

AD-A034 524

ROCK ISLAND ARSENAL ILL GENERAL THOMAS J RODMAN LAB
MEDIT - A PROGRAM TO EDIT COMPUTER SOURCE PROGRAMS.(U)
APR 76 J J HURT

F/G 9/2

UNCLASSIFIED

RIA-R-TR-76-010

NL

| OF |
AD
A034524



END

DATE
FILMED

2-77

ADA 034524

1
b.s.



R-TR-76-010

MEDIT

A PROGRAM TO EDIT
COMPUTER SOURCE PROGRAMS

by

JAMES J. HURT

1 APRIL 1976



PREPARED BY

RESEARCH DIRECTORATE

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION**

PREPARED FOR

RESEARCH DIRECTORATE
**GENERAL THOMAS J. RODMAN LABORATORY
ROCK ISLAND ARSENAL
ROCK ISLAND, ILLINOIS 61201**

DISCLAIMER

The findings of this report are not to be construed as an official department of the Army position, unless so designated by other authorized documents.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-TR-76-010	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MEDIT - A Program to Edit Computer Source Programs	5. TYPE OF REPORT & PERIOD COVERED 9 Final report	
7. AUTHOR(s) James J. Hurt	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Research Directorate, SARRI-LR-S GEN Thomas J. Rodman Laboratory Rock Island Arsenal, Rock Island, IL 61201	8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS DAMA-ARZ-B Chief of Research and Development HQ, Department of the Army	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1T161101A91A	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE 1 Apr 76	
	13. NUMBER OF PAGES 27	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Software Maintenance ADP Utilities		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A portable and easily used source text editor is described. This computer utility provides a method to maintain and modify computer source text in a manner that is very easy to use, provides a complete audit trail of changes, and can be used on a wide variety of computers, eg IBM-360-370, CDC-6000, UNIVAC-1100, et al.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

1 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DDC
 RECEIVED
 JAN 18 1977
 C

408 247
6pg

PREFACE

MEDIT is a computer program designed to assist a programmer in the maintenance of a large source library. Such a program is provided as an utility program by every major computer manufacturer, therefore some justification must be given for the subject of this report.

The author is familiar with the following vendor supplied utility programs:

- a. IEBUPDAT for IBM-360 and IBM-370 computers.
- b. IEBUPDTE for IBM-360 and IBM-370 computers.
- c. SETUP for CDC-6000 computers.
- d. UPDATE for CDC-6000 computers.
- e. EDIT, written by the Systems Research Group, The Ohio State University.

In addition, the author is cognizant of similar utility programs supplied by UNIVAC, Burroughs, Digital Equipment Corporation, and Computer Sciences Corporation. The following comparisons are based on the author's experience:

- i. EDIT is the easiest to use.
- ii. UPDATE provides the most useful features.
- iii. None of these are portable - none can be easily transferred to another vendor's hardware.

The intent in writing MEDIT was to provide a source library maintenance utility that was easy to use (as is EDIT), that provided useful capabilities (as UPDATE), and that can be easily transported from one machine to another. The actual program described must be viewed as Version 1 of the desired program. MEDIT is as easy to use as EDIT but does not provide all the useful features of UPDATE. Future versions of this program will include some of these missing features.

THIS IS A		✓
COPY		□
REPRODUCTION		□
BY DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. CODE	SPECIAL
A		

iii The following page is blank.

TABLE OF CONTENTS

	<u>PAGE</u>
1.0 INTRODUCTION	1-1
2.0 CREATING A BASE MODULE	2-1
3.0 CHANGING AND TESTING	3-1
4.0 CREATING A NEW BASE MODULE	4-1
A.0 APPENDIX - LISTING OF THE MEDIT PROGRAM	A-1

LIST OF FIGURES

<u>Figure</u>	<u>PAGE</u>
2.1 First Part of an Edit Command Deck to Create a Base Module .	2-2
2.2 First Page of MEDIT Output when Base Module is Created . . .	2-3
2.3 Job Control to Create a Base Module	2-4
3.1 Sample Edit Command Deck	3-2
3.2 MEDIT Output from Sample Edit Command Deck	3-3
3.3 Job Control to Edit a Module	3-6
4.1 Job Control to Update Base Module	4-2

1.0 INTRODUCTION

Basically, there are two techniques of developing a computer program. The first technique is to maintain a deck of the program's source statements and make changes to the program by manual insertion, removal, or replacement of cards in this deck. This manual technique is adequate if the number of cards in the deck is small, say less than 500. When the source deck becomes large, or several people are making simultaneous changes, this manual technique becomes inadequate: every computer run involves handling a large number of cards; no audit trail is kept of the changes that have been made; and locating the correct cards to change is difficult, time consuming, and prone to error.

The second technique is to use a computer program called an editor to make the desired changes. The source deck is stored on disk or tape so that large numbers of cards do not have to be handled. The source cards are numbered so that location of a certain card is easy and natural. The changes, and only the changes, are punched on cards. These cards serve as a record of the changes that have been made. These editor programs usually permit the separation of the entire source program into independent subunits (for example, a subroutine or function) and permit reference to one or more of these subunits without having to refer to the others. The complete source program is called a library and each of the subunits is called a module.

