XEDIT VERSION 3
REFERENCE MANUAL

CDC® OPERATING SYSTEM:
NOS 1
## XEDIT Command Summary

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD (A)</td>
<td>5-1</td>
<td>MODIFY (M)</td>
<td>5-6</td>
</tr>
<tr>
<td>ADDLN (ALN)</td>
<td>7-1</td>
<td>NEXT (N)</td>
<td>4-3</td>
</tr>
<tr>
<td>ADDLNS (ALNS)</td>
<td>7-1</td>
<td>NOBELLS (NB)</td>
<td>3-3</td>
</tr>
<tr>
<td>BOTTOM (B)</td>
<td>4-3</td>
<td>OCTCHANGE (OC)</td>
<td>8-2</td>
</tr>
<tr>
<td>BRIEF (BR)</td>
<td>3-2</td>
<td>PRINT (P)</td>
<td>4-4</td>
</tr>
<tr>
<td>CHANGE (C)</td>
<td>5-1</td>
<td>QMOD (QM)</td>
<td>5-6</td>
</tr>
<tr>
<td>CHANGES (CS)</td>
<td>5-1</td>
<td>QUIT (Q)</td>
<td>12-1</td>
</tr>
<tr>
<td>COPY</td>
<td>9-1</td>
<td>READ</td>
<td>9-3</td>
</tr>
<tr>
<td>COPYD</td>
<td>9-2</td>
<td>READP</td>
<td>9-2</td>
</tr>
<tr>
<td>§</td>
<td>6-4</td>
<td>REPLACE (R)</td>
<td>6-3</td>
</tr>
<tr>
<td>DBADL (DBL)</td>
<td>7-4</td>
<td>REPLACELN (RLN)</td>
<td>7-3</td>
</tr>
<tr>
<td>DEFTAB (DT)</td>
<td>11-1</td>
<td>RESTORE (REST)</td>
<td>8-2</td>
</tr>
<tr>
<td>DELETE (D)</td>
<td>6-1</td>
<td>RMARGIN (RM)</td>
<td>11-2</td>
</tr>
<tr>
<td>DELETELN (DLN)</td>
<td>7-1</td>
<td>STOP</td>
<td>13-1</td>
</tr>
<tr>
<td>DELIMIT (DEL)</td>
<td>10-1</td>
<td>TABS (TAB)</td>
<td>11-1</td>
</tr>
<tr>
<td>DEOF (DF)</td>
<td>8-3</td>
<td>TEOF</td>
<td>3-3</td>
</tr>
<tr>
<td>DEOR (DR)</td>
<td>8-3</td>
<td>TEOR</td>
<td>3-3</td>
</tr>
<tr>
<td>DLBLANKS (DLB)</td>
<td>8-1</td>
<td>TOP (T)</td>
<td>4-3</td>
</tr>
<tr>
<td>eEDIT</td>
<td>6-5</td>
<td>TOPNULL (TN)</td>
<td>8-1</td>
</tr>
<tr>
<td>END (E)</td>
<td>12-1</td>
<td>TRIM</td>
<td>5-4</td>
</tr>
<tr>
<td>EXPLAIN</td>
<td>3-4</td>
<td>TRUNCATE (TRUNC)</td>
<td>11-3</td>
</tr>
<tr>
<td>FBADL (FBL)</td>
<td>7-3</td>
<td>VERIFY (V)</td>
<td>3-2</td>
</tr>
<tr>
<td>FILE (F)</td>
<td>9-3</td>
<td>WEOF (WF)</td>
<td>8-4</td>
</tr>
<tr>
<td>FINDLL (FLL)</td>
<td>11-2</td>
<td>WEOR (WR)</td>
<td>8-4</td>
</tr>
<tr>
<td>HELP (H)</td>
<td>3-4</td>
<td>WHERE</td>
<td>4-4</td>
</tr>
<tr>
<td>INPUTe</td>
<td>6-5</td>
<td>WMARGIN (WM)</td>
<td>5-4</td>
</tr>
<tr>
<td>INSERT (I)</td>
<td>6-3</td>
<td>Y</td>
<td>10-2</td>
</tr>
<tr>
<td>INSERTB (IB)</td>
<td>6-3</td>
<td>YQMOD (YQM)</td>
<td>5-7</td>
</tr>
<tr>
<td>linemumber</td>
<td>4-1</td>
<td>Z</td>
<td>10-2</td>
</tr>
<tr>
<td>LISTAB (LT)</td>
<td>11-1</td>
<td>.</td>
<td>10-2</td>
</tr>
<tr>
<td>LOCATE (L)</td>
<td>4-1</td>
<td>-</td>
<td>10-2</td>
</tr>
</tbody>
</table>
XEDIT VERSION 3
REFERENCE MANUAL

CDC® OPERATING SYSTEM:
NOS 1
<table>
<thead>
<tr>
<th>REVISION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Manual released.</td>
</tr>
<tr>
<td>(6-2-'78)</td>
<td></td>
</tr>
</tbody>
</table>
**LIST OF EFFECTIVE PAGES**

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

<table>
<thead>
<tr>
<th>PAGE</th>
<th>REV</th>
<th>PAGE</th>
<th>REV</th>
<th>PAGE</th>
<th>REV</th>
<th>PAGE</th>
<th>REV</th>
<th>PAGE</th>
<th>REV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front Cover</td>
<td>-</td>
<td>A-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td></td>
<td>A-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td></td>
<td>A-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Title Page</td>
<td></td>
<td>A-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ii</td>
<td>A</td>
<td>B-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iii/iv</td>
<td>A</td>
<td>B-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v/vi</td>
<td>A</td>
<td>B-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>vii</td>
<td>A</td>
<td>B-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>viii</td>
<td>A</td>
<td>B-5</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-1</td>
<td>A</td>
<td>B-6</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-1</td>
<td>A</td>
<td>B-7</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-2</td>
<td>A</td>
<td>B-8</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-3</td>
<td>A</td>
<td>C-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-4</td>
<td>A</td>
<td>C-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-1</td>
<td>A</td>
<td>C-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-2</td>
<td>A</td>
<td>C-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-3</td>
<td>A</td>
<td>C-5</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>A</td>
<td>C-6</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-1</td>
<td>A</td>
<td>C-7</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-2</td>
<td>A</td>
<td>C-8</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-3</td>
<td>A</td>
<td>C-9</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4-4</td>
<td>A</td>
<td>C-10</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-1</td>
<td>A</td>
<td>C-11</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-2</td>
<td>A</td>
<td>C-12</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-3</td>
<td>A</td>
<td>C-13</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>A</td>
<td>Index-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-5</td>
<td>A</td>
<td>Index-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-6</td>
<td>A</td>
<td>Index-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-7</td>
<td>A</td>
<td>Comment</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-8</td>
<td>A</td>
<td>Sheet</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-1</td>
<td>A</td>
<td>Back Cover</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-5</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-5</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-4</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-2</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12-3</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PREFACE

XEDIT is an extended interactive text editor developed by the University of Minnesota and available for use the the CONTROL DATA® Network Operating System (NOS). NOS provides network capabilities for time-sharing and transaction processing, in addition to local and remote batch processing, on CDC® CYBER 170 Series, Models 171, 172, 173, 174, and 175 Computer Systems, CDC® CYBER 70 Series, Models 71, 72, 73, and 74 Computer Systems, and CDC® 6000 Series Computer Systems.

This manual describes XEDIT use. Each section explains and demonstrates a number of similar XEDIT commands.

This manual assumes that the user is familiar with NOS operation and file concepts. For information on time-sharing processing, the user should refer to the IAF Version 1 Reference Manual or the NOS 1 Time-Sharing User's Reference Manual. For batch processing information, the user should refer to the NOS 1 Batch User's Guide and NOS 1 Reference Manual, volume 1.

RELATED PUBLICATIONS

The following manuals may be of interest to the XEDIT user:

<table>
<thead>
<tr>
<th>Control Data Publication</th>
<th>Publication Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Products</td>
<td></td>
</tr>
<tr>
<td>Interactive Facility Version 1 Reference Manual</td>
<td>60455250</td>
</tr>
<tr>
<td>Network Products</td>
<td></td>
</tr>
<tr>
<td>Interactive Facility Version 1 User's Guide</td>
<td>60455260</td>
</tr>
<tr>
<td>Network Products</td>
<td></td>
</tr>
<tr>
<td>Network Terminal User's Instant</td>
<td>60455270</td>
</tr>
<tr>
<td>NOS Version 1 Batch User's Guide</td>
<td>60436300</td>
</tr>
<tr>
<td>NOS Version 1 Text Editor Reference Manual</td>
<td>60436100</td>
</tr>
<tr>
<td>NOS Version 1 Reference Manual, Volume 1</td>
<td>60435400</td>
</tr>
<tr>
<td>NOS Version 1 Time-Sharing User's Guide</td>
<td>60436400</td>
</tr>
<tr>
<td>NOS Version 1 Terminal User's Instant</td>
<td>60435800</td>
</tr>
</tbody>
</table>

In the examples in this manual, user input is printed in lowercase letters, and XEDIT output is printed in.uppercase letters.

The symbol in this manual is used to denote the terminal key that must be pressed in order to send an input line to the system. For most time-sharing terminals, this is the carriage return key. IAF users should refer to the IAF Reference Manual to determine which key applies to them.

DISCLAIMER

This product is intended for use only as described in this document. Control Data Corporation cannot be responsible for the proper functioning of undescribed features or undefined parameters.
CONTENTS

1. INTRODUCTION
   XEDIT Features

2. CALLING XEDIT
   Calling XEDIT for Time-Sharing Jobs
   Calling XEDIT for Batch Jobs
   XEDIT Control Statement Parameters

3. XEDIT CONVENTIONS
   General Rules for XEDIT Input
   XEDIT Command Syntax
   XEDIT Command Parameters
   Entering Editing Data
   String Delimiters
   Suppressing Default Conditions
   Suppressing Verification
   Suppressing Printing of File Marks
   Suppressing Terminal Bell Ringing
   Interrupting XEDIT Processing
   On-Line Reference Aids
   Asking for Error Message Explanations
   Asking for Command Descriptions

4. POSITIONING THE FILE POINTER
   Pointer Movement by Command
   Prefixing
   Pointer Movement Commands
   Locating Lines Via Line Numbers
   Locating Lines Via Specified Strings
   Advancing and Reversing the Pointer
   Positioning Pointer at Top or Bottom of File
   Positioning Pointer by Listing Lines
   Printing the Current Line Count

5. STRING EDITING
   Adding a String to the End of the Line
   Replacing, Deleting, and Inserting Strings
   Replacing Strings by Context
   Deleting Strings by Context
   Inserting Strings at the Beginning of a Line
   Restricting the String Search
   Ignoring Trailing Blanks
   Defining a Window
   Using a Window
   Modifying Strings Character-by-Character

6. LINE EDITING
   Deleting Lines
   Replacing Lines
   Inserting Lines
   INSERT Command

7. EDITING LINE NUMBERS
   Adding Line Numbers
   ADDLN Command
   ADDLNS Command
   Deleting Line Numbers
   Replacing Line Numbers
   Finding Lines Without Line Numbers
   Deleting Lines Without Line Numbers

8. MISCELLANEOUS EDITING COMMANDS
   Deleting Leading Blanks
   Inserting a Blank Line at the Top of a File
   Converting Octal Strings
   Canceling Editing Operations
   Deleting and Inserting Records and File Marks
   DEOR Command
   DEOF Command
   WEOR Command
   WEOF Command

9. MANIPULATING FILES
   Copying Lines Onto Another File
   Merging Files Into the Edited File
   Saving an Intermediate XEDIT File

10. COMMAND SEQUENCING
    Submitting Multiple Entries in a Single Line
    Using DELIMIT Command
    Using Z and Y Commands
    Repeating Commands

11. USING TABS AND MARGINS
    Tab Control
    Defining Tab Character
    Defining Tab Positions
    Listing Tab Settings
    Margin and Truncation Control
    Setting the Margin
    Finding Long Lines
    Truncating Long Lines

12. TERMINATING XEDIT
    Terminating XEDIT Without Saving Editing Changes
    Terminating XEDIT With Editing Changes Saved
APPENDIXES

A NOS TIME-SHARING 64-CHARACTER SET A-1
B MESSAGES B-1
C EXAMPLES OF XEDIT C-1

INDEX

FIGURES

2-1 Editing a Primary File 2-1
2-2 Editing a Working File 2-1
2-3 Editing a Direct Access File 2-2
2-4 Batch Job Using XEDIT 2-2
3-1 Entry of Editing Data 3-2
3-2 Verify Versus Brief Mode 3-3
3-3 Suppressing Verification 3-3
3-4 Suppressing Printing of File Marks 3-3
3-5 Suppressing Terminal Bell Ringing 3-4
3-6 Error Message Explanation 3-4
3-7 Command Description 3-4
4-1 Locating Lines Via Line Numbers 4-1
4-2 LOCATE With a Specified String 4-2
4-3 LOCATE With Two Specified Strings 4-2
4-4 LOCATE With String Not to be Found 4-3
4-5 The NEXT Command 4-3
4-6 TOP and BOTTOM 4-4
4-7 The PRINT Command 4-4
4-8 The WHERE Command 4-4
5-1 The ADD Command 5-1
5-2 Use of CHANGE to Modify a Line 5-2
5-3 Selective Deletion via CHANGE Command 5-3
5-4 Use of CHANGE to Insert a String 5-4
5-5 The TRIM Command 5-4
5-6 Use of the Suffix Characters 5-5
5-7 Restricting Field of Search 5-6
5-8 The MODIFY Command 5-7
5-9 The QMOD Command 5-7
5-10 The QMOD Command 5-8
6-1 Deleting Selected Consecutive Lines 6-1
6-2 Deleting Lines Containing a Specified String 6-2
6-3 Deleting Lines Containing Two Specified Strings 6-2
6-4 The REPLACE Command 6-3
6-5 The INSERT Command 6-4
6-6 The INSERTBE Command 6-4
6-7 The INPUT Command 6-5
6-8 The $ Command 6-5
7-1 The ADDLN Command 7-2
7-2 The ADDLNS Command 7-2
7-3 The DELETELN Command 7-2
7-4 The REPLACELN Command 7-3
7-5 The FBADL Command 7-3
7-6 The DBADL Command 7-4
8-1 The DLBLANS Command 8-1
8-2 The TOPNULL Command 8-2
8-3 The OCTCHANGE Command 8-3
8-4 The RESTORE Command 8-3
8-5 The DEOR Command 8-4
8-6 The DEOF Command 8-4
8-7 The WEOR Command 8-5
8-8 The WEOF Command 8-5
9-1 The COPY Command 9-2
9-2 The READP Command 9-3
9-3 The FILE Command 9-4
10-1 The DELIMIT Command 10-2
10-2 The Z Command 10-3
10-3 Repeating a Command 10-3
10-4 Repeating a Command Sequence 10-3
11-1 Defining a Tab Character 11-1
11-2 Defining Tab Stops 11-2
11-3 Listing the Tab Settings 11-2
11-4 The RMMARGIN and FINDLL Commands 11-3
11-5 The TRUNCATE Command 11-3
12-1 The STOP Command 12-1
12-2 The END Command Using REPLACE Mode 12-2
12-3 The END Command Using RL Mode 12-3

TABLE

5-1 Modification Directives 5-8
XEDIT can be used to manipulate the following types of files:

- Primary files
- Working files
- Indirect access files
- Direct access files

These files can have one or more of the following characteristics:

- Multifile and/or multirecord files (that is, files containing more than one end-of-file or end-of-record mark)
- Files containing either source programs, data, or text (listable files)
- Files prepared in either normal or ASCII mode (the user issues the ASCII command)

The user should not use XEDIT to edit binary files.

XEDIT FEATURES

The following are features of XEDIT.

- Simple command formats
- Multiple commands in a single line
- Verification of user entries; XEDIT automatically lists the file lines modified by a command
- Internal interrupt processing; control is not returned to the operating system
- Editing on the basis of line numbers
- Availability of permanent file commands; a file can be stored without leaving XEDIT
- Easy line modification; strings may be edited on a character-by-character basis
- Editing multifile and/or multirecord files
- Tab control capability
- Window capability; the scope of all string search commands may be restricted to a specific set of columns
- Available for batch processing

Appendix C contains sample editing sessions using these XEDIT features.
CALLING XEDIT

XEDIT can be called from a time-sharing terminal, where editing directives can be given interactively, or in a batch job, where a set of editing directives are processed without further user direction. XEDIT makes certain changes in its processing depending on whether a job is of time-sharing origin or of batch origin.

CALLING XEDIT FOR TIME-SHARING JOBS

To call XEDIT from a time-sharing terminal, the user must complete the following steps,†

1. Log-in to a time-sharing terminal.
2. Access the file to be edited.
3. Select the batch subsystem, ††

   old_address
   READY.
   batch
   $RFL,0.
   /attach,xedit/un=library
   /xedit
   XEDIT 3.0
   ??

   A copy of a permanent indirect access file called address becomes the primary file.
   The batch subsystem is called. XEDIT is attached and executed following the slash (/) prompt.

   Figure 2-1. Editing a Primary File

   get,bfile
   READY.
   $RFL,0.
   /attach,xedit/un=library
   /xedit,bfile
   XEDIT3.0
   ??

   A copy of a permanent indirect access file called bfile becomes a working file. The batch subsystem is called.

   XEDIT is called following the slash (/) prompt.

   Figure 2-2. Editing a Working File

† Users of certain terminals should issue a ROUT command, or if using IAF, a terminal definition command, before calling XEDIT since their terminal carriage return must operate at the correct speed if proper character alignment is to be made. This is necessary for successful execution of the MODIFY, QMOD, and YQMOD commands. Refer to the Time-Sharing User's Reference Manual for a description of the ROUT command and to the IAF Reference Manual for a description of the terminal definition command.

ATTACH(XEDIT/UN=LIBRARY)

and the XEDIT control statement in his control statement record. Assuming that the L, L, and FR XEDIT parameters are omitted as described in the following section, XEDIT takes its editing directives from a job deck record following the control statement record and leaves its output on file OUTPUT. The final command in the editing directives record must be END or QUIT; otherwise, XEDIT aborts.

**NOTE**

If the NEW or OLD control statements are used to create or retrieve the file to be edited, the /ND parameter must be specified on the NEW or OLD control statement so that XEDIT can take its directives from a later record in the job deck. (The NEW and OLD statements return all files including file INPUT if /ND is omitted.)

Figure 2-4 is an example of a batch job that uses XEDIT.

**XEDIT CONTROL STATEMENT PARAMETERS**

Additional parameters can be specified in the XEDIT control statement. The format used is:

XEDIT(ifn1, parameter1, parameter2, ..., parameter, /dcs)

A comma can be substituted for the left parenthesis and a period for the right parenthesis. The file name ifn must be the first parameter listed. All other parameters are order-independent.

**In brief, the parameters of the XEDIT control statement are:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ifn1</td>
<td>Name of the file to be edited</td>
</tr>
<tr>
<td>P</td>
<td>Retrieve and edit permanent file ifn</td>
</tr>
<tr>
<td>C</td>
<td>Create a new file ifn1 (creation mode)</td>
</tr>
<tr>
<td>I=ifn2</td>
<td>Take editing directives from the file specified</td>
</tr>
<tr>
<td>I=0</td>
<td>Take all editing directives from the trailing delimited command sequence (following, if the FR parameter is specified, processing of directives from the first line of the file to be edited)</td>
</tr>
<tr>
<td>dcs</td>
<td>Trailing delimited command sequence</td>
</tr>
<tr>
<td>FR</td>
<td>Take the first editing command from the first line of the file to be edited</td>
</tr>
<tr>
<td>NH</td>
<td>Suppress printing of the XEDIT header</td>
</tr>
<tr>
<td>L=ifn3</td>
<td>Put all XEDIT output on the specified file</td>
</tr>
<tr>
<td>L=0</td>
<td>No output is generated</td>
</tr>
<tr>
<td>B</td>
<td>The job is processed as a batch origin job</td>
</tr>
</tbody>
</table>

The following information describes the effects of specifying or omitting a parameter and examples of how it could be used.

The user names the job. He provides user identification and validation information. He accesses an indirect access file called BATCHFL. He attaches and calls XEDIT to edit the file BATCHFL. He replaces the edited file.

