement

```
                    CLOSE Statement (1 of 2)
CLOSE file-name-1 [{REEL,UNIT} [(WITH) NO REWIND (FOR) REMOVAL]
    [(WITH) {NO REWIND,LOCK}]]
    [file-name-2 [{REEL,UNIT} [(WITH) NO REWIND (FOR) REMOVAL]
    [(WITH) {NO REWIND,LOCK}]]]...
```

CLOSE2, CLOSEREL, or CLOSEIND for CLOSE RELATIVE or CLOSE INDEXED

SCREEN CLOSE Statement

```
                    CLOSE Statement (2 of 2)
CLOSE file-name-1 [(WITH) LOCK]
    [file-name-2 [(WITH) LOCK]]...
```

Evaluating an Arithmetic Expression and Storing the Result (COMPUTE Statement)

COMPUTE

SCREEN COMPUTE Statement

```
                    COMPUTE Statement
COMPUTE identifier-1 [ROUNDED] [identifier-2 [ROUNDED]] ...
    = arithmetic-expression [(ON) SIZE ERROR imperative-statement]
```

Incorporating Text into a COBOL Source Program (COPY Statement)

COPY

SCREEN COPY Statement

```
                    COPY Statement
COPY text-name [{OF,IN} library-name]
    [REPLACING {==pseudo-text-1==,identifier-1,literal-1,word-1}
     BY {==pseudo-text-2==,identifier-2,literal-2,word-2}]...
```

Logically Removing a Record from a Mass Storage File (DELETE Statement)

DELETE

SCREEN DELETE Statement

```
                    DELETE Statement
DELETE file-name RECORD [INVALID (KEY) imperative-statement]
```

Notifying MCS* to Inhibit the Transfer of Data (DISABLE Statement)

DISABLE

SCREEN DISABLE Statement

```
                                DISABLE Statement
DISABLE {INPUT [TERMINAL],OUTPUT} cd-name
    (WITH) KEY {identifier-1,literal-1}
```

Outputting Low-Volume Data on a System Logical Device (DISPLAY Statement)

DISPLAY

SCREEN DISPLAY Statement

```
                                DISPLAY Statement
DISPLAY {identifier-1,literal-1} [,{identifier-2,literal-2}] ...
    [UPON mnemonic-name [USING {identifier-3,literal-3}]]
```

Dividing One Numeric Data Item into Another and Storing the Quotient and Remainder (DIVIDE Statement)

DIVIDE or DIVIDEIN

SCREEN DIVIDE Statement

```
                                DIVIDE Statement (1 of 5)
DIVIDE {identifier-1,literal-1} INTO identifier-2 [ROUNDED]
    [,{identifier-3 [ROUNDED]] ...
    [;(ON) SIZE ERROR imperative-statement]
```

DIVIDE2 or DIVIDEIG

SCREEN DIVIDE Statement

```
                                DIVIDE Statement (2 of 5)
DIVIDE {identifier-1,literal-1} INTO {identifier-2,literal-2}
    GIVING identifier-3 [ROUNDED] [identifier-4 [ROUNDED]] ...
    [;(ON) SIZE ERROR imperative-statement]
```

DIVIDE3 or DIVIDEBG

SCREEN DIVIDE Statement

```
                                DIVIDE Statement (3 of 5)
DIVIDE {identifier-1,literal-1} BY {identifier-2,literal-2}
    GIVING identifier-3 [ROUNDED] [identifier-4 [ROUNDED]] ...
    [;(ON) SIZE ERROR imperative-statement]
```

*MCS stands for message control system.

DIVIDE4 or DIVIDIGR

SCREEN DIVIDE Statement

```
                                DIVIDE Statement (4 of 5)
DIVIDE {identifier-1,literal-1} INTO {identifier-2,literal-2}
    GIVING identifier-3 [ROUNDED] REMAINDER identifier-4
    [;(ON) SIZE ERROR imperative-statement]
```

DIVIDE5 or DIVIDBGR

SCREEN DIVIDE Statement

```
                                DIVIDE Statement (5 of 5)
DIVIDE {identifier-1,literal-1} BY {identifier-2,literal-2}
    GIVING identifier-3 [ROUNDED] REMAINDER identifier-4
    [;(ON) SIZE ERROR imperative-statement]
```

Notifying MCS* to Allow the Transfer of Data (ENABLE Statement)

ENABLE

SCREEN ENABLE Statement

```
                                ENABLE Statement
ENABLE {INPUT [TERMINAL],OUTPUT} cd-name
    (WITH) KEY {identifier-1,literal-1}
```

Displaying the Current Values of Data Items (EXHIBIT Statement)

EXHIBIT

SCREEN EXHIBIT Statement

```
                                EXHIBIT Statement
EXHIBIT {NAMED,CHANGED NAMED,CHANGED} {identifier,nonnumeric-literal}
```

Providing a Common End Point or a Logical End (EXIT Statement)

EXIT

SCREEN EXIT Statement

```
                                EXIT Statement
EXIT [PROGRAM]
```

*MCS stands for message control system.

Transferring Control out of Normal Sequence (GO TO Statement)

GO or GOTO

SCREEN GO TO Statement

```
                                GO TO Statement (1 of 3)
GO (TO) {procedure-name-1,.}
```

GO2 or GOTODEPE

SCREEN GO TO Statement

```
                                GO TO Statement (2 of 3)
GO (TO) procedure-name-1 [,procedure-name-2] ... procedure-name-n
    DEPENDING (ON) identifier
```

GO3 or GOTOMORE

SCREEN GO TO Statement

```
                                GO TO Statement (3 of 3)
GO (TO) MORE-LABELS
```

Evaluating a Condition (IF Statement)

SCREEN IF Statement

```
                                    IF Statement
IF condition [{THEN,;}] {statement-1,NEXT SENTENCE}
    {;ELSE statement-2,ELSE NEXT SENTENCE}
```

Tallying and/or Replacing Characters in a Data Item (INSPECT Statement)

INSPECT or INSPECTT

SCREEN INSPECT Statement

```
                                INSPECT Statement (1 of 3)
INSPECT identifier-1 TALLYING {identifier-2 FOR
    {{{ALL,LEADING {identifier-3,literal-1}},CHARACTERS},
    [{BEFORE,AFTER} (INITIAL) {identifier-4,literal-2}]}...}...
```

INSPECT2 or INSPECTR

SCREEN INSPECT Statement

```
                                INSPECT Statement (2 of 3)
INSPECT identifier-1 REPLACING {CHARACTERS BY {identifier-6,literal-4}
    [{BEFORE,AFTER} (INITIAL) {identifier-7,literal-5}]
    {{ALL,LEADING,FIRST} {{identifier-5,literal-3} BY
    {identifier-6,literal-4},
    [{BEFORE,AFTER} (INITIAL) {identifier-7,literal-5}]}...}...}
```