The computer program MEDIT is designed for the maintenance of a large library consisting of one or more modules. The use of MEDIT proceeds in several steps. The first step, described in Chapter 2, is to use MEDIT to create a base module of the original source decks. This base module consists of a tape or disk file with 80 characters per record (card image) and with sequence numbers in columns 77 through 80. The second step, described in Chapter 3, is to make and test changes to this base module. This consists of running MEDIT to create a test module consisting of the base module as altered by edit commands, then compiling and executing this test module. Once the changes have proven successful, they can be made permanent by creating a new base module as described in Chapter 4. Creating a new base module should be avoided if at all possible. The time to create a new base module is either:

- a. When production runs of the test version are nearly completed and the user is very sure that no additional changes will be required; or
- b. When the edit command deck exceeds 1000 cards, approximately one-half a box of cards.

The base module is a sequential file of card images with sequence numbers in columns 77 - 80. Columns 1 - 72 contain data or source statements. This file can be used as the source for a compiler. An MEDIT run produces a module called a test module that reflects every change to the base module and is suitable for compilation. Columns 73 - 76 are not used in the base module. These columns are used in the test module.

The input to MEDIT consists of control cards that give a detailed description of the changes to be made to the base module. A MEDIT run provides a listing of all the control cards and a detailed listing of all the changes made on this run. The first control card serves as a title that is printed at the top of every page of the listing.

The format for an edit command is:

Columns 1 - 2:

Two characters that identify this card as an edit command. While the two characters .. are used in this report, any other combination of two characters may be used. Examples are /. and ./ and \$\$ and **. Do not mix the characters in any one MEDIT run -- use the same characters throughout.

Column 3:

One character that identifies the type of MEDIT command. Chapter 3 lists the valid MEDIT commands and their meanings.

Columns 4 - k:

One or two decimal numbers. A comma separates the numbers if there are two numbers.

Columns k+1-72:

See a description of the E-Command. Placing comments here that describe the change being made is highly recommended.

Columns 73 - 80:

Not used by MEDIT and not listed.

Any card that is not a MEDIT command, i.e., any card whose first two characters disagree with columns 1 - 2 of an edit command, is treated as

a data card to be inserted into the test module. The placement of inserted cards is determined by the latest preceeding edit command.

1-3 The following page is blank.

2.0 CREATING A BASE MODULE

Before a module can be updated, it must be created. The program MEDIT is written to make library creation easy. A module is created by reading the deck of cards in the desired order. Figure 2.1 shows a sample edit command deck to create a base module. Figure 2.2 shows the output from MEDIT that results from these edit command cards.

The base module is a copy of the control deck except that the cards in each deck are numbered sequentially in columns 77 - 80 for later reference. If this base module is then compiled, a complete listing of the base module together with sequence numbers will be obtained. This listing should be kept as a reference for as long as the base module is kept. The binary deck output from this compilation may also be saved.

In summary, the MEDIT run to create a library should result in:

- a. A base module with sequence numbers;
- b. A listing of the contents of the base module; and
- c. The compiled version of the base module.