The second record of the job deck contains the editing directives to be used.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ln1</td>
<td>If specified, the file named is rewound, sorted if it is a sequenced primary file, edited as directed, and rewound after processing. If omitted, the primary file is edited. If there is no primary file, the message EMPTY FILE/CREATION MODE ASSUMED is printed and execution begins under creation mode (refer to explanation of the C parameter). If the file name is omitted, the separators that would appear before and after the file name must be included before other parameters are specified. Example of use: To call XEDIT to edit a local file called ADDRESS: XEDIT(ADDRESS)</td>
</tr>
<tr>
<td>P</td>
<td>If specified, file ln1 is retrieved from the user's permanent file catalog as a working file. Direct access files are attached in write mode. If the file ln1 is not found in the user's permanent file catalog, XEDIT prints an informative message and requests the name of the file to be edited. The user should then enter the file name followed by a C or P parameter, if appropriate. If omitted, the file ln1 is assumed to be a local file. Example of use: To call XEDIT to edit a permanent file called ADDR: XEDIT(ADDR, P)</td>
</tr>
<tr>
<td>C</td>
<td>If specified, XEDIT begins execution in creation mode. In creation mode, only the following commands are valid: <code>INSERT</code> <code>TEOR</code> <code>BRIEF</code> <code>LISTAB</code> <code>TOPNULL</code> <code>DEPTAB</code> <code>NOBELLS</code> <code>TRIM</code> <code>DELIMIT</code> <code>RMASS</code> <code>VERIFY</code> <code>EXPLAIN</code> <code>STOP</code> <code>WHERE</code> <code>HELP</code> <code>TABS</code> <code>Y</code> <code>INPUT</code> <code>TEOF</code> <code>Z</code> XEDIT leaves creation mode when a line of text exists within the file. Normal editing can follow. Upon exit, a working file is generated with the name, ln1, specified in the XEDIT control statement. If a file name ln1, is not specified, the file takes the name of the primary file, or if there is no primary file, TAPE1.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=ln1 or i=0</td>
<td>If omitted, the file to be edited must contain at least one line of text; otherwise, the message EMPTY FILE/CREATION MODE ASSUMED is printed. Example of use: To create a file called ADDR: XEDIT(ADDR, C)</td>
</tr>
</tbody>
</table>

If i=ln1 is specified, XEDIT takes its editing directives from the specified file after processing commands from the first line of the file to be edited (if the FR parameter is specified) and/or a trailing delimited command sequence. The last command in the specified directives file must be END, QUIT, or STOP; otherwise, XEDIT aborts.

If i=0 is specified, XEDIT takes its editing directives from the first line of the file to be edited (if the FR parameter is specified) and from a trailing delimited command sequence. The last command in the delimited command sequence (dcg) must be END, QUIT, or STOP; otherwise, XEDIT aborts.

If it is omitted, XEDIT takes editing directives for a batch job from the next record of the job deck. For a time-sharing job, XEDIT takes directives from terminal input or the file INPUT.

Examples of use: To call XEDIT to edit a local file called ADDR with directives from a previously created file called INFILE: XEDIT(ADDR, I=INFILE) To call XEDIT to edit a local file called ADDR using only directives from the trailing dcg: XEDIT(ADDR, I=0, PRINTS, END, ADDR, RL)

If specified, XEDIT takes its editing directives from the dcg before taking directives from the input file. Its format is the same as that of a multiple command interactive entry (refer to section 10) except the delimiter character is assumed to be a semicolon unless specified otherwise by the first character of the sequence. The characters which can begin XEDIT commands (letters, numbers, +, /, *) and the characters -, *, comma, and space are not recognized as command delimiters.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>If omitted, XEDIT takes its directives from the first line of the file to be edited, if the FR parameter is specified, and from the default or specified input file, if I=0 is not specified.</td>
<td></td>
</tr>
<tr>
<td>Example of use: To call XEDIT to print out the file ADDR before processing the editing directives from the input file B:</td>
<td></td>
</tr>
<tr>
<td>XEDIT(ADDR, I=B)PRINT*</td>
<td></td>
</tr>
<tr>
<td>FR</td>
<td>If specified, XEDIT takes its first editing command from the first line of the file to be edited. The command is assumed to begin with the first nonblank character following two consecutive blanks. XEDIT then takes its commands from the dcs, if one is specified, and then from the input file.</td>
</tr>
<tr>
<td>If omitted, XEDIT takes its editing directives from the dcs, if one is specified, and then from the input file.</td>
<td></td>
</tr>
<tr>
<td>Example of use: To call XEDIT to edit a file called ADDR, and to take its first command from the first line of ADDR:</td>
<td></td>
</tr>
<tr>
<td>XEDIT(ADDR, FR)</td>
<td></td>
</tr>
<tr>
<td>The following are examples of possible first lines in the file to be edited which would set the delimiter character, tab character, and tab settings.</td>
<td></td>
</tr>
</tbody>
</table>
| COMPASS: | * Y/DELIMIT;/ 
DEFTAB$; 
TABS 10 20 30 |
| FORTRAN: | 00100C Y/DELIMIT;/ 
DEFTAB$; 
TABS 10 20 30 |
| MODIFY: | * Y/DELIMIT;/ 
DEFTAB$; 
TABS 10 20 30 |
| BASIC: | 00100 REM Y/ 
DELIMIT;/DEFTAB$; 
TABS 10 20 30 |
<p>| NH | If specified, the XEDIT header message (for example, XEDIT 3.0.6) is not printed. |
| If omitted, the header message is printed to verify entry into XEDIT. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example of use: To call XEDIT to edit the primary file without printing the header message:</td>
<td></td>
</tr>
<tr>
<td>XEDIT(NH)</td>
<td></td>
</tr>
<tr>
<td>L=1fn$_3$ or</td>
<td>If L=1fn$_3$ is specified, all XEDIT output goes on the specified file. The file is not rewound before or after processing. If 1fn$_3$ is to be printed on a batch line printer, the file should be shifted by one character position to avoid printer carriage control characters. Refer to the COPYSBF control statement in the NOS Reference Manual, volume 1.</td>
</tr>
<tr>
<td>L=0</td>
<td>If L=0 is specified, XEDIT generates no output.</td>
</tr>
<tr>
<td>If omitted, all XEDIT output goes on the default file OUTPUT.</td>
<td></td>
</tr>
<tr>
<td>Examples of use: To call XEDIT to edit the file ADDR and leave its output on file OUTFILE:</td>
<td></td>
</tr>
<tr>
<td>XEDIT(ADDR, L=OUTFILE) REWIND(OUTFILE) COPYSBF(OUTFILE, OUTF) REWIND(OUTF) ROUTE(OUTF, DC=LP)</td>
<td></td>
</tr>
<tr>
<td>To call XEDIT to edit the file ADDR without producing any output:</td>
<td></td>
</tr>
<tr>
<td>XEDIT(ADDR, L=0)</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>If specified, XEDIT processes the job as a batch origin job regardless of its method of entry. Any error in the editing directives causes XEDIT to abort.</td>
</tr>
<tr>
<td>If omitted, jobs are processed according to their origin. Jobs run under the batch subsystem are of time-sharing origin; jobs entered via the SUBMIT control statement are of batch origin.</td>
<td></td>
</tr>
<tr>
<td>Example of use: To call XEDIT to edit file ADDR if the directives taken from the input file INFFILE are correct. If any error exists within the directives, XEDIT aborts, and file ADDR remains in its unedited form.</td>
<td></td>
</tr>
<tr>
<td>XEDIT(ADDR, I=INFFILE, B)</td>
<td></td>
</tr>
</tbody>
</table>
All XEDIT commands have the same basic format as explained in General Rules for XEDIT Input. This section also explains how XEDIT provides the means to suppress default conditions, interrupt XEDIT processing without leaving XEDIT, and obtain explanations of commands and error messages during the editing session.

**GENERAL RULES FOR XEDIT INPUT**

All XEDIT commands follow the same basic pattern in their syntax and parameters and in their required editing data. Knowledge of this basic pattern leads to fewer user errors while editing.

**XEDIT COMMAND SYNTAX**

XEDIT prints a double question mark (??) wherever it expects the user to enter an XEDIT command.†

The syntax for an XEDIT command is:

```
prefix|command
```

The optional prefix characters are:

<table>
<thead>
<tr>
<th>prefix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>Advances the pointer one line before processing the command.</td>
</tr>
<tr>
<td>^ or ++</td>
<td>Repositions the pointer to the beginning of the file before processing the prefixed command. This character is octal 76.</td>
</tr>
<tr>
<td>X</td>
<td>Temporarily suppresses VERIFY or BRIEF mode while processing the prefixed command.</td>
</tr>
<tr>
<td>+</td>
<td>Informs XEDIT that editing data exists on the same line as the command itself when used in conjunction with the DELIMIT, Y, or Z commands; the ADD, INSERT, INSERTB, MODIFY, QMOD, REPLACE, and YQMOD commands are the only commands that use the + prefix. It is ignored when used with other commands. (Refer to section 10.)</td>
</tr>
</tbody>
</table>

Prefix characters can be used in any order and combination and on any legal XEDIT command.

An optional line number prefix (ln) specifies the line number that begins the line at which the prefixed command is to begin processing. If the specified line is not found, the command is not processed, an informative message is printed, and if editing in verify mode, the line at the current pointer position is also printed.

A legal command conforms to the following conventions:

- The command must be spelled correctly.
- Command parameters must follow the sequence shown in that command’s description.
- Command parameters must stay within their maximum and minimum numerical limits. When an asterisk is substituted for an n or m parameter, it has a value of 99999.
- The command must be entered at the position where the terminal print element stopped after printing the double question mark.
- No embedded blanks can appear within a command or between a line number prefix and a command.
- The command must be separated from its parameters by either a comma, a space, or for string parameters, the string delimiter.
- Extra spaces may appear between command parameters.

If the user does not enter a valid command after a double question mark prompt (??), XEDIT prints an error message and another double question mark prompt (??).

**XEDIT COMMAND PARAMETERS**

In this manual, most XEDIT command formats use the following symbols:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Number of file lines that the command should process. This number should be a positive integer less than 100000, its default value is 1. The user can enter an asterisk (*) for n when he wants the command processed until the file pointer reaches the end-of-information. (XEDIT prints an END OF FILE message.)</td>
</tr>
</tbody>
</table>

† A double question mark is not printed if the user has assigned the input or output file to a mass storage device or to a file via the XEDIT control statement parameters I=lfn or I=0.
### SUPPRESSING DEFAULT CONDITIONS

The XEDIT user may find that he does not want certain aids provided by XEDIT. Therefore, XEDIT has commands to suppress verification, terminal bell ringing, and the printing of file and record marks.

### SUPPRESSING VERIFICATION

By default, XEDIT assumes that the user wants verification of the effect of an XEDIT command on his file. For example, suppose a user wants to delete two lines in the file, beginning at the current file pointer position. To do so, the user would issue a DELETE command. XEDIT automatically lists the lines which are deleted so that the user can verify that the command performed the task which was intended. This automatic verification occurs in verify mode.

While verify mode is the default condition for XEDIT, the user can suppress verification if he wants and edit under brief mode. He can select verification or no verification by entering one of the following commands.

<table>
<thead>
<tr>
<th>Action</th>
<th>Legal Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification</td>
<td>VERIFY ☑️</td>
</tr>
<tr>
<td></td>
<td>VERIFY+ ☑️</td>
</tr>
<tr>
<td></td>
<td>BRIEF- ☒️</td>
</tr>
<tr>
<td></td>
<td>V ☒️</td>
</tr>
<tr>
<td>No Verification</td>
<td>BRIEF ☒️</td>
</tr>
<tr>
<td></td>
<td>BRIEF+ ☐️</td>
</tr>
<tr>
<td></td>
<td>VERIFY- ☒️</td>
</tr>
<tr>
<td></td>
<td>BR ☒️</td>
</tr>
</tbody>
</table>

Figure 3-2 illustrates the difference between verify mode and brief mode. In both instances, the user issues a CHANGE command to delete the string ZIP from every line in the file.

When editing under either verify or brief mode, a user may want to process a single command under the alternate mode without having to issue a BRIEF or VERIFY command. This can be done by prefixing the command with an $, as shown in figure 3-3.

### STRING DELIMITERS

Certain XEDIT commands allow the user to replace sequences of numbers or characters (strings) that appear within a file line. When a user enters a string command, he separates the string from the command itself, other parameters, and other strings by using a string delimiter. A string delimiter can be any character that is not part of the string. The character must be within the NOS Time-Sharing character set (refer to appendix A) and cannot be a space, number, asterisk, or comma (or a $ under IAP). If the delimiter is alphabetic, a space must separate the command and the delimiter. In this manual, delimiters are represented by a slash (/) character.

For example, if a user wants to locate the next line containing the string IF N = ? any of the following LOCATE commands are legal and will accomplish his purpose.

- LOCATE/IF N = /
- LOCATE"IF N = 
- LOCATE ZIP N = Z

XEDIT sends a double question mark; user responds by issuing a MODIFY command to alter line 110.

- XEDIT prints line 110
- XEDIT prints a single question mark: user supplies editing data.
- XEDIT prints the modified version of line 110.

Figure 3-1. Entry of Editing Data
Verify Mode

?? change/zip/*
00110 1544 WILSHIRE ST 55722
00170 P.O.B 55 55703
00200 8710 14TH ST 55713
END OF FILE
??

Brief Mode

?? change/zip/*
END OF FILE
??

Figure 3-2. Verify Versus Brief Mode

?? verify
?? *locate/zip/
00110 1544 WILSHIRE ST ZIP CODE 55722
?? xchange/zip code/zip/
??

The user issues a VERIFY command.
The user issues a LOCATE command.
XEDIT prints the located line.
The user issues a CHANGE command with an x prefix; XEDIT suppresses its verification.

Figure 3-3. Suppressing Verification

SUPPRESSING PRINTING OF FILE MARKS

XEDIT prints the messages --EOR-- and --EOF-- when it reads an end-of-record mark or end-of-file mark while processing a command. A user may turn off and turn on the printing of these file marks by choosing one of the following commands.

<table>
<thead>
<tr>
<th>Format</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEOF+</td>
<td>Turns on the printing of --EOF-- messages</td>
</tr>
<tr>
<td>TEOF-</td>
<td>Turns off the printing of --EOF-- messages</td>
</tr>
<tr>
<td>TEOF</td>
<td>Toggles printing of --EOF-- messages</td>
</tr>
<tr>
<td>TEOR+</td>
<td>Turns on the printing of --EOR-- messages</td>
</tr>
<tr>
<td>TEOR-</td>
<td>Turns off the printing of --EOR-- messages</td>
</tr>
<tr>
<td>TEOR</td>
<td>Toggles printing of --EOR-- messages</td>
</tr>
</tbody>
</table>

Figure 3-4 shows suppression of the printing of file marks.

SUPPRESSING TERMINAL BELL RINGING

The NOBELLS command allows the user to suppress bell ringing when XEDIT messages are printed. Its format is:

NOBELLS

or

NB

Figure 3-5 tests this command.

?? print*
EXECUTIVE MANAGEMENT*
--EOR--
00100 A.B. KRAMER HQR24
00110 J.J. THOMPSON SWP19
--EOR--
--EOF--
00120 B.C. MILLER HQR44
END OF FILE
??

?? print*
EXECUTIVE MANAGEMENT*
--EOR--
00100 A.B. KRAMER HQR24
00110 J.J. THOMPSON SWP19
00120 B.C. MILLER HQR44
END OF FILE
??

The user lists a file containing end-of-record and end-of-file marks.

A TEOR command and a TEOF command are issued to suppress the printing of the --EOR-- and --EOF-- messages. The file is listed again.

Figure 3-4. Suppressing Printing of File Marks
INTERRUPTING XEDIT PROCESSING

The user can interrupt XEDIT processing without terminating XEDIT execution. To stop the printing of XEDIT output, the user presses the BREAK key (or its equivalent) on non-IAF terminals and the BREAK key (or its equivalent), the colon (:) or the right parentheses [)], and the \( \text{\textregistered} \) on IAF terminals. XEDIT prints a double question mark to request entry of the next editing command.

If the interruption occurs while a Z, Y, or delimited command sequence is being processed, the remaining component commands in the command list are skipped.

To stop XEDIT requests for editing data (single question mark prompts), the user presses the carriage return.

ON-LINE REFERENCE AIDS

The user can issue a HELP command for a description of a command or an EXPLAIN command for an explanation of an XEDIT message.

ASKING FOR ERROR MESSAGE EXPLANATIONS

The EXPLAIN command lets a user request an explanation of an error message that has just been printed. The messages explained by the EXPLAIN command are listed in appendix B. Its format is:

```
EXPLAIN \( \text{\textregistered} \)
```

Figure 3-6 gives the explanation for the error message, NO SUCH COMMAND.

ASKING FOR COMMAND DESCRIBITIONS

The HELP command enables a user to request information about a specific command. If he issues the HELP command without specifying a command or command abbreviation, XEDIT prints a list of all its commands. Its format is:

```
HELP, command \( \text{\textregistered} \)
```

or

```
H, command \( \text{\textregistered} \)
```

Figure 3-7 shows the description of the HELP command.

```
?? get, plan
NO SUCH COMMAND
?? explain

EXPLANATION OF-
NO SUCH COMMAND
THE COMMAND IS ILLEGAL OR AN IMPROPER SEPARATOR
WAS USED AFTER THE COMMAND.
??
```

Figure 3-6. Error Message Explanation

```
?? help, print
PRINT $ [%]
```

The user enters an illegal command. XEDIT prints the appropriate error message. The user requests an explanation. XEDIT prints the explanation of the NO SUCH COMMAND message.

Action- Prints $ lines starting at the current pointer position. The pointer is left positioned at the last line printed.

Figure 3-7. Command Description
During editing, XEDIT maintains a pointer positioned at the file line currently being processed. The pointer is positioned according to these rules.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Pointer Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>XEDIT is called.</td>
<td>Beginning-of-information.</td>
</tr>
<tr>
<td>A command is processed.</td>
<td>Last line processed by the command (usually the last line printed in verify mode).</td>
</tr>
<tr>
<td>Interruption of XEDIT output</td>
<td>Last line processed.</td>
</tr>
<tr>
<td>Command processing reaches the end-of-information.</td>
<td>The message END OF FILE is printed, further processing of the command stops, and the pointer is repositioned at the beginning of the file.</td>
</tr>
</tbody>
</table>

**LOCATING LINES VIA LINE NUMBERS**

When editing a file sequenced by line numbers, the user can advance the pointer to a line identified by a specific line number by entering the line number after a command prompt. XEDIT begins its search for the specified line number at the current pointer position. The search is circular; the top of the file may be passed in order to position the pointer at the specified line. Processing of the command ends when the pointer is positioned at the line having the specified line number. If no such line exists within the line number sequence, the pointer is positioned at the line with the next highest line number. Its format is:

```
line number @
```

Figure 4-1 demonstrates this command.

**LOCATING LINES VIA SPECIFIED STRINGS**

The user can advance the pointer to a line containing a specific string of alphanumeric characters by issuing a LOCATE command. LOCATE commands take three different forms. One form applies when the user wants to find a line containing a specified string. The second form applies when the user wants XEDIT to find a line with the first string followed by a second string. The third form applies when XEDIT is to find a line with the first string not followed by a second string.

When the user issues a LOCATE command with verify mode in effect, XEDIT prints the n lines containing the specified string. The pointer is positioned at the last line printed.

When the user wants to locate a line containing a specific string, the format is:

```
LOCATE/string/n @
```

or

```
L/string/n @
```

The user issues a line number command to locate line 120.
XEDIT verifies that the pointer is positioned at line 120.
The user enters line number 140.
XEDIT positions the pointer at line 150 because line 140 does not exist in this file and 150 is the next highest line number.
The user enters a 190 to find line 190.

The user enters a 170 to find line 170.
XEDIT performs a circular search to find line 170 and prints the line.

Figure 4-1. Locating Lines Via Line Numbers
string

String of alphanumerics characters which XEDIT attempts to locate. If the user omits the terminating delimiter, XEDIT prints an informative message and processes the command as if a delimiter existed after the last nonblank character.

n
Number of lines to be located containing the specified string. The file pointer is positioned at the last line located. Highest allowable value is 99999; default value is 1. If n is 0, the search is restricted to the current line.

The LOCATE command is used in figure 4-2 to find lines containing a specified string.

When the user wants to locate a line that contains a certain string followed by another string, the format is:

LOCATE/string1...string2/n

or

L/string1...string2/n

string1...string2 string1 followed by string2. If the user omits the terminating delimiter, XEDIT assumes one exists after the last nonblank character.

n
Number of lines to be located containing the specified strings. Highest allowable value is 99999; default value is 1. If n is 0, the search is restricted to the current line.

The user issues a prefixed LOCATE command to position the pointer at the first line containing the string ZIP.

XEDIT verifies that line 110 is located.

The user issues a slash-prefixed LOCATE command to advance the pointer one line before the search.