INSPECT3 or INSPCTTR

SCREEN INSPECT Statement

```
                    INSPECT Statement (3 of 3)
INSPECT identifier-1 TALLYING (see Format 1 for TALLYING description)
    REPLACING (see Format 2 for REPLACING description)
```

Combining Sorted Files and Outputting Records in Merged Order (MERGE Statement)

MERGE

SCREEN MERGE Statement

```
                          MERGE Statement
MERGE file-name-1 ON {ASCENDING,DESCENDING} (KEY) data-name-1
    [,data-name-2]... ON {ASCENDING,DESCENDING} (KEY) data-name-3
    [,data-name-4]... [(COLLATING) SEQUENCE (IS) alphabet-name]
    USING file-name-2, file-name-3 [,file-name-4]...
    {OUTPUT PROCEDURE (IS) section-name-1 [THRU section-name-2],
     GIVING file-name-5}
```

Transferring Data to One or More Data Areas (MOVE Statement)

MOVE

SCREEN MOVE Statement

```
                    MOVE Statement (1 of 2)
MOVE {identifier-1,literal} TO {identifier-2 [,identifier-3]...
```

MOVE2 or MOVECORR

SCREEN MOVE Statement

```
                    MOVE Statement (2 of 2)
MOVE {CORRESPONDING,CORR} identifier-1 TO identifier-2
```

Multiplying Numeric Data Items and Storing the Result (MULTIPLY Statement)

MULTIPLY

SCREEN MULTIPLY Statement

```
                    MULTIPLY Statement (1 of 2)
MULTIPLY {identifier-1,literal-1} BY identifier-2 [ROUNDED]
    [,identifier-3 [ROUNDED]]...
    [(ON) SIZE ERROR imperative-statement]
```

MULTIPL2 or MLTPLYBG

SCREEN MULTIPLY Statement

```
                    MULTIPLY Statement (2 of 2)
MULTIPLY {identifier-1,literal-1} BY {identifier-2,literal-2}
    GIVING identifier-3 [ROUNDED] [,identifier-4 [ROUNDED]]...
    [(ON) SIZE ERROR imperative-statement]
```

Specifying the Condition for Executing the Statements (ON Statement)

ON

SCREEN ON Statement

```
                    ON Statement
ON integer-1 [AND (EVERY) integer-2] [UNTIL integer-3]
    {statement-1,NEXT SENTENCE} ELSE {statement-2,NEXT SENTENCE}
```

Initiating the Processing of Files (OPEN Statement)

OPEN or OPENSEQU for OPEN SEQUENTIAL

SCREEN OPEN Statement

```
                    OPEN Statement (1 of 2)
OPEN {INPUT file-name-1 [REVERSED] [(WITH) NO REWIND]
  [,file-name-2 [REVERSED] [(WITH) NO REWIND]]...
  OUTPUT file-name-3 [(WITH) NO REWIND] [file-name-4 [NO REWIND]]...
  I-O file-name-5 [,file-name-6]...
  EXTEND file-name-7 [,file-name-8]...}
```

OPEN2, OPENRELA, or OPENINDE for OPEN RELATIVE or OPEN INDEXED

SCREEN OPEN Statement

```
                    OPEN Statement (2 of 2)
OPEN {INPUT file-name-1 [,file-name-2]...
      OUTPUT file-name-3 [,file-name-4]...
      I-O file-name-5 [,file-name-6]...}
      OUTPUT file-name-3 [,file-name-4]...
      I-O file-name-5 [,file-name-6]...
```

Transferring Control out of Normal Sequence and Returning Control after Transfer Execution (PERFORM Statement)

PERFORM or PERFORMT

SCREEN PERFORM Statement

```
                        PERFORM Statement (1 of 4)
PERFORM procedure-name-1 [{THROUGH,THRU} procedure-name-2]
```

PERFORM2 or PERFORMN

SCREEN PERFORM Statement

```
                        PERFORM Statement (2 of 4)
PERFORM procedure-name-1 [{THROUGH,THRU} procedure-name-2]
        {identifier-1,integer-1} TIMES
```

PERFORM3 or PERFORMU

SCREEN PERFORM Statement

```
                        PERFORM Statement (3 of 4)
PERFORM procedure-name-1 [{THROUGH,THRU} procedure-name-2]
        UNTIL condition-1
```

PERFORM4 or PERFORMV

SCREEN PERFORM Statement

```
                        PERFORM Statement (4 of 4)
PERFORM procedure-name-1 [{THROUGH,THRU} procedure-name-2] VARYING
     {identifier-2,index-name-1} FROM {identifier-3,index-name-2,literal-1}
     BY {identifier-4,literal-2} UNTIL condition-1
     [AFTER {identifier-5,index-name-3} FROM {identifier-6,index-name-4,literal-3}
     BY {identifier-7,literal-4} UNTIL condition-2
     [AFTER {identifier-8,index-name-5} FROM {identifier-8,index-name-6,literal-5}
     BY {identifier-10,literal-6} UNTIL condition-3]]
```

Making Available a Record from a File (READ Statement)

READ or READSEQU for READ SEQUENTIAL

SCREEN READ Statement

```
                                READ Statement (1 of 4)
READ file-name RECORD [INTO identifier] [;AT END imperative-statement]
```

READ2 or READNEXT

SCREEN READ Statement

```
                                READ Statement (2 of 4)
READ file-name [NEXT] RECORD [INTO identifier]
     [;AT END imperative-statement]
```

READ3 or READINVA

SCREEN READ Statement

```
                                READ Statement (3 of 4)
READ file-name  RECORD [INTO identifier]
     [;INVALID (KEY) imperative-statement]
```

READ4 or READKEYI

SCREEN READ Statement

```
                                READ Statement (4 of 4)
READ file-name  RECORD [INTO identifier] [KEY (IS) data-name]
     [;INVALID (KEY) imperative-statement]
```

Making Available a Message or Segment from an MCS* Queue (RECEIVE Statement)

RECEIVE

SCREEN RECEIVE Statement

```
                                RECEIVE Statement
RECEIVE cd-name {MESSAGE,SEGMENT} INTO identifier-statement]
     [;NO DATA imperative-statement]
```

Transferring Records to a Sort Operation (RELEASE Statement)

RELEASE

SCREEN RELEASE Statement

```
                                RELEASE Statement
RELEASE record-name [FROM identifier]
```

*MCS stands for message control system.