The various parts of Figure 2.3 show the job control cards to create a base module without compilation at each installation where MEDIT has been implemented.

```

C.... MEDIT VERSION TWO MEDIT VERSION TWO MEDIT VERSION TWO
C.- UNIT USAGE ENVIRONMENT TYPE
C.- 1 BASE (SOURCE TO BE EDITED) F(80) TAPE IN
C- 2 TEST (RESULT OF EDITING BASE) F(80) TAPE OUT
C 5 CMND (EDIT COMMAND CARDS) F(80) CARD IN
C.... 6 LIST (RESULTS OF EDIT) FA(85) PRINT
C.- FIRST CMND CARD IS USED AS A TITLE
C- (PRINTED AT TOP OF EVERY PAGE OF LIST)
C- FIRST TWO COLUMNS OF FIRST CMND CARD IDENTIFY EDIT COMMANDS
C ANY CMND CARD NOT A VALID EDIT COMMAND IS INSERTED IN TEST
C.... THIRD COLUMN OF EACH EDIT COMMAND CARD IS TYPE
C.- VALID EDIT TYPES ARE-
C- +N INSERT AFTER CARD NUMBER N
C- -N DELETE CARD N
C -N,M DELETE CARDS N THROUGH M INCLUSIVE
C.... CN MAKE CARD N A COMMENT CARD
C- CN,M MAKE CARDS N THROUGH M COMMENT CARDS
C- UN UNCOMMENT CARD N
C- UN,M UNCOMMENT CARDS N THROUGH M INCLUSIVE
C ENXAXBX EDIT CARD N
C EN,MXAXBX EDIT CARDS N THROUGH M INCLUSIVE
C
C SWITCH THAT INDICATES ENDFILE ON BASE
LOGICAL*1 ENDSW
C DECLARE ALL OTHER VARIABLES TO BE TYPE INTEGER
INTEGER*2 NTYPE,TYPE(5)
INTEGER*2 CMND(72),TITLE(72),OLD(72),MED(6)
INTEGER*2 OLD,NEW,PAGE,LINE,MAXLIN,NERRS,CHAR(2),THIS,NCOL
INTEGER*2 SPACE,COMMA,DIGIT(10)
INTEGER*2 NUM,I
C***** FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****
INTEGER*2 NEWCD(73),LOLD,POLD,LNEW,RNEW,ICOL,JCOL,KCOL
C***** PRECEDING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****
C DATA INITIALIZATION
DATA SPACE/1H /,COMMA/1H /,MED/1HM,1HE,1HD,1HI,1HT,1H /
DATA DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
DATA OLD/0/,NEW/0/,NERRS/0/,PAGE/0/
DATA LINE/50/,MAXLIN/50/
DATA NTYPE/5/,TYPE/1H-,1H.,1HC,1HU,1HE/,ENDSW/.FALSE./
C FORMAT FOR BASE AND INSERT TO TEST
1000 FORMAT(72A1,4X,I4)
C FORMAT FOR TEST
2000 FORMAT(72A1,2I4)
C FORMAT FOR CMND
5000 FORMAT(80A1)
C FORMAT FOR TITLE LINE
6001 FORMAT(1H1,72A1,4X,4HPAGE,I4/)
C FORMAT FOR COPY ACTION (TYPE -, C, U, OR E)
6002 FORMAT(1X,A1,1X,72A1,2I4,1X,A1)
C FORMAT FOR CMND CARD AND INSERT
6003 FORMAT(1X,A1,1X,72A1,4X,I4,1X,A1)
C ERROR MESSAGE FORMATS
6101 FORMAT(6H ERR #,I4,1X,18(1H*),20H INVALID EDIT TYPE. ,36(1H*))
6102 FORMAT(6H ERR #,I4,1X,18(1H*),26H COMMAND OUT OF SEQUENCE. ,
1 30(1H*))

```

Figure 2.1 First Part of an Edit Command Deck to Create a Base Module

C....	MEDIT VERSION TWO	MEDIT VERSION TWO	MEDIT VERSION TWO	PAGE	1
+ C....	MEDIT VERSION TWO	MEDIT VERSION TWO	MEDIT VERSION TWO	1	+
+ C.-	UNIT USAGE		ENVIRONMENT TYPE	2	+
+ C.-	1 BASE (SOURCE TO BE EDITED)	F(R0)	TAPE IN	3	+
+ C-	2 TEST (RESULT OF EDITING BASE)	F(R0)	TAPE OUT	4	+
+ C	5 CMND (EDIT COMMAND CARDS)	F(R0)	CARD IN	5	+
+ C....	6 LIST (RESULTS OF EDIT)	FA(85)	PRINT	6	+
+ C.-	FIRST CMND CARD IS USED AS A TITLE			7	+
+ C.-	(PRINTED AT TOP OF EVERY PAGE OF LIST)			8	+
+ C-	FIRST TWO COLUMNS OF FIRST CMND CARD IDENTIFY EDIT COMMANDS			9	+
+ C	ANY CMND CARD NOT A VALID EDIT COMMAND IS INSERTED IN TEST			10	+
+ C....	THIRD COLUMN OF EACH EDIT COMMAND CARD IS TYPE			11	+
+ C.-	VALID EDIT TYPES ARE-			12	+
+ C.-	+N INSERT AFTER CARD NUMBER N			13	+
+ C-	-N DELETE CARD N			14	+
+ C	-N,M DELETE CARDS N THROUGH M INCLUSIVE			15	+
+ C....	CN MAKE CARD N A COMMENT CARD			16	+
+ C.-	CN,M MAKE CARDS N THROUGH M COMMENT CARDS			17	+
+ C.-	UN UNCOMMENT CARD N			18	+
+ C-	UN,M UNCOMMENT CARDS N THROUGH M INCLUSIVE			19	+
+ C	ENXAXBX EDIT CARD N			20	+
+ C	EN,MXAXBX EDIT CARDS N THROUGH M INCLUSIVE			21	+
+ C				22	+
+ C	SWITCH THAT INDICATES ENDFILE ON BASE			23	+
+ C	LOGICAL*1 END5W			24	+
+ C	DECLARE ALL OTHER VARIABLES TO BE TYPE INTEGER			25	+
+ C	INTEGER*2 NTYPE,TYPE(5)			26	+
+ C	INTEGER*2 CMND(72),TITLE(72),OLD(72),MED(6)			27	+
+ C	INTEGER*2 OLD,NEW,PAGE,LIN,MAXLIN,NERRS,CHAR(2),THIS,NCOL			28	+
+ C	INTEGER*2 SPACE,COMMA,DIGIT(10)			29	+
+ C	INTEGER*2 NUM,I			30	+
+ C*****	FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****			31	+
+ C	INTEGER*2 NEWCD(73),LOLD,ROLD,LNEW,RNEW,ICOL,JCOL,KCOL			32	+
+ C*****	PRECEDING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****			33	+
+ C	DATA INITIALIZATION			34	+
+ C	DATA SPACE/1H/,COMMA/1H/,MED/1HM,1HE,1HD,1HI,1HT,1H/			35	+
+ C	DATA DIGIT/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/			36	+
+ C	DATA OLD/0/,NEW/0/,NERRS/0/,PAGE/0/			37	+
+ C	DATA LINE/50/,MAXLIN/50/			38	+
+ C	DATA NTYPE/5/,TYPE/1H-,1H-,1HC,1HL,1HE/,ENDSW/.FALSE./			39	+
+ C	FORMAT FOR BASE AND INSERT TO TEST			40	+
+ C	1000 FORMAT(72A1,4X,I4)			41	+
+ C	FORMAT FOR TEST			42	+
+ C	2000 FORMAT(72A1,2I4)			43	+
+ C	FORMAT FOR CMND			44	+
+ C	5000 FORMAT(80A1)			45	+
+ C	FORMAT FOR TITLE LINE			46	+
+ C	6001 FORMAT(1H1,72A1,4X,4HPAGE,I4/)			47	+
+ C	FORMAT FOR COPY ACTION (TYPE -. C, U, OR E)			48	+
+ C	6002 FORMAT(1X,41,1X,72A1,2I4,1X,A1)			49	+
+ C	FORMAT FOR CMND CARD AND INSERT			50	+