XEDIT prints the next two lines containing the string ZIP.

The user issues an abbreviated LOCATE command with a zero count parameter.
The string 8711 is not found on the current line.

Figure 4-2. LOCATE With a Specified String

The user issues a LOCATE command to find a line containing the string WILSHIRE followed by the string ZIP 55711.

XEDIT reads the end-of-information mark without finding the specified string.

The user issues a LOCATE command to find a line containing the string WILSHIRE followed by the string 55722.

XEDIT prints line 110.

Figure 4-3. LOCATE With Two Specified Strings
ADVANCING AND REVERSING THE POINTER

When the user wants to advance the pointer toward the end of the file, he enters a NEXT command with the following format:

NEXT n

or

N n

n Number of lines that the pointer is advanced. Highest allowable value is 99999; default value is 1.

When the user wants to reverse the pointer toward the beginning-of-information, he enters a NEXT command specifying a negative number of lines as follows:

NEXT -n

or

N-n

n Number of lines that the pointer should be moved toward the beginning-of-information. Processing ends when the pointer reaches the beginning-of-information. Maximum value is 99999. If n is 0 or if n is omitted, no pointer movement is performed.

Reverse pointer movements using the NEXT command are much slower than forward pointer movements. Figure 4-5 shows pointer movement in both directions.

POSITIONING POINTER AT TOP OR BOTTOM OF FILE

A user can position the pointer at the beginning-of-information by issuing a TOP command in the following format.

TOP

or

T

The user can move the pointer to the bottom of the current record in a file by issuing a BOTTOM command in the following format.

BOTTOM

or

B

After positioning the file pointer at the bottom of the current record, the pointer can be moved to the bottom of the next record by entering a NEXT command and another BOTTOM command.

Figure 4-6 uses both the BOTTOM and the TOP commands.

The user issues a PRINT command to list six lines. XEDIT prints six lines.

The user issues a NEXT command to reverse the pointer three lines. XEDIT verifies where the pointer is positioned.

The user issues a NEXT command to advance the pointer two lines. XEDIT verifies where the pointer is positioned.

Figure 4-5. The NEXT Command
The user positions the pointer at line 160. XEDIT prints the location of the pointer.
The user issues a TOP command to position the pointer at the first line in the file; no automatic verification is done.
The user issues a PRINT command.
XEDIT prints the first line in the file.
The user issues a BOTTOM command to move the pointer to the last line in the current record of the file.
XEDIT prints the bottom line of the record.

Figure 4-6. TOP and BOTTOM

POSITIONING POINTER BY LISTING LINES

A user can list lines from a file with the PRINT command. This command begins its listing at the current pointer position. When the listing is finished, the pointer is positioned at the last line printed. PRINT commands have the following format:

PRINT n  

or

P n  

n
Number of lines which the user wants printed. Maximum value is 99999; default value is 1. If n is *, every line of the file from the current pointer position to the end of information is printed. If n is 0, the current line is printed.

?? print
00160 Q.E. SMITH
?? print 4
00160 Q.E. SMITH
00170 P O.B. 55 ZIP 55
190 P.T. BEE
00200 8710 14TH ST ZIP 55713
?? print*
00200 8710 14TH ST ZIP 55713
00210 EXT 18
END OF FILE
?? print*

The user issues a PRINT command to list the current line. XEDIT prints line 160. The user issues a PRINT command to list four lines. Next, the user lists all lines from the current pointer position to the end-of-information. XEDIT prints lines 160 through 210 and the message END OF FILE. The final PRINT command lists the entire file because the pointer was automatically repositioned to the beginning-of-information by the last command.

?? print*
00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP 55722
00120 EXTENSION 6533
00160 Q.E. SMITH
00170 P O.B. 55 ZIP 55
190 P.T. BEE
00200 8710 14TH ST ZIP 55713
00210 EXT 18
END OF FILE
??

Figure 4-7. The PRINT Command

PRINTING THE CURRENT LINE COUNT

The WHERE command gives the user the current line count. Calculated and printed by XEDIT, this is the number of lines between the beginning-of-information and the current pointer position. The format is:

WHERE 

or

W  

Figure 4-8 uses the BOTTOM and WHERE commands.

?? bottom
ATTACH, XEDIT 64/UN=XSCB.
?? where 5
??

The user issues a BOTTOM command and a WHERE command to find the number of lines in the record.

Figure 4-8. The WHERE Command
The user can modify specific strings of alphanumeric characters within a file line by issuing string editing commands. The following section describes these commands in detail.

**ADDING A STRING TO THE END OF A LINE**

The user can append a string to the end of one or more file lines with an ADD command. After receiving an ADD command, XEDIT prints a single question mark (?) to prompt entry of the string to be appended. After entry of the string, XEDIT prints the lines to which the entered string has been appended starting at the current pointer position. The ADD command has the following format.

```
ADD n  
```

or

```
A n  
```

where:

- **n** Number of consecutive lines to which the specified string should be appended. Maximum value is 99999; default value is 1; an * can be entered for n when the string is to be appended to every line from the current pointer position to the end of information. When n is 0, the string is appended to the current line.

Figure 5-1 is an example of the use of the ADD command.

**REPLACING, DELETING, AND INSERTING STRINGS**

CHANGE and CHANGES commands can:

- Replace one string with a different string
- Delete strings from file lines
- Insert strings at the beginning of a line or at the beginning of a window set by the WMARGIN command

This windowing feature can also restrict the columns in which the change can occur. If the user prefers to change strings by visual character-by-character alignment instead of by context, he can use the MODIFY command described later in this section.

**REPLACING STRINGS BY CONTEXT**

The user can replace one string with a different string of any length and character content with a CHANGE or CHANGES command. However, to use the edited file within NOS, the user must not create a file line longer than 150 characters. The search for the first specified string starts at the line designated by the current pointer position. These formats are:

```
CHANGE/string1/string2/n  
```

or

```
C/string1/string2/n  
```

or

```
C/string1a...string1b/string2/n  
```

or

```
CHANGES/string1/string2/m  
```

or

```
CS/string1/string2/m  
```

or

```
CS/string1a...string1b/string2/m  
```

?? 160add
?  jr.
00160  O.S. SMITH JR.
??

The user positions the pointer to line 160 and issues an ADD command. XEDIT prints a single ?. The user enters a blank space and then the characters JR. XEDIT prints the edited line.

Figure 5-1. The ADD Command
string1 String to be replaced.
string1a...string1b stringla followed by stringlb. All characters between and including these two strings are replaced by string2.
string2 String to replace string1 (can be any length).

n Number of lines containing an occurrence of string1 which should be changed. XEDIT changes every occurrence of string1 in the specified lines. If n is omitted, XEDIT changes the first line found containing string1. Maximum value is 99999. If n is *, XEDIT replaces every occurrence of string1 between the current pointer position and the end of information. If n is 0, the search for string1 is restricted to the current line.

m Number of occurrences of string1 or string1a...string1b that are to be replaced. If the user omits the m value, XEDIT replaces the next occurrence of the string.

DELETING STRINGS BY CONTEXT

CHANGE or CHANGES commands can also be used to delete a string by omitting string2. These commands begin editing at the current line position and continue through n file lines or m occurrences of string1. These formats are:

**CHANGE/string1//n**

or

**C/string1//n**

or

**C/string1a...string1b//n**

or

**CHANGES/string1//m**

or

**CS/string1//m**

or

**CS/string1a...string1b//m**

---

Figure 5-2. Use of CHANGE to Modify a Line
string1 String of characters to be deleted.
string1a...string1b string1a followed by string1b. All characters within and between these two strings in a line are deleted.

n Number of lines containing string1 that should be processed by this command. Maximum value is 99999; default value is 1. If n is *, every line is deleted from the current pointer position to the end of information. If n is 0, XEDIT does not advance the pointer, and only the current line may be deleted.

m Number of occurrences of string1 or string1a...string1b to be deleted. If the user omits the m value, XEDIT deletes the next occurrence of the string.

The CHANGE command can delete selected lines as shown in figure 5-3.

**INSERTING STRINGS AT THE BEGINNING OF A LINE**

A user can insert a string before the first (leftmost) character of a line using a CHANGE or CHANGES command. To do this, he omits string1. The format is:

CHANGE///string2/n

or

C///string2/n

or

CHANGES///string2/n

or

CS///string2/n

The user lists the file.

The user issues a CHANGE command to delete every occurrence of the string ZIP from every line in the file.

XEDIT prints the edited lines.

The user then corrects the name in line 190 by issuing a CHANGES command.

Figure 5-3. Selective Deletion via CHANGE Command
string2 String of characters which the user wants to insert before the leftmost character of a line.

n Number of lines before which string2 is to be inserted. Maximum value is 99999; default value is 1. An # indicates that the insertion should occur before every line from the current pointer position to the end-of-information. If n is 0, XEDIT does not advance the pointer, and the insertion occurs at the current pointer position.

Figure 5-4 shows insertion of a string using the CHANGE command.

RESTRICTING THE STRING SEARCH

The user can restrict the search for a string by issuing one of the following commands.

TRIM Ignores trailing blanks on string search commands.

WMARGIN Defines column settings which restrict the scope of a string search command when used with the W or A command suffix characters.

IGNORING TRAILING BLANKS

A user can tell XEDIT to ignore trailing blanks on string searches by issuing the TRIM command. The TRIM command has three formats.

TRIM+ Ignores trailing blanks.

TRIM- Uses trailing blanks.

TRIM Toggles the TRIM setting.

The commands whose searches are limited by the TRIM command are:

CHANGE

CHANGES

COPY

COPYD

DELETE

LOCATE

OCTCHANGE

Lines that are entirely blank are not searched when XEDIT ignores trailing blanks. The TRIM command alters searches as shown in figure 5-5.

**print3**

00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP 55722
00120 EXT 6533
?? "change/\ext/3"
EX00100 M.T. JONES
EX00110 1544 WILSHIRE ST ZIP 55722
EX00120 EXT 6533
??

(1) The user lists three lines.

(2) The user inserts the string EX before each of the three lines.

Figure 5-4. Use of CHANGE to Insert a String

**print**

*EXECUTIVE MANAGEMENT*

EX00100 A.B. KRAMER HQR24130
EX00110 J.J. JOHNSON SWP19130
EX00130 B.C. MILLER HQR44130
END OF FILE
?? locate/130 / 2
EX00100 A.B. KRAMER HQR24130
?? trim
?? locate/130 / 2
EX00130 B.C. MILLER HQR44130
??

(1) The user lists the file.

(2) The user attempts to find manager number 130 but instead finds manager number 100 with the facility number HQR24130.

(3) The user issues a TRIM command.

(4) XEDIT now locates the desired line.

Figure 5-5. The TRIM Command
DEFINING A WINDOW

A user can restrict the scope of a string search to a specified range of columns through use of the WMARGIN command. Once the window columns have been defined, the user can ask for a search restricted to those columns. The user makes that request by appending a W or an A suffix character to the end of any string search command (for example, CHANGEW and LOCATEA). To set the left and right window margin columns, the WMARGIN command is used as follows:

WMARGIN lm rm

or

WM lm rm

lm  Column position setting of the left window margin
rm  Column position setting of the right window margin

The left margin must be less than or equal to the right margin.

USING A WINDOW

Once a window has been defined by the WMARGIN command, the user can request that a search be restricted to those columns by appending either a W or A suffix character to the end of any string search command. The legal commands are:

?? p  1
ABCDFGHJKLM
?? wmargin 4 11  2
?? lw/ef...ij/  3
ABCDFGHJKLM
?? lw/ef---m/
ABCDGFLHIJKLMNOP
?? la/klm/
ABCDGFLHIJKLMNOP
?? la/klm/
ABCDGFLHIJKLMNOP
?? la/ef---m/
END OF FILE
?? la/ef---h/
END OF FILE
?? la/ef...klm/
ABCDGFLHIJKLMNOP
?? la/klm/
ABCDGFLHIJKLMNOP
?? la/ef---m/
END OF FILE
?? la/ef...ij/
END OF FILE
?? la/def---zz/
ABCDGFLHIJKLMNOP
?? lw---lm/
ABCDGFLHIJKLMNOP
?? la/def---lm/
ABCDGFLHIJKLMNOP
?? ow/ef...ij/xx/
ABCDXJKL
?? ca/xx/jkl/
ABCDXJKL
?? ow/xx/
ABCDXJKL
??

A suffix character can also be appended to the corresponding command abbreviations.

The W or window suffix character requires that all characters of the string specified by the command reside within the window; otherwise, XEDIT does not find the string.

The A or anchor suffix character requires that the first character of the string specified by the command reside within the window; otherwise, XEDIT does not find the string. All other characters can extend beyond the window.

These suffix characters are used 14 ways in figure 5-6.

The windowing restriction is used to search an employee directory in figure 5-7.

Figure 5-6. Use of the Suffix Characters
MODIFYING STRINGS CHARACTER-BY-CHARACTER

Three XEDIT commands enable users to alter the contents of a line on a character-by-character basis.† They are MODIFY, QMOD, and YQMOD. These commands differ only in the alignment guides they provide for the modification directives and in the number of lines a single command can modify. The alignment guide for the MODIFY command is the contents of the line at the current pointer position. MODIFY can change only one line at a time. QMOD provides a row of column numbers as an alignment guide for the modification directives and executes those directives on the number of lines specified. YQMOD does not provide an alignment guide but is otherwise identical to QMOD. The procedure for using these commands is:

1. Position the file pointer at the first line to be modified.
2. Issue a modifying command (MODIFY, QMOD, or YQMOD).
3. XEDIT prints the alignment guide, and on the subsequent line, a single question mark.
4. Align and enter the appropriate modification directives under the character or column to be modified (refer to table 5-1).
5. XEDIT prints the modified line or lines.

If no modification directives are entered after the single question mark, no changes are made and no verification is given. The prefixed command XYQMOD suppresses the printing of the alignment guide and the modified lines.

† Since correct alignment is crucial for the successful processing of these commands, the user should be certain that his terminal carriage operates at the correct speed. Adjustment can be made on non-IAF terminals with the ROUT command as described in the NOS Time-Sharing User's Reference Manual and on IAF terminals with the terminal definition command as described in the IAP Reference Manual.

The user lists the file.

He defines the window as columns 10 through 15.

To locate employee number 208, the user enters an abbreviated LOCATE command with the suffix W so that XEDIT searches only the employee number field.

XEDIT finds employee number 208, Jack Johnson.

To find the first employee whose last name is Scott, the user redefines the window to be column 17 only.

The user issues an abbreviated LOCATE command with the suffix A.

XEDIT correctly finds the name Frank Scott.

Figure 5-7. Restricting Field of Search

NOTE

The user should not allow a modified line to exceed 150 characters.

The following are the formats of the three modification commands and examples of their use.

MODIFY

or

M

Figure 5-8 shows the MODIFY command.

QMOD n

or

QM n

n Number of lines starting at the current pointer position that is to be modified. Highest allowable value is 99999; default value is 1. If n is 0, every line in the file between the current pointer position and the end-of-information is modified. If n is 0, the line at the current pointer position is modified.

QMOD is often used when each line of a file consists of a series of fields each covering a specified range of columns as shown in figure 5-6.
10 modify ①
10 THIS STRING TO BE MODIFIED ②
10 THIS IS THE STRING TO BE MODIFIED ③

① The user issues a MODIFY command with a line number prefix to modify line 10.
② XEDIT lists line 10.
③ The user enters MODIFY directives to alter line 10. XEDIT verifies the changes in line 10.

Figure 5-8. The MODIFY Command

?? print④
00100 PART NO= 749322 QUAN= 757 VALUE= 945.50
00110 PART NO= 749323 QUAN= 1298 VALUE= 23.95
00120 PART NO= 749324 QUAN= 446 VALUE= 138.05
00130 PART NO= 749325 QUAN= 15 VALUE= 1650.50
00140 PART NO= 749326 QUAN= 376 VALUE= 182.75
END OF FILE

?? qmod④
0 1 2 3 4 5 6
123456789 0123456789 01 234567890 1234567890 1234567890 1234567890

④ XEDIT prompts for a QMOD command.

In the first entry, the user issues a PRINT command to list the entire file. XEDIT lists the file.
In the next entry, the user issues a QMOD command with n equal to *, indicating that his subsequent modification directives should change every file line. XEDIT prints a sequence of column numbers and a single question mark prompt. The user enters appropriate modification directives in the directives line (a hyphen in front of column 20 and the word COST replacing the word VALUE at column 42). XEDIT verifies the modifications in the file.

Figure 5-9. The QMOD Command

YQMOD n ⑤
or
YQM n ⑤

n Number of lines to be modified by the line of modification directives. Highest allowable value is 99999; default value is 1. If n is *, all lines from the current pointer position to the end of information are modified. If n is 0, the line at the current pointer position is modified.

Figure 5-10 shows a similar editing operation performed with the YQMOD command.
The user lists the entire file and then enters a YQMOD command specifying that every line in the file is to be modified. XEDIT prints a single question mark. The user counts over seven spaces and inserts the word FACTORY before column 7. XEDIT verifies the modification made in the file.

Figure 5-10. The YQMOD Command

### TABLE 5-1. MODIFICATION DIRECTIVES

<table>
<thead>
<tr>
<th>Directive</th>
<th>Function</th>
<th>Example</th>
</tr>
</thead>
</table>
| `string#` | XEDIT inserts the string of alphanumeric characters between the `#` and the `#` before the character pointed to by the `#`. Issuing a `#` combination places a `#` in the file line. If an `&` or a `#` is part of the string entered, that special character is treated as a normal character instead of a directive. If a `#` appears before a string but the terminating `#` is missing, XEDIT assumes the `#` should appear after the last nonblank character in the directives line. | `?? 200modify 00200 8710 14TH ST ZIP 55713
? 170modify 00170 PO 55 ZIP 55703
??` |
| `#` | When a `#` is entered without any trailing characters, XEDIT inserts a blank in front of the character above the `#`. When a `#` is entered without a preceding `#`, XEDIT deletes the character above the `#` and closes up the space left by the deletion. | `??` |
| Blank space | The character above a blank is left unchanged. | `??` |
| `&` | XEDIT deletes the character above an `&` and replaces it with a blank. | `??` |
| Other alphanumeric characters | The character in the file line is replaced by the character in the directives line. | `160 00160 O.E. SMITH
??` |
A user can change entire lines within a file with XEDIT line editing commands. These commands are discussed in the following section.

**DELETING LINES**

A user can delete entire lines from a file by using any of the following criteria,

- A sequence of lines starting at the current pointer position
- Selective lines on the basis of a specified string
- A single line with a specified string without moving the file pointer if the string is not found
- Selective lines if a specified string occurs within a specified set of columns (refer to the WMARGIN command)
- Selective lines when they are copied to another file (refer to the COPYD command)

The first three listed use the DELETE command. The DELETE command leaves the file pointer positioned at the line after the last deleted line. The formats are:

```
DELETE n
```

or

```
D n
```

```
?? 130 ①
00130 R.P. PEEPERS ②
?? delete 3
00170 R.P. PEEPERS
00140 116 WEST ELM DRIVE ZIP 55648
00150 EXT 3222
?? 190 ③
00190 P. T. BEE
?? x/delete
END OF FILE
??
```

① The user positions the file pointer at line 130 and issues a DELETE command to delete three lines.
② XEDIT verifies the lines deleted.
③ The user next positions the pointer to line 190 and then issues a DELETE command to remove all lines from line 190 to the end-of-information.

XEDIT indicates that the end-of-information was encountered but does not verify which lines were deleted because an x prefix was used on the DELETE command.

This format of the DELETE command is used in figure 6-1.

```
DELETE/string/n ④
```

or

```
D/string/n ④
```

```
string  String of characters that determines if a line should be deleted.
```

```
?? Number of lines containing the specified string that is to be deleted.
```

This second format of the DELETE command is used in figure 6-2.

Figure 6-1. Deleting Selected Consecutive Lines
The user issues a DELETE command to delete every line in the file containing the string EXT starting at the current pointer position.

XEDIT verifies the lines deleted.

DELETestring1...string2/n
or
D/string1...string2/n

string1...string2 string1 followed by string2. If XEDIT finds string1 followed by string2 in a line, that line is deleted. Any number of other characters may separate the two strings within the line.

n Number of lines containing string1 followed by string2 that should be deleted. Maximum value is 99999; default value is 1. If n is *, every line containing string1 followed by string2 from the current pointer position to the end-of-information is deleted. If n is 0, the line at the current pointer position is deleted if it contains string1 not followed by string2.

Figure 6-3 shows line deletion on the basis of two specified strings.