Obtaining Sorted Records from a Sort Operation or Merged Records during a Merge Operation (RETURN Statement)

RETURN

SCREEN RETURN Statement

```
                    RETURN Statement
RETURN file-name RECORD [INTO identifier] ;AT END imperative-statement
```

Logically Replacing a Record in a Mass Storage File (REWRITE Statement)

REWRITE

SCREEN REWRITE Statement

```
                    REWRITE Statement
REWRITE record-name [FROM identifier]
    [INVALID (KEY) imperative-statement]
```

Searching for a Table Element That Satisfies the Specified Condition (SEARCH Statement)

SEARCH or SEARCHVA

SCREEN SEARCH Statement

```
                    SEARCH Statement (1 of 2)
SEARCH identifier-1 [VARYING {identifier-2,index-name-1}]
  [;AT END imperative-statement-1]
  ;WHEN condition-1 {imperative-statement-2,NEXT SENTENCE}
  [;WHEN condition-2 {imperative-statement-3,NEXT SENTENCE}]...
```

SEARCH2 or SEARCHAL

SCREEN SEARCH Statement

```
                    SEARCH Statement (2 of 2)
SEARCH ALL identifier-1 [;AT END imperative-statement-1]
  WHEN {data-name-1 {(IS) EQUAL (TO),(IS) =}
    {identifier-3,literal-1,arithmetic-expression-1},condition-name-1}
  [AND {data-name-2 {(IS) EQUAL (TO),(IS) =}
    {identifier-4,literal-2,arithmetic-expression-2},condition-name-2}]
  {imperative-statement,NEXT SENTENCE}
```

Releasing a Message or Segment to Output MCS* Queues (SEND Statement)

SEND or SENDFROM

SCREEN SEND Statement

```
                                    SEND Statement (1 of 2)
SEND cd-name FROM identifier-1
```

SEND2 or SENDFRWI

SCREEN SEND Statement

```
                                    SEND Statement (2 of 2)
SEND cd-name [FROM identifier-1]
    {(WITH) identifier-2,(WITH) ESI,(WITH) EMI,(WITH) EGI}
    [{BEFORE,AFTER} ADVANCING
    {{{identifier-3,integer} [LINE,LINES]},PAGE}]
```

Establishing Reference Points for Table Handling Operations (SET Statement)

SET or SETTO

SCREEN SET Statement

```
                                    SET Statement (1 of 2)
SET {identifier-1 [,identifier-2]...,index-name-1 [,index-name-2]...}
    TO {identifier-3,index-name-3,integer-1}
```

SET2, SETUPBY, or SETDOWNB

SCREEN SET Statement

```
                                    SET Statement (2 of 2)
SET index-name-4 [,index-name-5]...
    {UP BY,DOWN BY} {identifier-4,integer-2}
```

Creating a Sort File and Outputting Records in Sorted Order (SORT Statement)

SORT

SCREEN SORT Statement

```
                                    SORT Statement
SORT file-name-1 ON {ASCENDING,DESCENDING} (KEY) data-name-1
    [,data-name-2]... [ON {ASCENDING,DESCENDING} (KEY) data-name-3
    [,data-name-4]...] [COLLATING SEQUENCE (IS) alphabet-name]
    {INPUT PROCEDURE (IS) section-name-1 [THRU section-name-2],
     USING file-name-2 [,file-name-3]...}
    {OUTPUT PROCEDURE (IS) section-name-3 [THRU section-name-4],
     GIVING file-name-4}
```

*MCS stands for message control system.

Logically Positioning a Relative or Indexed File and Retrieving Records Sequentially (START Statement)

START

SCREEN START Statement

```
                                START Statement
START file-name [KEY {(IS) EQUAL (TO),(IS) =,(IS) GREATER (THAN),(IS) >,
                      (IS) NOT LESS (THAN),(IS) NOT < } ]
     [;INVALID KEY imperative-statement]
```

Terminating or Temporarily Suspending the Program Execution (STOP Statement)

STOP

SCREEN STOP Statement

```
                                STOP Statement
STOP {RUN,literal}
```

Juxtaposing Two or More Data Items into a Single One (STRING Statement)

STRING

SCREEN STRING Statement

```
                                STRING Statement
STRING {identifier-1 [identifier-2],literal-1 [literal-2]}...
  DELIMITED (BY) {identifier-3,literal-3,SIZE}
       [{identifier-4 [identifier-5],literal-4 [literal-5]}...
        DELIMITED (BY) {identifier-6,literal-6,SIZE}]...
  INTO identifier-7 [(WITH) POINTER identifier-8]
  [;ON OVERFLOW imperative statement]
```

Subtracting Numeric Data Items from Specified Items and Storing the Result (SUBTRACT Statement)

SUBTRACT or SUBTRCTF

SCREEN SUBTRACT Statement

```
                          SUBTRACT Statement (1 of 3)
SUBTRACT {identifier-1,literal-1} [,{identifier-2,literal-2}]...
    FROM identifier-m [ROUNDED] [,identifier-n [ROUNDED]]
    [(ON) SIZE ERROR imperative-statement]
```

SUBTRAC2 or SUBTRCTG

SCREEN SUBTRACT Statement

```
                          SUBTRACT Statement (2 of 3)
SUBTRACT {identifier-1,literal-1} [,{identifier-2,literal-2}]...
    FROM {identifier-m,literal-m} GIVING
         identifier-n [ROUNDED] [,identifier-o [ROUNDED]]
    [(ON) SIZE ERROR imperative-statement]
```

SUBTRAC3 or SUBTRCTC

SCREEN SUBTRACT Statement

```
                    SUBTRACT Statement (3 of 3)
SUBTRACT {CORRESPONDING,CORR} identifier-1 FROM identifier-2 [ROUNDED]
    [(ON) SIZE ERROR imperative-statement]
```

Initiating or Terminating the Trace Function (TRACE Statement)

TRACE

SCREEN TRACE Statement

```
                    TRACE Statement
{READY,RESET} TRACE
```

Altering Characters of an Identifier (TRANSFORM Statement)

TRANSFOR or TRNSFMFT

SCREEN TRANSFORM Statement

```
                    TRANSFORM Statement (1 of 2)
TRANSFORM identifier-1 [,identifier-2]... CHARACTERS
    FROM {identifier-3,nonnumeric-literal-1,figurative-constant-1}
    TO {identifier-4,nonnumeric-literal-2,figurative-constant-2}
```

TRANSFO2 or TRNSFMOB

SCREEN TRANSFORM Statement

```
                    TRANSFORM Statement (2 of 2)
TRANSFORM identifier-1 [,identifier-2]... CHARACTERS
    {ON,BY} identifier-5
```

Separating Characters in a Field and Placing Them into Multiple Fields (UNSTRING STATEMENT)

UNSTRING

SCREEN UNSTRING Statement

```
                    UNSTRING Statement
UNSTRING identifier-1 [DELIMITED (BY) [ALL] {identifier-2,literal-1}
                    [OR [ALL] {identifier-3,literal-2}]...]
        INTO identifier-4 [,DELIMITER IN identifier-5]
                    [,COUNT IN identifier-6]
            [identifier-7 [,DELIMITER IN identifier-8]
                    [,COUNT IN identifier-9]]...
        [(WITH) POINTER identifier-10] [TALLYING IN identifier-11]
        [;ON OVERFLOW imperative-statement]
```