Figure 2.2 First Page of MEDIT Output When Base Module is Created

FIGURE 2.3 - JOB CONTROL TO CREATE A BASE MODULE

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4LTD JOB (1T01,M999),'VALID JOB CARD'
/*SETUP      DMSENN
//NEW EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE1,
//          JUNIT=2314,VOL=SER=DMSENN
//EDIT.SYSIN DD *
...
SOURCE DECK
...
/*
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//          VOL=SER=DMSENN,
//          JUNIT=2314,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
DATA
...
/*
```

AVAILABLE PROCS-

```
MEDIT - FOR MEDIT ONLY
MEDITA, MEDITAG, MEDITAL, MEDITALG - FOR ASSEMBLER-G
MEDITF, MEDITF3, MEDITFL, MEDITFLG - FOR FORTRAN-G
MEDITH, MEDITHL, MEDITHL3 - FOR FORTRAN-H
MEDITP, MEDITP3, MEDITPL, MEDITPLG - FOR PL/I
MEDITWAT - FOR WATFIV
```

```
SU=FIX G MEANS LOAD-AND-GO
SU=FIX L MEANS LINK-EDIT
SU=FIX LG MEANS LINK-EDIT AND GO
NOTE - PROCS AUTOMATICALLY USE RODMAN CAL-COMP LIBRARY.
```

PART 2 - AT USAMSSA (PENTAGON)

```
//ZXNAVERO JOB (2401,1Z07,1,20,909,1,1,P),
//          '1H VALID JOB CARD U',CLASS=0
/*ROUTE XEQ CAA-RIA
//NEW EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE1,
//          JUNIT=3330,VOL=SER=USAABC
//EDIT.SYSIN DD *
...
SOURCE DECK
...
/*
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//          VOL=SER=USAABC,
//          JUNIT=3330,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
...
DATA
...
/*
//MARSJATA DD RECORD=Z30001-76
```

AVAILABLE PROCS SAME AS AT AVSCOM.

Figure 2.3 Job Control to Create a Base Module

FIGURE 2.3 - JOB CONTROL TO CREATE A BASE MODULE (CONT)

PART 3 - AT HOPKINS (APL)

```
//AAWCXXXX JOB (6401XXXX,C,U,N),,VALID JOB CARD
//NEW EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.SOURCE.BASE1,
//      JNIT=3330,VOL=SER=JSAAWC
//EDIT.SYSIN DD *
      ...
      SOJRCE DECK
      ...
/*
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,
//      VOL=SER=USAAWC,
//      JNIT=3330,SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ...
      DATA
      ...
/*