DELETestring1---string2/n
or
D/string1---string2/n

string1---string2 string1 is not followed by string2. If XEDIT finds string1 without string2 following it in the same line, that line is deleted.

n Number of lines containing string1 not followed by string2 that should be deleted. Maximum value is 99999; default value is 1. If n is *, every line containing string1 not followed by string2 from the current pointer position to the end-of-information is deleted. If n is 0, the line at the current pointer position is deleted if it contains string1 followed by string2.

The user lists the entire file.

The user then issues a DELETE command to remove every line in the file that contains the string WILSHIRE ST followed by the string 55722.

XEDIT verifies which lines are deleted.
REPLACING LINES

The REPLACE command allows the user to replace a specific number of existing file lines starting at the current pointer position with the same number of substitute lines. After the user issues the command, XEDIT prints a single question mark to inform the user that he should enter his first substitute line. XEDIT continues to print single question mark prompts until the specified number of substitute lines has been entered. If the user enters a carriage return immediately after the ? prompt, XEDIT stops processing the REPLACE command and requests the next command. To replace a line with a blank line, the user must enter a space before the carriage return. The format is:

REPLACE n

or

R n

n Number of consecutive lines starting at the current pointer position that is to be replaced. Maximum value is 99999; default value is 1. If n is 9, every line from the current pointer position to the end of information is replaced. If n is 0, the line at the current pointer position is replaced.

Figure 6-4 is an example of the use of the REPLACE command.

INSERTING LINES

The user can insert entire new lines into a file, without affecting any existing lines, in three different ways.

- INSERT commands insert a specific number of lines after the current pointer position
- INSERTB commands insert a specific number of lines before the current pointer position
- Entering INPUT mode inserts any number of lines after the current pointer position

INSERT COMMAND

The INSERT command lets a user insert a specific number of lines into a file immediately after the line at the current pointer position. When the user enters the INSERT command, XEDIT prints a single question mark to prompt entry of the first line to be inserted. XEDIT continues to print single question mark prompts until the specified number of lines to be inserted has been entered. If the user enters a carriage return immediately after the ? prompt, XEDIT stops processing the INSERT command and requests the next command. To insert a blank line, the user enters a space before the carriage return. After processing an INSERT command, the file pointer is positioned at the last line inserted. The format is:

INSERT n

or

I n

n Number of lines which the user wants to insert. Maximum value is 99999; default value is 1. If n is 0, XEDIT assumes that one line is to be inserted.

Figure 6-5 demonstrates the INSERT command.

INSERTB COMMAND

When the user wants to insert a specific number of lines into a file before the line at the current pointer position, he issues an INSERTB command which indicates how many lines are to be inserted. Entry of the lines to be inserted is the same as for the INSERT command. The first line entered is the first line of the insertion sequence within the file. After processing an INSERTB command, the pointer is at the same position it occupied before the command was issued. The format is:

INSERTB n

or

IB n

n Number of lines which the user wants to insert. Maximum value is 99999; default value is 1. If n is 0, XEDIT assumes that one line is to be inserted.

Figure 6-6 shows the use of the INSERTB command.

?? 170 ①
00170 322 WILSHIRE ST ZIP CODE 55723
?? replace2
?? 00170 18 park place apt 111 zip 55704
?? 00180 ext 8886 ②
??

① The user positions the pointer at line 170 and then issues a REPLACE command to replace line 170 and the line following it. XEDIT prints a single ?.
② XEDIT prints a second ? prompt.
③ The user enters a modified address.
④ The user enters a modified extension number.

Figure 6-4. The REPLACE Command

60455730 A 6-3
?? 120
00120 EXT 6533
?? insert 3
? 00130 a.b. newton
? 00140 166 haskell circle zip 55713
? 00150 ext 227
?? ▲ print #4
### NAMES/ADDRESSES ARE FICTITIOUS ###
00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP CODE 55722
00120 EXT 6533
00130 A.B. NEWTON
00140 166 HASKELL CIRCLE ZIP 55713
00150 EXT 227
00160 T.G. SLATER
00170 322 WILSHIRE ST ZIP CODE 55723
00180 EXT 67
00190 R.C. CARTER
00200 6100 WILSHIRE ST ZIP CODE 55722
00210 EXT 1101
END OF FILE
??

1. The user positions the pointer at line 120.
2. He then issues an INSERT command to insert three lines after line 120.
3. XEDIT prints a single question mark and the user enters a line containing an employee name.
4. The user enters a line containing an address.
5. The user enters a line containing a phone extension.
6. The user lists the entire file to check if the new lines, 130 to 150, were inserted correctly.

Figure 6-5. The INSERT Command

?? 160
00160 T.G. SLATER
?? insertb 3
? 00157 A.P. macdonald
? 00158 1313 lemontree ave zip 55722
? 00159 ext 5339
?? ▲ print6
00140 166 HASKELL CIRCLE ZIP 55713
00150 EXT 227
00157 A.P. MACDONALD
00158 1313 LEMONTREE AVE ZIP 55722
00159 EXT 5339
00160 T.G. SLATER
??

1. The user positions the pointer at line 160.
2. He then issues an INSERTB command to insert three lines before line 160.
3. The first line entered is line 157 which is to be the top line of the inserted sequence.
4. The following two lines are inserted.
5. The user lists the sequence to check if the lines were inserted correctly.

Figure 6-6. The INSERTB Command

INPUT MODE

Input mode enables a user to insert an unspecified number of lines following the current file pointer position. To enter input mode, the user performs one of the following.

- Presses carriage return immediately after a ?? prompt
- Enters the INPUT command
- Enters the INPUT command followed by the desired escape character

XEDIT prints the message INPUT, and on the following line, a single question mark. This indicates that the user is in input mode and can enter the first line to be inserted. XEDIT continues to print ? prompts until the user leaves input mode.

The user can leave input mode by entering one of the following after a single question mark prompt.

- A carriage return
- The EDIT command prefixed by the escape character

Batch origin jobs cannot use the carriage return method. They must enter input mode by the INPUT command followed by an escape character and leave input mode by the EDIT command prefixed by the escape character.

If an escape character is specified when entering input mode, the user can process XEDIT commands prefixed by the escape character while in input mode. He is restricted, however, to commands which do not move the file pointer (such as REPLACE instead of DELETE). Whenever the
escape character is in the first character position of a line, XEDIT attempts to process the rest of the line as a command. If the command entered after the escape character would move the file pointer, the message COMMAND NOT VALID is printed. If the command specifies a repetition parameter (n or m) other than null or zero, the message ARGUMENT ERROR is printed, and the command is not processed. Processing continues in input mode.

The escape character is recognized as such only when it is in the first column of a line. Therefore, if the tab character is the same as the escape character, the tab character is effective in all columns except the first. If placed there, the rest of the line is interpreted as a command.

The following formats are valid for the INPUT and EDIT commands.

```
INPUT e ②
eEDIT ③
```

① The escape character may be any character except space or the command delimiter set by the DELIMIT command (refer to section 10). When INPUT, is entered, a comma becomes the escape character.

Figure 6-7 uses the INPUT command to enter input mode.

Figure 6-8 uses ② to enter input mode.

```
?? 120①
00120 EXT 6533
?? input $ ②
INPUT ③
? 00121 m.c. greenway ④
? $change/eee/ee/ ⑤
00121 M.C. GREENWAY ⑥
? 00122 867 mission st zip 55744
? $edit? ⑦
EDIT
??
```

① The user positions the pointer at line 120.
② The user then issues the INPUT command with a dollar sign ($) as the escape character.
③ XEDIT indicates that it is ready to accept lines to be inserted after the current pointer position.
④ After entering a line, the user notices he misspelled GREENWAY and issues a CHANGE command prefixed by the escape character ($).
⑤ XEDIT verifies that the change was made.
⑥ The user inserts another line of text.
⑦ The user enters the EDIT command prefixed by the escape character to leave input mode.

Figure 6-7. The INPUT Command

```
?? 120①
00120 EXT 6533
??
INPUT③
? 00122 m.c. greenway④
? 00123 867 mission st zip 55744
? 00123 ext 1500
? 00124 special class**supervisor
? 00125 emp: 12
? 00126 list: c
?④
EDIT
?? t?⑤
?? p*
### NAMES/ADDRESSES ARE FICTITIOUS ###
00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP CODE 55722
00120 EXT 6533
00121 M.C. GREENWAY
00122 867 MISSION ST ZIP 55744
00123 EXT 1500
00124 SPECIAL CLASS**SUPERVISOR
00125 EMP: 12
00126 LIST: C
00130 A.B. NEWTON
00140 166 HASKELL CIRCLE ZIP 55713
00150 EXT 227
00157 A.P. MACDONALD
00158 1313 LEMON
??⑧
```

① The user positions the pointer to line 120.
② The user presses carriage return to enter input mode.
③ The user inserts a series of lines.

④ The user presses carriage return to leave input mode.
⑤ The user issues abbreviated TOP and PRINT commands to list the entire file.

⑥ The user interrupts the listing.

Figure 6-8. The ② Command
The user can issue commands to modify the line numbers that sequence a file. These general conventions apply to all line numbering commands.

- Before any line numbering command is processed, the file pointer is automatically set to the beginning of the file.
- After processing all line numbers within a file and reading the end-of-information mark, XEDIT prints the message END OF FILE to indicate that the specified line number editing is finished. The pointer is automatically repositioned to the beginning of the file.
- A line numbering command must not cause the edited file line to exceed 150 characters if the file is to be processed under NOS.
- No line number can exceed a value of 99999. If a line numbering command would break this rule, the file is restored to its original condition, an informative message is printed, and the pointer is positioned at the beginning of the file.
- Line numbering commands are primarily intended for editing line-numbered card-image text files and programs written in time-sharing FORTRAN Extended. These commands are not practical for use on BASIC programs, since BASIC branch line numbers are not modified by these commands.
- XEDIT does not verify the effects of any line numbering command.

ADDLN Command

Two XEDIT commands allow users to add line numbers to a file.

- ADDLN: Inserts a line number before each line
- ADDLNS: Inserts a line number and a blank space before each line

If line numbers already exist in the file, ADDLN and ADDLNS insert an additional set of line numbers.

ADDLN Command

When a user wants to add a line number without a trailing space to each line in a file, he can enter an ADDLN command in the following format.

```
ADDLN ln n
```

or

```
ALN ln n
```

ln Line number to be inserted before the first line in a file. No line number can exceed 99999. Default is 1.

n Number by which each line number in the file is incremented. No line number can exceed a value of 99999. Default is 1.

Figure 7-1 is an example of this command.

ADDLNS Command

When the user wants to add both a line number and a trailing blank space to each line in a file, he enters an ADDLNS command in the following format.

```
ADDLNS ln n
```

or

```
ALNS ln n
```

ln Line number to be inserted before the first line in the file. No line number can exceed 99999. Default is 1.

n Number by which each line number is incremented. No line number can exceed 99999. Default is 1.

Figure 7-2 demonstrates the ADDLNS command.

DELETING LINE NUMBERS

The DELETELN command enables the user to remove every line number from a file. It has no effect on lines without line numbers. The format is:

```
DELETELN
```

or

```
DLN
```

Figure 7-3 shows a use of this command.
?? p* ①
M.T. JONES
1544 WILSHIRE ST ZIP CODE 55722
EXT 6533
A.B. NEWTON
166 HASKELL CIRCLE ZIP 55713
EXT 227
A.P. MACDONALD
1313 LEMONTREE AVE ZIP 55722
EXT 5339
END OF FILE
?? addln 00010 10 ②
END OF FILE
?? p* ③
00010 M.T. JONES
00020 1544 WILSHIRE ST ZIP CODE 55722
00030 EXT 6533
00040 A.B. NEWTON
00050 166 HASKELL CIRCLE ZIP 55713
00060 EXT 227
00070 A.P. MACDONALD
00080 1313 LEMONTREE AVE ZIP 55722
00090 EXT 5339
END OF FILE
??

① The user lists a file in which the first character of each file line is a blank.
② He inserts a set of line numbers beginning with 00010 and increasing by 10.
③ He then lists the file to check the effect of the ADDLN command.

Figure 7-1. The ADDLN Command

?? p* ①
Q.E. SMITH
18 PARK PLACE APT 111 ZIP 55704
EXT 8866
P.T. BEE
8710 14TH ST ZIP 55713
EXT 18
END OF FILE
?? addlns 00200 10 ②
END OF FILE
?? p* ③
00200 Q.E. SMITH
00210 18 PARK PLACE APT 111 ZIP 55704
00220 EXT 8866
00230 P.T. BEE
00240 8710 14TH ST ZIP 55713
00250 EXT 18
END OF FILE
??

① The user lists a file in which all lines are left-justified.
② He inserts a set of line numbers beginning with 00200 and increasing by 10.
③ He then lists the file to check the effect of the ADDLNS command.

Figure 7-2. The ADDLNS Command

?? p* ①
00200 Q.E. SMITH
00210 18 PARK PLACE APT 111 ZIP 55704
00220 EXT 8866
00230 P.T. BEE
00240 8710 14TH ST ZIP 55713
00250 EXT 18
END OF FILE
?? delete ②
END OF FILE
?? p* ③
Q.E. SMITH
18 PARK PLACE APT 111 ZIP 55704
EXT 8866
P.T. BEE
8710 14TH ST ZIP 55713
EXT 18
END OF FILE
??

① The user lists a file with line numbers.
② He deletes every line number in the file.
③ He then lists the file to check the effect of the DELETeln command.

Figure 7-3. The DELETeln Command
REPLACING LINE NUMBERS

When the user wants to replace an existing set of line numbers with a different set of line numbers, he enters a REPLACELN command in the following format.

REPLACELN ln n
or
RLN ln n

ln  Line number to replace the first line number in the file. No line number can exceed 99999. Default is 1.

n  Number by which each line number is incremented. No line number can exceed 99999. Default is 1.

The REPLACELN command shown in figure 7-4 only changes a line if it is preceded by a line number.

FINDING LINES WITHOUT LINE NUMBERS

When a user wants to locate a specific number of lines without line numbers, he enters a FBADL command. This command is a mnemonic for find bad lines, bad lines being lines without line numbers. This command should be used in verify mode so that the lines located are printed. Its format is:

FBADL n
or
FBL n

n  Number of lines without line numbers that the user wants to locate. If n is 0, XEDIT locates every line in the file not preceded by a line number. Maximum value is 99999; default is 1. The file pointer is positioned at the last line located unless the end-of-information mark is read. If n is 0, XEDIT positions the pointer at the first line located.

Figure 7-5 demonstrates the FBADL command.

?? p 1
00100 Q.E. SMITH
00110 18 PARK PLACE APT 111 ZIP 55704
00120 EXT 8866
00130 P.T. BEE
00140 8710 14TH ST ZIP 55713
00150 EXT 18
END OF FILE
?? replace ln 00500 5
END OF FILE
?? p 2
00500 Q.E. SMITH
00505 18 PARK PLACE APT 111 ZIP 55704
00510 EXT 8866
00515 P.T. BEE
00520 8710 14TH ST ZIP 55713
00525 EXT 18
END OF FILE
??

1 The user lists a file with line numbers.

2 He issues a REPLACELN command specifying that the first line number should be 00500 and the line numbers should be incremented by 5.

3 He lists the file to check the results of the REPLACELN command.

Figure 7-4. The REPLACELN Command

?? fbad1 1
END OF FILE
?? top
?? fbad1
### NAMES/ADDRESSES ARE FICTIONAL ###
END OF FILE
??

1 The user issues a FBADL command to find all lines without line numbers in the file.

2 Because it is in verify mode, XEDIT prints the line found. It also prints the END OF FILE message to indicate that the entire file was searched.

Figure 7-5. The FBADL Command
DELETING LINES WITHOUT LINE NUMBERS

The user can delete lines which do not begin with a line number by entering a DBADL command. Starting at the current pointer position, XEDIT locates and deletes the number of lines without line numbers specified in the command. When XEDIT operates in verify mode, the editor lists each line that is deleted. The format is:

DBADL n ②

or

DBL n ②

This command is used in figure 7-6.

?? dbad1* ①
### NAMES/ADDRESSES ARE FICTITIOUS ###
END OF FILE③
??

① The user issues a DBADL command to delete all lines without line numbers.
② XEDIT prints the line it deleted and the END OF FILE message indicating that the entire file was searched.

Figure 7-6. The DBADL Command
The following XEDIT commands perform various editing functions.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLBLANKS</td>
<td>Deletes leading blanks from file lines</td>
</tr>
<tr>
<td>TOPNULL</td>
<td>Inserts a blank line at the beginning of the file</td>
</tr>
<tr>
<td>OCTCHANGE</td>
<td>Converts display code strings</td>
</tr>
<tr>
<td>RESTORE</td>
<td>Cancels the effect of the most recent editing commands</td>
</tr>
<tr>
<td>DEOF</td>
<td>Deletes end-of-file marks from the file</td>
</tr>
<tr>
<td>DEOR</td>
<td>Deletes end-of-record marks from the file</td>
</tr>
<tr>
<td>WEOF</td>
<td>Writes end-of-file marks onto the file</td>
</tr>
<tr>
<td>WEOR</td>
<td>Writes end-of-record marks onto the file</td>
</tr>
</tbody>
</table>

DELETING LEADING BLANKS

The DLBLANKS command deletes leading blanks from a specified number of lines in the file between the current pointer position and the end-of-information. Blank lines are deleted completely. The format is:

```
?? p* ⬅️
### NAMES/ADDRESSES ARE FICTITIOUS ###
Q.E. SMITH
18 PARK PLACE APT 111 ZIP 55704
EXT 8866
P.T. BEE
8710 14TH ST ZIP 55713
EXT 18
END OF FILE
?? dblank#* ⬅️
END OF FILE
?? p*  
### NAMES/ADDRESSES ARE FICTITIOUS ###
Q.E. SMITH
18 PARK PLACE APT 111 ZIP 55704
EXT 8866
P.T. BEE
8710 14TH ST ZIP 55713
EXT 18
END OF FILE
??
```

The user lists the file.

The user deletes all leading blanks from the file and lists it again for verification.

Figure 8-1. The DLBLANKS Command
CONVERTING OCTAL STRINGS
The OCTCHANGE command is used to replace one 
string of octal code (internal display code) with
another string of octal code. This enables the user
to enter certain terminal line controls (for example,
line feeds and carriage returns) into his file that
otherwise cannot be specified.

```
OCTCHANGE oct1 oct2 n
```

or

```
OC oct1 oct2 n
```

oct1 Octal display code that represents the
existing string to be replaced within
the user's file. Each display code
character must be represented by an
even number of octal digits. Refer
to appendix A for a list of valid display
codes and their graphic equivalences.
oct2 Octal display code that represents the
string that replaces oct1. Each
display code character must be represented
by an even number of octal
digits. Unpredictable results occur
when characters are changed to 00
codes. Refer to appendix A for a list
of valid display codes and their
graphic equivalences.
n Numer of lines to be processed by
the command. Maximum value is
99999; default is 1. If n is 0, every
line between the current pointer
position and the end-of-information is
processed. If n is 0, XEDIT does not
advance the pointer and processes
only the current line.

Figure 8-3 demonstrates the OCTCHANGE
command.

CANCELING EDITING OPERATIONS
The user issues a RESTORE command to cancel all editing
operations performed since the pointer was
positioned at the beginning of the file. This command
is entered when the user determines that one
or more recently issued commands were incorrectly entered or did not accomplish what the user originally intended. Changes made before the following events cannot be restored.

- An END OF FILE message is printed.
- A TOP command is processed
- An up-arrow (↑) command prefix is processed
- A NEXT -n command is processed
- A find line number (ln) command is processed causing a circular search

The format is:

```
RESTORE
```

or

```
REST
```

Figure 8-4 uses the RESTORE command.

DELETING AND INSERTING RECORDS
AND FILE Marks
The user can both insert and delete record and file
marks. DEOR and DEOF commands enable the
user to delete these marks, while WEOR and WEOF
commands allow the user to insert them.

Figure 8-2. The TONULL Command
batch
$RFL,0.
/attach,xedit/un=library
/xedit,pass,c ①
XEDIT 3.0
?? ②
INPUT
? today's password is:
? orange
?
EDIT
?? "print" ③
today's password is:
orange
END OF FILE
?? next
orange
?? xadd ④
?? "$w$w$w$w$w$m$m$m$m"
?? "print" ⑤
today's password is:
orange"$w$w$w$w$w$m$m$m$m"
END OF FILE
?? octchange 53 7655 ⑥
?? "print" ③
today's password is:
?? "m$m$m$m"
END OF FILE
??