Specifying Procedures for I/O Error or Tape Label Handling or Identifying User Items to Be Monitored by a Debugging Section (USE Statement)

USE or USEAFTER

SCREEN USE Statement

```
                                USE Statement (1 of 3)
USE AFTER (STANDARD) {EXCEPTION,ERROR} PROCEDURE (ON)
    {file-name-1 [,file-name-2]...,INPUT,OUTPUT,I-O,EXTEND}
```

USE2 or USEFORDE

SCREEN USE Statement

```
                                USE Statement (2 of 3)
USE (FOR) DEBUGGING (ON) {cd-name-1,[ALL (REFERENCES OF)] identifier-2,
                          file-name-1,procedure-name-1,ALL PROCEDURES}
    [{cd-name-2,[ALL (REFERENCES OF)] identifier-3,file-name-2,
      procedure-name-2,ALL PROCEDURES}]...
```

USE3 or USELABEL

SCREEN USE Statement

```
                                USE Statement (3 of 3)
USE {AFTER,BEFORE} (STANDARD) [BEGINNING,ENDING] [FILE,REEL]
     LABEL PROCEDURE (ON) {file-name-1 [,file-name-2]...,INPUT,OUTPUT}
```

Releasing a Logical Record for an Output or I/O File (WRITE Statement)

WRITE or WRITESEQ for WRITE SEQUENTIAL

SCREEN WRITE Statement

```
                                WRITE Statement (1 of 2)
WRITE record-name [FROM identifier-1]
    [{BEFORE,AFTER} ADVANCING {{identifier-2,integer} [{LINE,LINES}],
                               mnemonic-name,PAGE}]
    [;AT {END-OF-PAGE,EOP} imperative-statement]
```

WRITE2, WRITEREL, or WRITEIND for WRITE RELATIVE or WRITE INDEXED

SCREEN WRITE Statement

```
                                WRITE Statement (2 of 2)
WRITE record-name [FROM identifier]
    [;INVALID (KEY) imperative-statement]
```

Indicating the Location Where a Debugging Package Is to Be Executed (*DEBUG Statement)

*DEBUG

SCREEN *DEBUG Statement

```
                                        *DEBUG Statement

    *DEBUG procedure-name
```

Appendix B. General Editor Command Summary

B.1. SUMMARY OF EDT COMMANDS

Table B-1 summarizes the formats and explanations for the EDT commands. The commands are listed in alphabetical order.

Table B-1. EDT Command Summary (Part 1 of 6)

Command	Format	Explanation
@	@{line-number [increment] / + / -}[:{data / command}]	Sets the current line number and increment for data and command lines keyed in at the workstation
CHANGE	@C ['search-string'[*n]] TO 'change-string'[*n]	Replaces an existing string in the current work-space file with a new string
COPY	@CO [line-range]['search-string'[*n]] TO destination	Copies lines in the current work-space file to new line locations without deleting the original lines
DELETE	@D [line-range]['search-string'[*n]]	Erases specified lines from the current work-space file
FIND	@FIN 'search-string'[*n]	Locates the first occurrence of a string in the work-space file and assigns its corresponding line number to the variable ? and the column numbers of the first and last columns it occupies to [and] respectively

Table B-1. EDT Command Summary (Part 2 of 6)

Command	Format	Explanation
FSTATUS	To specify file parameters for any file for which you want a list of modules, use this format: @FS[MODULE=module-name][,TYPE={module-type / S}] ,FILENAME={filename / 'filename' / "filename"}[,RDPASS=password] ,VSN=volume [,DEVICE={did / DISK / DISKETTE}]	Creates in the work-space file a list of all modules contained in a specified program library
INSERT	@I 'change-string'[*n]	Inserts a specified string into lines in the current work-space file
LIST	@L [line-range]['search-string'[*n]][IMMEDIATE]	Prints specified lines from the current work-space file on the printer
MOVE	@M [line-range]['search-string'[*n]] TO destination	Transfers specified lines to new line locations in the work-space file and deletes the original lines and line numbers
NUMBER	@NU 'sequence-string'[*n][BY increment]	Inserts sequence numbers into input lines
PRINT	@P [line-range]['search-string'[*n]]	Displays specified lines from the current work-space file on the workstation screen
PUNCH	@PU [line-range]['search-string'[*n]][IMMEDIATE]	Reproduces specified lines from the current work-space file on cards
READ	To read a SAT or MIRAM library module from disk or format label diskette to the current work-space file, use this format: @READ MODULE=module-name[,TYPE={module-type / S}] [,TRUNC={YES / NO}],FILENAME={filename / 'filename' / "filename"} [,RDPASS=password],VSN=volume [,DEVICE={did / DISK / DISKETTE}] △[{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOW△first-col-no:last-col-no}]	Reads a copy of a library module or program library into the work-space file

Table B–1. EDT Command Summary (Part 3 of 6)

Command	Format	Explanation
READ (cont)	To read a MIRAM data file from disk or format label diskette to the current work-space file, use this format: @READ FILENAME={filename / 'filename' / "filename"} [,RDPASS=password] ,VSN=volume [,KEYNO={n / **0**}] [,DEVICE={did / **DISK** / **DISKETTE**}] [,BFSZ=n] [,TRUNC={YES / **NO**}] Δ [{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOWΔfirst-col-no:last-col-no}] To read a unit record file from a data set label diskette or from the card reader, use this format: @READ FILENAME={filename / 'filename' / "filename"},VSN=volume ,DEVICE={did / DISKETTE / RDR} [,TRUNC={YES / **NO**}] Δ [{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOWΔfirst-col-no:last-col-no}] To read a file from a tape, use this format: @READ FILENAME={filename / 'filename' / "filename"} [,RDPASS=password] ,VSN=volume,DEVICE={did / TAPE} [,BKNO={YES / **NO**}] [,TRUNC={YES / **NO**}] Δ [{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOWΔfirst-col-no:last-col-no}]	

Table B–1. EDT Command Summary (Part 4 of 6)

Command	Format	Explanation
READ (cont)	To read a file from the spool file to the current work-space file, use this format: @READ [JOB=jobname][,HOLD={L / N / Y}] [,FILENAME={filename / 'filename' / "filename"}][,ACCT=acct-no] ,QUEUE={LOG / PRINT / PUNCH / RDR}[,ALL={YES / NO}][,SKIP={n / 0}] [,TRUNC={YES / NO}]Δ[{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOWΔfirst-col-no:last-col-no}] To read the same module or file last accessed through a previous @READ or @WRITE command, use this format: @READ To read the same module or file last accessed through a previous @READ or @WRITE command but read now with a KEY, KKEY, or SHOW parameter or any valid EDT command specified, use this format: @READΔ;Δ[{KEY=start-col-no:end-col-no / KKEY=start-col-no:end-col-no / SHOWΔfirst-col-no:last-col-no}] [valid EDT command]	
REMOVE	@REM 'search-string'[*n]	Deletes a specified string from lines in the work-space file
SEQUENCE	@SEQ {'sequence-string'[*n] / *}BY increment	Inserts sequence numbers into existing lines in the current work-space file
UPDATE	@U [line-range]['search-string'[*n]]	Displays specified lines from the work-space file one at a time for you to edit or change

