Horst

SPERRY✦UNIVAC

This Library Memo announces the release and availability of Updating Package A to "SPERRY UNIVAC Operating System/3 (OS/3) Report Program Generator II (RPG II) Editor User Guide/Programmer Reference", UP-8803 Rev. 2.

This maintenance update for release 8.0 explains how to enter an & option in a /COPY modifier statement. Also, sample initial display screens now illustrate the position of the release level and version number of the RPG II editor being used.

Copies of Updating Package A are now available for requisitioning. Either the updating package only or the complete manual with the updating package may be requisitioned by your local Sperry Univac representative. To receive only the updating package, order UP-8803 Rev. 2—A. To receive the complete manual, order UP-8803 Rev. 2.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00, A10, B00, B10, 18, 18U, 19, 19U, 20, 20U, 21, 21U, 28U, 29U, 75, 75U, 76 and 76U (Package A to UP-8803 Rev. 2, 17 pages plus Memo)	Library Memo for UP-8803 Rev. 2—A RELEASE DATE: January, 1983

SPERRY✛UNIVAC
COMPUTER SYSTEMS

This Library Memo announces the release and availability of "SPERRY UNIVAC® Operating System/3 (OS/3) Report Program Generator II (RPG II) Editor User Guide/Programmer Reference", UP-8803 Rev. 2.

This revision documents the following changes to the RPG II editor for release 8.0:

■ addition of Auto Report specification screens,

■ changes to the layout of the file specification screen,

■ addition of the currency symbol field on the header specification screens,

■ clarifications and editorial changes to descriptive text, and

■ addition of RPG II Directive Summary (Appendix C)

Destruction Notice: If you are going to OS/3 release 8.0, use this revision and destroy all previous copies. If you are not going to OS/3 release 8.0, retain the copy you are now using and store this revision for future use.
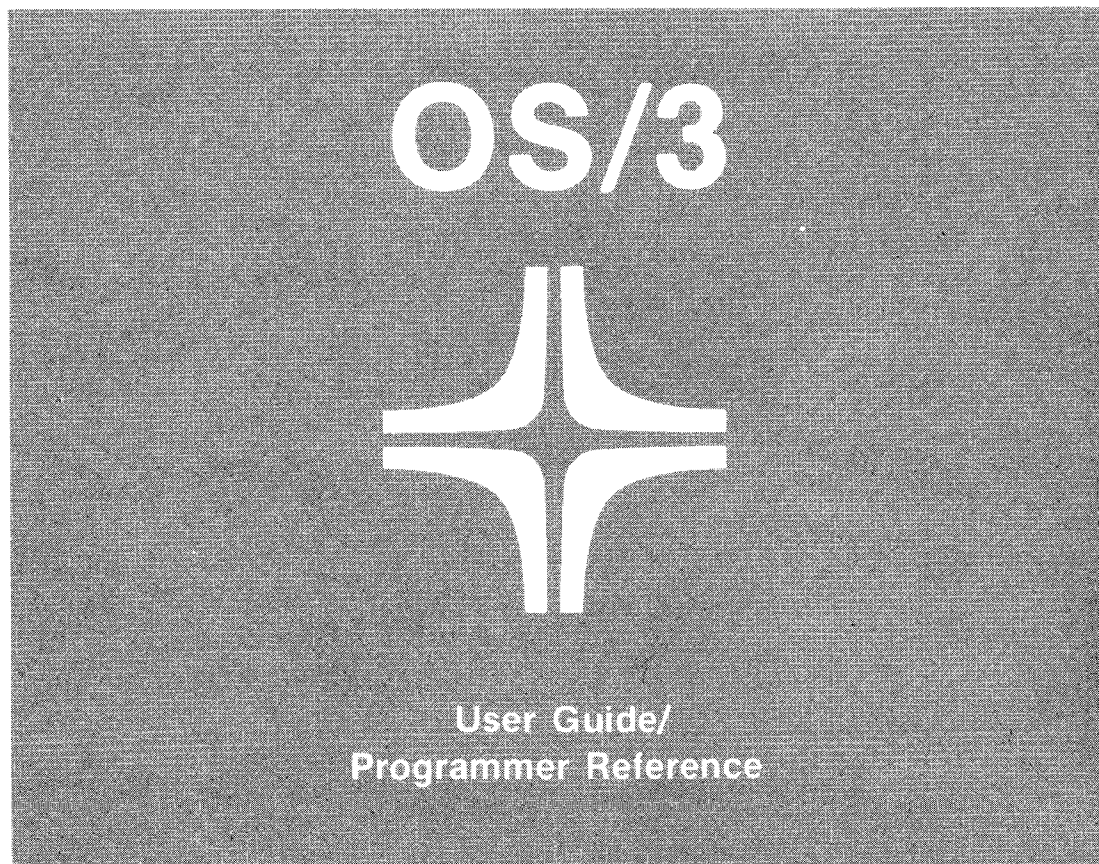
Copies of UP-8803 Rev. 1 will be available for 6 months after the release of 8.0. Should you need additional copies of this edition, you should order them within 90 days of the release of 8.0. When ordering the previous edition of a manual, be sure to identify the exact revision and update packages desired and indicate that they are needed to support an earlier release.

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, A10, B00, B10, 18, 18U, 19, 19U, 20, 20U, 21, 21U, 28U, 29U, 75, 75U, 76 and 76U (Cover and 94 pages)	Library Memo for UP-8803 Rev. 2
		RELEASE DATE: September, 1982

Report Program Generator II (RPG II) Editor

OS/3

User Guide/
Programmer Reference

PAGE STATUS SUMMARY

ISSUE: Update A — UP-8803 Rev. 2
RELEASE LEVEL: 8.0 Forward

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover/Disclaimer		Orig.						
PSS	1	A						
Preface	1	Orig.						
Contents	1 thru 3							
1	1 thru 4	Orig.						
2	1,2 3 thru 5 6 7, 8	A Orig. A Orig.						
3	1 thru 4 5 6 thru 14 15 16 thru 22 23 24 thru 27	Orig. A Orig. A Orig. A Orig.						
4	1 2 2a 3 thru 10 11 12 thru 15	A Orig. Orig. Orig. A Orig.						
5	1 thru 3	Orig.						
Appendix A	1 thru 12	Orig.						
Appendix B	1 thru 11	Orig.						
Appendix C	1	Orig.						
Index	1 thru 5	Orig.						
User Comment Sheet								

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 the programmer in the use of the SPERRY UNIVAC Operating System/3 (OS/3). Specifically, this manual describes the OS/3 RPG II editor and its effective use. The intended audience is the novice programmer who has little knowledge in data processing or the more experienced programmer who may or may not be familiar with the RPG II editor.

This manual comprises:

- Section 1. Introduction

  Explains what the RPG II editor is, what it's used for, and how it operates with the general editor.

- Section 2. Using the RPG II Editor

  Describes how to activate and terminate the RPG II editor.

- Section 3. Creating an RPG II Source Program

  Explains how to create RPG II source programs using the RPG II editor's display formats.

- Section 4. Updating an RPG II Source Program

  Explains how to update RPG II source programs by using both the RPG II editor's display formats and commands and EDT commands.

- Section 5. Error Detection and Recovery

  Describes the types of errors that may occur during an RPG II editor session.

- Appendixes

  Contain the specification screens in positional, formatted, and free-form display formats; a table summarizing the EDT commands; and a table summarizing directives.

Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

B. EDT COMMANDS

C. RPG II DIRECTIVE SUMMARY

INDEX

USER COMMENT SHEET

FIGURES

TABLES

1.   Introduction

1.1.   WHAT IS THE RPG II EDITOR?

Specialized language editor

The SPERRY UNIVAC Operating System/3 (OS/3) Report Program Generator II (RPG II) editor is a specialized language (subeditor) editor of the OS/3 general editor (EDT) used to create and update RPG II source programs interactively from a workstation. Because it is a subeditor of EDT, the RPG II editor uses many EDT facilities, as well as its own, when creating and updating programs. Using the RPG II editor enables you to receive prompt screens and immediate syntax checking, which are helpful when either creating or updating your program. In addition to the RPG II editor, EDT provides you with a feature known as *screen mode processing,* whereby an RPG II template screen is displayed for your input. Screen mode processing does not provide syntax checking but it is useful when making minor corrections to an existing program.

EDT overview

You must be familiar with EDT before you can use the RPG II editor. For convenience, a brief overview of EDT is provided here; however, you should see the OS/3 general editor user guide/programmer reference, UP-8828 (current version) for a complete discussion of EDT.

EDT functions

The OS/3 general editor is a user-oriented interactive program used to create and update library modules and data files from a workstation. EDT enables you to add to, delete from, and modify the contents of your files.

Using the work-space file

EDT works with your files in a temporary disk file called the work-space file. Each time EDT is initiated, the file is created and lasts for the duration of the EDT session. When you read or enter RPG II source statements or data into the work-space file, they are automatically assigned temporary line numbers. You use these line numbers similar to the sequence numbers on the RPG II coding forms (columns 1–5) to reference and manipulate statements in your program during an EDT session.

Saving the
work-space file

The work-space file can hold only one program or data file at a time. Therefore, the content of the work-space file must be saved on disk or diskette before the RPG II editor is terminated or another file is created or read from disk; otherwise, the content of the work-space file will be lost.

Creating a
new program

To create an RPG II program, enter source statements into the work-space file one at a time. When the program is completed, you can store the new program in a SAT file on either disk or format label diskette. You may also list the file on the printer or punch it on cards. Figure 1-1 summarizes this procedure.

Updating an existing
program

To update an existing program, you must read the program from the program library to place a copy in the work-space file. You then do all the updating with the work-space copy. When the updating is completed, you can either overwrite the original library version with the work-space version or store the work-space version in a new file. You can also list the file on the printer or punch it on cards. Figure 1-2 summarizes this procedure.

1.2. COMMAND CONVENTIONS

The conventions used to illustrate the commands presented in this manual are:

Commands and parameters	@RPG	Commands and parameters in capital letters must be keyed in exactly as shown.
Abbreviated commands	@PRINT	Underscoring indicates commands may be abbreviated; only the underscored characters are required as keyins.
User-defined variables	@PRINT line-range	Parameters constructed in lowercase letters designate user-defined variables.
Optional parameters	@PRINT [line-range]	Optional parameters are enclosed in brackets.
User entries	FIELD NAME: NAMEIN	User entries are identified by reverse type.
Default values	SELECT MODE (C)	Default values automatically generated by the RPG II editor are enclosed in parentheses and shaded. (See 2.2.)
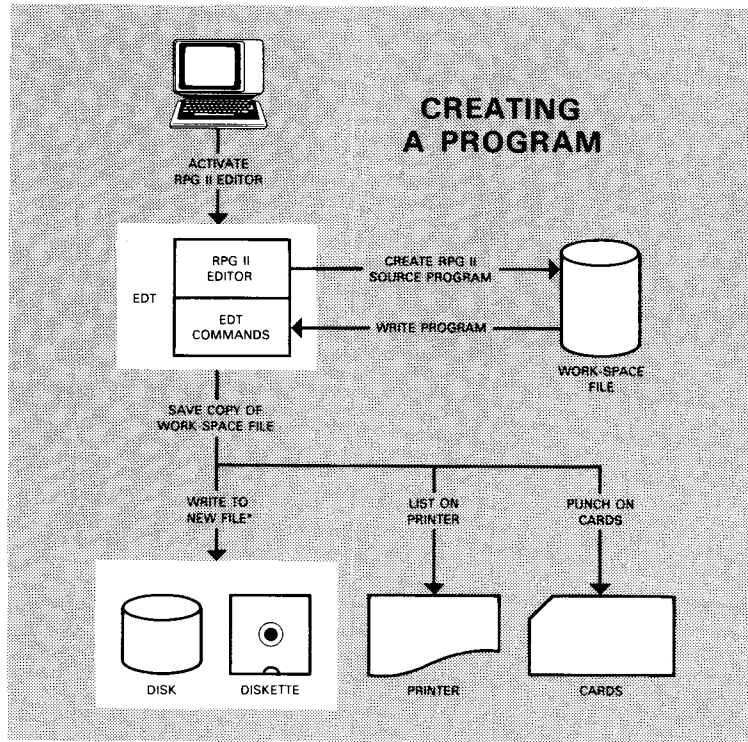
*RPG II programs must be written to SAT files to be compiled.

Figure 1-1. Creating an RPG II Program
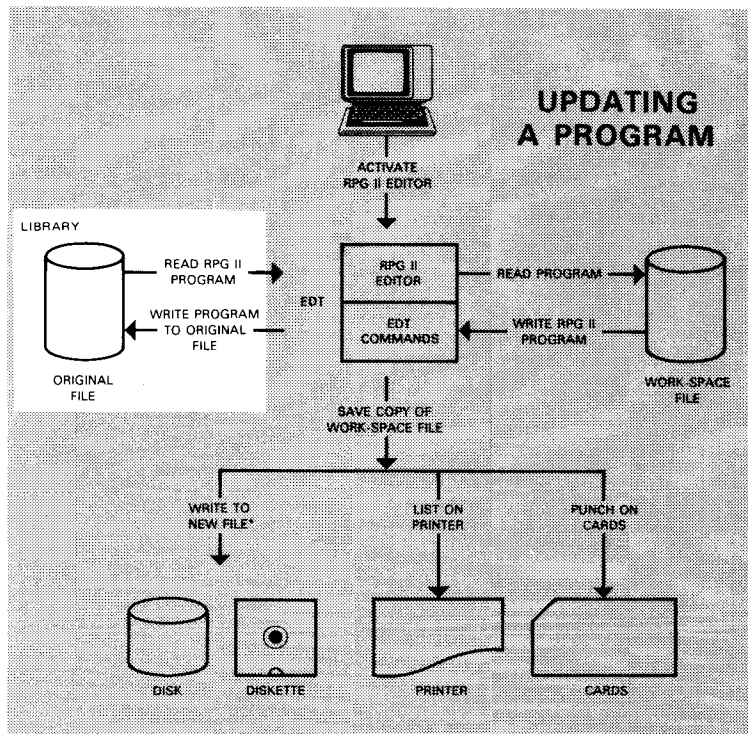
*RPG II programs must be written to SAT files to be compiled.