AVAILABLE PROCS SAME AS AT AVSCOM.
```

Figure 2.3 Job Control to Create a Base Module (cont)

3.0 CHANGING AND TESTING

Once a base module has been created, changes to it may be required. The program MEDIT can be used to make these changes and produce an updated or test version of the module. This test module should be used to compile and test the changes.

The MEDIT commands to make changes to the base module are:

- | | |
|---------------------------------|---|
| a. <code>..+n</code> | to insert cards after card n |
| b. <code>..-n,m</code> | to delete cards n through m inclusive |
| c. <code>..-n</code> | same as <code>..-n,n</code> |
| d. <code>..Cn,m</code> | to place a 'C' in column 1 of cards n through m inclusive |
| e. <code>..Cn</code> | same as <code>..Cn,n</code> |
| f. <code>..Un,m</code> | to place a space in column 1 of cards n through m inclusive |
| g. <code>..Un</code> | Same as <code>..Un,n</code> |
| h. <code>..En,m!old!new!</code> | to edit test on cards n through m inclusive
(See below) |
| i. <code>..En!old!new!</code> | same as <code>..En,n!old!new!</code> |

Any card that is not an MEDIT command is treated as a card to be inserted. All card numbers refer to the sequence numbers in the base module and **must** appear in numerical order. Cards may be inserted in place of deleted cards by coding the appropriate `..-` card followed by the card(s) to be inserted. Cards can be inserted before the first card in the deck by placing them immediately after a `..+0` card at the beginning of the edit command deck. A sample to illustrate all this is given in Figure 3.1. MEDIT output from an update run is shown in Figure 3.2.

A short summary of the logic of MEDIT will serve as a detailed description of the various MEDIT commands. In the main loop of MEDIT, an input card is read. If this card is not an MEDIT command (columns 1 and 2 are tested), then this card is written to the test module. If the input card is an MEDIT command, then the appropriate action is taken as follows.

If the input card is an "insert" command (`..+n`), then the base module is copied to the test module with the base module card numbered n being the last card copied. In conjunction with the treatment of non-MEDIT

```

..+0      CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE
PROGRAM MEDIT(RASE,TEST,INPUT,OUTPUT,
1      TAPE1=RASE,TAPE2=TEST,TAPE5=INPUT,TAPE6=OUTPUT)
..-24
      LOGICAL END$W
..E26.32!INTEGER*2!INTEGER !
..E34!50!55!
..+39
C
..-65
      READ(5,5000) CHND
      IF (EOF(5).NE.0) GO TO 900
..-88
      100 READ(5,5000) CHND
      IF (EOF(5).NE.0) GO TO 900
..-248
      READ(1,1000) OLD$D,OLD
      IF (EOF(1).NE.0) GO TO 192
..-283,284
      401 READ(1,1000) OLD$D,OLD
      IF (EOF(1).EQ.0) GO TO 400

```

Figure 3.1 Sample Edit Command Deck

```

..MEDIT   CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE                                PAGE   1

  ..+0      CONVERT MEDIT VERSION 2 TO CDC-6000 SCOPE
+ PROGRAM MEDIT(BASE,TEST,INPUT,OUTPUT)                                           1 +
+ 1  TAPE1=BASE,TAPE2=TEST,TAPE5=INPUT,TAPE6=OUTPUT)                             2 +
  ..-24
- LOGICAL*1 FENDSW                                                                24 25 -
+ LOGICAL FENDSW                                                                26 +
  ..F24,32!INTEGER*2!INTEGER !
E  INTEGER *TYPE,TYPE(5)                                                         26 28 E
E  INTEGER CMND(72),TITLE(72),CLDCD(72),MED(6)                                  27 29 F
E  INTEGER OLD,NEW,PAGE,LINE,MAXLIN,NEWS,CHAR(2),THIS,NCOL                     28 30 E
E  INTEGER SPACE,COMMA,DIGIT(10)                                                29 31 F
E  INTEGER NUK,I                                                                30 32 F
E C***** FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY *****                31 33 F
E  INTEGER NENCD(73),IOLD,ROLD,LNEW,RNEW,ICOL,JCOL,KCOL                       32 34 F
  ..ERR150!55!
E  DATA LINE/55/,MAXLIN/55/                                                    38 40 E
  ..+39
+ C                                                                                42 +
  ..-65
- READ(5,5000,FID=900) CMND                                                       65 67 -
+ READ(5,5000) CMND                                                             68 +
+ IF(EOF(5).NE.0) GO TO 900                                                       69 +
  ..-PR
- 100 READ(5,5000,FID=900) CMND                                                    88 91 -
+ 100 READ(5,5000) CMND                                                           92 +
+ IF(EOF(5).NE.0) GO TO 900                                                       93 +
  ..-248
- READ(1,1000,FID=192) CLDCD,CLD                                                  248 252 -
+ READ(1,1000) CLDCD,CLD                                                         253 +
+ IF(EOF(1).NE.0) GO TO 192                                                       254 +
  ..-283,284
- 401 READ(1,1000,FID=402) CLDCD,CLD                                              283 288 -
- GO TO 400                                                                      284 288 -
+ 401 READ(1,1000) CLDCD,CLD                                                     289 +
+ IF(EOF(1).EQ.0) GO TO 400                                                       290 +