Table B–1. EDT Command Summary (Part 5 of 6)

Command	Format	Explanation
WRITE	To write the current work-space file to a SAT or MIRAM library module on a disk or format label diskette, use this format: @WRITE MODULE=module-name[,TYPE={module-type | S}] ,FILENAME={filename | 'filename' | "filename"}[,WRPASS=password] [,DEVICE={did | DISK | DISKETTE}] ,VSN=volume [,CONTIG={YES | NO}][,INC={n | 1}][,RCSZ=n][,SIZE=n] [,SAT={YES | NO}] To write the current work-space file to a MIRAM data file on a disk or format label diskette, use this format: @WRITE FILENAME={filename | 'filename' | "filename"}[,WRPASS=password] ,VSN=volume[,CONTIG={YES | NO}][,DEVICE={did | DISK | DISKETTE}] [,INC={n | 1}][,INIT={YES | NO}][,EXTEND={YES | NO}] [,KEYi={start-col-no:end-col-no | (start-col-no:end-col-no,{DUP | NODUP},{CHG | NOCHG})}] ,SIZE=n [,RCB={YES | NO}][,RCFM={FIX | VAR}],RCSZ=n [,SCSZ={n | 256}][,BFSZ=n] To write the current work-space file to a unit record file (i.e., to the printer, card punch, or to a data set label diskette), use this format: @WRITE FILENAME={filename | 'filename' | "filename"},VSN=volume ,DEVICE={did | DISKETTE | PRINT | PUNCH}[,RCFM={FIX | VAR}][,RCSZ=n]	Writes a copy of the current work-space file to: a program library or data file on disk, diskette, or tape, or to the spool file

Table B-1. EDT Command Summary (Part 6 of 6)

Command	Format	Explanation
WRITE (cont)	To write the current work-space file to a tape, use this format: @WRITE FILENAME={filename / 'filename' / "filename"}[,WRPASS=password] ,VSN=volume,DEVICE={did / TAPE}[,BFSZ=n][,BKNO={YES / NO}] [,RCFM={FIXUNB / FIXBLK / VARUNB / VARBLK / UNDEF}][,RCSZ=n][,INIT={YES / NO}] [,EXTEND={YES / NO}] To write the current work-space file to the spool file, use this format: @WRITE [JOB=jobname][,HOLD={YES / NO}] [,FILENAME={filename / 'filename' / "filename"}][,ACCT=acct-no] ,QUEUE={PRINT / PUNCH / RDR}[,COPIES={n / 1}] To write to the same module or file last accessed through a previous @READ or @WRITE command, use this format: @WRITE To write to the same module or file last accessed through a previous @READ or @WRITE command, but written to now with any valid EDT command specified, use this format: @WRITE△;△valid EDT command	

B.2. SUMMARY OF EDT PROCEDURE FILE COMMANDS

Table B-2 summarizes the formats and explanations for the EDT procedure file commands. The commands are listed in alphabetical order.

Table B-2. EDT Procedure File Command Summary

Command	Format	Explanation
DO	@DO proc-number [{PRINT / NOPRINT / REVERT}]	Executes a procedure file
END	@E	Terminates procedure file definition
GOTO	@G {label / line}	Permits branching within a procedure file
INPUT	@INP file-parameters [{PRINT / NOPRINT / REVERT}]	Loads and executes a procedure file
NOP	@NOP [comment]	Enters extra lines for branching or comments into a procedure file
PROC	@PRO [proc-number]	Begins procedure file definition
RETURN	@RET	Terminates procedure file execution

B.3. SUMMARY OF EDT VARIABLE COMMANDS

Table B-3 summarizes the formats and explanations for the EDT variable commands. The commands are listed in alphabetical order.

Table B-3. EDT Variable Command Summary

Command	Format	Explanation
ASSIGN	@AS Gn={'string'[*n] / n(x:y) / n[±m] / Gm / LEN(n)}	Assigns values to EDT variables
DISPLAY	@DI {'string'[*n] / n(x:y) / n[±m] / Gm / LEN(n)}	Displays a specified expression or the value of a specified expression from the work-space file on the workstation screen
IF	@IF.condition.command or @IF expression relation expression command	Permits an EDT command or EDT procedure file command to be executed based on some condition

B.4. SUMMARY OF EDT DIRECTIVE COMMANDS

Table B–4 summarizes the formats and explanations for the directive commands. The commands are listed in alphabetical order.

Table B–4. EDT Directive Command Summary (Part 1 of 2)

Command	Format	Explanation
CHECK	@CHE [{ON / OFF}]	Determines if processed lines are to be displayed on the workstation screen
COBOL	@COB	Activates the COBOL editor
DROP	@DR	Deletes all lines in the entire EDT work-space file
EFP	@EFP	Activates the error file processor
FORMAT	@FORMAT parameter string (for RPGEDT) @FORMAT (for COBEDT)	Used only in conjunction with either RPGEDT or COBEDT. See the appropriate subeditor manual for information on the @FORMAT directive.
HALT	@H	Terminates the EDT session
RPG	@RPG	Activates the RPG II editor
SET	@S [CHAR=tab-character,TABS={columns / OFF}] [,LINE=length][,EXCLUDE={exclusion-character / OFF}] [,ATSIGN=command-trigger][,COLON=range-separator] [,ENCOL=end-column][,BUFFER={record-size / ON}] [,WIDTH=device-size][,CLEAR][,STRIP={ON / OFF}] [,DISPLAY] [,SCRDSPLY={TRUNCATE / FOLD}] [,ROLL={15 (if SCRDSPLY=TRUNCATE) / 0 (if SCRDSPLY=FOLD) / 1–15}] [,MODE={LINE / SCREEN}][,LANGUAGE={FREEFORM / FORTRAN / COBOL / RPG}] [,RECENTRY={SINGLE / MULT}][,SCRFORM={UNDERLINE / BLANK}]	Defines various parameters to EDT that collectively make up your EDT environment

Table B-4. EDT Directive Command Summary (Part 2 of 2)

Command	Format	Explanation
SYSTEM	@SY [workstation-command]	Permits workstation commands to be issued during an EDT session or temporarily returns you to system mode

B.5. SUMMARY OF EDT SCREEN COMMANDS

Table B-5 summarizes the formats and explanations for the screen commands. It lists the commands in alphabetical order.