Figure 1-2. Updating an RPG II Program

1.3. WORKSTATION CONSIDERATIONS

Logon

Interactive program

Since the RPG II editor is an interactive program, you access it via the workstation. If you are the first user, turn on the workstation and then log on. The logon procedure is described in full detail in the interactive services commands and facilities user guide, UP-8845 (current version).

Screen Display

Protected fields are unalterable

Screens displayed by the RPG II editor contain protected and unprotected fields. The protected fields are unalterable fields.

Unprotected fields are used for entries

Entries may be made in unprotected fields. When an entry is made in an unprotected field on the screen, the underscores are replaced by the entry. If the entry does not completely fill the field, the entry is padded with blank spaces where the underscores are. This is helpful to show whether an entry is left-justified or right-justified within the field.

Moving the Cursor

Keys that move the cursor

During an RPG II editor session, the cursor can be repositioned on the workstation screen by pressing any of the following workstation keys:

- Tab keys (to tab forward and tab backward)

- Cursor scan keys [→] [←] [↑] [↓]

- Space bar (to type a space)

- Cursor-to-home key (to reposition the cursor to the top left corner of the workstation screen)

2. Using the RPG II Editor

2.1. ACTIVATING THE RPG II EDITOR

Using the RPG II editor with EDT

Because the RPG II editor is actually a specialized language editor (subeditor) of EDT, the RPG II editor can be used only while EDT is activated.

Activating both EDT and the RPG II editor

If EDT is not already activated, you activate both EDT and the RPG II editor by keying in ☞ **EDT @RPG**

Activating the RPG editor only

If you are already in an EDT session, you key in ☞ **@RPG**

2.2. INITIAL DISPLAY

First workstation screen

Once the RPG II editor is activated, the first workstation screen you see is the initial display. This screen identifies the release level and version number of the RPG II editor that you are using. You also use this screen to select the starting conditions for your RPG II editor session.

INITIAL DISPLAY

```
                    RPGEDT VERSION XX.XX/XX
        SELECT MODE   (C)
        C = CREATE     U = UPDATE
        SELECT FORMAT TYPE    (1)
         1 = POSITIONAL     2 = FORMATTED     3 = FREEFORM
        SPECIFICATION TYPE DISPLAY?  (Y)     Y = YES     N = NO
```

Screen uses

Use this screen to specify:

1. whether the RPG II editor will be used to create or update an RPG II source program;

2. which display format will be used to display the specification screens; and

3. whether the SPECIFICATION TYPE DISPLAY will be displayed next to show you the types of screens available with the RPG II editor.

Default values

No entries need to be specified if you want to use the following default values:

- create mode;

- positional format; and

- no specification type display.

Selecting all
default values

Select all the default values by moving the cursor past the N that is displayed and then pressing the transmit (XMIT) key. After the screen is transmitted, a header specification screen is displayed.

Selecting Create Mode

Used to create
program statements

Keyin C on the
initial display

C

Select create mode on the initial display when you want to create a new RPG II source program. To select create mode, move the cursor past the displayed C to the next unprotected field. Create mode (being the default value) is automatically selected for you. Remember, the cursor may be moved by pressing the tab key or the appropriate cursor scan keys.
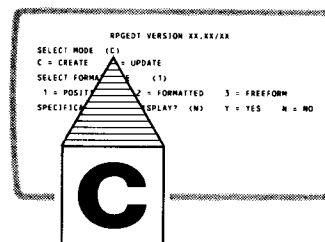
Selecting Update Mode

Used to update
program statements

Keyin U on the
initial display

U

Select update mode (command mode) on the initial display when you want to update an existing RPG II source program. To select update mode, key in U over the displayed C on the initial display.
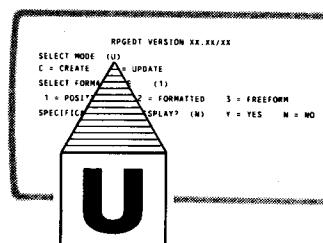
Selecting a Format Type

Available formats
for specification
screens

The RPG II editor gives you the option of selecting any one of three formats in which to display your specification screens.

3	Formats	
1 Positional	**2** Formatted	**3** Free-form
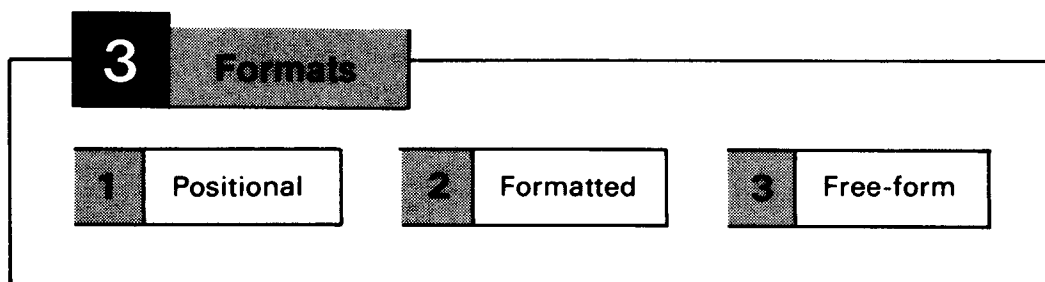
Key in 1, 2, or 3

You select a format type by keying in its number: 1, 2, or 3. Because 1 is the default value, you may select that just by moving the cursor to the next unprotected field.

Duration of use

Once you select a format, it will be used for all specification screens until a different format is selected with an @FORMAT command. (See 4.2 for instructions.)

Available specification screens

Appendix A contains all the available specification screens in formatted, positional, and free-form formats.

FORMATTED FORMAT
HEADER SPECIFICATION SCREEN

```
                                                              LINE -    1.0000
   1 SEQUENCE NUMBER: _____      6 FORM TYPE  H

   7 COMPILATION MODE: _         8 ERROR DUMP: _       9. OPERATOR CONTROL: _

  15 DEBUG: _                   18 CURRENCY SYMBOL: _

  21 INVERTED PRINT: _          26 ALTSEQ: _

  31 BINARY SEARCH: _           40 SIGN HANDLING: _

  41 FORMS ALIGNMENT: _         42 INDICATOR INIT.: _

  43 FILE TRANSLATION: _        70 CCA NAME: ____

  74 SUBROUTINE: _              75 PROGRAM ID: _____

     NEXT SPECIFICATION TYPE, ST, OR CMD:    (___)
  ----------------------------------------------------------------------------
  ----------------------------------------------------------------------------
```

Template format

Shows field names and starting column positions

The *formatted* format is a template display format designed for the novice user. When a specification screen is displayed in this format, each new field on the specification screen is labeled by its name and beginning column number. There is a different screen for each specification type.

POSITIONAL FORMAT
HEADER SPECIFICATION SCREEN

```
                                                              LINE -    1.0000
                          1 1 2 2 3 4 4 4 4 7     7 7
     1       6 7 8 9 5 8 1 6 1 0 1 2 3 0     4 5
     ----- H - - - - - - - - - - - ---- - ------
     NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
  ----------------------------------------------------------------------------
  ----------------------------------------------------------------------------
```

Tabular format

Shows starting column positions only

The *positional* format is a tabular display format designed mainly for the experienced user. When a specification screen is displayed in positional format, each new field on the specification screen is identified by its beginning column number. There is a different screen for each specification type.

FREE-FORM FORMAT
SPECIFICATION SCREEN

```
                                                          LINE -    1.0000
          1          2          3          4          5          6
1234567890123456789012345678901234567890123456789012345678901234
----------------------------------------------------------------
                                          1          1          1
6      7       8         9         0          1          2
5678901234567890123456789012345678901234567890123456789012345678
----------------------------------------------------------------

NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
----------------------------------------------------------------
----------------------------------------------------------------
```

Display format

Shows 128 column positions

> The *free-form* format is a display format also designed for the experienced user. It is a 2-line (64 characters/line) format composed of 128 column numbers with an underline under each for data to be keyed in. The same screen is used for all specification types.

Entering data and comments

You also select a free-form screen to enter data or comments when using positional or formatted format.

Selecting the Specification Type Display

The third question on the initial display asks whether you want to see the specification type display immediately after the initial display. You select N if you don't want to see the specification type display by moving the cursor past the displayed N (the default value); or you select Y (by keying in Y over the displayed N) if you want to see the display.

Selecting

N or **Y**

SPECIFICATION TYPE DISPLAY

```
ENTER SPECIFICATION TYPE:   (__)

                    ····  SPECIFICATION TYPES  ····
         H  - HEADER          IF - INPUT FIELD          A  - ALTSEQ
         F  - FILE            C  - CALCULATION          FT - 'FILES
         E  - EXTENSION       OR - OUTPUT FILE          EQ - 'EQUATE
         L  - LINE COUNTER    OF - OUTPUT FIELD         FF - FREE FORM/
         IR - INPUT RECORD    T  - TELECOMMUNICATION         COMMENTS
         AU - AUTO REPORT     AC - AUTO REPORT (/COPY)  ·· - TABLE/ARRAY
              OPTIONS                                        DELIMITER
```

**Types of available
specification screens**

The specification type display lists all the types of screens the RPG II editor provides for you to use when creating an RPG II source program. Table 2-1 lists the use of each type of display.

**Requesting the first
specification type**

When the specification type display is displayed immediately after the initial display, you may key in an entry in the provided underlines, or leave the field blank and then move the cursor to the bottom of the screen and press the transmit key. If you leave the field blank, the header specification screen will be displayed next. Otherwise, the type of screen you specified will be displayed.

NOTE:

You may request to see the specification type display periodically throughout an RPG II editor session. In this case, when the display is presented, you may specify the next screen by keying in the appropriate abbreviation in the provided underlines and then transmitting the selection.

Transmitting the Initial Display Selections

Changing selections

Once you make all your selections on the initial display, make sure they are correct. If you want to change any of them, you may do so at this time by positioning the cursor over the selections you want to change and then keying in the new selections. When your selections are correct, move the cursor to the bottom of the workstation screen (past the last selection on the initial display).

**Transmitting
selections**

Then press the transmit key (XMIT) to transfer your selections to the RPG II editor.

XMIT

Table 2-1. Uses for RPG II Editor Screens

Screen Type	Abbreviation	Use
HEADER	H	To enter a control specification in your program
FILE	F	To enter a file specification in your program
EXTENSION	E	To enter a file extension specification in your program
LINE COUNTER	L	To enter a line counter specification in your program
INPUT RECORD	IR	To enter the first 42 columns of an input specification in your program
INPUT FIELD	IF	To enter columns 1-5 and 43-70 of an input specification in your program
CALCULATION	C	To enter a calculation specification in your program
OUTPUT FILE	OR	To enter the first 31 columns of an output specification in your program
OUTPUT FIELD	OF	To enter columns 1-5 and 23-70 of an output specification in your program
TELECOMMUNICATION	T	To enter a telecommunication specification in your program
AUTO REPORT	AU	To enter an auto report option specification in your program
AUTO REPORT (/COPY)	AC	To enter auto report (/COPY) statements in your program
TABLE/ARRAY DELIMITER	**	To input data to your program in a table or array
ALTSEQ	A	To use an alternate collating sequence in your program
*FILES	FT	To input file translation table records for all files in your program
*EQUATE	EQ	To input file translation table records for selected files in your program
FREE-FORM/ COMMENTS	FF	To include data, comments, RPG II compiler directives, or any source statements in your program

NOTES:

1.   When ** (TABLE/ARRAY DELIMITER) is selected on the specification type display, ** is automatically placed in the work-space file to designate the subsequent entries as data. Free-form screens are then displayed for you to enter data.

2.   For more information on where to enter RPG II statements and data on these screens, see the corresponding topic in the OS/3 RPG II user guide, UP-8067 (current version).

3.   The RPG II editor cannot be used to enter an '&' option in a /COPY modifier statement for a file description specification. To do this, you must terminate the RPG II editor with @RPG END and enter the statement using EDT.

The Next Display after the Initial Display

The type of screen that is displayed after the initial display depends on the entries you made on the initial display. The possibilities for the next display and the kind of entry you can make on it are as follows:

Entries	Next Display	Next Entry
C 1 or 2 N	Header specification screen	Key in source statement.
C 3 N	Free-form specification screen	Key in source statement.
U 1, 2, or 3 N	Work-space line number 1.0000	Key in EDT command.
C 1, 2, or 3 Y	Specification type display	Place cursor at bottom of screen and transmit. Key in source statement on next screen.
U 1, 2, or 3 Y	Specification type display	Place cursor at bottom of screen and transmit. Key in EDT command on line 1.0000.

2.3. TERMINATING THE RPG II EDITOR

Returning to
update mode

You can terminate the RPG II editor by returning to update mode and keying in either of the following commands to the right of the current work-space line number:

Keyin choices

      @RPG END   or   @HALT

@RPG END

```
23.0000▷@RPG END
```

Key in @RPG END to terminate the RPG II editor but not EDT.