NO ERRORS ENCOUNTERED.  NORMAL END TO MEDIT.

```

Figure 3.2 MEDIT Output from Sample Edit Command Deck

commands, this will cause cards to be inserted in the appropriate position in the test module.

If the input card has any of the other forms (`..-n` or `..-n,m` or `..Cn` or `..Cn,m` or `..Un` or `..Un,m` or `..En` or `..En,m`), then the base module is copied to the test module with the base module card numbered `n-1` being the last card copied. This positions both the base and test modules for the appropriate action.

If the input card is a "delete" command (`..-n` or `..-n,m`), then the base module is read until the appropriate base module card numbered `m` has been read. Since these cards are not copied to the test module, they are deleted from the test module. In conjunction with the treatment of non-MEDIT commands, this can be used to insert cards in place of the deleted cards.

If the input card is a "comment" command (`..Cn` or `..Cn,m`), then the base module is copied to the test module with the character `C` replacing the first column of each card. This converts each of the cards to a FORTRAN comment card in the test module.

If the input card is an "uncomment" command (`..Un` or `..Un,m`), then the base module is copied to the test module with the blank character replacing the first column of each card. This converts each of the cards to a FORTRAN statement in the test module. The "uncomment" command should be used only on cards that have been "commented" when the base module was created.

If the input card is an "edit text" command (`..En` or `..En,m`), then the first character after the line number on the input card is used as a character string delimiter to determine the "old string" and "new string" on the input card. In the example "edit text" command:

`..E54,59!THIS!THOSE!`

The exclamation point (!) is the delimiter, the "old string" is the characters `THIS` and the "new string" is the characters `THOSE`. Any character except a comma may be used as a delimiter. The delimiter cannot be used in either the "old string" or the "new string."

The basic operation of the "edit text" operation is to copy cards from the base module to the test module. However, before each card is copied, it is searched for occurrences of the "old string." Every occurrence

of the "old string" is replaced by the "new string" before the card is copied. Thus, in the example "edit text" command, every occurrence of the characters THIS on cards 54 through 59 will be replaced by the characters THOSE in the test module.

Making changes to a base module and testing these changes proceeds as follows:

- a. An edit command deck is used to make changes to the base module producing a test module. Any old binary decks for this module should be purged.

- b. This test module is compiled and tested. The resulting binary deck may be saved but the test module is not.

- c. If this test module is still not correct, additional changes are added to the edit command deck and the process repeated from step a. above.

Note that the test module is not saved, only the compiled version of the test module. Any new changes will be made to the base module so that the edit command deck will serve as a detailed record of every change. Returning to the base module for every change also prevents a proliferation of versions of the program and the attendant confusion and waste of computer storage (disk or tape).

The test module which is produced by a MEDIT run has two sets of sequence numbers. One set, in columns 77 - 80, are sequential starting with 1. The other set, in columns 73 - 76, are copied from the base module. For cards that have been inserted, columns 73 - 76 are blank. This set of sequence numbers are copied from the base module to facilitate making additional changes since all edit commands refer to the sequence numbers in the base module. The listing of the test module can be used to locate additional changes to the base module and the corresponding edit command cards placed appropriately in the edit command deck.

The various parts of Figure 3.3 show the job control to make changes to a library at each installation where MEDIT has been implemented. Please note that the module of the changed source is not saved. These changes should be made for every test run until a new base module is created (Chapter 4). A new base module should not be created until all potential changes have been proven correct.

FIGURE 3.3 - JOB CONTROL TO EDIT A MODULE

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4LID JOB (1T01,M999),'A VALID JOB CARD'
//EDIT EXEC MEDITFLG
//EDIT.BASE DD DISP=OLD,DSN=YOUR.SOURCE.BASE1,
//      UNIT=2314,VOL=SER=DMSEVN
//EDIT.SYSIN DD *
..*0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
      ***
      EDIT COMMAND DECK
      ***
/*
//EDIT.PURGE DD DISP=(MOD,DELETE),DSN=YOUR.BINARY.BASE1,
//      UNIT=2314,VOL=SER=DMSEVN --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE1,
//      UNIT=2314,VOL=SER=DMSEVN,
//      SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ***
      DATA
      ***
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

PART 2 - AT USAMSSA (PENTAGON)

```
//ZXVAMERO JOB (2401,1707,1,20,999,1,1,P),
//      '1H VALID JOB CARD U',CLASS=0
/*ROJTE XEQ CAA-RIA
//EDIT EXEC MEDITFLG
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..*0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
      ***
      EDIT COMMAND DECK
      ***
/*
//EDIT.PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,
//      DISP=(MOD,DELETE) --- TO DELETE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,
//      UNIT=3330,VOL=SER=USAAWC,
//      SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ***
      DATA
      ***
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 3.3 Job Control to Edit a Module

FIGURE 3.3 - JOB CONTROL TO EDIT A MODULE (CONT)