Table B-5. EDT Screen Command Summary

Command	Format	Explanation
BLOCK	@BL	Displays a freeform screen that allows you to switch to block mode for entering multiple commands or data
HELP	@HE [error message code]	Displays help screens for any EDT error messages
PARAMS	@PA	Displays a screen showing the parameters on the @SET directive (those that make up your EDT environment)
PROMPT	@PROM [edt command]	Displays the EDT command menu screen or help screens for any of the EDT commands (meaning EDT commands, modifiers, directives, procedure file commands, variables, and screen commands)
RESTORE	@RES	Returns you to the point in your EDT session where you originally entered a screen command
ROLL	@RO	Displays freeform screens, showing the EDT work-space file, where you can update lines or simply view them

B.6. SUMMARY OF EFP COMMANDS

Table B-6 summarizes the formats and explanations for the EFP commands. It lists the commands in alphabetical order, not the order in which you may necessarily use them.

Table B-6. EFP Command Summary

Command	Format	Explanation

EFP	To correct and display COBOL and RPG II errors and FORTRAN IV errors for one source module at a time, use:
	
	`@EF[X]Δ[program-unit-name]Δ`
	`[error-range]Δ`
	`['search-string']`
	
	To correct and display FORTRAN IV errors for compilations that process multiple source modules, use:
	
	`@EF SOU source-module-name,`
	`source-file-name,`
	`vsn`
		Displays errors in your error file along with the source lines that contain those errors. Note that EFP is both an EDT directive and an EFP command.

END	`@EF END`	Terminates the error file processor
SUMMARY	`@EF SUM`	Displays an error file summary for the module you're correcting

Appendix C. Workstation Command List

For a quick reference, all the commands you may enter from a workstation are listed here. For detailed descriptions of these commands, see the current versions of the interactive services commands and facilities user guide/programmer reference and its summary.

```
ALLOCATE△{ST}  ,FILENAME={filename  }[,RDPASS=password][,WRPASS=password]
         {IR}            {'filename'}
         {IS}            {"filename"}
         {DA}
         {SQ}
         {NI}
         {MI}
  ,VSN=volume[,CONTIG={YES}][,INC={n}],SIZE=n
              {NO }         {1}
```

```
ASK△[user-id],'text'
```

```
BEGIN△JBQ[,{A}]
          {H}
          {N}
          {P}
```

```
BEGIN△SPL,[{ALL   }][,ACCT=acctno][,CART=cartridge-id][,DEV={768 }]
          {LOG    }                                       {770 }
          {PRINT  }                                       {773 }
          {PUNCH  }                                       {776 }
          {RDR    }                                       {778 }
                                                          {789 }
                                                          {9300}
  [,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]
```

```
BRKPT△{P}[,{PR}]  [,ACCT=acctno][,CART=cartridge-id]
      {I} {PU}
  [,DEV={768 }]  [,FILE=filename][,FORM=formname],JOB=jobname
        {770 }
        {773 }
        {776 }
        {778 }
        {789 }
        {9300}
```

```
BRKPT△LOG
```

```
CANCEL△jobname [,{D}]
               {N}
```

```
CHANGE△jobname,{H}
               {N}
               {P}
```

```
COMMENT△MODULE=modulename [,TYPE={module-type}],FILENAME={filename  }
                              {S           }           {'filename' }
                                                       {"filename" }
  [,RDPASS=password][,WRPASS=password],VSN=volume△text
```

```
CONNECT△job[,filename]
```

Copying Modules:

```
COPY△MODULE=modulename [,TYPE={module-type}],FILENAME={filename  }
                       {S           }           {'filename' }
                                                {"filename" }
  [,RDPASS=password],VSN=volume△TO△MODULE=modulename [,TYPE={module-type}]
                                                     {S           }
  ,FILENAME={filename  } [,RCSZ=n][,WRPASS=password],VSN=volume
            {'filename' }
            {"filename" }
  [,CONTIG={YES}] [,INC={n}] [,SIZE=n] [,SAT={YES}] △[NUMBER][,HEX][,WAIT]
           {NO }       {1}               {NO }
```

Copying MIRAM Files:

```
COPY△FILENAME={filename|'filename'|"filename"},RDPASS=password,VSN=volume[,KEYNO={n|0}]
  [,DEVICE={did|DISK|DISKETTE}]△TO△FILENAME={filename|'filename'|"filename"}[,WRPASS=password],VSN=volume
  [,CONTIG={YES|NO}][,INC={n|1}][,KEYNO={n|0}][,KEYi={n:m|(n:m,{DUP|NDUP},{CHG|NCHG})}][,SIZE=n]
  [,INIT={YES|NO}][,RCB={YES|NO}][,RCFM={FIX|VAR}][,RCSZ=n][,EXTEND={YES|NO}][,BFSZ=n]
  [,SCSZ={n|256}][,DEVICE={did|DISK|DISKETTE}]△[NUMBER][,HEX][,WAIT]
```

Copying Spool Files:

```
COPY△[JOB=jobname][,HOLD={L|N|W}][,FILENAME={filename|'filename'|"filename"}]
  [,ACCT=acct],QUEUE={LOG|PRINT|PUNCH|RDR}
  [,ALL={YES|NO}][,SKIP={n|W}]△TO△[JOB=jobname][,HOLD={N|Y}][,FILENAME={filename|'filename'|"filename"}]
  ,QUEUE={PRINT|PUNCH|RDR}△[NUMBER][,HEX][,WAIT]
```

Copying Tape Files:

```
COPY△[FILENAME={filename|'filename'|"filename"}][,RDPASS=password],VSN=volume,DEVICE={did|TAPE}[,BKNO={YES|NO}]
  △TO△[FILENAME={filename|'filename'|"filename"}][,WRPASS=password],VSN=volume,DEVICE={did|TAPE}
  [,INIT={YES|NO}][,EXTEND={YES|NO}][,BFSZ=n][,BKNO={YES|NO}][,RCFM={FIXUNB|FIXBLK|VARUNB|VARBLK|UNDEF}]
  [,RCSZ=n]△[NUMBER][,HEX][,WAIT]
```

Copying Unit Record Files:

```
COPY△DEVICE={did      },FILENAME={filename   },VSN=volume △TO△DEVICE={did      }
            {DISKETTE}           {'filename'}                        {DISKETTE}
            {RDR     }           {"filename"}                        {PRINT    }
                                                                     {PUNCH    }

 [,RCFM={FIX}][,RCSZ=n] ,FILENAME={filename   },VSN=volume△[NUMBER][,HEX][,WAIT]
        {VAR}                     {'filename'}
                                  {"filename"}
```

```
DDP△command-string
```

```
DEFKEY△{F#nn}{,command string}
       {MW  }
```