@HALT

```
23.0000▷@HALT
```
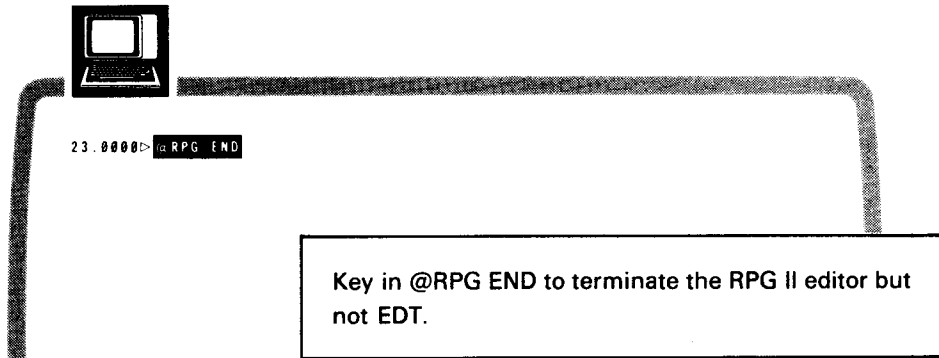
Key in @HALT to terminate RPG II editor and EDT.
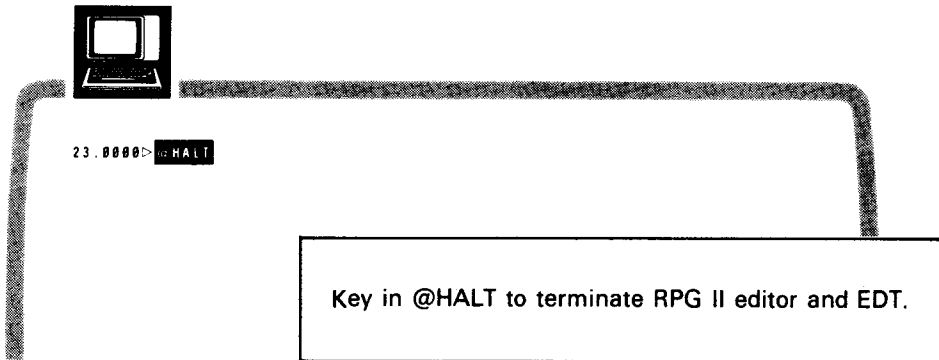
3. Creating an RPG II Source Program (Create Mode)

3.1. USING FORMATTED FORMAT

Selecting Formatted Format

Selections on the initial display

required

C and **2**

optional

N or **Y**

The first specification screen
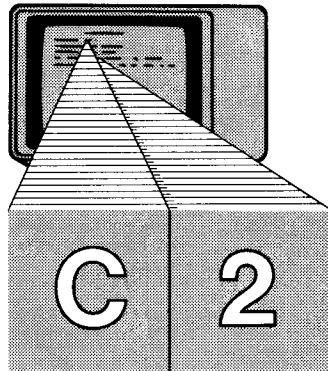
If you want to create an RPG II program using the formatted format, you must select C and 2 on the initial display. You may also select N or Y depending on whether or not you want to see a list of the specification types. The first specification screen displayed after the initial display (other than the specification type display) is always a header specification screen when in create mode.

Filling in a Formatted Screen

Positioning the cursor
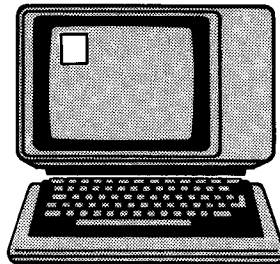
Changing the line number

When any specification screen is first displayed, the cursor is in the top left corner of the screen. At this time, you may position the cursor over the line number in the top right corner of the screen to change the line number, if necessary. To change the line number, you key in the new line number over the existing one. (The current work-space line number can be a number between 0.0001 and 9999.9999.)

Advancing the cursor

Keying in the statement

After the line number is set, you must then advance the cursor to the field on the specification screen where you want to enter your statement. You may then key in your statement on the appropriate underlines. Remember, to advance the cursor from field to field (if it doesn't automatically advance to the correct field), press the tab key, space bar, or the appropriate cursor scan keys.

Checking for
errors

Specifying the next
screen

After you finish keying in your statement, check it over for errors. If you want to change any input, reposition the cursor to the field you want to change and then key in the new entry. When the statement is correct, move the cursor to the next screen identification line:

NEXT SPECIFICATION TYPE, ST, OR CMD: (_ _ _)

Here, you specify what is to be displayed next. You have four choices:

Entry   **Next Display**

Selected specification screen

NOTE:

The entry for this option is one specification type abbreviation (Table 2—1).

Specification type display

Next work-space line number with RPG II editor in EDT command mode

New specification screen of the same type

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.

Transmitting the Screen

Transmitting the completed screen

XMIT

Transmitting the last source statement

CMD

After you make your choice, press the transmit (XMIT) key. This enters your source statement in the work-space file and tells the RPG II editor what to display next. Continue filling in the screens and transmitting them until all of your source statements are keyed in. Key in CMD on the next screen identification line of the specification containing your last source statement. This places the RPG II editor in the update mode (EDT command mode) so you can store your program in a program library, list it on the printer, display it on the workstation screen, or alter it.

Temporarily Changing to Free-Form Format

Uses for a free-form screen

Selecting specification type FF

Selecting the next specification screen

When using the formatted format type, you may occasionally need a free-form screen to enter a comment, data, RPG directives, or even another source statement. There is a specification type called free form/comments (abbreviated FF), which you can use in this situation to temporarily change the format type. When you enter FF on the next screen identification line of a formatted screen, the next screen will be in free-form format. There you can key in the comment, data, or source statement. (See free-form format under selecting format type in 2.2 for instructions.) Before transmitting the screen, move the cursor to the next screen identification line on the free-form screen. You then have five choices for the field there:

5 OPTIONS

Entry   Next Display

▢

Formatted format screen for the same specification type specified in column 6 of the currently displayed free-form screen

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.

FF

Another free-form screen (You must specify FF as the next specification type as many times as you wish. When you specify another type, however, the editor will use formatted format for the next screen.)

FF (crossed out)

Selected specification screen in formatted format

NOTE:

For this specification option, use any statement specification type other than FF.

ST

Specification type display

CMD

Current work-space line number with the editor in update mode (EDT command mode). The current display format type remains formatted.

Under the following formatted display example, we illustrate the use of a free-form screen to enter a comment in a program.

NOTE:

If you want to change the current display format type for all specification types during the creation of an RPG II program, enter the update mode (key in CMD on the next screen identification line of a specification screen) and then key in an @FORMAT command. The @FORMAT command is discussed in 4.2.

An Example Using Formatted Format

The following example shows you how you can create an RPG II source program via specification screens displayed in formatted formats. This example explains the creation of an RPG II program beginning with the initial display, traveling through the input of source statements on the specification screens, and ending with storing the program on a program library.

INITIAL DISPLAY

SCREEN 1

```
                    RPGEDT VERSION XX.XX/XX
        SELECT MODE   (C)
        C = CREATE    U = UPDATE
        SELECT FORMAT TYPE    (2)
          1 = POSITIONAL    2 = FORMATTED    3 = FREEFORM
        SPECIFICATION TYPE DISPLAY?  (N)    Y = YES    N = NO
```

In this example, we select C, 2, and N; then we transmit these selections to the RPG II editor by pressing the XMIT key. A header specification screen in formatted format is automatically displayed.

HEADER SPECIFICATION

SCREEN 2

```
                                                      LINE -    1.0000
         1 SEQUENCE NUMBER:  __0.0      6 FORM TYPE   H

         7 COMPILATION MODE:_          8 ERROR DUMP: D      9 OPERATOR CONTROL: _

        15 DEBUG: I                   18 CURRENCY SYMBOL: _

        21 INVERTED PRINT: _          26 ALTSEQ: _

        31 BINARY SEARCH: _           40 SIGN HANDLING: _

        41 FORMS ALIGNMENT: _         42 INDICATOR INIT.: _

        43 FILE TRANSLATION: _        70 CCA NAME: ____

        74 SUBROUTINE: _              75 PROGRAM ID: _____

           NEXT SPECIFICATION TYPE, ST, OR CMD:    (FF )
        -------------------------------------------------------------------
        -------------------------------------------------------------------
```

Here, we key in our source statement on the appropriate underlines. On the next screen identification line, we specify what is to be displayed next. Because we want to enter a comment with this statement, we specify the free-form screen as the next display by keying in FF. Then we press the XMIT key. A free-form format is displayed.

FREE-FORM FORMAT

SCREEN 3

```
                                                      LINE -    2.0000
            1         2         3         4         5         6
   12345678901234567890123456789012345678901234567890123456789 01234
   ------- F*THIS IS A PERSONNEL PROGRAM -------------------------------
                                           1         1         1
    6    7         8         9         0         1         2
   5678901234567890123456789012345678901234567890123456789012345678

   NEXT SPECIFICATION TYPE ST.OR CMD:   (F__)
   -----------------------------------------------------------------------
   -----------------------------------------------------------------------
```

We key in the comment: THIS IS A PERSONNEL PROGRAM. Note that an asterisk (*) is entered in column 7 to identify our statement as a comment, and a valid form type abbreviation (F in this case) is entered in column 6 to conform to RPG II programming language specifications. On the next screen identification line, again we specify what is to be displayed next. Because we keyed in F on the next screen identification line, after we transmit the screen, a file specification (in formatted format) is displayed.

FILE SPECIFICATION

SCREEN 4

```
                                                      LINE -    3.0000
   1 SEQUENCE NUMBER: __020  6 FORM TYPE F  7 FILENAME:CDFILE__   15 FILE TYPE:I
   16 FILE DESIGNATION: P  17 EOF: E  18 SEQ: A  19 FILE FORMAT: F  20 BLK LEN:____
   24 RECORD LEN: ____  28 FILE PROCESSING MODE: _  29 KEY OR FIELD LENGTH: __
   31 RECORD ADDRESS TYPE: _  32 FILE ORGANIZATION: _  33 OVERFLOW INDICATOR: __
   35 KEY FLD STARTING LOC: ____  39 EXTENSION/LINE CTR CODE: _  40 DEVICE: READER_
   53 CONT LINES: _  54 OPTION: _____  60 ENTRY/STORAGE/KEY LOC: _____
   66 FILE ADDITION AND CYL OVF % /KEY-LENGTH: __  68 # OF EXTENTS/KEY OPTIONS: __
   70 REWIND: _          71 FILE CONDITIONERS: __
                 NEXT SPECIFICATION TYPE. ST. OR CMD: (___)
   -----------------------------------------------------------------------
   -----------------------------------------------------------------------
```

Again, we key in our source statement, specify the next screen display, and transmit. Because we transmitted the screen without keying in an entry on the next screen identification line, another file specification screen in formatted format is displayed.

FILE SPECIFICATION

```
                                                          LINE -    4.0000
  1 SEQUENCE NUMBER: __030   6 FORM TYPE F   7 FILENAME:PRFILE__    15 FILE TYPE:O

 16 FILE DESIGNATION: _   17 EOF: _   18 SEQ: _   19 FILE FORMAT: F   20 BLK LEN: ____

 24 RECORD LEN: _132  28 FILE PROCESSING MODE: _   29 KEY OR FIELD LENGTH: __

 31 RECORD ADDRESS TYPE: _   32 FILE ORGANIZATION: _   33 OVERFLOW INDICATOR: __

 35 KEY FLD STARTING LOC: ____   39 EXTENSION/LINE CTR CODE: _   40 DEVICE: PRINTER

 53 CONT LINES: _   54 OPTION: _____   60 ENTRY/STORAGE/KEY LOC: _____

 66 FILE ADDITION AND CYL OVF % /KEY-LENGTH: __   68 # OF EXTENTS/KEY OPTIONS: __

 70 REWIND: _              71 FILE CONDITIONERS: __

              NEXT SPECIFICATION TYPE, ST, OR CMD: (ST )
```

Like the previous file specification screen, again we key in our source statement, specify the next screen display, and then press the XMIT key. Because we keyed in ST on this screen, the next screen we see is the specification type display.

SPECIFICATION TYPE DISPLAY

```
ENTER SPECIFICATION TYPE:  (IR)
                    ····   SPECIFICATION TYPES   ····
      H  - HEADER          IF - INPUT FIELD        A  - ALTSEQ
      F  - FILE            C  - CALCULATION        FT - 'FILES
      E  - EXTENSION       OR - OUTPUT FILE        EQ - 'EQUATE
      L  - LINE COUNTER    OF - OUTPUT FIELD       FF - FREE FORM/
      IR - INPUT RECORD    T  - TELECOMMUNICATION       COMMENTS
      AU - AUTO REPORT     AC - AUTO REPORT (/COPY)  ·· - TABLE/ARRAY
           OPTIONS                                      DELIMITER
```

We now select the specification screen to be displayed next. We key in the appropriate abbreviation and then press the XMIT key. Because we keyed in IR, an input specification screen (record identification) is displayed.

INPUT SPECIFICATION
RECORD IDENTIFICATION

SCREEN
7

```
                                                          LINE -    5.0000
1 SEQUENCE NUMBER: __040        6 FORM TYPE  IR (RECORD ID)

7 FILENAME: CDFILE_  14 _    15 SEQUENCE: AA      17 NUMBER: _   18 OPTION: _

19 RECORD IDENTIFYING INDICATOR. DS. OR '': 01

    RECORD                      POSITION    N · NOT  C/Z/D    CHARACTER

    IDENTIFICATION              21 __15     25 N     26 C      27  1

    CODES                       28 ____     32 _     33 _      34 _

                                35 ____     39 _     40 _      41 _

42 STACKER SELECT: _

              NEXT SPECIFICATION TYPE, ST, OR, CMD:    (If_ )
```

NOTE:

The remainder of the specification screens in this example (except for the final screen) operate in this manner. We key in the source statement, specify the next screen display (on the next screen identification line), and then transmit.