① The user calls XEDIT to create a file called pass.
② He enters input mode and inserts two lines into the file.
③ He lists the file and then positions the pointer at the second line.
④ A string of characters is appended to the end of the line.
⑤ The edited file is printed.
⑥ The user next issues an OCTCHANGE command to replace each dollar sign (octal code 53) with a carriage return (octal code 7655).
⑦ XEDIT verifies the effect of the OCTCHANGE command.
⑧ The user lists the file again.

Figure 8-3. The OCTCHANGE Command

?? xlocate/smith/①
?? delete 3 ②
00160 Q.E. SMITH ③
00170 322 WILSHIRE ST ZIP CODE 5572:③ XEDIT verifies the lines deleted.
00180 EXT 67?
?? restore ④
?? 160 ⑤
00160 Q.E. SMITH
??

① The user sets the pointer to the first line containing the string SMITH.
② He then deletes three lines.
③ Seeing that he deleted information on the wrong SMITH, the user issues a RESTORE command.
④ To verify the restoration, the user locates line 160.

Figure 8-4. The RESTORE Command

DEOR COMMAND

The DEOR command deletes a specified number of end-of-record marks from the file. Its format is:

DEOR m ⑥

or

DR m ⑥

m Number of end-of-record marks to be deleted starting at the current pointer position. Maximum value is 999999; default value is 1. If m is 0, every end-of-record mark between the current pointer position and the end-of-information is deleted. If m is 0, the next end-of-record mark is deleted.

Figure 8-5 shows use of the DEOR command.

DEOF COMMAND

The DEOF command deletes a specified number of end-of-file marks from the file. Its format is:

DEOF m ⑥

or

DF m ⑥

m Number of end-of-file marks to be deleted starting at the current pointer position. Maximum value is 999999; default value is 1. If m is 0, every end-of-file mark between the current pointer position and the end-of-information is deleted. If m is 0, the next end-of-file mark is deleted.

The DEOF command is used in figure 8-6.
The user lists the file.

The user deletes the end-of-record marks.
XEDIT verifies that two end-of-record marks were found in the file and deleted.

Figure 8-5. The DEOR Command

The user lists the file.

He deletes all end-of-file marks in the file.

Figure 8-6. The DEOF Command

--- NOTE ---

The DEOF command deletes only end-of-file marks. It does not delete end-of-record marks (such as those created by the WEOF command).

--- WEOR COMMAND ---

To insert an end-of-record mark into the file in front of the current pointer position, the user enters a WEOR command in the following format.

WEOR

or

WR

Figure 8-7 demonstrates the WEOR command.

--- NOTE ---

To insert an end-of-file mark into the file in front of the current pointer position, the user enters a WEOF command in the following format.

WEOF

or

WF

Figure 8-8 shows the WEOF command.

The system inserts an end-of-record mark before the end-of-file mark if one does not already exist.
The file pointer is positioned at the line containing the word EXECUTIVE.

The user issues a WEOR command to insert an end-of-record mark before that line.

The user lists the file.

Figure 8-7. The WEOR Command

The file pointer is positioned at the line containing the word EXECUTIVE.

The user issues a WEOF command to insert an end-of-file mark before that line.

The user lists the file.

Figure 8-8. The WEOF Command
The user can manipulate files during editing with the following XEDIT commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY</td>
<td>Copies a specified set of lines from the edit file onto another file</td>
</tr>
<tr>
<td>COPYD</td>
<td>Copies a specified set of lines from the edit file onto another file and deletes those lines from the edit file</td>
</tr>
<tr>
<td>READ</td>
<td>Copies the contents of the specified local files onto the edit file</td>
</tr>
<tr>
<td>READP</td>
<td>Copies the contents of the specified permanent files onto the edit file</td>
</tr>
<tr>
<td>FILE</td>
<td>Saves an intermediate version of the edit file</td>
</tr>
</tbody>
</table>

The following are the formats of the COPY command and their functions.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPY name n</td>
<td>Copies n lines from the edit file onto file name</td>
</tr>
<tr>
<td>COPY name / string/n</td>
<td>Copies onto file name all lines from the current pointer position through the nth line containing the specified string or until the end-of-information mark is read.</td>
</tr>
<tr>
<td>COPY name / string1...string2/n</td>
<td>Copies onto file name all lines from the current pointer position through the nth line containing string1 followed by string2 or until the end-of-information mark is read.</td>
</tr>
<tr>
<td>COPY name / string1---string2/n</td>
<td>Copies onto file name all lines from the current pointer position through the nth line containing string1 not followed by string2 or until the end-of-information mark is read.</td>
</tr>
<tr>
<td>COPY name / ---string2/n</td>
<td>Copies onto file name all lines from the current pointer position through the nth line not containing string2 or until the end-of-information mark is read.</td>
</tr>
</tbody>
</table>

**COPYING LINES ONTO ANOTHER FILE**

When the user wants to copy lines from the edit file onto another local file, he can enter either a COPY or a COPYD command. Both commands perform the same function, except that the copied lines are deleted from the edit file when the COPYD command is used. The following rules apply to both COPY and COPYD commands.

- Copying begins at the current pointer position and includes all lines in the file from the current pointer position through the line containing the nth occurrence of the specified string. If XEDIT reads the end-of-information mark while processing these commands, all lines in the file between the initial pointer position and the end-of-information are copied to the local file.
- After the copying is completed, the pointer is positioned at the last line copied or at the beginning of the file if the end-of-information is read.
- When verify mode is in effect, XEDIT prints only those lines containing the specified string.
- Before copying begins, XEDIT rewinds the local file. Later copies onto the same file are added to the end of that file.
- To specify what columns the specified string should occur in, the windowing feature can be used with the COPY and COPYD commands. (Refer to the WMARGIN command.)
MERGING FILES INTO THE EDITED FILE

To merge one or more files into the edited file, the user enters a READ or READP command. While both commands function the same, READ commands copy local files, and READP commands copy permanent files. The following general rules govern use of both READ and READP commands.

- The specified local or permanent file is copied onto the edited file after the current pointer position, starting with the first record, and continuing until two consecutive end-of-record marks, an end-of-file mark, or an end-of-information mark is detected. Embedded end-of-record marks are maintained.
- Once each copying operation is completed, the pointer is positioned at the last line that was copied onto the edited file.
- XEDIT rewinds the specified files before and after the copying operation.
- If a specified local or permanent file cannot be read (for example, it does not exist or the user entered the wrong file name), XEDIT prints an error message and ends processing of the READ or READP command without copying any other files.

The format of the READP command is:

\[
\text{READP } \text{name}_1 \text{name}_2 \ldots \text{name}_n
\]

\[
\text{name}_1 \ldots \text{name}_n
\]

Names of the files to be copied onto the edited file in the sequential order in which they are to be copied. A space must separate each name from the preceding name.

This format is used in figure 9-2.

\[
\text{COPYD} \text{name} \text{n}
\]

\[
\text{COPYD} \text{name} /\text{string}/\text{n}
\]

\[
\text{COPYD} \text{name} /\text{string}1\ldots\text{string2}/\text{n}
\]

\[
\text{COPYD} \text{name} /\text{string}1\ldots\text{string2}/\text{n}
\]

\[
\text{COPYD} \text{name} /\text{---string2}/\text{n}
\]

When the user enters the COPYD command with name specified as NULL, XEDIT deletes the lines without saving them on a local file.

?? locate/*division(1)
*DIVISION 1*
?? copy div1 /*end/ (2)
*END OF DIVISION*
?? end,r1 (3)
DIV REPLACED
DIV IS A LOCAL FILE
/lnh,f=div1 (4)
*DIVISION 1*
00170 A.P. NEWTON
00140 166 HASKELL CIRCLE ZIP 55713
00150 EXT 227
00157 A.P. MACDONALD
00158 1313 LEMON TREE AVE ZIP 55722
00159 EXT 5339
00170 C.E. SMITH
00170 18 PARK PLACE APT. 111 ZIP 55704
00180 EXT 8866
*END OF DIVISION*
/

?? The user locates the string #DIVISION.
?? The user copies all lines from the current pointer position until the next line containing the string END onto the local file DIV1.
?? The user terminates XEDIT execution.
?? The user lists the local file DIV1.

Figure 9-1. The COPY Command
The READ command uses the same format and parameters as the READP command. It copies specified local files onto the edit file. To move a group of lines from one part of the file to another, the user can copy the lines onto a local file using the COPY or COPYD commands and then insert the lines at the desired location by positioning the pointer and entering a READ command to copy the local file onto the edit file. The format of the READ command is:

```
READ name1 name2...namen
```

SAVING AN INTERMEDIATE XEDIT FILE

The FILE command enables the user to save an intermediate version of the edit file. When the FILE command is entered, the following procedure occurs.

1. XEDIT saves a copy of the edit file as specified by the mode parameter.
2. XEDIT prints one or more messages verifying its action.
3. XEDIT requests the next editing command.

The format for the FILE command is:

```
FILE, name, mode
```

-or-

```
F, name, mode
```

**name** Name to be given to the copy of the edit file; it can be from one to seven alphanumeric characters in length. If **name** is omitted, the file is given the name given the edit file when XEDIT was called. The command before and after **name** must be included although **name** is omitted (for example, FILE, , SAVE).

**mode** Type of NOS file to be created. The possible modes are:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S or SAVE</td>
<td>Saves the edit file as a new indirect permanent file.</td>
</tr>
<tr>
<td>R or REPLACE</td>
<td>Replaces the existing indirect access file <strong>name</strong> with the edit file (same as SAVE if no file with that name exists).</td>
</tr>
<tr>
<td>L or LOCAL</td>
<td>Makes the edit file a local file; LOCAL mode is illegal for direct access files.</td>
</tr>
<tr>
<td>C or COPY</td>
<td>Copies the edit file to file <strong>name</strong>. COPY mode is default for direct access files attached in write mode.</td>
</tr>
<tr>
<td>RL</td>
<td>Combines the functions of REPLACE and LOCAL modes.</td>
</tr>
<tr>
<td>SL</td>
<td>Combines the functions of SAVE and LOCAL modes.</td>
</tr>
</tbody>
</table>

If the mode parameter is omitted, LOCAL mode is assumed for working files and COPY mode for direct access files.
Figure 9-3 demonstrates the FILE command.

1. The user positions the pointer at line 120.
2. He changes the extension number.
3. He replaces the indirect access permanent file called ADDRESS with a copy of the edited file.

Figure 9-3. The FILE Command
The commands described in this section enable the user to repeat a sequence of editing operations at specified points in the file.

SUBMITTING MULTIPLE ENTRIES IN A SINGLE LINE
The user can issue more than one command in a single line by using either of the following methods.

- By specifying with a DELIMIT command the delimiter character to be used to separate a sequence of commands and editing data entered on a single line.
- By using the Z or Y command formats to enter several commands and their editing data on the same line. The command sequence can be repeated using the -n command.

USING DELIMIT COMMAND
The following are basic rules governing use of the DELIMIT command and subsequent delimited sequences.

- Any valid command can appear within the line of delimited commands and editing data.
- The delimiter specified in the DELIMIT command is used to separate commands and editing data.
- The delimit character specified in the DELIMIT command remains in effect until another DELIMIT command is issued.
- The delimit character used in a Y or Z command must not be the same character as the delimit character specified by the last DELIMIT command.
- Both commands and editing data (for example, input related to an ADD command) can appear within a delimited command sequence. To include editing data for a command on the same line as the command, the user should prefix the command with a +. Otherwise, editing data is requested as needed via a single question mark prompt.

DELIMIT commands should be in the following format.

```
DELIMIT char @
```

or

```
DEL char @
```

Character to be used as the delimiter; it must not be alphabetic or a space. To specify a comma as a delimiter, the command DELIMIT,, is entered. If the character is omitted from the command, XEDIT clears the effect of the previous DELIMIT command.

Assuming that the last DELIMIT command specified a semicolon as the delimiter, a delimited command sequence should use the following format.

```
cmd_1;cmd_2;...;cmd_n
```

- **cmd**: A legal command (or editing data required by a + prefixed command). Each command is processed in order from left to right. The DELIMIT character must not be included in cmd.

If any command is in error, all remaining commands in the list are ignored and the editor requests the next command.

XEDIT expects editing data as the next delimited item following a + prefixed command. Otherwise, editing data is requested as needed via a single question mark prompt.

Figure 10-1 demonstrates the DELIMIT command.

USING Z AND Y COMMANDS
The Z and Y commands can be used to enter a delimited sequence of commands and editing data on a single line. The only difference in the Z and the Y command operation is that for the Z command, XEDIT prints the component commands of the sequence as they are processed, and for the Y command, it does not print them.

The following basic rules apply to both Z and Y commands.

- Any valid XEDIT command except Z and Y commands can be included in the list of component commands. The Y or Z command delimiter should never be the same as the delimiter specified by the last DELIMIT command. If a DELIMIT command is included in a Z or Y command, the DELIMIT character takes effect only after XEDIT has processed all component commands in the Y or Z command sequence.

- If any command in the sequence is in error, all remaining commands in the sequence are ignored, and XEDIT requests the next command.
The user lists two lines.

He sets the delimiter character as &.

He enters a delimited command sequence containing:

- An abbreviated + and x prefixed ADD command
- The string to be appended by the ADD command
- An abbreviated + prefixed INSERT command
- The line to be inserted
- An abbreviated PRINT command to list three lines beginning at line 220

Figure 10-1. The DELIMIT Command

- Both commands and editing data can appear as part of the command sequence. XEDIT expects editing data as the next delimited item following a + prefixed command.

- The command delimiter can be any alphabetic, numeric or special character not appearing within a component command or editing data item and which is not the delimiter character specified in the last DELIMIT command.

It is suggested that dollar signs or semicolons be used as command delimiters and slashes as string delimiters because those characters are never found in XEDIT commands.

Assuming that $ has been chosen as the command delimiter, Y and Z commands are in the following format.

\[
Z\text{cmd}_1\text{cmd}_2\ldots\text{cmd}_n \quad \text{or} \\
Y\text{cmd}_1\text{cmd}_2\ldots\text{cmd}_n \\
\$ \quad \text{Command delimiter.}
\]

\[
\text{cmd} \quad \text{A legal command (or editing data required by a + prefixed command). Each command is processed in order from left-to-right. The command delimiter must not be included in cmd.}
\]

A Z command is used in figure 10-2.

REPEATING COMMANDS

The user can repeat the processing of the preceding command at a different pointer position. To do so, he enters a period followed by the number of lines the pointer should be advanced before the preceding command is processed again. To repeat the processing of a Z or Y command, the user enters a minus sign followed by the number of lines the pointer should be advanced. The formats are:

\[
\text{.n} \quad \text{or} \\
-n \quad \text{Process the preceding command (other than Z or Y commands)} \\
- \quad \text{Process the preceding Z or Y command} \\
n \quad \text{Number of lines that the pointer should be advanced before processing the preceding command. Default value is 1. If n is 0, the last command is processed without advancing the pointer.}
\]

Figure 10-3 shows how to repeat a command.

Figure 10-4 uses a delimited command sequence and the repeat command.
?? z$locate/smith/++add$ jr.$delete/bee/
?? LOCATE/SMITH
?? $ADD
00160 O.E. SMITH
?? DELETE/BEER
00190 P.T. BEE
??

1 The user enters a Z command containing three component commands: LOCATE, ADD, and DELETE.
2 XEDIT prints each processed command and verification of its action.

Figure 10-2. The Z Command

?? print 1
00130 A.B. MACDONALD
?? .3 2
00157 A.P. MACDONALD
?? .3 3
00180 EXT 67
??

1 The user prints a line.
2 He enters a .3 to advance the pointer three lines and print that line.
3 He once again advances the pointer three lines and prints the line.

Figure 10-3. Repeating a Command

?? del $ 1
?? l/address/++insert$zip code:
NAME:
ADDRESS:
ZIP CODE:
END OF FILE
?? $ADD/ &.8 $, 4
? q.e. smith jr. 5
NAME: Q.F. SMITH JR.
? po 55
ADDRESS: POB 55
? 55703
ZIP CODE: 55703
END OF FILE
??

1 The user sets the delimiter as $.
2 He enters a delimited command sequence containing two commands, LOCATE and INSERT.
3 He lists the file.

4 He then enters a delimited command sequence consisting of the ADD command processed four times on consecutive lines.
5 For each prompt, he enters an appropriate string and XEDIT prints the appended line.

Figure 10-4. Repeating a Command Sequence
The commands in this section provide valuable formatting tools for the user. Tab control commands ensure proper spacing of paragraphs, tables, and programs. Margin control commands enable the user to set a right margin, locate lines that extend beyond that margin, and truncate them.

**TAB CONTROL**

The user can control tab settings during editing by employing the following XEDIT commands.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFTAB</td>
<td>Defines a tab character for use in entering editing data with the INSERT, INSERTB, REPLACE, INPUT, and commands.</td>
</tr>
<tr>
<td>TABS</td>
<td>Defines tab stop positions.</td>
</tr>
<tr>
<td>LISTAB</td>
<td>Lists the current tab character and tab stop positions.</td>
</tr>
</tbody>
</table>

**DEFINING TAB CHARACTER**

The user can define a tab character by issuing a DEFTAB command in the following format.

```
DEFTAB char
```

or

```
DT char
```

- **char** The tab character to be used in entering editing data with the INSERT, INSERTB, REPLACE, INPUT, and commands. If the user issues a DEFTAB command without specifying a tab character, XEDIT clears the effect of the previous DEFTAB command. There is no default tab character. To specify a comma as the tab character, DEFTAB,, is entered.

Figure 11-1 demonstrates tab character definition and use.

**DEFINING TAB POSITIONS**

The user can define up to eight tab column positions by issuing a TABS command in the following format.

```
TABS t1 t2 t3... t8
```

or

```
TAB t1 t2 t3... t8
```

- **t** Tab column positions with increasing values from left to right; default values are 11, 18, and 30. When a TABS command is entered without any t parameters, XEDIT clears the effect of the previous TABS command. Any tab characters included in the input line beyond the last tab stop are copied to the file.

Figure 11-2 shows the setting of tab stops.

**LISTING TAB SETTINGS**

The user can list the current tab character and tab stop column positions by issuing a LISTAB command in the following format.

```
LISTAB
```

or

```
LT
```

Figure 11-3 uses the LISTAB command.

??? deftab $ ① ① The user issues a DEFTAB command to establish the dollar sign ($) as the tab character.
??? insert ② A line is inserted into the file using the tab character to space the column headings according to the default tab column positions, 11, 18, and 30.
? $mtd$ytd$total ② The user lists the line for verification.
? print ②

Figure 11-1. Defining a Tab Character
MARGIN AND TRUNCATION CONTROL

The following commands can be used to control the length of file lines.

- **RMARGIN** Sets the right margin position
- **FINDLL** Finds lines longer than the current right margin setting
- **TRUNCATE** Truncates lines longer than the current right margin setting

The user can employ these commands to ensure that no file lines are longer than 150 characters, the maximum length allowed by most NOS line processing routines.

**SETTING THE MARGIN**

The RMARGIN command sets the column position of the right margin of a file. Lines extending beyond that column can then be located by the FINDLL command and truncated by the TRUNCATE command. After the RMARGIN command is executed, the pointer retains its original position.

```
RMARGIN m
```

or

```
RM m
```

The column position of the right margin of the file. It cannot be less than 10.

**FINDING LONG LINES**

To locate a specific number of long lines, the user issues a FINDLL command. Long lines are lines longer than the current right margin setting (refer to the RMARGIN command). This command should be entered while editing in verify mode so that the located long lines are listed. The format for the FINDLL command is:

```
FINDLL n
```

or

```
FLL n
```

where:

- **n** Number of lines longer than the current RMARGIN setting to be located. If n is 0, every line between the current pointer position and the end of information is found. The maximum value is 39999; default value is 1. If the file pointer is located at the last long line found unless the end-of-information is read so that the pointer is positioned at the beginning of the file. If n is 0, $XEDIT$ finds the next long line.
Figure 11-4 uses both the RMARGIN and the FINDLL commands.

TRUNCATING LONG LINES

The TRUNCATE command truncates a specified number of lines at the right margin set by the RMARGIN command. The format is:

TRUNCATE n ②

or

TRUNC n ②

Figure 11-5 shows use of this command.

?? print ①
*EXECUTIVE MANAGEMENT*
EX100 S.W. KRAMER  HOR24
EX120 J.J. JOHNSON  HOR44
EX125 R.E. MILLER  HOR44
EX130 F.R. SCOTT  END OF FILE
?? rmargin 25 ②
?? findll*
EX100 S.W. KRAMER  HOR24
EX125 R.E. MILLER  HOR44
END OF FILE
??

① The user lists the file.

② He sets the right margin at column 25 and then lists all lines longer than 25 characters.