Deleting Function Key Definitions:

```
DEFKEY△{F#mm}
       {MW  }
```

```
DEFKEY DISPLAY
```

```
DELETE△SPL,{ALL   }[,ACCT=acctno][,CART=cartridge-id] [,DEV={768 }]
           {LOG   }                                    [      {770 }]
           {PRINT }                                    [      {773 }]
           {PUNCH }                                    [      {776 }]
           {RDR   }                                    [      {778 }]
                                                       [      {789 }]
                                                       [      {9300}]

  [,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]
```

Deleting a Specific Job:

```
DELETE△jobname[,LOG]
```

Deleting All Jobs from One or All Job Queues:

```
DELETE△JBQ,{A}[,LOG]
           {H}
           {N}
           {P}
```

```
DISPLAY△JBQ,  { A }
              { H }
              { N }
              { P }
```

```
DISPLAY△ACT [,{ ALL   }] [,ACCT=acctno][,CART=cartridge-id] [,DEV={ 768 }]
             { PRINT  }                                       {  770 }
             { PUNCH  }                                       {  773 }
                                                              {  776 }
                                                              {  778 }
                                                              {  789 }
                                                              { 9300 }
  [,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]
```

```
DISPLAY△JS[,jobname]
```

```
DISPLAY△LOG
```

```
DISPLAY△SPL[,{ ALL   }] [,ACCT=acctno][,CART=cartridge-id] [,DEV={ 768 }]
            { LOG    }                                       {  770 }
            { PRINT  }                                       {  773 }
            { PUNCH  }                                       {  776 }
            { RDR    }                                       {  778 }
                                                             {  789 }
                                                             { 9300 }
  [,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]
```

```
EDT△[initial command]
```

To Run a Command Stream from a Library File:

```
ENTER△MODULE=modulename[,TYPE={ module-type }] ,FILENAME={ filename   }
                           {  ALL           }             { 'filename' }
                                                          { "filename" }
  [,RDPASS=password],VSN=volume
```

To Run a Command Stream from a Spooled File:

```
ENTER△[,HOLD={N}] [,FILENAME={ filename   }] ,QUEUE=RDR
          {Y}                { 'filename' }
                             { "filename" }
```

To Run a Command Stream from a Diskette:

```
ENTER△[FILENAME={ filename   }] [,RDPASS=password],VSN=volume [,DEVICE={ did      }]
                { 'filename' }                                         { DISKETTE }
                { "filename" }
```

To Run a Command Stream from a Tape:

```
ENTER△ [FILENAME={filename   }] [,RDPASS=password],VSN=volume [,DEVICE={did }]
                 {'filename' }                                        {TAPE}
                 {"filename" }
```

To Run a Command Stream from a Card Reader:

```
ENTER△ [DEVICE={did}]
               {RDR}
```

Erasing a Library Module:

```
ERASE△MODULE=modulename [,TYPE={module-type}],FILENAME={filename   }
                            {■              }          {'filename' }
                                                       {"filename" }
  [,WRPASS=password],VSN=volume
```

Erasing Library and MIRAM Files:

```
ERASE△FILENAME={filename   } [,WRPASS=password],VSN=volume
               {'filename' }
               {"filename" }
```

```
EXECUTE jobname
```

```
FILE {([did],label)} △[{(:alt-filename                                  )}]
     {(RDR,label)   }  [{(:(alt-filename,{RES}))                        )}]
                       [{                 {RUN}                         }]
                       [{                 {vsn}                         }]
                       [{(:(alt-filename,[{RES}] write-pass))            }]
                       [{                 [{RUN}]                       }]
                       [{                 {vsn}                         }]
```

```
FREE
```

```
FSTATUS△[MODULE=modulename] [,TYPE={module-type}],FILENAME={filename   }
                                   {■           }          {'filename' }
                                                           {"filename" }
   [,RDPASS=password],VSN=volume△[LONG]
```

```
GO△jobname
```

```
HELP△{command     }
     {message-no  }
```

```
HOLD△SPL,[(ALL    )][,ACCT=acctno][,CART=cartridge-id] [,DEV=(768 )]
         [(PRINT  )]                                      [(770 )]
         [(PUNCH  )]                                      [(773 )]
         [(LOG    )]                                      [(776 )]
         [(RDR    )]                                      [(778 )]
                                                          [(789 )]
                                                          [(9300)]
    [,FILE=filename][,FORM=formname][,JOB=jobname][,STEP=stepno]
```

Holding All Jobs or All Jobs on a Particular Job Queue:

```
HOLD△JBQ[,(    )]
         [(H   )]
         [(N   )]
         [(P   )]
```

Holding an Individual Job:

```
HOLD△jobname
```

```
LOGON△user-id[,acct][,password][,exec-pro] [,BULLETIN=(NO  )][,LOG=(    )]
                                            [(    )]          [(NO  )]
```

```
LOGOFF
```

```
MENU△[menu-name]
```

```
OCL{([did],label)}△[jobname-library-unit[(new-name)]] [,(PRE  )]
   {(RDR,label)  }                                     [(HIGH )]
                                                       [(     )]

  [,key-1=val-1,...,key-n=val-n]
  OV△jobname-library-unit[(new-name)] [,(PRE  )][,key-1=val-1,...,key-n=val-n]
                                      [(HIGH )]
                                      [(     )]
```

```
PAUSE△jobname
{PR}△[function-code][,ACCT=actno][,CART=cartridge-id]
{PU}
  [,FILE=filename][,FORM=formname][,JOB=jobname]
```

Printing a Library Module:

```
PRINT△MODULE=modulename[,TYPE=(module-type)] ,FILENAME=(filename  )
                       [(     )]               ('filename' )
                                               ("filename ")

  [,RDPASS=password],VSN=volume [,COPIES=(n)]△[DIRECT][,NUMBER][,HEX][,WAIT]
                                [(    )]
```

Printing a Library or MIRAM File:

```
PRINT△FILENAME={filename  }[,RDPASS=password],VSN=volume [,KEYNO={n}]
               {'filename'}                                    {■}
               {"filename"}

[,COPIES={n}][,DEVICE={did      }] △[DIRECT][,NUMBER][,HEX][,WAIT]
          {■}         {DISKETTE}
                      {DISK     }
```

Printing a Spool File:

```
PRINT△[JOB=jobname] [,HOLD={L}][,FILENAME={filename  }][,ACCT=acct]
                      {N}    {'filename'}
                      {Y}    {"filename"}

  ,QUEUE={LOG   }[,ALL={YES}][,COPIES={n}][,SKIP={n}]△[DIRECT][,NUMBER][,HEX][,WAIT]
         {PRINT }      {■■■}          {■}         {■}
         {PUNCH }
         {RDR   }
```

Printing a Tape File:

```
PRINT△ [FILENAME={filename  }][,RDPASS=password],VSN=volume ,DEVICE={did }
               {'filename'}                                        {TAPE}
               {"filename"}

  [,BKNO={YES}] △[DIRECT][,NUMBER][,HEX][,WAIT]
         {■■■}
```