PART 3 - AT HOPKINS (APL)

```
//AA#CNAME JOB (6401,USER,C,U,N),'VALID JOB CARD'  
//EDIT EXEC MEJITFLG  
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1  
//EDIT.SYSIN DD *  
..*0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE  
    ***  
    EDIT COMMAND DECK  
    ***  
/*  
//EDIT.PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,  
//  DISP=(OLD,DELETE) --- TO PURGE OLD BINARY  
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE1,  
//  UNIT=3330,VOL=SER=USAAWC,  
//  SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY  
//GO.SYSIN DD *  
    ***  
    DATA  
    ***  
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 3.3 Job Control to Edit a Module (cont)

4.0 CREATING A NEW BASE MODULE

After changes have been tested and proven, these changes can be made permanent by creating a base module that reflects all these changes. This is accomplished by changing the job control to save the test module and using the same MEDIT command deck that resulted in the proven test module.

The changes to the old base module are transferred to the new base module when the test module is saved. The new base module is renumbered with the new sequence numbers in columns 77 - 80.

The cost of an edit and compile run is determined by the length of the base module and not by the number of changes being made. Therefore, the creation of a new base module should be avoided whenever possible. The time to create a new base module is either:

a. When production runs of the test version are nearly completed and the user is very sure that no additional changes will be required; or

b. When the edit command deck exceeds 1000 cards, approximately one-half a box of cards.

When a new base module is created, the listing and binary (compiled) deck should also be kept as a reference for as long as this base module is kept. After the new base module has been successfully created and compiled, the old base module should be purged together with its listing and binary (compiled) deck.

The various parts of Figure 4.1 give the job control for updating a library at the various installations where MEDIT has been implemented.

FIGURE 4.1 - JOB CONTROL TO UPDATE BASE MODULE.

PART 1 - AT USA AVSCOM, ST. LOUIS

```
//MV4LID JOB (1T01,M999),'A VALID JOB CARD'
/*SETUP      DMSENN
//EDIT EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,KEEP),DSN=YOUR.SOURCE.BASE2,
//      UNIT=2314,VOL=SER=DMSENN,SPACE=(TRK,(1,1),RLSE)
//EDIT.BASE DD DISP=OLD,DSN=YOUR.SOURCE.BASE1,
//      UNIT=2314,VOL=SER=DMSENN
//EDIT.SYSIN DD *
..*0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
      ...
      EDIT COMMAND DECK
      ...
/*
//EDIT.PURGE DD DISP=(MOD,DELETE),DSN=YOUR.BINARY.BASE1,
//      UNIT=2314,VOL=SER=DMSENN --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,KEEP),DSN=YOUR.BINARY.BASE2,
//      UNIT=2314,VOL=SER=DMSENN,
//      SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ...
      DATA
      ...
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

PART 2 - AT USAMSSA (PENTAGON)

```
//ZXVAMERO JOB (2401,1Z07,1,20,999,1,1,P),
//      '1H VALID JOB CARD U',CLASS=0
/*ROJTE XEQ CAA-RIA
//EDIT EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.SOURCE.BASE2,
//      UNIT=3330,VOL=SER=USAAWC,SPACE=(TRK,(1,1),RLSE)
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..*0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
      ...
      EDIT COMMAND DECK
      ...
/*
//EDIT.PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,
//      DISP=(MOD,DELETE) --- TO DELETE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE2,
//      UNIT=3330,VOL=SER=USAAWC,
//      SPACE=(TRK,(1,1),RLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ...
      DATA
      ...
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 4.1 Job Control to Update Base Module

FIGURE 4.1 - JOB CONTROL TO UPDATE BASE MODULE (CONT).

PART 3 - AT HOPKINS (APL)

```
//AAWCNAME JOR (6401,USER,C,U,N),'VALID JOB CARD'
//EDIT EXEC MEDITFLG
//EDIT.TEST DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.SOURCE.BASE2,
//      UNIT=3330,VOL=SER=USAAWC,SPACE=(TRK,(1,1),RLSE)
//EDIT.BASE DD DISP=OLD,DSN=AAWC.YOUR.SOURCE.BASE1
//EDIT.SYSIN DD *
..+0      TITLE CARD - APPEARS AT TOP OF EVERY PAGE
      ...
      EDIT COMMAND DECK
      ...
/*
//EDIT.PURGE DD DSN=AAWC.YOUR.BINARY.BASE1,
//      DISP=(OLD,DELETE) --- TO PURGE OLD BINARY
//FORT.SYSLIN DD DISP=(NEW,CATLG),DSN=AAWC.YOUR.BINARY.BASE2,
//      UNIT=3330,VOL=SER=USAAWC,
//      SPACE=(TRK,(1,1),PLSE) --- TO SAVE BINARY
//GO.SYSIN DD *
      ...
      DATA
      ...
/*
```

PROCS SAME AS AT AVSCOM. PROCS LISTED IN FIG. 2.3.

Figure 4.1 Job Control to Update Base Module (cont)

NOTE: This is a listing of the CDC-6000 version of MEDIT. This source was the result of the editing described in Figures 3,1 and 3,2,

A-1