Figure 11-4. The RMARGIN and FINDLL Commands

?? print ①
*EXECUTIVE MANAGEMENT*
EX100 S.W. KRAMER  HOR24
EX120 J.J. JOHNSON  HOR44
EX125 R.E. MILLER  HOR44
EX130 F.R. SCOTT  END OF FILE
?? rmargin 25 ②
?? truncate* ③
*EXECUTIVE MANAGEMENT*
EX100 S.W. KRAMER
EX120 J.J. JOHNSON
EX125 R.E. MILLER
EX130 F.R. SCOTT
END OF FILE
??

① The user lists the file.

② He sets the right margin at column 25.

③ He then truncates all lines in the file longer than 25 characters.

Figure 11-5. The TRUNCATE Command
The user issues one of the following commands to end his editing session.

- END or QUIT Terminates editing and saves the edit file
- STOP Terminates editing and does not save the edit file

TERMINATING XEDIT WITHOUT SAVING EDITING CHANGES

The user issues a STOP command to terminate XEDIT without saving the edited version of the file. The file is left exactly as it appeared before XEDIT was called into execution, assuming that no FILE commands were processed during the session. If the STOP command is issued in batch mode, the first statement after the EXIT control statement is executed; otherwise, the job is terminated.

The STOP command has the following format.

STOP ☞

Figure 12-1 uses the STOP command.

TERMINATING XEDIT WITH EDITING CHANGES SAVED

When the user wants to terminate XEDIT execution and keep the edited file, he issues a QUIT or END command. Both END and QUIT perform exactly the same function and use identical parameters. The following general rules apply to these commands.

- When the command is entered, the edit file is written onto the specified file name in accordance with the specified mode parameter. XEDIT then terminates its execution.
- The edit file becomes the primary file only if the filename specified matches the name of the primary file.
- The user should enter SAVE and REPLACE mode parameters with care. SAVE and REPLACE parameters do not write the edit file modifications onto the existing local version of the file. Only permanent files are affected by SAVE and REPLACE parameters.

The END and QUIT commands should be entered in the following format.

END, filename, mode ☞

or

E, filename, mode ☞

or

QUIT, filename, mode ☞

or

Q, filename, mode ☞

filename Name (one to seven alphanumeric characters in length) to be given the edit file upon leaving XEDIT. If no file name is entered, XEDIT uses the name specified for the edit file when XEDIT was called. When filename is omitted, the commas before and after filename must be entered.

mode Type of file the edit file is to become. Valid entries for this parameter are:

1. The user deletes three lines.
2. He then attempts to restore the deleted lines but fails to do so because the end-of-information mark was read during the deletion.
3. The user then terminates XEDIT execution to return the file to its original state.
4. When he calls XEDIT again, line 190 is included in the file.

Figure 12-1. The STOP Command
<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Saves the edit file as a new indirect access permanent file</td>
</tr>
<tr>
<td>SAVE</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Replaces the existing indirect access permanent file name with the edit file</td>
</tr>
<tr>
<td>REPLACE</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Rewrites the edit file onto the direct access file of the same name. This mode is default for direct access files attached in write mode</td>
</tr>
<tr>
<td>COPY</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Makes the edit file a local file</td>
</tr>
<tr>
<td>LOCAL</td>
<td></td>
</tr>
<tr>
<td>RL</td>
<td>Combines the functions of REPLACE and LOCAL modes</td>
</tr>
<tr>
<td>SL</td>
<td>Combines the functions of SAVE and LOCAL modes</td>
</tr>
</tbody>
</table>

If the mode parameter is omitted, LOCAL mode is assumed for working files, and COPY mode is assumed for direct access files.

Figures 12-2 and 12-3 show use of the END command with REPLACE and RL specified, respectively.

---

Figure 12-2. The END Command Using REPLACE Mode

---

(a) The user positions the pointer at line 120. He changes the string EXT to EXTENSION in all lines from line 120 to the end-of-information.

(b) He then terminates XEDIT and replaces the indirect access file called ADDRESS with the edit file.

(c) He then lists the local version of the edit file which does not contain the edited changes.
The user positions the pointer at line 120.

He changes the string EXT to EXTENSION in all lines from 120 to the end-of-information.

He then terminates XEDIT and replaces both the indirect access permanent file called ADDRESS and the local file with the edit file.

He lists the local file containing the edited changes.

Figure 12-3. The END Command Using RL Mode
The character sets for ALGOL and COBOL are listed in their respective reference manuals.

Table A-1. NOS 64-Character Set

<table>
<thead>
<tr>
<th>CHAR</th>
<th>CODE (7-BIT OCTAL)</th>
<th>CODE (7-BIT HEXADECIMAL)</th>
<th>INTERNAL DISPLAY CODE (6/12-BIT OCTAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td>072</td>
<td>3A</td>
<td>001</td>
</tr>
<tr>
<td>A</td>
<td>101</td>
<td>41</td>
<td>01</td>
</tr>
<tr>
<td>B</td>
<td>102</td>
<td>42</td>
<td>02</td>
</tr>
<tr>
<td>C</td>
<td>103</td>
<td>43</td>
<td>03</td>
</tr>
<tr>
<td>D</td>
<td>104</td>
<td>44</td>
<td>04</td>
</tr>
<tr>
<td>E</td>
<td>105</td>
<td>45</td>
<td>05</td>
</tr>
<tr>
<td>F</td>
<td>106</td>
<td>46</td>
<td>06</td>
</tr>
<tr>
<td>G</td>
<td>107</td>
<td>47</td>
<td>07</td>
</tr>
<tr>
<td>H</td>
<td>110</td>
<td>48</td>
<td>10</td>
</tr>
<tr>
<td>I</td>
<td>111</td>
<td>49</td>
<td>11</td>
</tr>
<tr>
<td>J</td>
<td>112</td>
<td>4A</td>
<td>12</td>
</tr>
<tr>
<td>K</td>
<td>113</td>
<td>4B</td>
<td>13</td>
</tr>
<tr>
<td>L</td>
<td>114</td>
<td>4C</td>
<td>14</td>
</tr>
<tr>
<td>M</td>
<td>115</td>
<td>4D</td>
<td>15</td>
</tr>
<tr>
<td>N</td>
<td>116</td>
<td>4E</td>
<td>16</td>
</tr>
<tr>
<td>O</td>
<td>117</td>
<td>4F</td>
<td>17</td>
</tr>
<tr>
<td>P</td>
<td>120</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Q</td>
<td>121</td>
<td>51</td>
<td>21</td>
</tr>
<tr>
<td>R</td>
<td>122</td>
<td>52</td>
<td>22</td>
</tr>
<tr>
<td>S</td>
<td>123</td>
<td>53</td>
<td>23</td>
</tr>
<tr>
<td>T</td>
<td>124</td>
<td>54</td>
<td>24</td>
</tr>
<tr>
<td>U</td>
<td>125</td>
<td>55</td>
<td>25</td>
</tr>
<tr>
<td>V</td>
<td>126</td>
<td>56</td>
<td>26</td>
</tr>
<tr>
<td>W</td>
<td>127</td>
<td>57</td>
<td>27</td>
</tr>
<tr>
<td>X</td>
<td>130</td>
<td>58</td>
<td>30</td>
</tr>
<tr>
<td>Y</td>
<td>131</td>
<td>59</td>
<td>31</td>
</tr>
<tr>
<td>Z</td>
<td>132</td>
<td>5A</td>
<td>32</td>
</tr>
<tr>
<td>a</td>
<td>060</td>
<td>30</td>
<td>33</td>
</tr>
<tr>
<td>b</td>
<td>061</td>
<td>31</td>
<td>34</td>
</tr>
<tr>
<td>c</td>
<td>062</td>
<td>32</td>
<td>35</td>
</tr>
<tr>
<td>d</td>
<td>063</td>
<td>33</td>
<td>36</td>
</tr>
<tr>
<td>e</td>
<td>064</td>
<td>34</td>
<td>37</td>
</tr>
<tr>
<td>f</td>
<td>065</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td>g</td>
<td>066</td>
<td>36</td>
<td>41</td>
</tr>
<tr>
<td>h</td>
<td>067</td>
<td>37</td>
<td>42</td>
</tr>
<tr>
<td>i</td>
<td>070</td>
<td>38</td>
<td>43</td>
</tr>
<tr>
<td>j</td>
<td>071</td>
<td>39</td>
<td>44</td>
</tr>
<tr>
<td>k</td>
<td>053</td>
<td>28</td>
<td>45</td>
</tr>
<tr>
<td>l</td>
<td>055</td>
<td>20</td>
<td>46</td>
</tr>
<tr>
<td>?</td>
<td>052</td>
<td>2A</td>
<td>47</td>
</tr>
<tr>
<td>/</td>
<td>037</td>
<td>2F</td>
<td>50</td>
</tr>
<tr>
<td>(</td>
<td>050</td>
<td>28</td>
<td>51</td>
</tr>
<tr>
<td>)</td>
<td>051</td>
<td>29</td>
<td>52</td>
</tr>
<tr>
<td>$</td>
<td>044</td>
<td>24</td>
<td>53</td>
</tr>
<tr>
<td>=</td>
<td>075</td>
<td>3D</td>
<td>54</td>
</tr>
</tbody>
</table>

1 In the 63-character set, this display code represents a null character. Also, use of the colon in program and data files may cause problems. This is particularly true when it is used in print and format statements.

<table>
<thead>
<tr>
<th>CHAR</th>
<th>CODE (7-BIT OCTAL)</th>
<th>CODE (7-BIT HEXADECIMAL)</th>
<th>INTERNAL DISPLAY CODE (6/12-BIT OCTAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>空间</td>
<td>040</td>
<td>20</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>054</td>
<td>2C</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>056</td>
<td>2E</td>
<td>57</td>
</tr>
<tr>
<td>*</td>
<td>043</td>
<td>23</td>
<td>60</td>
</tr>
<tr>
<td>[</td>
<td>133</td>
<td>5B</td>
<td>61</td>
</tr>
<tr>
<td>]</td>
<td>135</td>
<td>5D</td>
<td>62</td>
</tr>
<tr>
<td>%</td>
<td>045</td>
<td>25</td>
<td>63</td>
</tr>
<tr>
<td>=</td>
<td>042</td>
<td>22</td>
<td>64</td>
</tr>
<tr>
<td>-</td>
<td>13711</td>
<td>5F</td>
<td>65</td>
</tr>
<tr>
<td>;</td>
<td>041</td>
<td>21</td>
<td>66</td>
</tr>
<tr>
<td>a</td>
<td>046</td>
<td>26</td>
<td>67</td>
</tr>
<tr>
<td>'</td>
<td>047</td>
<td>27</td>
<td>70</td>
</tr>
<tr>
<td>?</td>
<td>077</td>
<td>3F</td>
<td>71</td>
</tr>
<tr>
<td>&lt;</td>
<td>074</td>
<td>3C</td>
<td>72</td>
</tr>
<tr>
<td>&gt;</td>
<td>076</td>
<td>3E</td>
<td>73</td>
</tr>
<tr>
<td>@</td>
<td>100</td>
<td>40</td>
<td>74</td>
</tr>
<tr>
<td>\</td>
<td>134</td>
<td>5C</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>136</td>
<td>5E</td>
</tr>
<tr>
<td>i</td>
<td>073</td>
<td>38</td>
<td>77</td>
</tr>
<tr>
<td>NULL</td>
<td></td>
<td></td>
<td>7600</td>
</tr>
<tr>
<td>a</td>
<td>141</td>
<td>61</td>
<td>7601</td>
</tr>
<tr>
<td>b</td>
<td>142</td>
<td>62</td>
<td>7602</td>
</tr>
<tr>
<td>c</td>
<td>143</td>
<td>63</td>
<td>7603</td>
</tr>
<tr>
<td>d</td>
<td>144</td>
<td>64</td>
<td>7604</td>
</tr>
<tr>
<td>e</td>
<td>145</td>
<td>65</td>
<td>7605</td>
</tr>
<tr>
<td>f</td>
<td>146</td>
<td>66</td>
<td>7606</td>
</tr>
<tr>
<td>g</td>
<td>147</td>
<td>67</td>
<td>7607</td>
</tr>
<tr>
<td>h</td>
<td>150</td>
<td>68</td>
<td>7610</td>
</tr>
<tr>
<td>i</td>
<td>151</td>
<td>69</td>
<td>7611</td>
</tr>
<tr>
<td>j</td>
<td>152</td>
<td>6A</td>
<td>7612</td>
</tr>
<tr>
<td>k</td>
<td>153</td>
<td>6B</td>
<td>7613</td>
</tr>
<tr>
<td>l</td>
<td>154</td>
<td>6C</td>
<td>7614</td>
</tr>
<tr>
<td>m</td>
<td>155</td>
<td>6D</td>
<td>7615</td>
</tr>
<tr>
<td>n</td>
<td>156</td>
<td>6E</td>
<td>7616</td>
</tr>
<tr>
<td>o</td>
<td>157</td>
<td>6F</td>
<td>7617</td>
</tr>
<tr>
<td>p</td>
<td>160</td>
<td>70</td>
<td>7620</td>
</tr>
<tr>
<td>q</td>
<td>161</td>
<td>71</td>
<td>7621</td>
</tr>
<tr>
<td>r</td>
<td>162</td>
<td>72</td>
<td>7622</td>
</tr>
<tr>
<td>s</td>
<td>163</td>
<td>73</td>
<td>7623</td>
</tr>
<tr>
<td>t</td>
<td>164</td>
<td>74</td>
<td>7624</td>
</tr>
<tr>
<td>u</td>
<td>165</td>
<td>75</td>
<td>7625</td>
</tr>
<tr>
<td>v</td>
<td>166</td>
<td>76</td>
<td>7626</td>
</tr>
<tr>
<td>w</td>
<td>167</td>
<td>77</td>
<td>7627</td>
</tr>
<tr>
<td>x</td>
<td>170</td>
<td>78</td>
<td>7630</td>
</tr>
<tr>
<td>y</td>
<td>171</td>
<td>79</td>
<td>7631</td>
</tr>
</tbody>
</table>

1 In the 63-character set, this display code represents a colon (:) 7-bit ASCII code 072, 7-bit hexadecimal code 3A.

II On TTY models having no underline, the backarrow (\) take its place.
<table>
<thead>
<tr>
<th>CHAR</th>
<th>CODE (7-BIT OCTAL)</th>
<th>CODE (7-BIT HEXADECIMAL)</th>
<th>INTERNAL DISPLAY CODE (6/12-BIT OCTAL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>172</td>
<td>7A</td>
<td>7632</td>
</tr>
<tr>
<td>[</td>
<td>173</td>
<td>7B</td>
<td>7633</td>
</tr>
<tr>
<td>]</td>
<td>174</td>
<td>7C</td>
<td>7634</td>
</tr>
<tr>
<td></td>
<td>175</td>
<td>7D</td>
<td>7635</td>
</tr>
<tr>
<td>~</td>
<td>176</td>
<td>7E</td>
<td>7636</td>
</tr>
<tr>
<td>DEL</td>
<td>177</td>
<td>7F</td>
<td>7637</td>
</tr>
<tr>
<td>NUL</td>
<td>000</td>
<td>00</td>
<td>7640</td>
</tr>
<tr>
<td>SOH</td>
<td>001</td>
<td>01</td>
<td>7641</td>
</tr>
<tr>
<td>STX</td>
<td>002</td>
<td>02</td>
<td>7642</td>
</tr>
<tr>
<td>ETX</td>
<td>003</td>
<td>03</td>
<td>7643</td>
</tr>
<tr>
<td>EOT</td>
<td>004</td>
<td>04</td>
<td>7644</td>
</tr>
<tr>
<td>ENQ</td>
<td>005</td>
<td>05</td>
<td>7645</td>
</tr>
<tr>
<td>ACK</td>
<td>006</td>
<td>06</td>
<td>7646</td>
</tr>
<tr>
<td>BEL</td>
<td>007</td>
<td>07</td>
<td>7647</td>
</tr>
<tr>
<td>BS</td>
<td>010</td>
<td>08</td>
<td>7650</td>
</tr>
<tr>
<td>HT</td>
<td>011</td>
<td>09</td>
<td>7651</td>
</tr>
<tr>
<td>LF</td>
<td>012</td>
<td>0A</td>
<td>7652</td>
</tr>
<tr>
<td>VT</td>
<td>013</td>
<td>0B</td>
<td>7653</td>
</tr>
<tr>
<td>FF</td>
<td>014</td>
<td>0C</td>
<td>7654</td>
</tr>
<tr>
<td>CR</td>
<td>015</td>
<td>0D</td>
<td>7655</td>
</tr>
<tr>
<td>SO</td>
<td>016</td>
<td>0E</td>
<td>7656</td>
</tr>
<tr>
<td>SI</td>
<td>017</td>
<td>0F</td>
<td>7657</td>
</tr>
<tr>
<td>DLE</td>
<td>020</td>
<td>10</td>
<td>7660</td>
</tr>
<tr>
<td>DC1</td>
<td>021</td>
<td>11</td>
<td>7661</td>
</tr>
<tr>
<td>DC2</td>
<td>022</td>
<td>12</td>
<td>7662</td>
</tr>
<tr>
<td>DC3</td>
<td>023</td>
<td>13</td>
<td>7663</td>
</tr>
<tr>
<td>DC4</td>
<td>024</td>
<td>14</td>
<td>7664</td>
</tr>
<tr>
<td>NAK</td>
<td>025</td>
<td>15</td>
<td>7665</td>
</tr>
<tr>
<td>SYN</td>
<td>026</td>
<td>16</td>
<td>7666</td>
</tr>
<tr>
<td>ETB</td>
<td>027</td>
<td>17</td>
<td>7667</td>
</tr>
<tr>
<td>CAN</td>
<td>030</td>
<td>18</td>
<td>7670</td>
</tr>
<tr>
<td>EM</td>
<td>031</td>
<td>19</td>
<td>7671</td>
</tr>
<tr>
<td>SUB</td>
<td>032</td>
<td>1A</td>
<td>7672</td>
</tr>
<tr>
<td>ESC</td>
<td>033</td>
<td>1B</td>
<td>7673</td>
</tr>
<tr>
<td>FS</td>
<td>034</td>
<td>1C</td>
<td>7674</td>
</tr>
<tr>
<td>GS</td>
<td>035</td>
<td>1D</td>
<td>7675</td>
</tr>
<tr>
<td>RS</td>
<td>036</td>
<td>1E</td>
<td>7676</td>
</tr>
<tr>
<td>US</td>
<td>037</td>
<td>1F</td>
<td>7677</td>
</tr>
<tr>
<td>NULL</td>
<td>—</td>
<td>—</td>
<td>7400</td>
</tr>
<tr>
<td>®</td>
<td>100</td>
<td>40</td>
<td>7401</td>
</tr>
<tr>
<td>®</td>
<td>136</td>
<td>5E</td>
<td>7402</td>
</tr>
<tr>
<td>®</td>
<td>072</td>
<td>3A</td>
<td>7403</td>
</tr>
<tr>
<td>NULL</td>
<td>—</td>
<td>—</td>
<td>7404</td>
</tr>
<tr>
<td>NULL</td>
<td>—</td>
<td>—</td>
<td>7405</td>
</tr>
<tr>
<td>NULL</td>
<td>—</td>
<td>—</td>
<td>7406</td>
</tr>
<tr>
<td>®</td>
<td>140</td>
<td>60</td>
<td>7407</td>
</tr>
</tbody>
</table>
This appendix contains an alphabetical listing of the messages that may appear in a user's output file. Lowercase characters are used to identify variable names or fields. All messages beginning with variable names or characters follow those beginning with A through Z and 0 through 9. These messages are then alphabetized according to the first nonvariable word or character. Messages beginning with any special characters (such as hyphens or asterisks) are alphabetized as if the special character were not present. For example, the message

filename SAVED

is listed after the messages beginning with A through Z and 0 through 9 and is alphabetized with the messages whose first nonvariable word or character begins with S.
<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>SIGNIFICANCE</th>
<th>ACTION</th>
<th>ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORTED</td>
<td>XEDIT terminated its execution and recorded only those editing changes performed during the editing session which were written to files via FILE, COPY, or COPYD commands.</td>
<td>STOP command has been executed. Recall XEDIT if more editing is to be done.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>ARGUMENT ERROR</td>
<td>One of the following:</td>
<td>Reenter last command with correct parameters.</td>
<td>XEDIT</td>
</tr>
<tr>
<td></td>
<td>- One string field is needed but is missing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Two string fields are needed but the second is missing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- While in INPUT mode the user entered a line count or string count parameter value other than zero on a command that might move the pointer, or used the up-arrow or slash (/) prefix characters on a command.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Margin specifications on WMARGIN command are in wrong order.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The Y or Z command list is missing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- In OCRCHANGE an even number of digits was not specified for the octal number.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAD FILE NAME</td>
<td>User did not specify a required file name or specified one that contains bad characters, is more than seven characters in length, or is reserved by XEDIT.</td>
<td>Reenter command with correct file name specified.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>BAD TEXT LINE ENCOUNTERED</td>
<td>The last word of a nonempty record does not contain an end of line byte. XEDIT aborts.</td>
<td>Most often user has forgotten to issue a PACK command after leaving TEXT mode. PACK the file and recall XEDIT.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>BATCH ABORT-COMMAND ERROR</td>
<td>Command syntax or parameter error (other than DELIMITER) was encountered in batch mode. XEDIT aborts.</td>
<td>Correct error and resubmit job.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>CANNOT EDIT EXECUTE ONLY FILES</td>
<td>User attempted to edit a file which is accessed in execute only mode.</td>
<td>Ensure that file name specified is correct. If so, access file in write mode.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>COMMAND NOT VALID</td>
<td>User issued an invalid command under INPUT or CREATION mode as follows:</td>
<td>Enter a valid command such as REPLACE under INPUT mode or INPUT under CREATION mode.</td>
<td>XEDIT</td>
</tr>
</tbody>
</table>
**MESSAGE**