Printing a Unit Record File:

```
PRINT△ DEVICE={did      },FILENAME={filename  },VSN=volume
              {DISKETTE}           {'filename'}
              {RDR     }           {"filename"}

  △[DIRECT][,NUMBER][,HEX][,WAIT]
```

Punching a Library Module:

```
PUNCH△ MODULE=modulename [,TYPE={module-type}],FILENAME={filename  }
                              {■■■■■■■■■■■}           {'filename'}
                                                      {"filename"}

  [,RDPASS=password],VSN=volume [,COPIES={n}]△[DIRECT][,WAIT]
                                         {■}
```

Punching a MIRAM File:

```
PUNCH△FILENAME={filename  }[,RDPASS=password],VSN=volume [,KEYNO={n}]
               {'filename'}                                  {■}
               {"filename"}
  [,COPIES={n}] [,DEVICE={did      }]△[DIRECT][,WAIT]
           {1}          {DISKETTE}
                        {DISK     }
```

Punching a Spool File:

```
PUNCH△[JOB=jobname] [,HOLD={L}] [,FILENAME={filename  }][,ACCT=acct]
                          {N}              {'filename'}
                          {Y}              {"filename"}
  ,QUEUE={LOG  } [,ALL={YES}] [,COPIES={n}] [,SKIP={n}]△[DIRECT][,WAIT]
         {PRINT}       {■■ }          {■}         {■}
         {PUNCH}
         {RDR  }
```

Punching a Tape File:

```
PUNCH△[FILENAME={filename  }][,RDPASS=password],VSN=volume
              {'filename'}
              {"filename"}
  ,DEVICE={did } [,BKNO={YES}]△[DIRECT][,WAIT]
          {TAPE}        {■■ }
```

Punching a Unit Record File:

```
PUNCH△DEVICE={did     },FILENAME={filename  },VSN=volume △[DIRECT][,WAIT]
             {DISKETTE}          {'filename'}
             {RDR    }          {"filename"}
```

```
RECALL△[{LAST△nn          }]
        {hh:mm:ss-hh:mm:ss}
```

```
RECOVER△MODULE=modulename [,TYPE={module-type}],FILENAME={filename  }
                              {■              }          {'filename'}
                                                         {"filename"}
  [,RDPASS=password][,WRPASS=password],VSN=volume
```

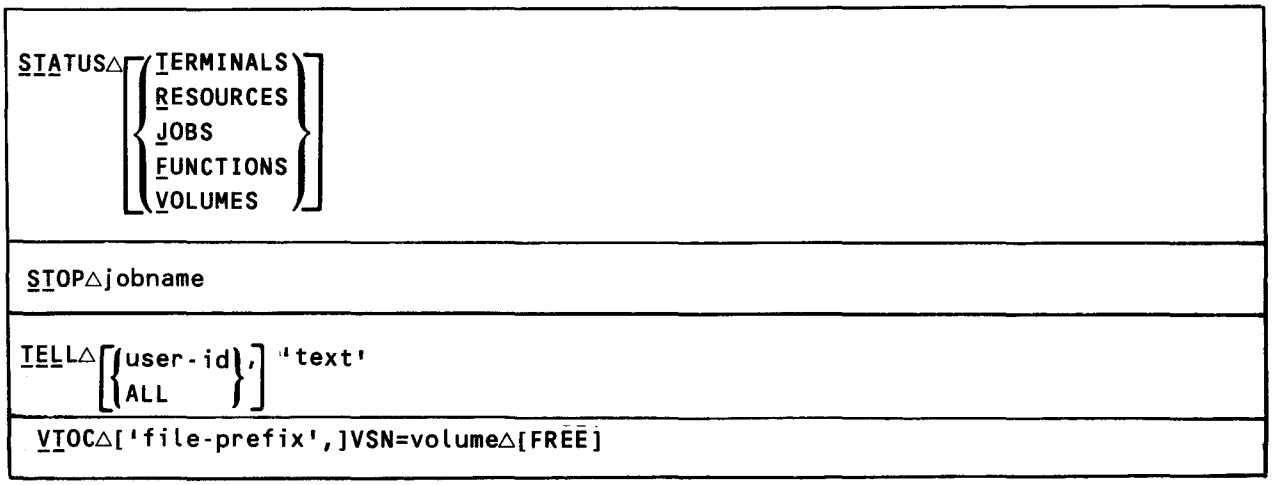
```
REMARK△text
```

RESUME

```
RUN{([did],label)}△[{jobname     }][(new-name)][{:alt-filename                                  }]
   {(RDR,label)  }  {module-name }              {:(alt-filename,{RES}))                         }
                                                {               {RUN}                           }
                                                {               {vsn}                           }
                                                {:(alt-filename,[{RES}],read-pass)              }
                                                {               [{RUN}]                         }
                                                {               [{vsn}]                         }
    [,{PRE }][,key-1=val-1,...,key-n=val-n]
      {HIGH}
      {NOR }

RV△{jobname     }[(new-name)][{:alt-filename                                  }][,{PRE }]
   {module-name }             {:(alt-filename,{RES}))                         }  {HIGH}
                              {               {RUN}                           }  {NOR }
                              {               {vsn}                           }
                              {:(alt-filename,[{RES}],read-pass)              }
                              {               [{RUN}]                         }
                              {               [{vsn}]                         }
  [,key-1=val-1,...,key-n=val-n]
```

```
SI{([did],label)}△[{jobname     }][(new-name)][,{PRE }]
  {(RDR,label)  }  {module-name }              {HIGH}
                                               {NOR }

SC△{jobname     }[(new-name)][,{PRE }]
  {module-name }              {HIGH}
                              {NOR }
```

```
SCREEN△[,{SCROLL  }]  ,{UPPER}  [,XMIT={VAR }]  [,XFER={VAR }]  [,SPEED={9600 }]
       {ROLL    }      {LOWER}         {CHAN}          {CHAN}          {4800 }
       {NP      }                      {ALL }          {ALL }          {2400 }
       {WRAP    }                                                      {1200 }
       {NOROLL  }                                                      {600  }
                                                                       {300  }

[,SPACEBAR={DESTRUCT   }]  [,LINES={24}]  [,KEYBOARD={STANDARD}]  [,INTENSITY={NORMAL  }]
           {NONDESTRUCT}           {12}            {KATAKANA}              {LOW     }
                                                                           {REVERSE }
```


STATUS△[{TERMINALS / RESOURCES / JOBS / FUNCTIONS / VOLUMES}]
STOP△jobname
TELL△[{user-id / ALL},] 'text'
VTOC△['file-prefix',]VSN=volume△[FREE]

Index

SPERRY UNIVAC

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

_____

(Document Title)

_____ _____ _____

(Document No.) *(Revision No.)* *(Update No.)*

Comments:

Cut along line.

From:

_____

(Name of User)

_____

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

CUT

FOLD