INPUT SPECIFICATION
FIELD DESCRIPTION

SCREEN
8

```
                                                          LINE -    6.0000
 1 SEQUENCE NUMBER: __050       6 FORM TYPE   IF

43 DATA FORMAT: _               44 FROM: __15    48 TO: __45

52 DECIMAL POSITIONS: _         53 FIELD NAME: NAMEIN

59 CONTROL LEVEL: __            61 MATCHING/CHAINING FIELDS: __

63 FIELD RECORD RELATION: __

65 PLUS: __       67 MINUS: __       69 ZERO/BLANK: __

    NEXT SPECIFICATION TYPE, ST, OR CMD: (C__ )
```

CALCULATION SPECIFICATION

```
                                                          LINE -    7.0000
 1 SEQUENCE NUMBER: __060        6 FORM TYPE  C   7 CONTROL LEVEL: __

   INDICATORS: 9: ___   12: ___   15: ___              18 FACTOR 1: _____

28 OPERATION: MOVE      33 FACTOR 2: NAMEIN____         43 RESULT FIELD: NAMEOT

49 RESULT FIELD LENGTH: _31     52 DECIMAL POSITIONS: _      53 HALF ADJUST: _

                         ARITHMETIC:      PLUS          MINUS    ZERO
                         COMPARE   :      1>2           1<2      1=2
               LOOKUP (FACTOR 2) IS:      HIGH          LOW      EQUAL
   RESULTING INDICATORS:                  54 __         56 __   58   __

60 COMMENTS:_____      NEXT SPECIFICATION TYPE, ST, OR CMD: (OR )
--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

OUTPUT SPECIFICATION
FILE IDENTIFICATION AND CONTROL

```
                                                          LINE -    8.0000
 1 SEQUENCE NUMBER: __070        6 FORM TYPE  OR

 7 FILENAME: PRFILE              14 _ (AND/OR COL 14-16)

15 TYPE (H/D/T/E): D             16 STACKER/FETCH: _ (ADD/DEL COL 16-18)

17 SPACE BEFORE/AFTER: 1_        19 SKIP BEFORE: __

21 SKIP AFTER: __

   OUTPUT INDICATORS: 23_01  26 ___   29 ___

32 AUTO REPORT (*AUTO): _____

   NEXT SPECIFICATION TYPE, ST, OR CMD:  (OF )
--------------------------------------------------------------------------
--------------------------------------------------------------------------
```

SCREEN 11

OUTPUT SPECIFICATION
FIELD DESCRIPTION AND CONTROL

```
                                                          LINE -    9.0000
    1 SEQUENCE NUMBER:  __080      6 FORM TYPE  OF

      OUTPUT INDICATORS - 23 ___ 26 ___  29 ___
   32 FIELD NAME: NAMEOT
   38 EDIT CODES: _
   39 BLANK AFTER: B
   40 END POSITION: __48
   44 DATA FORMAT: _
   45 CONSTANT OR EDIT WORD: _____
      NEXT SPECIFICATION TYPE, ST, OR CMD:   (CMD)
   ------------------------------------------------------------------------
   ------------------------------------------------------------------------
```

On the last output screen, we key in the final statement of the program. However, this time instead of specifying a particular specification screen or the specification type display on the next screen identification line, we key in CMD. This puts the RPG II editor into the update mode (EDT command mode) so we can store our program. Then we press the XMIT key. The next work-space line (10.0000) is then displayed by itself on the next screen.

END

UPDATE MODE

```
10.0000▷@WRITE MO=MYPROG,FIL=LISTFILE,VSN D00029,SA YES,SI 2
10.0000▷@RPG END
```

On this screen, we first key in an EDT @WRITE command to store a copy of our newly created RPG II source program. We store it as the module MYPROG on the program library LISTFILE (which is a SAT file) on disk volume D00029. For more detailed information on how to save newly created source programs, see the WRITE command in the general editor user guide/programmer reference, UP-8828 (current version). After the @WRITE command is executed, we terminate the RPG II editor session by keying in the command @RPG END.

3.2. USING POSITIONAL FORMAT

Selecting Positional Format

Selections on the initial display

required

🄲 and 🄻

optional

🄽 or 🆈
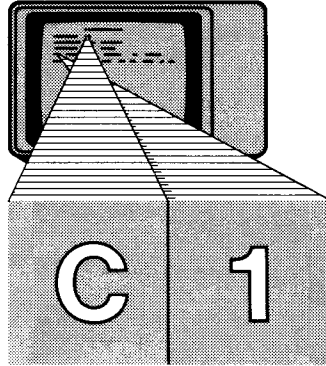
For the most part, the procedure for creating RPG II source programs using positional format is exactly the same as using formatted format except that different screens are used. If you want to create an RPG II program using positional format, you must select C and 1 (the default values) on the initial display. You may also select N or Y, depending on whether or not you want to see a list of the specification types. The first specification screen displayed after the initial display (other than the specification type display) is always a header specification screen when in create mode.

Filling in a Positional Screen

Positioning the cursor

Changing the line number

When any specification screen is first displayed, the cursor is in the top left corner of the screen. At this time, you may position the cursor over the line number in the top right corner of the screen to change the line number, if necessary. To change the line number, you key in the new line number over the existing one. (The current work-space line number can be a number between 0.0001 and 9999.9999.)

Keying in the statement

After the line number is set, you must then advance the cursor to the field on the specification screen where you want to enter your statement. You may then key in your statement on the appropriate underlines. Remember, to advance the cursor from field to field (if it doesn't automatically advance to the correct field), press the tab key, space bar, or the appropriate cursor scan keys.

Checking for errors

After you finish keying in your statement, check it over for errors. If you want to change any of your input, reposition the cursor to the field you want to change and then key in the new entry. When the statement is correct, move the cursor to the next screen identification line on the specification screen:

Specifying the next display

NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

Here, you specify what is to be displayed next. You have four choices:

4 OPTIONS

Entry   **Next Display**

**H
:
FF**   Selected specification screen

NOTE:

The entry for this option is one specification type abbreviation (Table 2—1).

ST   Specification type display

CMD   Next work-space line number with RPG II editor in EDT command mode

☐   New specification screen of the same type

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.
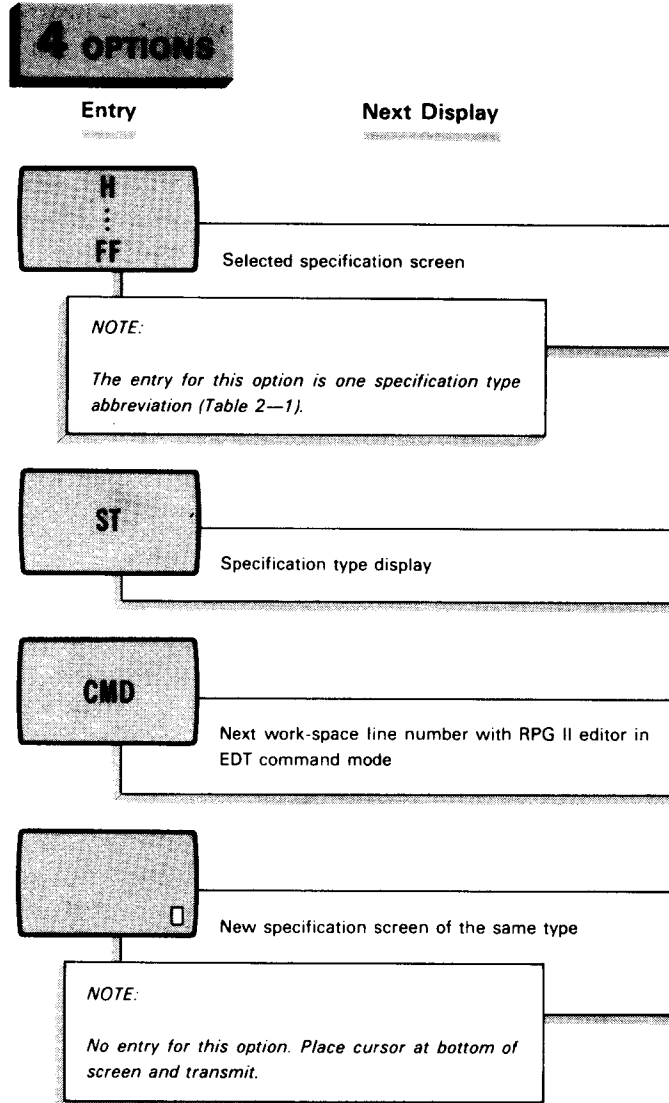
Transmitting the Screen

Transmitting the completed screen

XMIT

After you make your choice, press the transmit (XMIT) key. This enters your source statement in the work-space file and tells the RPG II editor what to display next.

Transmitting the last statement

CMD

Continue filling in screens and transmitting them until all of your source statements are keyed in. On the last specification screen containing your final source statement, key in CMD on the next screen identification line. This places the RPG II editor in the update mode (EDT command mode) so you can store your program in a program library, list it on the printer, display it on the workstation screen, or alter it.

Temporarily Changing to Free-Form Format

Uses for a free-form screen

Selecting specification type FF

Selecting the next specification type

When using the positional format type, you may occasionally need a free-form screen to enter a comment, data, RPG directives, or even another source statement. There is a specification type called free form/comments (abbreviated FF), which you can use in this situation to temporarily change the format type. When you enter FF on the next screen identification line of a positional screen, the next screen will be in free-form format. There you can key in the comment, data, or source statement. (See free-form format under selecting format type in 2.2 for instructions.) Then move the cursor to the next screen identification line of the free-form screen. For the entry field, you then have five choices:

5 OPTIONS

Entry   **Next Display**

[□]

Positional format screen for the same specification type specified in column 6 of the currently displayed free-form screen

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.

[FF]

Another free-form screen (You must specify FF as the next specification type as many times as you wish. When you specify another type, however, the editor will use formatted format for the next screen.)

[FF] (crossed out)
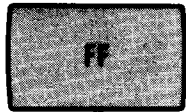
Selected specification screen in positional format

NOTE:

For this specification option, use any statement specification type other than FF.

[ST]

Specification type display (The current display format type remains positional.)
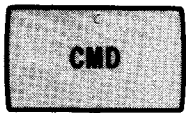
[CMD]

Current work-space line number with the editor in update mode (EDT command mode). The current display format type remains positional.

NOTE:

If you want to change the current display format type for all specification types during the creation of an RPG II program, enter the update mode (key in CMD on the next screen identification line) and then key in an @FORMAT command. The @FORMAT command is discussed in 4.2.

An Example Using Positional Format

The following example shows you how you can create an RPG II source program via specification screens displayed in positional format. This example explains the creation of an RPG II program beginning with the initial display, traveling through the input of source statements on specification screens, and ending with storing the program on a program library. Note that the program created here is the same as the one created in 3.1 when we created a program via specification screens displayed in formatted format. You may want to compare the two methods.

INITIAL DISPLAY

SCREEN 1

```
                    RPGEDT VERSION XX.XX/XX
SELECT MODE    (C)
C = CREATE    U = UPDATE
SELECT FORMAT TYPE    (1)
 1 = POSITIONAL    2 = FORMATTED    3 = FREEFORM
SPECIFICATION TYPE DISPLAY?  (N)    Y = YES    N = NO
```

For this example, we select the default values C, 1, and N; then we transmit these selections to the RPG II editor by pressing the XMIT key. (Note that we have to move the cursor past the displayed N before we transmit the selections.) Because the default values are selected, the next screen is a header specification screen in positional format.

HEADER SPECIFICATION

SCREEN 2

```
                                                    LINE -    1.0000
                1 1 2 2 3 4 4 4 4 7    7 7
1    6 7 8 9 5 8 1 6 1 0 1 2 3 0    4 5
__010 H _ 0 _ 0 _ _ _ _ _ _ _ _ ____ _ _____
NEXT SPECIFICATION TYPE, ST, OR CMD:  (ST )
```

Now, we key in our source statement on the appropriate underlines. On the next screen identification line, we specify what is to be displayed next. ST on this screen indicates that the next screen is to be the specification type display. When we press the XMIT key, a specification type display is displayed.

SCREEN 3

SPECIFICATION TYPE DISPLAY

```
ENTER SPECIFICATION TYPE:  (F )

            ····  SPECIFICATION TYPES  ····
 H  - HEADER           IF - INPUT FIELD         A  - ALTSEQ
 F  - FILE             C  - CALCULATION         FT - 'FILES
 E  - EXTENSION        OR - OUTPUT FILE          EQ - 'EQUATE
 L  - LINE COUNTER     OF - OUTPUT FIELD        FF - FREE FORM/
 IR - INPUT RECORD     T  - TELECOMMUNICATION        COMMENTS
 AU - AUTO REPORT      AC - AUTO REPORT (/COPY)  ** - TABLE/ARRAY
      OPTIONS                                         DELIMITER
```

> Here, we select the specification screen to be displayed next by keying in the appropriate abbreviation and pressing the XMIT key. Because we keyed in F, a file specification screen (in positional format) is displayed.

SCREEN 4

FILE SPECIFICATION

```
                                                LINE -   2.0000
                     1 1 1 1 1 2    2    2 2  3 3
 1      6 7          5 6 7 8 9 0    4    8 9  1 2
 __020 F CDFILE__    I P C A F____ ____ __ __ _

 3 3    3 4          5 5      6       6 6  7 7
 3 5    9 0          3 4      0    ,  6 7  8 0 1
 __ ____ _ READER__  _ _____ _____ _ __ _ __
 NEXT SPECIFICATION TYPE. ST. OR CMD:  (___)
```

> Again we key in the source statement, specify the next screen display, and transmit. Because we transmitted the screen without keying in an entry on the next screen identification line, another file specification screen (in positional format) is displayed.

FILE SPECIFICATION

SCREEN 5

```
                                                    LINE -    3.0000
                        1 1 1 1 1 2    2    2 2  3 3
        1      6 7      5 6 7 8 9 0    4    8 9  1 2
        __030 F PRFILE__ O _ _ _ F ____ 132 _ __ _ _
        3  3    3 4      5 5      6     6 6 6  7 7
        3  5    9 0      3 4      0     6 7 8  0 1
        __ ____ _ PRINTER _ _____ _____ _ _ __ _ __
        NEXT SPECIFICATION TYPE. ST. OR CMD:  ( IR )
```

As on the first file specification screen, we again key in the source statement, specify the next screen display, and transmit. Because we keyed in IR on this screen, the next screen is an input specification screen (record identification).

INPUT SPECIFICATION
RECORD IDENTIFICATION

SCREEN 6

```
                                            LINE -    4.0000
                        1 1 1 1
        1      6 7      5 7 8 9
        __040 IR CDFILE __ AA _ _ 01
        2   2  2    3   3   3   4
        1   5  8    2   5   9   2
        __15 NC1 ____ ___ ____ ___ _
        NEXT SPECIFICATION TYPE. ST. OR CMD:  ( IF )
```

NOTE:

The remainder of the specification screens in this example (except for the final screen) operate in this manner. We key in the source statement, specify the next screen display (on the next screen identification line), and press the XMIT key. Note that we advance the cursor (if it doesn't automatically advance to the appropriate field) by pressing the tab key. We can also use the space bar and cursor scan keys.

SCREEN 7

INPUT SPECIFICATION
FIELD DESCRIPTION

```
                                                        LINE -    5 0000

                    4 4     4     5 5       5  6  6   6
        1      6  3 4     8     2 3       9  1  3   5PMMZZ
        __050  IF  _  __15  __45  _ NAMEIN  __  __  __  _____

        NEXT SPECIFICATION TYPE. ST. OR CMD:  ( C __ )
```

SCREEN 8

CALCULATION SPECIFICATION

```
                                                        LINE -    6 0000

                                  1              2
        1      6  7  9   N   N   8-FACTOR 1 8-OP
        __060  C  __  ___  ___  ___  _____   MOVE

        3       4      4   5 5 5       6
        3-FACTOR 2 3-NAME  9    2 3 4PMMZZ  0  (COMMENTS)
        NAMEIN  ____  NAMEOT  _31  _  _  _____  _____