- **COMMAND STACKING ERROR**
  - User attempted to recursively call an input source (for example, a Y or Z command is used within another Y or Z command).
  - Enter legal command.
  - XEDIT

- **DELMITTER**
  - Warning message indicating that user omitted closing delimiter of a delimited string parameter. XEDIT executes the command by assuming that a string delimiter appears after the last nonblank character of the string.
  - Continue editing.
  - XEDIT

- **EDIT**
  - User has been returned to EDIT mode.
  - Enter next editing command.
  - XEDIT

- **EMPTY FILE/CREATION MODE ASSUMED**
  - User entered XEDIT without specifying a file to be edited, no primary file is present as default, or the file specified by name or by default is empty. XEDIT automatically enters CREATION mode.
  - Enter a command valid under CREATION mode.
  - XEDIT

- **END OF FILE**
  - XEDIT read the end-of-information mark and terminates execution of the last command. The file pointer is positioned at the beginning-of-information.
  - Enter next editing command.
  - XEDIT

- **END OF INPUT ENCOUNTERED—ABORTED**
  - While in batch mode an end-of-record, end-of-file, or end-of-information mark was encountered on the input file. XEDIT aborts.
  - Enter an END or QUIT command as the last command in the input file. Resubmit job.
  - XEDIT

--- **EOF---**

  - XEDIT read an end-of-file mark. Execution of the command continues until normal termination.
  - The end-of-file mark can be deleted with the DEOF command and the printing of the message can be turned off by the TEOF command.
  - XEDIT

--- **EOR---**

  - XEDIT read an end-of-record mark. Execution of the command continues until normal termination.
  - The end-of-record mark can be deleted with the DEOR command and the printing of the message can be turned off by the
<table>
<thead>
<tr>
<th>MESSAGE</th>
<th>SIGNIFICANCE</th>
<th>ACTION</th>
<th>ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR IN XEDIT ARGUMENTS</td>
<td>User issued an XEDIT control statement with illegal options specified.</td>
<td>TEOR command.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>FILE FUNCTION ILLEGAL</td>
<td>User is working with direct access files, entered END, FILE, or QUIT command, and specified the LOCAL option or specified the COPY option for a file not attached in WRITE mode.</td>
<td>Correct the XEDIT control statement.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>FILE NAME CONFLICT</td>
<td>User issued more than one XEDIT control statement parameter which specified the same file name.</td>
<td>Change the mode option on the command.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>FILE NOT XEDITABLE</td>
<td>User specified a non-empty file which does not contain a legal line. It cannot be edited by XEDIT.</td>
<td>Correct the XEDIT control statement and reenter it.</td>
<td>XEDIT</td>
</tr>
<tr>
<td><em>PR</em> COMMAND STACKING ERROR</td>
<td>XEDIT internal error. The PR XEDIT control statement option has not been processed.</td>
<td>Inform site analyst.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>ILLEGAL PARAMETER</td>
<td>User specified one of the following: - An alphabetic DELIMIT command delimiter. - Extra data after the last parameter. - A parameter on a command that does not require a parameter. - An RMARGIN command less than 10 or greater than 160. - Tab stop values not in increasing order or not between 1 and 160.</td>
<td>Correct command and reenter.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>INPUT</td>
<td>User has entered INPUT mode.</td>
<td>After the prompting question marks appear, enter lines to be inserted.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>LINE NUMBER NOT FOUND, COMMAND NOT EXECUTED</td>
<td>The line number prefix specified a line number not in the file. XEDIT does not execute the command and positions the pointer at the next highest line number.</td>
<td>None.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>LINE NUMBER TOO LARGE</td>
<td>A line number formed by the ALN, ALNS, or RLN commands exceeded 99999.</td>
<td>Reenter the command specifying a smaller starting line number or a smaller increment.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>MESSAGE</td>
<td>SIGNIFICANCE</td>
<td>ACTION</td>
<td>ROUTINE</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>LOCAL FILE ERROR</td>
<td>XEDIT internal error.</td>
<td>Inform site analyst.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>NO SUCH COMMAND</td>
<td>User entered an illegal command or an improper separator after the command.</td>
<td>Enter legal command.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>NULL LINE-IN CREATION MODE</td>
<td>User issued a PRINT command on an empty file.</td>
<td>Enter a command valid</td>
<td>XEDIT</td>
</tr>
<tr>
<td>STRING NOT FOUND</td>
<td>The specified string could not be found.</td>
<td>under CREATION mode.</td>
<td></td>
</tr>
<tr>
<td>TRUNCATION AT n</td>
<td>A processed file line (the nth line in the file) exceeds the maximum line</td>
<td>Extend search for string or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>width. XEDIT truncates the line.</td>
<td>continue with next editing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>command.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>XEDIT version</td>
<td>XEDIT header line listing current version.</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XEDIT</td>
<td></td>
</tr>
<tr>
<td>XEDIT INTERNAL ERROR (ERD)</td>
<td>An XEDIT internal error has occurred.</td>
<td>Inform site analyst.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>NOTIFY CONSULTANT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YOU DELETED THE ENTIRE FILE</td>
<td>XEDIT cannot locate a line at which to position the pointer; XEDIT aborts.</td>
<td>Recall XEDIT. CREATION</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mode is assumed until a line</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>of text exists within the file.</td>
<td></td>
</tr>
<tr>
<td>O-IN CREATION MODE</td>
<td>User issued a UNHEE command; however, no lines exist within the file.</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>XEDIT</td>
<td></td>
</tr>
<tr>
<td>fileam ALREADY PERMANENT</td>
<td>User specified the S or SAVE option on an END, FILE, or QUIT command and</td>
<td>Enter the END, FILE, or QUIT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the named file is already permanent.</td>
<td>command with an R or REPLACE</td>
<td></td>
</tr>
<tr>
<td>fileam CANNOT BE ACCESSSED</td>
<td>User requested a file which is not local, is an execute only or append only</td>
<td>Ensure that the file to be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>file, or is not available from the user's permanent file area.</td>
<td>edited is local and can be</td>
<td></td>
</tr>
<tr>
<td>fileam IS A LOCAL FILE</td>
<td>User specified the L or LOCAL option on an END, FILE, or QUIT command.</td>
<td>None.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The edited file is now a local file.</td>
<td>XEDIT</td>
<td></td>
</tr>
<tr>
<td>MESSAGE</td>
<td>SIGNIFICANCE</td>
<td>ACTION</td>
<td>ROUTINE</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>filename NOT FOUND</td>
<td>User requested a file which could not be found in the user's permanent file area. XEDIT requests the name of a file to edit.</td>
<td>Enter correct name of the file to be edited. If file is permanent, enter ,P after the name. If file is to be created, enter ,C after the name.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>NAME EDIT FILE?</td>
<td></td>
<td>None.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>filename REPLACED</td>
<td>User specified the R or REPLACE option on an END, FILE, or QUIT command. The edited file has replaced the permanent file of the same name in the user's permanent file area.</td>
<td>None.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>filename REWRITTEN</td>
<td>An END, FILE, or QUIT command has been executed for a direct access file. The edited file has been copied back to the permanent file of the same name in the user's permanent file area.</td>
<td>None.</td>
<td>XEDIT</td>
</tr>
<tr>
<td>filename SAVED</td>
<td>User specified the S or SAVE option on an END, FILE, or QUIT command. The edited file is now an indirect access permanent file.</td>
<td>None.</td>
<td>XEDIT</td>
</tr>
</tbody>
</table>
Other terminal error messages not listed are issued by NOS. Refer to the Time-Sharing User's Reference Manual or IAF Reference Manual for explanations.

An on-line explanation of any of these messages can be obtained by entering the EXPLAIN command after the message is printed. Refer to the EXPLAIN command for further details.
Figures C-1 through C-10 are sample editing sessions which demonstrate a collection of similar XEDIT commands.

old,savings
READY.
batch
$RFL,0.
/attach,xedit/un=library  
/xedit
XEDIT 3.0
?? help,print  
PRINT $ [P]

Action- Prints $ lines starting at the current pointer position. The pointer is left positioned at the last line printed.  
?? h tn  
TOPNULL

Action- Same as the TOP command except that a blank line is inserted as the first line of the file. If the edit file has leading record marks before the first text line, this command can be used to position the pointer before those marks.  
??

1 User calls XEDIT into execution.
2 User issues HELP command; wants information about PRINT command.
3 XEDIT prints information about the PRINT command.
4 User issues an abbreviated HELP command; wants information about the TOPNULL command.
5 XEDIT prints information about TOPNULL.
6 XEDIT indicates user should enter another XEDIT command.

Figure C-1. HELP Command
EXAMPLES OF XEDIT USE

Figures C-1 through C-10 are sample editing sessions which demonstrate a collection of similar XEDIT commands.

old,savings
READY.
bach
$RFL,0.
/attach,xedit/un=library ①
/xedit
XEDIT 3.0
?? help,print ②
PRINT $ [P]
=
Action: Prints $ lines starting at the current pointer position. The pointer is left positioned at the last line printed. ③
?? h tn ④
TOPNULL
=
Action: Same as the TOP command except that a blank line is inserted as the first line of the file. If the edit file has leading record marks before the first text line, this command can be used to position the pointer before those marks. ⑤
?? ⑥

① User calls XEDIT into execution.
② User issues HELP command; wants information about PRINT command.
③ XEDIT prints information about the PRINT command.
④ User issues an abbreviated HELP command; wants information about the TOPNULL command.
⑤ XEDIT prints information about TOPNULL.
⑥ XEDIT indicates user should enter another XEDIT command.

Figures C-11 and C-12 are sample batch processing control statement records which use XEDIT to edit a permanent file and to create a file.

Figure C-1. HELP Command
old, saving
READY.
bach
$RFL,0.
/attach,xedit/un=library
/xedit

XEDIT 3.0
?? print* ①
00100 PRINT TAB(10),"DAYDREAM SAVINGS PLAN"
00110 PRINT
00120 PRINT "DAYS","DAILY AMOUNT","TOTAL SAVINGS"
00130 LET N=1
00140 LET D=.01
00150 LET A=0
00160 LET A=A+D
00170 PRINT USING 00180,N,D,A
00180 ^##
00190 LET D=D+D
00200 LET N=N+1
00210 IF N<31 THEN 00160
00220 END
END OF FILE
?? print ④
00100 PRINT TAB(10),"DAYDREAM SAVINGS PLAN" ⑤
?? 170 ③
00170 PRINT USING 00180,N,D,A
?? x200 ②
?? next ⑥
00210 IF N<31 THEN 00160
?? next3 ②
00130 LET N=1
?? next -3 ⑥
00100 PRINT TAB(10),"DAYDREAM SAVINGS PLAN"
?? locate/let...+/#/ ⑧
00160 LET A=A+D
00190 LET D=D+D
0C290 LET N=N+1
END OF FILE
?? locate/let/3 ⑧
00130 LET N=1
00140 LET D=.01
00150 LET A=0
?? print ⑨
00150 LET A=0
?? bottom ⑨
00220 END
??

① User calls XEDIT.
② User issues PRINT command to list entire contents of the file (that is, n is entered as the n parameter).
③ XEDIT lists entire file.
④ User issues PRINT command; by default since no n parameter appears, XEDIT assumes only 1 line should be listed.
⑤ XEDIT prints the line indicated by the current pointer position.

NOTE
Since the first PRINT command reached an end-of-file, the pointer was repositioned to the top of the file.

⑥ User enters a particular line number (170) to place pointer at that position. By default, verify mode is in effect.
⑦ User enters line number 200 but indicates that XEDIT should not verify the current pointer position (that is, an X prefixes the user entry).
⑧ User issues NEXT command to advance the pointer by one line; XEDIT verifies the pointer position.
⑨ User issues an up-arrow (^) prefix to reposition pointer to the beginning of the file and to NEXT command. Advances the pointer by three lines.
⑩ User issues NEXT command to move the pointer backwards by three lines.
⑪ User issues LOCATE command to find the third line (from the current pointer position) containing the string LET.
⑫ User issues LOCATE command to find all lines in the file containing a combination of two strings (LET and +).
⑬ User issues PRINT command; pointer is positioned at the beginning of the file.
⑭ User issues BOTTOM command; pointer is now positioned at the last line in the file.

Figure C-2. Pointer Movement Commands
old, savings
READY.
batch
$RFL.O.
/attach,xedit/unslibrary
/xedit
XEDIT 3.0
?? print
 00100 PRINT TAB(10), "DATEDREAM SAVING PLAN"
 00110 PRINT
 00120 PRINT TAB(10) + "DAYS", "DAILY AMOUNT", "TOTAL SAVINGS"
 00130 LET N=1
 00140 LET D=0.01
 00150 LET A=0
 00160 LET A=A+D
 00170 PRINT USING 00180,N,D,A
 00180 ** ******* ** ******* **
 00190 LET D=D+D
 00200 LET N=N+1
 00210 IF N<31 THEN 00160
 00220 END
END OF FILE
?? locate/tab
 00100 PRINT TAB(10), "DATEDREAM SAVING PLAN"
?? change/(10)/ (130)/
 00100 PRINT TAB(13), "DATEDREAM SAVING PLAN"
?? change/<c<=/ 
 00210 IF N<31 THEN 00160
?? tc/let/**
 00130 N=1
 00140 D=0.01
 00150 A=0
 00160 A=A+D
 00190 D=D+D
 00200 N=N+1
END OF FILE
?? locate/nz/**
 00100 PRINT TAB(13), "DATEDREAM SAVING PLAN"
??
00130 N=1.75
?? 100 modify
 00100 PRINT TAB(13), "DATEDREAM SAVING PLAN" ?
??

1. User calls XEDIT to edit a file called SAVINGS.
2. User issues PRINT command to list entire file.
3. XEDIT lists the SAVINGS file.
4. User issues LOCATE command to move pointer to first line containing the string TAB; XEDIT verifies which line was located.
5. User issues CHANGE command to replace each reference to (10) with (13) in the first line containing a (10); XEDIT verifies how the change was made.
6. User issues CHANGE command to replace each reference to < with <= in the first line containing a < sign; XEDIT verifies how the change was made.
7. User issues an abbreviated CHANGE command with an up-arrow (^) prefix to delete all references to LET with a blank (in other words, delete the LET references) throughout the entire file.
8. XEDIT indicates which lines were affected by the preceding CHANGE command.
9. User issues LOCATE command to find the first line containing an N-string; XEDIT places pointer to line 130.
10. User issues ADD command to attach information onto the end of the line designated by the current pointer position (that is, line 130).
11. User specifies that .75 should be added to the end of line 130.
12. XEDIT verifies the change which resulted because of the above ADD command.
13. User issues MODIFY command to alter line 100.
14. XEDIT prints line 100.
15. User issues MODIFY directives to indicate how line 100 should be altered.
16. XEDIT verifies how line 100 was affected by the above MODIFY command.

Figure C-3. String Editing Commands
old,depart
READY.
bath
$RFL,0.
/attach,xedit/un=library
/xedit
XEDIT 3.0
?? print*
*DEPARTMENT 4208*
D4208 E927 EX100 KRAMER, SCOTT W. HQR24-130
D4208 E208 EX130 JOHNSON, JACK J. SWP19-250
D4208 E492 EX120 MILLER, RICK HQR44-140
D4208 B698 EX125 SCOTT, FRANK R. MXJ97-220
END OF FILE
?? locate/130
DELIMITER
D4208 E927 EX100 KRAMER, SCOTT W. HQR24-130
?? trim
?? 1locate/130 /
D4208 E208 EX130 JOHNSON, JACK J. SWP19-250
?? wmargin 7,10
?? 1dw/208/ 
D4208 E208 EX130 JOHNSON, JACK J. SWP19-250
?? w 18 18
?? 1la/scott/
D4208 B698 EX125 SCOTT, FRANK R. MXJ97-220
??

1 User calls XEDIT to edit a file called DEPART.
2 User issues PRINT command to list entire file; XEDIT lists file.
3 User attempts to find executive number 130 but instead finds executive 100 since his facility number HQR24-130 happened to also have the number 130 in it.
4 To correctly locate executive 130, the user issues a TRIM command to tell XEDIT to ignore the trailing spaces (blanks) to the right of all facility numbers such as HQR24-130.
5 The user intends to search for employee number 208, so the WMARGIN command is issued to restrict the search window to just the employee number field (columns 7 through 10 inclusively).
6 Once the window has been defined, the user issues an abbreviated LOCATE command (the window W post-fix) to specify that XEDIT should look at only the employee number field during the search. XEDIT responds by finding employee Jack Johnson number E208 as requested, instead of the first line with D4208 in it.
7 Next the user wants to find the first employee whose last name is Scott. Since the last name field begins in column 18, the user issues an abbreviated WMARGIN command to redefine the window to be just that column.
8 The user finally issues an abbreviated LOCATE command with the anchor prefix to tell XEDIT to begin the search at the top of the file. XEDIT correctly finds Frank Scott instead of Scott Kramer.

Figure C-4. String Search Control Commands
old, plan
READY...
batch
BPL, 0.
*attach, xedit/unilibrary
*xedit
XEDIT 3.0
?? print
TABLE ONE
?? listab
TABLE TWO
END OF FILE
?? deftab ;
?? tabs 20 30 40 50 60
?? listab
?? tabs 20 30 40 50 60
?? (7)
INPUT
?? ; profit picture
?? ************
?? ; q, q, q, n ; total
?? -------
?? ; sales revenue; $48,578; 51,027; 53,598; $56,300; $209,503
?? ; total expenses; 42,273; 43,025; 45,709; 47,548; $179,482
?? ; operating profit; 6,305; 7,057; 7,889; 8,752; $30,021
?? -------
?? (8)
EDIT
?? "next"
TABLE TWO
?? tabs 34 34 34
?? listab
?? tabs 34 34 34 60
?? (9)
INPUT
?? ; cash flow forecast
?? ************
?? ; july; aug; sept
?? ------
?? ; total cash available; $250,000; $852,400; $959,316
?? ; total cash required; 378,100; $386,600; $397,966
?? ; ending cash balance; 71,900; $33,760; $51,408
?? -------
?? (10)
EDIT
?? "print"
TABLE ONE

PROFIT PICTURE

<table>
<thead>
<tr>
<th></th>
<th>1Q</th>
<th>2Q</th>
<th>3Q</th>
<th>4Q</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>revenue</td>
<td>48,578</td>
<td>51,027</td>
<td>53,598</td>
<td>56,300</td>
<td>$209,503</td>
</tr>
<tr>
<td>expenses</td>
<td>42,273</td>
<td>43,025</td>
<td>45,709</td>
<td>47,548</td>
<td>$179,482</td>
</tr>
<tr>
<td>profit</td>
<td>6,305</td>
<td>7,057</td>
<td>7,889</td>
<td>8,752</td>
<td>$30,021</td>
</tr>
</tbody>
</table>

TABLE TWO

CASH FLOW FORECAST

<table>
<thead>
<tr>
<th></th>
<th>JULY</th>
<th>AUG</th>
<th>SEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>450,000</td>
<td>452,300</td>
<td>459,316</td>
</tr>
<tr>
<td>required</td>
<td>378,100</td>
<td>386,600</td>
<td>397,966</td>
</tr>
<tr>
<td>balance</td>
<td>71,900</td>
<td>33,760</td>
<td>51,408</td>
</tr>
</tbody>
</table>

END OF FILE
??

User calls XEDIT to edit a file called PLAN.
User issues PRINT command to list entire file.
XEDIT lists file.
User issues DEFTAB command to establish a semicolon (;) as the tab character.
User issues TABS command to establish tab stop column positions 20, 30, 40, 50 and 60.
User issues LISTAB command to list the current tab character and tab stop column positions.
User presses carriage return to insert an unknown number of lines after the TABLE ONE line.
User enters several new lines with the tab character used where tabbing is desired.
User presses carriage return to terminate the insertion of new lines.
User issues NEXT command to advance the pointer by one line; XEDIT verifies the pointer position.
User issues another TABS command to reset the tab stops to positions 24, 34, and 44.
User issues LISTAB command to list the current tab character and tab stop column positions.
User presses carriage return to insert an unknown number of lines after TABLE TWO line.
User adds several new lines with the tab character used where tabbing is desired.
User presses carriage return to terminate the insertion of new lines.
User issues a PRINT command with an up-arrow (↑) prefix to list the entire file.
XEDIT lists the file verifying that tabbing was correctly performed.

Figure C-5. Tab Control Commands
old, manager
READY.
bitch
%RFL,0.
/attach,xedit/un=library
/xedit
  XEDIT 3.0
?? print* ②

*EXECUTIVE MANAGEMENT*
EX100 S.W. KRAMER HQR24-130
EX110 J.J. JOHNSON SWP19-130
EX120 R.E. MILLER HQR44-130
EX125 F.R. SCOTT MXJ97
END OF FILE
?? rmargin 30 ④
?? findll*

*EXECUTIVE MANAGEMENT*
EX110 S.W. KRAMER HQR24-130
EX110 J.J. JOHNSON SWP19-130
EX120 R.E. MILLER HQR44-130
END OF FILE
?? trunc*

*EXECUTIVE MANAGEMENT*
EX100 S.W. KRAMER HQR24-13
EX110 J.J. JOHNSON SWP19-13
EX120 R.E. MILLER HQR44-13
EX125 F.R. SCOTT MXJ97
END OF FILE
??

① User calls XEDIT to edit a file called MANAGER.
② User issues PRINT command to list entire file.
③ XEDIT lists file.
④ User sets the right margin to column 30.
⑤ User requests XEDIT to find all lines that are larger than 30 characters (current RMARGIN setting) by issuing the FINDLL command; XEDIT responds accordingly.
⑥ User requests XEDIT to truncate all lines larger than 30 characters (current RMARGIN setting) by issuing the TRUNC command; XEDIT responds accordingly.

Figure C-6. Margin Control Commands
Figure C-7. Line Editing Commands
old,add5
READY.
batch
$RPL.0.
/attach,xedit/un=library
/xedit
XEDIT 3.0
?? print
### NAMES ADDRESSES ARE FICTITIOUS ###
00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP 55722
00120 EX6533
00160 Q.E. SMITH
^165 APT 23
00170 PQR 55 ZIP 99107
00180 EXT 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXT 18
END OF FILE
?? brief
?? change/zip/zip code/*
END OF FILE
?? change/ext/extension/*
END OF FILE
?? print
### NAMES ADDRESSES ARE FICTITIOUS ###
00100 M.T. JONES
00110 1544 WILSHIRE ST ZIP CODE 55722
00120 EX6533
00160 Q.E. SMITH
00165 APT 23
00170 PQR 55 ZIP CODE 99107
00180 EXTENSION 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXTENSION 18
END OF FILE
?? verify
?? change/zip code/zip/*
^1110 1544 WILSHIRE ST ZIP 55722
00170 PQR 55 ZIP 99107
END OF FILE
?? change/extension/ext/*
00180 EXT 8837
00210 EXT 18
END OF FILE
??

1. User calls XEDIT to edit a file called ADDRESS.
2. User issues PRINT command to list the entire file.
3. XEDIT lists file.
4. User issues BRIEF command so that XEDIT will not verify the effects of XEDIT commands.
5. User issues CHANGE command; XEDIT does not verify its results.
6. User issues another CHANGE command; XEDIT does not verify its results.
7. User issues PRINT command.
8. XEDIT lists edited file; indicates how the above commands altered the user's file.
9. User issues VERIFY command so that XEDIT will verify the effects of XEDIT commands.
10. User issues CHANGE command.
11. XEDIT verifies which lines were affected by the preceding CHANGE command.
12. User issues another CHANGE command.
13. XEDIT verifies which lines were affected by the preceding CHANGE command.

Figure C-8. VERIFY and BRIEF Commands (Sheet 1 of 2)
old,address
READY.
batch
SRML,0.
/attach,xedit/un=library
/xedit
XEDIT 3.0
?? print* ②
### NAMES/ADDRESSES ARE FICTIONAL ###
00100 M. JONES
00110 1544 WILSHIRE ST ZIP 90022
00120 (66) 573
00130 R. S. SMITH
00140 APT 23
00150 1075 ZIP 90017
00160 EXT 8837
00170 R. RFR
00180 5710 14TH ST
00190 EXT 18
END OF FILE
?? xchange/zip/zip code/* ④
END OF FILE
?? file,nesf,s ⑤
NEW SAVED ⑥
?? xchange/ext/extension/* ⑦
END OF FILE
?? file,replace ⑧
ANDS REPLACED ⑨
??

① User calls XEDIT to edit ADDRESS file.
② User issues PRINT command to list entire file.
③ XEDIT lists file.
④ User issues CHANGE command prefixed by an X; since verify mode is automatically in effect, the X prefix suppresses command verification.
⑤ User issues FILE command to save the editing changes on a new permanent file called NEWF.
⑥ XEDIT indicates that NEWF has been saved in the user's permanent file catalog.
⑦ User issues another CHANGE command with X prefix.
⑧ User issues another FILE command to save these editing changes; since the REPLACE option is selected and the frame parameter defaults to the primary local file, the change from both CHANGE commands now appear on the permanent copy of ADDRESS.
⑨ XEDIT indicates that the edited file has replaced the original file in the user's permanent file catalog.

Figure C-8. FILE Command (Sheet 2 of 2)
old.address
READY.
bach
$RFL,0.
/attach,xedit/un=library
/xedit
XEDIT 3.0
?? print# ②
### NAMES/ADDRESSES ARE FICTIONOUS ###
00100 M.T. JONES
00110 1544 WILSHIRE ST
00120 EXT 6533
00160 Q.E. SMITH
00170 POB 55
00180 EXT 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXT 18
END OF FILE
?? #locate/ext/$next-1$+add$xnex2 ④
?? LOCATE/EXT/
00120 EXT 6533
?? NEXT-1
00110 1544 WILSHIRE ST
?? +ADD
? 55722 ⑥
00110 1544 WILSHIRE ST 55722
?? XNEXT2 ⑦
?? - ⑨
?? LOCATE/EXT/
00180 EXT 8837
?? NEXT-1
00170 POB 55
?? +ADD
? 55703
00170 POB 555703
?? XNEXT2
?? - ⑩
?? LOCATE/EXT/
00210 EXT 18
?? NEXT-1
00200 8710 14TH ST
?? +ADD
? 55713
00200 8710 14TH ST 55713
?? XNEXT2
END OF FILE
?? $locate/ext/$next-1$+modify$xnex2 ⑪
00120 EXT 6533
00110 1544 WILSHIRE ST 55722
00110 1544 WILSHIRE ST 55722
?? - ⑯
00180 EXT 8837
00170 POB 5555703
00170 POB 5555703
?? - ⑰
00170 POB 55 ZIP 55703
?? - ⑱
00210 EXT 18
00200 8710 14TH ST 55713
00200 8710 14TH ST 55713
?? - ⑲
00200 8710 14TH ST ZIP 55713
END OF FILE
??

1. User calls XEDIT to edit ADDRESS file.
2. User issues PRINT command to list entire file.
3. XEDIT lists file.
4. User issues a Z command which contains 4 component commands; its objective is to add a numeric zip code to the file line containing an address. In this Z command, a $ acts as the delimeter which separates component commands. In addition, the plus (+) prefix on the ADD command tells XEDIT that ADD input data (in this case a zip code) will come from the next entry on the Z command sequence, rather than from normal input mode (a single question mark (?)).
5. XEDIT verifies that the first three component commands (LOCATE, NEXT, and ADD) are executed.
6. XEDIT verifies how the ADD command affected the file.
7. XEDIT verifies the 4th component command (NEXT) is executed.
8. User enters a - sign to advance the file pointer (one line by default) and reexecute the preceding Z command; user wants to modify the second address in the file.
9. XEDIT reexecutes the Z command sequence.
10. User enters a - sign to advance pointer and reexecute Z command; user wants to modify the third address in the file.
11. XEDIT reexecutes the Z command sequence.
12. User enters a Y command which is made up of 4 component commands; its objective is to add the word ZIP before each numeric zip code in the file. NOTE: Unlike the Z command, the Y command does not list each component command before it is executed.
13. XEDIT executes the MODIFY command (which is one of the component Y commands).
14. Since the plus (+) prefix was not used on the MODIFY command the user issues his MODIFY directives from normal input mode (a single question mark (?)) to alter line 110.
15. XEDIT verifies how these directives affected line 110.
16. User enters - sign to reexecute the preceding Y command after advancing the pointer one line; user wants to modify the second address in the file.
17. XEDIT reexecutes the Y sequences; user modifies the second address.
18. User enters - sign to reexecute Y command and modify the third address in the file.
19. XEDIT reexecutes the Y sequence; user modifies the third address.

Figure C-9. Z and Y Commands and + Prefix Character
old,address
READY.
batch
$RFL,0
/attach,xdit/un=library
/xdit
XEDIT 3.0
?? print

*** NAMES/ADDRESSES ARE FICTITIOUS ***
00100 M.T. JONES
00110 1544 WILSHIRE ST
00120 EXT 5533
00160 Q.S. SMITH
00170 P.OB 55
00180 EXT 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXT 18
END OF FILE
?? change/###/###/##
*** NAMES/ADDRESSES ARE FICTITIOUS ***
?? end,,r
ADDRESS REPLACED
/lnh
*** NAMES/ADDRESSES ARE FICTITIOUS ***
00100 M.T. JONES
00110 1544 WILSHIRE ST
00120 EXT 5533
00160 Q.S. SMITH
00170 P.OB 55
00180 EXT 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXT 18
/old,address
/lnh
*** NAMES/ADDRESSES ARE FICTITIOUS ***
00100 M.T. JONES
00110 1544 WILSHIRE ST
00120 EXT 5533
00160 Q.S. SMITH
00170 P.OB 55
00180 EXT 8837
00190 P.T. BEE
00200 8710 14TH ST
00210 EXT 18
xedit
*** NAMES/ADDRESSES ARE FICTITIOUS ***
?? print 5
ADDRESS IS A LOCAL FILE
/lnh
*** NAMES/ADDRESSES ARE FICTITIOUS ***
00100 M.T. JONES
00110 1544 WILSHIRE
old,address
/lnh
*** NAMES/ADDRESSES ARE FICTITIOUS ***
00100 M.T. JONES
00110 1544 WILSHIRE
stop
This batch job calls XEDIT to create a file called plan according to the directives in the second record of the job. All output goes on the file out which is rewound and printed on the central site line printer.

NEWPLAN.
USER(username,passwd,family)
CHARGE(chrgnum,projnum)
ATTACH.XEDIT/UN=LIBRARY
XEDIT(plan,L=out,C)
REWIND(out)
COPYSBF(out)
7/8/9
deftab*
tabs 20 30 40 50 60
input, *
*profit picture
*************
*1q*2q*3q*4q*total
*------------------*
sales revenue 48,578 51,027 53,598 56,300 $209,503
*total expenses 42,273 43,952 45,709 47,548 $179,482
*operating profit 6,305 7,075 7,889 8,752 $30,025
$edit
top
print*
end,r1
6/7/8/9

The following is the output from the batch job NEWPLAN.

XEDIT 3.0
INPUT
EDIT

PROFIT PICTURE
*************
1Q 2Q 3Q 4Q TOTAL

SALES REVENUE 48,578 51,027 53,598 56,300 $209,503
TOTAL EXPENSES 42,273 43,952 45,709 47,548 $179,482
OPERATING PROFIT 6,305 7,075 7,889 8,752 $30,025

END OF FILE
PLAN REPLACED
PLAN IS A LOCAL FILE

Figure C-11. Calling XEDIT Within a Batch Job to Create a File
The batch job FIXPLAN calls XEDIT to edit a permanent file called plan. All XEDIT output goes on a file called out which is rewound and printed on the central site line printer.

```
FIXPLAN.
USER(username, password, family)
CHANGE(chrgnum, projnum)
ATTACH,XEDIT/UN-LIBRARY>
EDIT(plan,N=out,P)$CHANGE/30,025/30,021/STOP$PRINT$SEND.,RL
REPLY(out)
COPYSRF(out)
6/7/84
```

The following is the listing of the file out.

```
XEDIT 3.0
OPERATING PROFIT 6,305 7,075 7,839 8,752 $30,021
PROFIT PICTURE
10 20 30 40 TOTAL
SALES REVENUE 48,578 51,027 53,598 56,300 $209,503
TOTAL EXPENSES 42,273 42,952 45,709 47,548 $179,482
OPERATING PROFIT 6,305 7,075 7,839 8,752 $30,021
END OF FILE
PLAN REPLACED
PLAN IS A LOCAL FILE
```

Figure C-12. Calling XEDIT Within a Batch Job to Edit a File
INDEX

A suffix 5-5
Accessing files 2-1
ADD command 5-1
ADDLN command 7-1
ADDLNS command 7-1
Alignment guide 5-6
Appending a string 5-1
ASCII files 1-1

B control statement parameter 2-4
Bad lines 7-3
Batch origin editing 2-2, 3
Input mode 6-4
Binary files 1-1
BOTTOM command 4-3
BREAK key 3-4
BRIEF command 3-2
Brief mode 3-2

C control statement parameter 2-3
Calling XEDIT 2-1
Cancellation of editing changes 8-2
Carriage return
  Speed adjustment 2-1; 5-6
  To enter and leave input mode 6-4
CHANGE command 5-1
CHANGES command 5-1
Character-by-character line modification 5-6
Command
  Descriptions 3-4
  Parameters 3-1
  Prompts 3-1, 2
  Repetition 10-2
  Syntax 3-1
Conversion of octal strings 8-2
COPY
  Command 9-1
  Mode parameter
    On END or QUIT commands 12-2
    On FILE command 9-3
COPYD command 9-2
Copying files 9-1
Creation mode 2-3

DBADL command 7-4
Dcs control statement parameter 2-3
Defining a
  Delimiter character 10-1
  Escape character 6-5
  Tab character 11-1
DEFTAB command 11-1
DELETE command 6-1, 2
DELETBLN command 7-1

Deleting
  Characters 5-8
  File marks 8-3
  Leading blanks 8-1
  Lines 6-1, 2
  Lines without line numbers 7-4
  Record marks 8-3
  Strings 5-2
DELIMIT command 10-1
Delimited command sequence 10-1
  Interrupting execution of 3-4
  On control statement 2-3
Delimiter character for
  dcs parameter 2-3
  Delimited command sequence 10-1
  String parameter 3-2
DEOF command 8-3
DEOR command 8-3
Direct access file editing 2-1
Display code conversion 8-2
DLBLANKS command 8-1

EDIT command 6-4, 5
END command 12-1
End-of-information 4-1
Entering editing data 3-2
EOF message 3-3
EOR message 3-3
Error message explanation 3-4
Escape character 6-4, 5
EXPLAIN command 3-4

FBADL command 7-3
FILE command 9-3
File
  Copying 9-1
  Line length 5-1, 6; 7-1; 11-2
  Manipulation 9-1
  Marks 1-1; 3-3; 8-2; 9-2
  Pointer in input mode 6-4, 5
  Pointer movement by prefix or command 4-1
  Saving 9-3
  Types 1-1
FINDLL command 11-2
First record editing directives 2-4
FR control statement parameter 2-4

HELP command 3-4

I control statement parameter 2-3
INPUT command 6-4, 5

60455730 A
Input file 2-3
Input mode 6-3,4,5
INSERT command 6-3
INSERTB command 6-3
Inserting
Blank line at top of file 8-1
Characters 5-8
File marks 8-4
Lines 6-3
Record marks 8-4
Strings 5-1,3
Interrupting XEDIT 3-4

L control statement parameter 2-4
Line
Deletion 6-1,2
Insertion 6-3
Listing 4-4
Replacement 6-3
Line number 4-1
Command 4-1
Deleting line without 7-4
Deletion 7-1
Insertion 7-1
Prefix 3-1
Replacement 7-3
Search for line without 7-3
LISTAB command 11-1
LOCAL mode parameter 9-3
On FILE 9-3
On END or QUIT 12-2
LOCATE command 4-1,2
Login procedure 2-1
Long lines 11-2

Margin setting 11-2
Merging files 9-2
Mode parameters
On FILE command 9-3
On END or QUIT commands 12-1,2
Modification directives 5-8
MODIFY command 5-6
Multiple command entries 10-1

NEXT command 4-3
NH parameter 2-4
NOBELLS command 3-3

Octal string conversion 8-2
OCTCHANG command 8-2
Output file 2-3, 4

PRINT command 4-4
Prompts 3-1, 2

QMOD command 5-6
QUIT command 12-1

READ command 9-3
READP command 9-2
Record mark 1-1
Deletion 8-3
Insertion 8-4
Suppression 3-3
Repeating commands 10-2
REPLACE
Command 6-3
Mode parameter
On END or QUIT command 12-2
On FILE command 9-3
Replacing
Characters 5-3
Line numbers 7-3
Lines 6-3
Strings 5-1
REPLACECCN command 7-3
RESTORE command 8-2
Reverse pointer movement 4-3
RL mode parameter
On END or QUIT commands 12-2
On FILE command 9-3
RMARGIN command 11-2

SAVE mode parameter
On END or QUIT commands 12-2
On FILE command 9-3
Setting
Right margin 11-2
Tab stops 11-1
SL mode parameter
On END or QUIT commands 12-2
On FILE command 9-3
String
Appending 5-1
Deletion 5-2
Delimiters 3-2
Insertion 5-3
Replacement 5-1
Search restrictions 5-4, 5
STOP command 12-1

Tab
Character
Definition 11-1
In input mode: 6-5
Stop setting 11-1
TABS command 11-1
TEOF command 3-3
TEOR command 3-3
Terminal bell suppression 3-3
Terminating XEDIT 12-1

Index-2

60455730 A
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time-sharing origin editing</td>
<td>2-1</td>
</tr>
<tr>
<td>TOP command</td>
<td>4-3</td>
</tr>
<tr>
<td>TOPNULL command</td>
<td>8-1</td>
</tr>
<tr>
<td>TRIM command</td>
<td>5-4</td>
</tr>
<tr>
<td>TRUNCATE command</td>
<td>11-3</td>
</tr>
<tr>
<td>VERIFY command</td>
<td>3-2</td>
</tr>
<tr>
<td>Verify mode</td>
<td>3-2</td>
</tr>
<tr>
<td>W suffix</td>
<td>5-5</td>
</tr>
<tr>
<td>WEOF command</td>
<td>8-4</td>
</tr>
<tr>
<td>WEOR command</td>
<td>8-4</td>
</tr>
<tr>
<td>WHERE command</td>
<td>4-4</td>
</tr>
<tr>
<td>Window restriction</td>
<td>5-5</td>
</tr>
<tr>
<td>WMARGIN command</td>
<td>5-5</td>
</tr>
<tr>
<td>Working file editing</td>
<td>2-1</td>
</tr>
<tr>
<td>X prefix</td>
<td>3-1,2</td>
</tr>
<tr>
<td>XEDIT Call</td>
<td>2-1</td>
</tr>
<tr>
<td>Control statement format</td>
<td>2-2</td>
</tr>
<tr>
<td>Input</td>
<td>2-3</td>
</tr>
<tr>
<td>Interruption</td>
<td>3-4</td>
</tr>
<tr>
<td>Output</td>
<td>2-4</td>
</tr>
<tr>
<td>Parameters</td>
<td>2-2,3,4</td>
</tr>
<tr>
<td>Termination</td>
<td>12-1</td>
</tr>
<tr>
<td>Y command</td>
<td>3-4;10-1,2</td>
</tr>
<tr>
<td>YQMOD command</td>
<td>5-6,7</td>
</tr>
<tr>
<td>Z command</td>
<td>3-4;10-1,2</td>
</tr>
<tr>
<td>- command</td>
<td>10-2</td>
</tr>
<tr>
<td>, command</td>
<td>10-2</td>
</tr>
</tbody>
</table>