        NEXT SPECIFICATION TYPE. ST. OR CMD:   ( OR  )
```

SCREEN 9

OUTPUT SPECIFICATION
FILE IDENTIFICATION AND CONTROL

```
                                                        LINE -    7 0000

                          1 1 1 1 1  2  2           3
        1      6  7       5 6 7 8 9  1  3   N   N    2
        __070  OR PRFILE  __ D _ 1  _  __  __  _01  ___  ___  _____

        NEXT SPECIFICATION TYPE. ST. OR CMD:   ( OF _ )
```

SCREEN
10

OUTPUT SPECIFICATION
FIELD DESCRIPTION AND CONTROL

```
                                                    LINE -    8.0000
                   2            3        3 3 4    4
        1     6  3    N    N    2        8 9 0    4
        _ 080  OF. ___  ___  ___  NAMEOT   B  _ 48 _
        4
        5 (CONSTANT OR EDIT WORD)
        ----------------------------
        NEXT SPECIFICATION TYPE. ST. OR CMD:   (CMD)
```

On the output screen, we key in the final statement of the program. However, this time, instead of specifying a particular specification screen or the specification type display on the next screen identification line, we key in CMD. This puts the RPG II editor in update mode (EDT command mode) so we can store our program. Then we press the XMIT key. The next work-space line number (by itself) is then displayed on the next screen.

END

UPDATE MODE

```
9.0000▷ @WRITE MO MYPROG.FIL LSTFILE,VSN D00028.SA YES.SI 3
```

On this screen, we key in an EDT @WRITE command to store a copy of our newly created RPG II source program. We store it as module MYPROG on program library LSTFILE (which is a SAT file) on disk volume D00028.

3.3. USING FREE-FORM FORMAT

Selecting Free-Form Format

Selections on the initial display

required

C and **3**

optional

N or **Y**
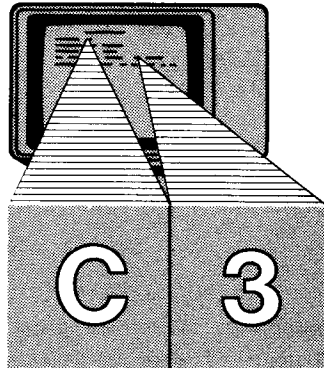
If you want to create an RPG II program using free-form format, you must select C and 3 on the initial display. You may also select N or Y, depending on whether or not you want to see a list of the specification types. Because the same free-form screen is used for all specification types, you should not need to see the specification type display.

Filling in a Free-Form Screen

Changing the line number
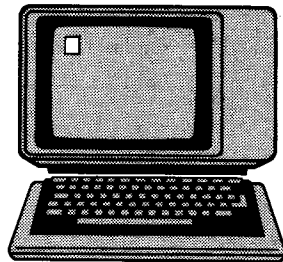
When the first free-form specification screen is displayed after the initial display, the cursor is in the top left corner of the screen. At this time, you may position the cursor over the line number in the top right corner of the screen to change the line number, if necessary. To change the line number, you key in the new line number over the existing one. (The current work-space line number can be a number between 0.0001 and 9999.9999.)

Uses for a free-form screen

After the line number is set, you must advance the cursor to the column position on the screen where you want to begin keying in your entry. This will depend on whether the entry is a source statement, a comment, or data.

*Entering source
statements*

When entering source statements, you must know the starting column position for each field; the free-form screen contains no field names or beginning column numbers to specify what each new field is and where it begins. Also, because the same free-form screen is displayed for all specification types, you must key in the form type (identified on the corresponding specification form) in column 6 on the screen. The valid entries are:

FORM TYPES

*Valid
form
types*

```
1 2 3 4 5 6
- - - - - H
```

H	Header Specification (Control Specification)	**E**	File Extension Specification
F	File Specification	**L**	Line Counter Specification
I	Input Specification		
C	Calculation Specification	**T**	Telecommunication Specification
O	Output Specification	**U**	Auto Report Options Specification

Entering comments

```
1 2 3 4 5 6 7
- - - - - H *
```

When entering comments, you must key in a valid form type in column 6, an asterisk (*) in column 7 (to indicate the entry is a comment and not a source statement), and then follow these with the comment. The comment is automatically assigned to the source statement preceding it.

Entering data

When entering data, simply key in the data on any of the provided underlines. However, be sure not to enter a valid form type abbreviation in column 6 because the RPG II editor will interpret the data as a source statement. Also, don't key in an asterisk in column 7 because the data will be interpreted as a comment.

Checking for errors

After you finish keying in your source statement, check it over for errors. If you want to change any input, reposition the cursor to the field you want to change and then key in the new entry. When the statement is correct, move the cursor to the next screen identification line on the free-form screen:

*Specifying the next
display*

NEXT SPECIFICATION TYPE, ST, OR CMD: (_ _ _)

Here, you specify what is to be displayed next. You have three possible entries:

3 OPTIONS

Entry   **Next Display**

ST

Specification type display

Another specification screen in free-form format

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.

CMD

Next work-space line number with RPG II editor in update mode (EDT command mode)

Transmitting the Screen

*Transmitting the
completed screen*

XMIT

*Transmitting the
last statement*

CMD

After you perform one of these options, press the XMIT key. This enters your source statement into the work-space file and tells the RPG II editor what to display next. Continue filling in screens and transmitting them until all of your source statements are keyed in. Key in CMD on the next screen identification line of the screen containing your last source statement. This places the RPG II editor in the update mode (EDT command mode) so you can store your program in a program library, list it on the printer, display it on the workstation screen, or alter it.

NOTE:

If you want to change your current display format type while you are creating your program, enter the update mode (key in CMD on the next screen identification line) and then key in an @FORMAT command. The @FORMAT command is discussed in 4.2.

An Example Using Free-Form Format

The following example shows you how you can create an RPG II source program via screens in the free-form format. This example explains the creation of an RPG II program starting with the initial display, traveling through the input of the source statements on the specification screens, and ending with storing the program on a program library. Note that the program created is the same as the examples in 3.1 and 3.2 when we created a program via specification screens displayed in formatted and positional formats. You may want to compare this method with the other two.

INITIAL DISPLAY

SCREEN 1

```
                        RPGEDT VERSION XX.XX/XX
        SELECT MODE   (C)
        C = CREATE      U = UPDATE
        SELECT FORMAT TYPE     (3)
        1 = POSITIONAL     2 = FORMATTED     3 = FREEFORM
        SPECIFICATION TYPE DISPLAY?  (N)      Y = YES    N = NO
```

In this example, we select C, 3, and N; then we transmit these selections to the RPG II editor by pressing the XMIT key. In turn, these selections cause a specification screen in free-form format to be displayed.

SCREEN
2

🖥 FREE-FORM SPECIFICATION

```
                                                              LINE -      1.0000
              1         2         3         4         5         6
    12345678901234567890123456789012345678901234567890123456789012341234
    _010H_D_____1_____
                                            1         1         1
    6     7        8        9        0         1         2
    567890123456789012345678901234567890123456789012345678901234567 8
    ----------------------------------------------------------------
    NEXT SPECIFICATION TYPE. ST. OR CMD: (___)
    ----------------------------------------------------------------
    ----------------------------------------------------------------
```

Here, the source statement is keyed in on the appropriate underlines. Because we key in H in column 6, this screen is now accepted as a header specification. When we move the cursor to the bottom of the screen and press the XMIT key, another free-form screen is displayed.

SCREEN
3

🖥 FREE-FORM SPECIFICATION

```
                                                              LINE -      2.0000
              1         2         3         4         5         6
    12345678901234567890123456789012345678901234567890123456789012341234
    _020FCDFILE__IPEAF_____READER_____
                                            1         1         1
    6     7        8        9        0         1         2
    567890123456789012345678901234567890123456789012345678901234567 8
    ----------------------------------------------------------------
    NEXT SPECIFICATION TYPE. ST. OR CMD: (___)
    ----------------------------------------------------------------
    ----------------------------------------------------------------
```

Next, we key in the source statement (along with an F in column 6). The F identifies our specification as a file specification. Then we move the cursor to the bottom of the screen and press the XMIT key. The next screen is another free-form screen.

SCREEN
4

FREE-FORM SPECIFICATION

```
                                                           LINE -    3.0000
                        1         2         3         4         5         6
              12345678901234567890123456789012345678901234567890123456789001234
              __030FPRFILE_O__F_____132_____PRINTER_____

                                                    1         1         1
               6    7         8         9          0         1         2
              56789012345678901234567890123456789012345678901234567890123456789012345678
              -----------------------------------------------------------------

              NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

              ----------------------------------------------------------------
              ----------------------------------------------------------------
```

> NOTE:
>
> The remainder of the screens in this example (except for the final screen) operate in the same manner. We key in the source statement (identifying the form type in column 6), move the cursor to the bottom of the screen, and transmit.

SCREEN
5

FREE-FORM SPECIFICATION

```
                                                           LINE -    4.0000
                        1         2         3         4         5         6
              12345678901234567890123456789012345678901234567890123456789001234
              __040ICDFILE__AA__01__15NC1_____

                                                    1         1         1
               6    7         8         9          0         1         2
              56789012345678901234567890123456789012345678901234567890123456789012345678
              -----------------------------------------------------------------

              NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

              ----------------------------------------------------------------
              ----------------------------------------------------------------
```

> NOTE:
>
> Only I (not IR or IF) is entered in column 6 to identify any input specification screen.

SCREEN 6

FREE-FORM SPECIFICATION

```
                                                          LINE -    5.0000
             1         2         3         4         5         6
    123456789012345678901234567890123456789012345678901234567890 1234
    _0501_____15__45_NAMEIN_____

                                         1         1         1
    6     7         8         9           0         1         2
    567890123456789012345678901234567890123456789012345678901 2345678
    --------------------------------------------------------

    NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
```

SCREEN 7

FREE-FORM SPECIFICATION

```
                                                          LINE -    6.0000
             1         2         3         4         5         6
    123456789012345678901234567890123456789012345678901234567890 1234
    _060C_____MOVE_NAMEIN____NAMEOT_31_____

                                         1         1         1
    6     7         8         9           0         1         2
    567890123456789012345678901234567890123456789012345678901 2345678
    --------------------------------------------------------

    NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
```

> **NOTE:**
>
> C is entered in column 6 to identify a calculation specification screen.

SCREEN 8

FREE-FORM SPECIFICATION

```
                                                          LINE -    7.0000
             1         2         3         4         5         6
    123456789012345678901234567890123456789012345678901234567890 1234
    _0700PRFILE_D_1_____01_____

                                         1         1         1
    6     7         8         9           0         1         2
    567890123456789012345678901234567890123456789012345678901 2345678
    --------------------------------------------------------

    NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
```

> **NOTE:**
>
> Only O (not OR or OF) is entered in column 6 to identify any output specification screen.

SCREEN
9

FREE-FORM SPECIFICATION

```
                                                          LINE -    8.0000
                 1         2         3         4         5         6
        1234567890123456789012345678901234567890123456789012345678901234
        __0800_____NAMEOT B 48_____

                                                      1         1         1
        6       7         8         9         0         1         2
        567890123456789012345678901234567890123456789012345678901234567 8

        ----------------------------------------------------------------
        NEXT SPECIFICATION TYPE, ST OR CMD: (CMD)
        -----------------------------------------------------------------
        -----------------------------------------------------------------
```

On this specification screen, we key in the final source statement of the program. However, instead of moving the cursor to the bottom of the screen and then pressing the XMIT key, we key in CMD on the next screen identification line of the screen and then transmit. This puts the RPG II editor in update mode (EDT command mode) so we can save the program. The next work-space line number (9.0000) by itself is then displayed.

END

UPDATE MODE

```
        9.0000▷ @WRITE MO MYPROG.FIL LIST.VSN D00031.TY S.SA YES.SI 2
        9.0000▷ @HALT
```

On this screen, we key in an EDT @WRITE command to store a copy of our newly created program. We store it as module MYPROG on program library LIST (which is a SAT file) on disk volume D00031.

After the @WRITE command is executed, the RPG II editor session (as well as the EDT session) is terminated by keying in the command @HALT.

4. Updating an RPG II Source Program (Update Mode)

4.1. PLACING THE RPG II EDITOR IN UPDATE MODE

When to change modes

You have created your program (in create mode), and you want to write it to a program library, update it, or update another program. You must place the RPG II editor in update mode (EDT command mode) in either of two ways, depending on whether the current display is the initial display or a specification screen.

Selecting Update Mode on the Initial Display

Completing the initial display

To select update mode on the initial display, key in U in place of the displayed C in the first field on the screen. In the next field, you choose the format type in which to display the statements you are going to update. The 1 for positional format is already displayed as the default value. To select formatted or free-form format, key in 2 or 3 in place of the displayed 1. You should not want to see the specification type display when using update mode. Therefore, leave the default value N in the third field. After making your selections, press the XMIT key to transmit your selections to the RPG II editor.

INITIAL DISPLAY

```
                   RPGEDT VERSION XX.XX/XX
SELECT MODE   (U)
C = CREATE     U = UPDATE
SELECT FORMAT TYPE    (1)
 1 = POSITIONAL    2 = FORMATTED    3 = FREEFORM
SPECIFICATION TYPE DISPLAY?  (N)    Y = YES    N = NO
```

The first work-space line number

The current work-space line number (by itself) is displayed on the next screen. In this event, that line number will always be 1.0000 because this is the beginning of a new RPG II editor session. When the line number is displayed, you may then begin the update session.

Loading the program into the work-space file

If you are updating an existing program, you must always begin by keying in an EDT @READ command to transfer a copy of the program you want to update from a program library to the work-space file. Once the program is loaded into the work-space file, you may then key in any of the EDT commands to update it.

Keying in EDT commands

These EDT commands (which are summarized in Appendix B) can be used to add, delete, and alter statements in your programs.

Switching from Create Mode to Update Mode

Keying in CMD

To switch the RPG II editor from create mode to update mode (EDT command mode), key in CMD on the next screen identification line on the currently displayed specification screen:

```
NEXT SPECIFICATION TYPE,ST,OR CMD:(CMD)
-------------------------------------------
-------------------------------------------
```
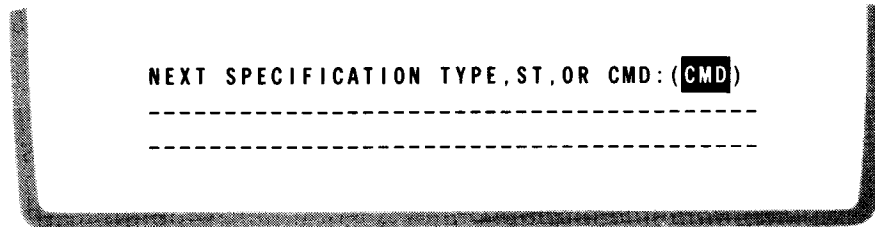
The current work-space line number

The next screen contains only the current work-space line number. This is the next higher line number after the one that appeared on the specification screen where you keyed in CMD.

Updating or saving the work-space file

When the line number is displayed, you may then begin the update session. At this time, you may either update source statements in the work-space file or save the content of the work-space file on disk or format label diskette.

Reading another program from disk

During update mode, you may also read a copy of another library program into the work-space file to update that. However, if you do not first save the content of the current work-space file, the source statements it contains will be lost when the new file is read.

Using EDT commands

Once you are ready to update the program, you may begin by keying in any of the EDT commands that play an important role in the update of a program. Again, you should understand the functions of these commands and how to use them in order to successfully update your own RPG II programs. For your convenience, the EDT commands used more often are summarized in Appendix B. However, for a detailed explanation of each command, see the OS/3 general editor user guide/programmer reference, UP-8828 (current version).

4.2. SPECIAL COMMANDS

Three special
function commands

The three commands (one RPG II editor command and two enhanced EDT commands) that perform special functions when used with the RPG II editor, are:

Command	Definition
@FORMAT	Lets you change the display format type and/or switch the RPG II editor from update mode to create mode.
@PRINT	Lets you display specified lines in your work-space file in positional, formatted, or free-form format.
@UPDATE	Lets you display, and then change, a source statement in the work-space file.

Changing the display
format type (aligned with @FORMAT row)

Displaying
statements (aligned with @PRINT row)

Updating
statements (aligned with @UPDATE row)

Changing the Format Type (@FORMAT)

Changing the display
format type
Switching to
create mode

The @FORMAT command enables you to change the display format type (specified on the initial display or the most recent @FORMAT command) and/or switch the RPG II editor from update mode to create mode. The valid syntax variations for this command are:

Syntax
variations

@FORMAT	Syntax
1	@FORMAT specification-type
2	@FORMAT specification-type, format-type
3	@FORMAT , format-type
4	@FORMAT , format-type, CMD

Parameters

Parameters	
specification-type	Specifies the next specification screen to be displayed. (You may specify any of the specification types or the specification type display (ST).) This parameter is required when switching from update mode to create mode.
format-type	Specifies the new format type to be displayed. Key in 1 to specify positional, 2 to specify formatted, and 3 to specify free-form. This parameter is required if you want to change the display format.
CMD	Specifies that the RPG II editor is to remain in update mode.

Parameter restrictions

NOTE:

The specification type and CMD parameter must not be specified in the same @FORMAT command.

Examples

@FORMAT COMMAND EXAMPLES	
@FO C	Switches the RPG II editor back into create mode (from update mode) and displays a calculation specification screen (in the current format type) for you to enter a statement
@FO F,2	Resets the display format type to formatted and switches the RPG II editor into create mode. Because F and 2 are specified, a file specification screen (in formatted format) is displayed next for you to enter a statement.
@FO ,1	Resets the display format type to positional without switching the RPG II editor back into create mode. When neither a specification type or CMD is included in the @FORMAT command, the current work-space number is presented next for you to enter a command.

@FORMAT COMMAND EXAMPLES	
@FO ,3,CMD	Resets the display format type to free-form and keeps the RPG II editor in update mode ready to accept the next command
@FO ST	Switches the RPG II editor into create mode and displays the specification type display for you to select a specification screen

Displaying Statements (@PRINT)

@PRINT function

The @PRINT command enables you to display specified lines of your work-space file in positional, formatted, or free-form format. (In this case, the free-form format is the traditional EDT display.) Statements contained in the lines are displayed in the display format type that was most recently specified, that is, the one specified on the initial display or in the most recent @FORMAT command.

@PRINT

Format:

```
@PRINT[line-range][,'search-string'[*n]]
```

Syntax

Parameters

This format is presented for convenience. If you need to know more about the parameters or the PRINT command itself, see the OS/3 general editor user guide/programmer reference, UP-8828 (current version).

How statements are displayed

If statements are displayed in either positional or formatted format, only one statement at a time can be displayed. However, if the free-form format (traditional EDT display) is used, more than one line can be displayed.

NOTE:

When the workstation screen becomes full with data lines during the execution of a PRINT command, output of data to the workstation screen halts. To continue the PRINT command (i.e., display the remaining specified lines), press F19 (function key 19). For more information on the use of F19, see the interactive services commands and facilities user guide/programmer reference, UP-8845 (current version).

@PRINT EXAMPLES

EXAMPLE 1

Example:
Displaying one
specification screen

```
                                        1   1 1 1                    LINE -    4.0000
                      1     6  7        5   7 8 9
                        040 IR CDFILE    AA      01

                      2     2  2    3   3    3   4
                      1     5  8    2   5    9   2
                        15 NC1

                      NEXT SPECIFICATION TYPE, ST. OR CMD:  (___)
```

> If 1 (for positional) is specified on the initial display,
> the command @PRINT 4 produces this screen.

EXAMPLE 2

Example:
Displaying successive
specification screens

```
                                                              LINE -    4.0000
      1 SEQUENCE NUMBER:    040        6 FORM TYPE   IR (RECORD ID)
      7 FILENAME: CDFILE    14        15 SEQUENCE: AA      17 NUMBER:    18 OPTION:
     19 RECORD IDENTIFYING INDICATOR, DS OR "": 01
        RECORD                        POSITION    N = NOT   C/Z/D     CHARACTER
        IDENTIFICATION                21    15    25 N      26 C      27 I
        CODES                         28          32        33        34
                                      35          39        40        41
     42 STACKER SELECT:
              NEXT SPECIFICATION TYPE, ST. OR CMD:  (   )
```

> If 2 (for formatted) is specified on the most recent
> @FORMAT command, the command @PRINT 4:5
> produces these two screens. Because only one line
> can be displayed on a single screen, press the XMIT
> key to display the second screen.

```
                                                              LINE -    5.0000
      1 SEQUENCE NUMBER:    050        6 FORM TYPE   IF
     43 DATA FORMAT:                  44 FROM:   15   48 TO:    45
     52 DECIMAL POSITIONS:            53 FIELD NAME: NAMEIN
     59 CONTROL LEVEL:                61 MATCHING/CHAINING FIELDS:
     63 FIELD RECORD RELATION:
     65 PLUS:        67 MINUS:           69 ZERO/BLANK:
        NEXT SPECIFICATION TYPE, ST. OR CMD:  (   )
```

Example:
Displaying lines in
free-form format

EXAMPLE 3

```
2.0000▷020FCDFILE   IPEAF                    READER
3.0000▷030FPRFILE   0   F     132            PRINTER
4.0000▷040ICDFILE   AA  01  15NCI
5.0000▷050I                                  15  45 NAMEIN
```

If 3 (for free-form) is specified on the most recent
@FORMAT command, the command @PRINT 2:5
produces this screen.

Updating Statements (@UPDATE)

@UPDATE
function

The @UPDATE command displays statements like the @PRINT
command and allows you to change them.

@UPDATE

Syntax

Format:

@UPDATE[line-range][,'search-string'[*n]]

Parameters

This format is presented for convenience. If you need to know more
about the parameters or the @UPDATE command itself, see the
OS/3 general editor user guide/programmer reference, UP-8828
(current version).

Keying in
changes

The @UPDATE command displays specification screens for the
specified lines in the work-space file in the current display format
type. These format types are positional, formatted, and free form.
(The free-form format is a 2-line, 64 characters/line format.) Make
any changes necessary by moving the cursor to the fields you want
to change and keying in the corrections.

*Transmitting
the
changes*

Before transmitting the corrected screen, move the cursor to the next screen identification line. For the entry field there, you have four choices:

4 OPTIONS

Entry   **Effect**

The new version of the statement overwrites the statement in the work-space file. The specification screen for the next statement specified in the UPDATE command is displayed.

NOTE:

No entry for this option. Place cursor at bottom of screen and transmit.

@NOCHANGE

Any changes just entered on the screen are ignored. The statement in the work-space file is not changed. The specification screen for the next statement specified in the UPDATE command is displayed.

@CONTINUE

If used alone, the UPDATE command terminates. If combined with another EDT command, only the UPDATE function terminates (i.e., no changes made are displayed).

@STRIKE

Deletes the statement from the work-space file

@UPDATE EXAMPLES

EXAMPLE 1

Example:
Deleting a
statement

```
                                                        LINE -    2.0000
                            1 1 1 1 1 2   2    2 2 3 3
          1      6 7        5 6 7 8 9 0   4    8 9 1 2
          020 F CDFILE      I P E A F
          3 3    3 4        5 5       6      6 6 6  7 7
          3 5    9 0        3 4       0      6 7 8  0 1
                      READER
        NEXT SPECIFICATION TYPE. ST. OR CMD:   (@ST)
```

If 1 (for positional) is specified on the initial display, the command @UPDATE 2 produces this screen. Because @ST is entered on the next screen identification line of this screen, the statement is deleted from the file.

EXAMPLE 2

Example:
Ignoring
changes

```
                                                        LINE -    7.0000
         1 SEQUENCE NUMBER:    070       6 FORM TYPE  C    7 CONTROL LEVEL:
           INDICATORS: 9:      12:       15:                18 FACTOR 1:
        28 OPERATION: MOVE     33 FACTOR 2: NAMEIN          43 RESULT FIELD: NAMEOT
        49 RESULT FIELD LENGTH  31     52 DECIMAL POSITIONS:     53 HALF ADJUST:
                                     ARITHMETIC:       PLUS       MINUS    ZERO
                                     COMPARE    :      1>2        1 2      1=2
                            LOOKUP (FACTOR 2) IS:      HIGH       LOW      EQUAL
           RESULTING INDICATORS:                     54         56       58
        60 COMMENTS:                         NEXT SPECIFICATION TYPE. ST. OR CMD: (@NO)
```

If 2 (for formatted) is specified on the most recent @FORMAT command, the command @UPDATE 7 produces this screen. Because @NO is entered on the next screen identification line, any changes made to the statement are ignored when the screen is transmitted.

Example:
Terminating
updates

EXAMPLE 3

```
                                                          LINE -    5.0000
            1         2         3         4         5         6
12345678901234567890123456789012345678901234567890123456789 01234
  0501                                            15  45 NAMEIN
                                  1         1         1
6    7        8         9         0         1         2
567890123456789012345678901234567890123456789012345678901 2345678
NEXT SPECIFICATION TYPE. ST. OR CMD:   (@CO)
```

If 3 (for free-form) is specified on the most recent @FORMAT command, the command @UPDATE 5,7,8 produces this screen. Because @CO is entered on the next screen identification line, the UPDATE command is terminated when this screen is transmitted. The remaining lines specified in the UPDATE command (lines 7 and 8) are not updated.

4.3. SAMPLE UPDATE SESSION

The program created in the formatted display example in 3.1 is updated in the following example. The specification screens displayed in the formatted display formats, the RPG II editor commands, and necessary EDT commands are used to update the program. The program was originally designed to read a list of names and then print them on the printer. However, it is updated here to include a phone number beside each name.

INITIAL DISPLAY

SCREEN 1

```
                    RPGEDT VERSION XX.XX/XX
SELECT MODE   (U)
C = CREATE    U = UPDATE
SELECT FORMAT TYPE    (2)
 1 = POSITIONAL    2 = FORMATTED    3 = FREEFORM
SPECIFICATION TYPE DISPLAY?   (N)    Y = YES    N = NO
```

For this update session, we select U, 2, and N. We transmit these selections to the RPG II editor by pressing the XMIT key. The next screen contains only the current work-space line number (1.0000).

SCREEN 2

```
 1.0000▷@READ M0 MYPROG.FIL LISTFILE.VSN D00029
10.0000▷@FORMAT IF
```

At this step, we key in the @READ command to transfer a copy of our program (MYPROG) into the work-space file from the program library (LISTFILE). Then we press the XMIT key, and the next work-space line number (10.0000) is displayed for us to key in another command.

***Adding
source statements***

We know that, in order for the program to print out a phone number beside each name, we must add three new source statements: one after line 5, one after line 7, and one after line 9. The first statement must be entered on an input specification screen (field description), so we issue the @FORMAT command to display an IF specification screen.

INPUT SPECIFICATION
FIELD DESCRIPTION

SCREEN 3

```
                                                              LINE -   5.5000
  1 SEQUENCE NUMBER: __055        6 FORM TYPE   IF

 43 DATA FORMAT: _                44 FROM: ___1   48 TO: ___8

 52 DECIMAL POSITIONS: _          53 FIELD NAME: PHONIN

 59 CONTROL LEVEL: __             61 MATCHING/CHAINING FIELDS: __

 63 FIELD RECORD RELATION: __

 65 PLUS: __      67 MINUS: __         69 ZERO/BLANK: __

    NEXT SPECIFICATION TYPE, ST, OR CMD:  (C__)
-----------------------------------------------------------------------
-----------------------------------------------------------------------
```

When the IF specification screen is displayed, we change the line number to 5.5 (a line number greater than 5) and then key in our source statement. Because the next new source statement of our program must be entered on a calculation specification screen, we enter C on the next screen identification line of this screen and then transmit.

CALCULATION SPECIFICATION

SCREEN 4

```
                                                              LINE -   7.5000
  1 SEQUENCE NUMBER: __075        6 FORM TYPE   C   7 CONTROL LEVEL: __

    INDICATORS: 9: ___   12: ___   15: ___              18 FACTOR 1: _____

 28 OPERATION: MOVE      33 FACTOR 2: PHONIN___     43 RESULT FIELD: PHONOT

 49 RESULT FIELD LENGTH: __8      52 DECIMAL POSITIONS: _      53 HALF ADJUST: _

                        ARITHMETIC:       PLUS         MINUS       ZERO
                        COMPARE    :      1>2          1<2         1=2
              LOOKUP (FACTOR 2) IS:       HIGH         LOW         EQUAL
    RESULTING INDICATORS:                 54__         56 __  58   __

 60 COMMENTS:_____     NEXT SPECIFICATION TYPE, ST, OR CMD: (OF_)
-----------------------------------------------------------------------
-----------------------------------------------------------------------
```

When the calculation specification screen is displayed, we change the line number to 7.5 (a line number greater than 7) and key in our source statement. Because the final new source statement of our program must be entered on an output specification screen (field description and control), we enter OF on the next screen identification line of this screen and then transmit.

SCREEN 5

OUTPUT SPECIFICATION
FIELD DESCRIPTION AND CONTROL

```
                                                         LINE -    9 5000

 1 SEQUENCE NUMBER: __095      6 FORM TYPE  OF

   OUTPUT INDICATORS - 23 ___  26 ___  29 ___

32 FIELD NAME: PHONOT
38 EDIT CODES: _

39 BLANK AFTER: B
40 END POSITION: ___8

44 DATA FORMAT: _

45 CONSTANT OR EDIT WORD: _____

   NEXT SPECIFICATION TYPE, ST, OR CMD:  (CMD)
```

*Keying in the last
new source statement*

Keying in CMD

END

When the OF specification screen is displayed, we again change the line number. This time we change it to 9.5 (a line number greater than line 9). Then we key in our new source statement. Because we want to return to update mode (EDT command mode) to save the updated version of our program, we key in CMD on the next screen identification line. A work-space line number is then displayed, ready to accept an EDT or RPG II editor command.

```
10.5000>@WRITE MO MYPROG.FIL LISTFILE.VSN D00029
IS100 FILE/MODULE ALREADY EXISTS; OK TO WRITE TO IT? (Y.N)>Y
10.5000>@RPG END
```

*Storing the corrected
program*

*Ending the
RPG II session*

Here, we enter a @WRITE command to store our updated program (MYPROG) as a module on the program library (LISTFILE), which is a SAT file. Because a copy of our program already exists on the file, we are asked if we want to overwrite the previous version. We answer yes (key in Y), which causes the new version of our program to be saved and the old version to be deleted. We end our update session by keying in @RPG END.

4.4. COMPILING AN RPG II PROGRAM

Creating and running
a job to compile a
program

After creating or updating an RPG II program, you might want to try to compile it immediately. To do this, you must first save the program in a SAT file on disk or format label diskette. Then you must write and run a job control stream that executes the RPG II compiler and uses your saved RPG II program as input. You begin with the RPG II editor in update mode and a complete RPG II program in the work-space file. The following entries save and compile the program we created in 3.1:

Workstation entries

```
1.   10.0000@ WRITE MO=MYPROG,FIL=LISTFILE,VSN=D00029,SAT=YES,SIZE=2
2.   10.0000@ RPG END
3.   10.0000@ DELETE
4.    1.0000// JOB CMPRPG
      2.0000//MYPROG RPG IN=(D00029,LISTFILE)
      3.0000/&
5.    4.0000@ WRITE MO=CMPRPG,FIL=MYJCS,VSN=D00029,SAT=YES,SIZE=1
6.    4.0000@ HALT
```

7.

```
FUNCTION  +  SYS
              MODE
```
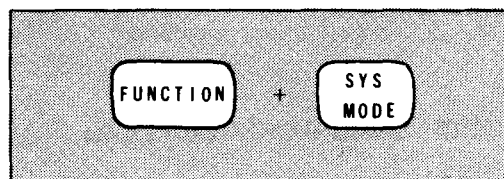
8.

```
RV CMPRPG:(MYJCS,D00029)
```

Entry descriptions

1. Save the program in a SAT file on disk. If the SAT file is already allocated, omit the SIZE parameter. After the @WRITE command is executed, a copy of the program exists on disk and another copy remains in the work-space file.

2. Terminate the RPG II editor, but not EDT.

3. Delete the copy of the program in the work-space file.

4. Use EDT to write a job control stream to compile the program.

5. Save the job control stream in a SAT file on disk.

6.   Terminate EDT.

7.   Put the workstation in system mode. Press the FUNCTION key
     and, while holding it down, press the SYS MODE key.

8.   Enter an RV command to run the job control stream that
     compiles the program.

Correcting compile errors After the job is executed, a message indicating successful or
unsuccessful compilation is displayed at the workstation. If there
are any errors, use EDT screen mode and/or the EDT error file
processor (EFP) to correct the errors. For information about using
EFT, see the general editor user guide/programmer reference, UP-
8828 (current version). If the errors are extensive, you might want
to use the RPG II editor in update mode to correct the program.

5. Error Detection and Recovery

Types of errors

Four types of errors may be encountered during an RPG II editor session.

4	Error Types
1	Syntactical and parameter value errors
2	Invalid specification type
3	EDT errors
4	Hardware and software errors

List of errors

A complete list of all errors encountered during an RPG II editor session appears in the OS/3 system messages programmer/ operator reference, UP-8076 (current version).

5.1. SYNTACTICAL AND PARAMETER VALUE ERRORS

Automatic checking

Error indication

Choosing to correct or ignore an error

The RPG II editor checks each new source statement (entered via specification screens) for syntactical and parameter value errors when it is transmitted to the work-space file. If this type of error is detected, an error message appears on the last two lines of the specification screen. The error message contains an error number, description, and recovery action. A blinking character depicts the location of the error. You must then either correct or ignore the error message to have the next screen displayed. To correct this type of error, reposition the cursor to the error and key in a correct entry; or if you choose not to correct it, you can key in IGN (for ignore) on the next screen identification line:

NEXT SPECIFICATION TYPE, ST, OR CMD: (IGN)

Note that if you ignore the error, the statement is written to the work-space file as is and the screen specified on the next screen identification line will be displayed next. The portion of the entry following the detected error is not checked for errors.

NOTES:

Handling multiple errors

1. *Multiple errors in a statement are handled one at a time.*

Handling compile errors

2. *Although the RPG II editor checks for syntactical errors, this does not eliminate compile errors. For example, if you key in FILEA on the file specification and FELEA (E entered instead of I) on the input specification, you will get a compile error even though the RPG II editor indicated the file name was syntactically correct.*

Handling syntax errors

3. *If any syntax errors are detected when source statements are entered or changed in command mode, the error message is displayed in the free-form format.*

5.2. INVALID SPECIFICATION TYPE

If your specification is correct but the entry on the next screen identification line is not valid, the RPG II editor displays the following error message on a screen by itself:


```
                                                LINE - line-number

     RED008-INVALID FORMAT NAME.


                                                PRESS-TRANSMIT TO CONTINUE.
```
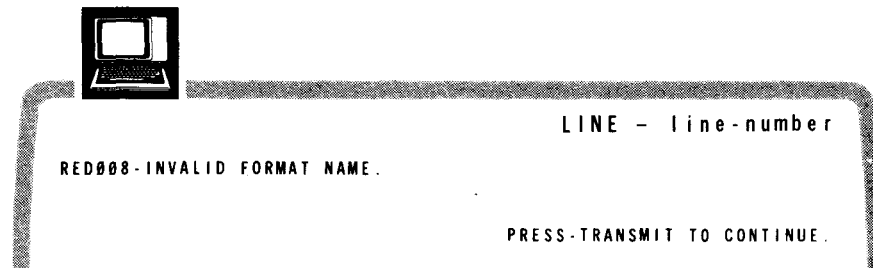
You must then press the XMIT key. The RPG II editor then displays the specification type display, where you must enter a valid specification type abbreviation for the next type of specification screen that you want displayed.

5.3. EDT ERRORS

*Error indication
and recovery*

Because the RPG II editor is a subeditor of EDT, EDT errors may also occur during an RPG II editor session. The error number and error message text of an EDT error are displayed on the workstation screen to help you correct the error; and the line number of the line containing the error is redisplayed for you to enter the correct command.

5.4. HARDWARE AND SOFTWARE ERRORS

If the RPG II editor detects a hardware or software error, it clears the screen and then displays the following error message:

```
                                              LINE -  line-number

RED035-HARDWARE OR UNRECOVERABLE ERROR. RPG EDT SESSION TERMINATED.
                                    PRESS -TRANSMIT- TO CONTINUE.
```
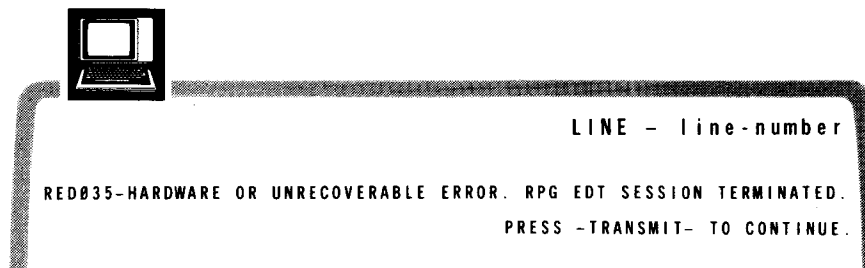
To respond to this message, press the XMIT key. If an EDT work-space line number is displayed next, then EDT is still active. You can enter @RPG to reactivate the RPG II editor. In this case, all or part of your work-space file may be lost. To determine what is still in the work-space file, enter an @PRINT command. Any portion of the program that you had already saved through an @WRITE command should still exist in a file on disk or diskette and may be retrieved by using an @READ command. If EDT is no longer active, the system operator may have to determine the cause of the error or help you reactivate the editor.

Appendix A. Specification Screens

A.1. SPECIFICATION SCREENS IN FORMATTED FORMAT

The following 12 specification screens show the formatted format.

CALCULATION SPECIFICATION

```
                                                    LINE –    1.0000
 1 SEQUENCE NUMBER: _____        6 FORM TYPE  C   7 CONTROL LEVEL: __
   INDICATORS: 9: ___  12: ___  15: ___           18 FACTOR 1: _____
28 OPERATION: _____   33 FACTOR 2: _____      43 RESULT FIELD: _____
49 RESULT FIELD LENGTH: ___     52 DECIMAL POSITIONS: _    53 HALF ADJUST: _
                         ARITHMETIC:      PLUS        MINUS    ZERO
                         COMPARE   :      1>2         1<2      1=2
              LOOKUP (FACTOR 2) IS:       HIGH        LOW      EQUAL
   RESULTING INDICATORS:                  54 __       56 __  58  __

60 COMMENTS:_____     NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
-------------------------------------------------------------------------
-------------------------------------------------------------------------
```

EXTENSION SPECIFICATION

```
                                                           LINE -    1.0000
 1 SEQUENCE NUMBER: _____      6 FORM TYPE  E

 7 RECORD SEQUENCE OF CHAINING FILE: __       9 NUMBER OF CHAINING FIELD:  __

11 FROM FILENAME: _____    19 TO FILENAME: _____

27 TABLE/ARRAY NAME: _____    33 NUMBER OF ENTRIES PER RECORD: ___

36 NUMBER OF ENTRIES PER TABLE/ARRAY: ____     40 LENGTH OF ENTRY: ___

43 DATA FORMAT: _  44 DECIMAL POSITIONS: _     45 SEQ.: _

46 ALTERNATING TABLE/ARRAYNAME: _____        52 LENGTH OF ENTRY: ___

55 DATA FORMAT: _  56 DECIMAL POSITIONS: _     57 SEQ.: _

58 COMMENTS:_____      NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```

FILE SPECIFICATION

```
                                                           LINE -    1.0000
 1 SEQUENCE NUMBER: _____   6 FORM TYPE F   7 FILENAME: _____   15 FILE TYPE: _

16 FILE DESIGNATION: _   17 EOF: _   18 SEQ: _   19 FILE FORMAT: _   20 BLK LEN: ____

24 RECORD LEN: ____   28 FILE PROCESSING MODE: _   29 KEY OR FIELD LENGTH: __

31 RECORD ADDRESS TYPE: _   32 FILE ORGANIZATION: _   33 OVERFLOW INDICATOR: __

35 KEY FLD STARTING LOC: ____   39 EXTENSION/LINE CTR CODE: _  40 DEVICE: _____

53 CONT LINES: _   54 OPTION: _____   60 ENTRY/STORAGE/KEY LOC: _____

66 FILE ADDITION AND CYL OVF % /KEY-LENGTH: __   69 # OF EXTENTS/KEY OPTIONS: __

70 REWIND: _           71 FILE CONDITIONERS: __

                  NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```

HEADER SPECIFICATION

```
                                                           LINE -    1.0000
  1 SEQUENCE NUMBER: _____     6 FORM TYPE   H

  7 COMPILATION MODE: _        8 ERROR DUMP: _     9 OPERATOR CONTROL: _

 15 DEBUG: _                  18 CURRENCY SYMBOL: _

 21 INVERTED PRINT: _         26 ALTSEQ: _

 31 BINARY SEARCH: _          40 SIGN HANDLING: _

 41 FORMS ALIGNMENT: _        42 INDICATOR INIT.: _

 43 FILE TRANSLATION: _       70 CCA NAME: ____

 74 SUBROUTINE: _             75 PROGRAM ID: _____

    NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)

-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
```

INPUT SPECIFICATION
RECORD DESCRIPTION

```
                                                           LINE -    1.0000
  1 SEQUENCE NUMBER: _____        6 FORM TYPE   IR (RECORD ID)

  7 FILENAME: _____ 14 _     15 SEQUENCE: __       17 NUMBER: _   18 OPTION: _

 19 RECORD IDENTIFYING INDICATOR, DS  OR **: __

    RECORD                     POSITION    N = NOT  C/Z/D     CHARACTER
    IDENTIFICATION             21 ____      25 _     26 _       27 _

    CODES                      28 ____      32 _     33 _       34 _

                               35 ____      39 _     40 _       41 _

 42 STACKER SELECT: _

                NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)

-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
```

INPUT SPECIFICATION
FIELD DESCRIPTION

```
                                                              LINE -   1.0000

  1 SEQUENCE NUMBER: _____      6 FORM TYPE   IF

43 DATA FORMAT: _             44 FROM: ____    48 TO: ____

52 DECIMAL POSITIONS: _       53 FIELD NAME: _____

59 CONTROL LEVEL: __          61 MATCHING/CHAINING FIELDS: __

63 FIELD RECORD RELATION: __

65 PLUS: __      67 MINUS: __      69 ZERO/BLANK: __

   NEXT SPECIFICATION TYPE, ST. OR CMD:  (___)
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

LINE COUNTER SPECIFICATION

```
                                                              LINE -   1.0000

  1 SEQUENCE NUMBER: _____

  6 FORM TYPE  L          7 FILENAME: _____

LINE #  CHANNEL #      LINE #  CHANNEL #     LINE #  CHANNEL #
15 ___   18 __         20 ___   23 __        25 ___   28 __

30 ___   33 __         35 ___   38 __        40 ___   43 __

45 ___   48 __         50 ___   53 __        55 ___   58 __

60 ___   63 __         65 ___   68 __        70 ___   73 __

   NEXT SPECIFICATION TYPE, ST. OR CMD:  (___)
------------------------------------------------------------------------------
------------------------------------------------------------------------------
```

OUTPUT SPECIFICATION
FILE IDENTIFICATION AND CONTROL

```
                                                      LINE -    1.0000

  1 SEQUENCE NUMBER: _ ____      6 FORM TYPE   OR

  7 FILENAME: _____           14 _ (AND/OR COL 14-16)

 15 TYPE (H/D/T/E): _           16 STACKER/FETCH: _ (ADD/DEL COL 16-18)

 17 SPACE BEFORE/AFTER: __      19 SKIP BEFORE: __

 21 SKIP AFTER: __

    OUTPUT INDICATORS: 23 ___  26 ___  29 ___

 32 AUTO REPORT ('AUTO): _____

    NEXT SPECIFICATION TYPE, ST. OR CMD:  (___)
```

OUTPUT SPECIFICATION
FIELD DESCRIPTION AND CONTROL

```
                                                      LINE -    1.0000

  1 SEQUENCE NUMBER: _____        6 FORM TYPE   OF

    OUTPUT INDICATORS - 23 ___  26 ___   29 ___

 32 FIELD NAME: _____

 38 EDIT CODES: _

 39 BLANK AFTER: _

 40 END POSITION: ____

 44 DATA FORMAT: _

 45 CONSTANT OR EDIT WORD: _____

    NEXT SPECIFICATION TYPE, ST.OR CMD:  (___)
```

TELECOMMUNICATION SPECIFICATION

```
                                                    LINE -    1.0000

  1 SEQUENCE NUMBER: _____     6 FORM TYPE: T

  7 FILENAME: _____       15 CONFIGURATION: _

 16 TYPE OF STATION: _       19 TRANSPARENCY: _    20 SWITCHED: _

 48 REMOTE TERMINAL: ____    53 PERMANENT ERROR INDICATOR: __

 55 WAIT TIME: ___           60 LAST FILE: _

 65 REMOTE DEVICE: _____    71 TERMINAL NAME: ____

    NEXT SPECIFICATION TYPE, ST. OR CMD:  (___)
 ---------------------------------------------------------------------
 ---------------------------------------------------------------------
```

AUTO REPORT OPTIONS SPECIFICATION

```
                                                    LINE -    1 0000

  1 SEQUENCE NUMBER: _____     6 FORM TYPE  U

  7 SAVE SOURCE: _             8 SOURCE LIBRARY NAME: _____

 27 DATE SUPPRESS: _          28 * SUPPRESS: _

 30 LIST OPTIONS: _

              NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
 ---------------------------------------------------------------------
 ---------------------------------------------------------------------
```

AUTO REPORT (/COPY) SPECIFICATION

```
                                                              LINE -      1.0000
 1 SEQUENCE NUMBER: _____   6 FORM TYPE: _

 7 /COPY
13 SOURCE LIBRARY NAME: _____
50 COMMENTS: _____
              NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```

A.2. SPECIFICATION SCREENS IN POSITIONAL FORMAT

The following 15 specification screens show the positional format.

ALTERNATE COLLATING SEQUENCE SPECIFICATION

```
                                                              LINE -      1.0000
               1   1   2   2   2   3   3   4   4   4   5   5   6
 1         9   3   7   1   5   9   3   7   1   5   9   3   7   1
ALTSEQ__  ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____

                                   1   1   1   1   1   1   1
 6   6   7   7   8   8   8   9   9   0   0   0   1   1   2   2
 5   9   3   7   1   5   9   3   7   1   5   9   3   7   1   5
---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
      NEXT SPECIFICATION TYPE, ST, OR CMD: (___)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```

CALCULATION SPECIFICATION

```
                                                 LINE -    1.0000
                                         1           2
     1      6 7  9   N   N    8-FACTOR 1 8-OP
     _____  C  __ ___ ___ ___ _____ _____
     3            4      4   5 5 5       6
     3-FACTOR 2 3-NAME 9   2 3 4PMMZZ 0 (COMMENTS)
     _____ _____ ___ _ _ _____ _____
          NEXT SPECIFICATION TYPE. ST. OR CMD: (___)

     _____
     _____
```

EQUATE SPECIFICATION

```
                                                       LINE -    1.0000
                   1  1   2   2   2   3   3   4   4   4   5   5   6
     1        9    3  7   1   5   9   3   7   1   5   9   3   7   1
     *EQUATE_ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
                                       1   1   1   1   1   1   1
     6    6    7    7    8    8    8   9   9   0   0   0   1   1   2   2
     5    9    3    7    1    5    9   3   7   1   5   9   3   7   1   5
     ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____
          NEXT SPECIFICATION TYPE. ST. OR CMD:  (___)

     _____
     _____
```

EXTENSION SPECIFICATION

```
                                              LINE -    1.0000
                 1       1     2      3  3
     1     6 7 9 1       9     7      3  6
     _____ E  __ __ _____ _____ _____ ___ ____
     4    4 4 4 4     5   5 5 5 5
     0    3 4 5 6     2   5 6 7 8 (COMMENTS)
     ___ _ _ _ _____ ___ _ _ _ _____
     NEXT SPECIFICATION TYPE. ST. OR CMD:  (___)

     _____
     _____
```

FILE SPECIFICATION

```
                                                              LINE -    1.0000

                   1 1 1 1 1 2     2     2 2   3 3
    1      6 7     5 6 7 8 9 0     4     8 9   1 2
    -----  F  --------  - -  -  -  ----  ----  -  --  - -

    3  3    3 4      5 5      6       6 6 6   7 7
    3  5    9 0      3 4      0       6 7 8   0 1
    -- ----  ---- -- -------- - ------ ------- - - -- - --

    NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)

    --------------------------------------------------------------------------
    --------------------------------------------------------------------------
```

FILE TRANSLATION SPECIFICATION

```
                                                              LINE -    1.0000

                 1    1    2    2    2    3    3    4    4    4    5    5    6
    1          9 3    7    1    5    9    3    7    1    5    9    3    7    1
    *FILES__ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____ ____

                                1    1    1    1    1    1    1
    6    6    7    7    8    8    8    9    9    0    0    0    1    1    2    2
    5    9    3    7    1    5    9    3    7    1    5    9    3    7    1    5
    ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
        NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)

    --------------------------------------------------------------------------
    --------------------------------------------------------------------------
```

HEADER SPECIFICATION

```
                                                              LINE -    1.0000
                     1 1 2 2 3 4 4 4 4 7     7 7
    1        6 7 8 9 5 8 1 6 1 0 1 2 3 0     4 5
    -----  H  - - - - - - - - - - - - ----  - -----
    NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)

    --------------------------------------------------------------------------
    --------------------------------------------------------------------------
```

INPUT SPECIFICATION
RECORD IDENTIFICATION

```
                              LINE -    1.0000
                  1  1 1 1
1     6  7        5  7 8 9
_____ IR _____ __ _ _ __
2   2   2    3   3    3   4
1   5   8    2   5    9   2
____ ___ ____ ___ ____ ___ _
NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
--------------------------------------------------------------------
--------------------------------------------------------------------
```

INPUT SPECIFICATION
FIELD DESCRIPTION

```
                                       LINE -    1.0000
        4 4    4    5 5    5  6  6  6
1   6   3 4    8    2 3    9  1  3  5PMMZZ
_____ IF  _ ____ ____ _ _____ __ __ __ _____
NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
--------------------------------------------------------------------
--------------------------------------------------------------------
```

LINE COUNTER SPECIFICATION

```
                                       LINE -    1.0000
              1     2    2    3    3
1   6 7       5     0    5    0    5
_____ L _____ _____ _____ _____ _____ _____
4   4   5    5    6    6    7
0   5   0    5    0    5    0
_____ _____ _____ _____ _____ _____ _____
NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
--------------------------------------------------------------------
--------------------------------------------------------------------
```

OUTPUT SPECIFICATION
FILE IDENTIFICATION AND CONTROL

```
                                                    LINE -    1.0000
                          1 1 1 1 1  2  2            3
         1     6  7       5 6 7 8 9  1  3   N   N    2
         _____  OR _____ _ _ _ __ __ ___ ___ ___ _____
         NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
```

OUTPUT SPECIFICATION
FIELD DESCRIPTION AND CONTROL

```
                                                    LINE -    1.0000
                 2           3      3 3 4    4
         1     6 3   N   N   2      8 9 0    4
         _____  OF ___ ___ ___ _____ _ _ ____ _
         4
         5 (CONSTANT OR EDIT WORD)
         --------------------------
         NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
```

TELECOMMUNICATION SPECIFICATION

```
                                                    LINE -    1.0000
                          1 1 1 2 4    5  5   6 6      7
         1     6 7        5 6 9 0 8    3  5   0 5      1
         _____  T _____ _ _ _ _ ____ __ ___ _ _____ ____
         NEXT SPECIFICATION TYPE, ST, OR CMD:  (___)
```

AUTO REPORT OPTIONS SPECIFICATION

```
                                                            LINE -    1.0000

                              2 2 3
1      6 7 8      '           7 8 0
-----  U _ -------------------- _ _ _
        NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
```

AUTO REPORT (/COPY) SPECIFICATION)

```
                                                            LINE -    1.0000
                         1
1      6 (SPEC) 7        3 (LIBRARY NAME)
-----  _       /COPY     -----------------
5
0      (COMMENTS)
---------------------------------
        NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
```

A.3. SPECIFICATION SCREEN IN FREE-FORM FORMAT

The following specification screen shows the free-form format.

FREE-FORM FORMAT

```
                                                            LINE -    1.0000
                   1         2         3         4         5         6
         123456789012345678901234567890123456789012345678901234567890123 4
         ---------------------------------------------------------------

                                          1         1         1
         6     7        8         9        0         1         2
         567890123456789012345678901234567890123456789012345678901234567 8
         ---------------------------------------------------------------

         NEXT SPECIFICATION TYPE, ST, OR CMD: (___)

         -------------------------------------------------------------------
         -------------------------------------------------------------------
```

Appendix B. EDT Commands

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}] Δ[{KEY=start-col-no:end-col-no | KKEY=start-col-no:end-col-no | SHOWΔfirst-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 / NDUP}, {CHG / NCHG})}] ,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=$\begin{Bmatrix} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{Bmatrix}$ [,WRPASS=password] ,VSN=volume,DEVICE=$\begin{Bmatrix} \text{did} \\ \text{TAPE} \end{Bmatrix}$ [,BFSZ=n] $\left[\text{,BKNO=}\begin{Bmatrix} \text{YES} \\ \textbf{NO} \end{Bmatrix}\right]$ $\left[\text{,RCFM=}\begin{Bmatrix} \textbf{FIXUNB} \\ \text{FIXBLK} \\ \text{VARUNB} \\ \text{VARBLK} \\ \text{UNDEF} \end{Bmatrix}\right]$ [,RCSZ=n] $\left[\text{,INIT=}\begin{Bmatrix} \text{YES} \\ \textbf{NO} \end{Bmatrix}\right]$ $\left[\text{,EXTEND=}\begin{Bmatrix} \textbf{YES} \\ \text{NO} \end{Bmatrix}\right]$ To write the current work-space file to the spool file, use this format: @WRITE [JOB=jobname] $\left[\text{,HOLD=}\begin{Bmatrix} \text{YES} \\ \text{NO} \end{Bmatrix}\right]$ $\left[\text{,FILENAME=}\begin{Bmatrix} \text{filename} \\ \text{'filename'} \\ \text{"filename"} \end{Bmatrix}\right]$ [,ACCT=acct-no] ,QUEUE=$\begin{Bmatrix} \text{PRINT} \\ \text{PUNCH} \\ \text{RDR} \end{Bmatrix}$ $\left[\text{,COPIES=}\begin{Bmatrix} \text{n} \\ \textbf{1} \end{Bmatrix}\right]$ 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: @WRITEΔ;Δvalid 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
A<u>S</u>SIGN	@AS Gn= $\begin{cases} \text{'string'[*n]} \\ n(x:y) \\ n[\pm m] \\ Gm \\ LEN(n) \end{cases}$	Assigns values to EDT variables
D<u>I</u>SPLAY	@DI $\begin{cases} \text{'string'[*n]} \\ n(x:y) \\ n[\pm m] \\ Gm \\ LEN(n) \end{cases}$	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 DIRECTIVES

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

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

Command	Format	Explanation
C<u>H</u>ECK	@CHE $\begin{bmatrix} \{ON\ \} \\ \{OFF\} \end{bmatrix}$	Determines if processed lines are to be displayed on the workstation screen
C<u>O</u>BOL	@COB	Activates the COBOL editor
D<u>R</u>OP	@DR	Deletes all lines in the entire EDT work-space file
EFP	@EFP	Activates the error file processor

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

Command	Format	Explanation
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 / OFF}] [,WIDTH=device-size][,CLEAR] [,STRIP={ON / OFF}] [,DISPLAY] [,SCRDSPLY={TRUNCATE / FOLD}] [,ROLL={15(if SCRDSPLY=TRUNCATE) / 8(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
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. 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)
PRO**M**PT	@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. RPG II Directive Summary

Table C-1 summarizes the formats and explanations of the RPG II directives. The directives are listed in alphabetical order.

Table C-1. RPG II Directive Summary

Command	Format	Explanation
/EJECT	/EJECT	Causes each specification to be printed on a new page in the source listing
/SPACEn	/SPACEn	Controls spacing between lines in source listing
/TITLE	/TITLE△title	Prints a title heading at the top of the source listing

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:

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

Cut along line.

FOLD

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

FOLD

CUT

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:

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

Cut along line.

FOLD

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

FOLD

CUT