```

115 LINE=THE+1
    GO TO 101
    C CMD CMD IS A VALID EDIT COMMAND
    C EVALUATE FIRST NUMBER
120 104 ASSIGN 107 TO IXT
    NCOL=3
    GO TO 100
    C IS CMD IN SEQUENCE
    C 107 IF (V.A.G.CLC) GO TO 108
    C NO. SIGNAL ERROR AND RESET NUM
    NREDS=NRDS+1
    WRITE(A,BL2) NRDS
    LINE=THE+1
    GO TO 100
    C COPY FROM BASE TO TEST
    C 108 ASSIGN 104 TO IXT
    C GET NUM IF THIS IS AN INSERT COMMAND.
    IF (I=1) THEN N=NUM+1
    GO TO 100
    C IS THIS COMMAND AN INSERT
    C 109 IF (I=1) GO TO 100
    C IF IT IS NOT INSERT
    C 110 IF (I=2) GO TO 111
    C YES. EVALUATE IT
    ASSIGN 111 TO IXT
    GO TO 100
    C LOOP TO ALTER BASE
    C 111 ASSIGN 100 TO IXT
    C IF (I=1) THEN ASSIGN 101 TO IXT
    C ***** FOLLOWING LOGIC IS FOR TEXT EDIT FEATURE ONLY
    C GET NCOL SWITCH IN CASE NOT AN EDIT COMMAND.
    NCOL=1
    C IS TYPE AN EDIT
    C IF (I=1) GO TO 112
    C YES. GET OLD AND NEW STRING LIMITS
    LOI=NCOL+1
    DO 142 PULL=OLD+72
    IF (PULL=OLD) -FO.CMD(NCOL) GO TO 143
    C 142 CONTINUE
    C INVALID DELIMITED. SIGNAL ERROR AND IGNORE CMD
    NREDS=NRDS+1
    WRITE(A,BL1) NRDS+NCOL.CMD(NCOL)
    LINE=THE+1
    GO TO 100
    C IS OLD STRING A NULL STRING
    C 143 IF (PULL=OLD) GO TO 144
    C YES. SIGNAL ERROR AND IGNORE
    NREDS=NRDS+1
    WRITE(A,BL2) NRDS
    LINE=THE+1
    GO TO 100
    C 144 UNFOLD-1
    UNFOLD-1
    DO 145 MAKE=IF+72
    IF (CMD(NCOL) -FO.CMD(NCOL)) GO TO 146

```

A-4

SYNAPTIC RFFEFUF:CF MIP (N=3)

A-6

PROGRAM - FMT	76/74	REF LINE	REFERENCES
STATEMENT LABELS			
10314 104	114	110	276
10320 107	123	119	
10330 104	130	123	
10334 109	135	130	293
0 110	134		
10343 111	144	138	140
10423 112	141	150	255
10440 120	142	143	
10442 130	142	143	
10444 140	144	143	
0 142	155	153	
10371 143	152	154	
10401 144	150	162	
0 145	174	170	
10421 144	178	171	
10455 151	194	234	
10444 152	207	202	
10444 153	206	200	205
10502 154	215	214	
10503 152	220	212	217
10505 154	222	211	
10507 157	222	204	207
10517 158	232	227	
10522 159	234	223	230
10547 160	251	246	
0 161	231	236	
0 162	195	194	
10530 160	241	144	267
10573 161	244	145	267
10554 162	257	254	
10554 200	240	240	106
10565 201	240	241	141
10567 300	255	279	
10572 301	270	272	
0 302	274	273	
10601 303	274	273	
10604 400	274	273	
10613 401	274	273	
0 402	274	273	
10621 403	274	273	
10623 404	274	273	
10624 701	274	273	
10627 602	274	273	
10624 1000	274	273	
10627 2000	274	273	
10641 3000	274	273	
10641 6001	274	273	
10647 6002	274	273	
10673 6003	274	273	
10677 6101	274	273	
10704 6102	274	273	
10715 6141	274	273	
10727 6142	274	273	
10737 6143	274	273	
10751 6150	274	273	
11122 6901	274	273	
11132 6902	274	273	

PROGRAM MERIT			74/74	CPT#2	PROPERTIES	
LOC#	LABEL	INDEX	PCOM-TJ	LENGTH		
10224	1	• KCOL	71 72	24	INSTACK	EXITS
10245	2	• T-15	74 76	24	INSTACK	EXITS
10247	4	• KCOL	87 84	24	INSTACK	EXITS
10304	105	• T-15	105 111	24	INSTACK	EXITS
10357	142	• KCOL	152 155	24	INSTACK	EXITS
10407	145	• KCOL	170 172	24	INSTACK	EXITS
10444	149	• I	184 195	24	INSTACK	EXITS
10440	153	• KCOL	200 208	74	INSTACK	EXITS
10474	155	• KCOL	212 220	104	CPT	
10525	161	• KCOL	234 237	34	INSTACK	EXITS
10575	302	• I	272 274	24	INSTACK	EXITS
STATISTICS						
PROGRAM LENGTH			14444	804		
BUFFER LENGTH			102059	4229		

A. Dept. of the Defense

Defense Communication Engineering Office
ATTN: Mr. Roy Rosner (1)
Reston, VA

Foreign Science and Technology Center
ATTN: Mr. Reg Barta (1)
ATTN: Mr. David Howell (1)
ATTN: Mr. Thomas Pruden (1)
Charlottesville, VA

Defense Documentation Center
ATTN: TIPCR (12)
Cameron Station
Alexandria, VA 22314

B. Dept. of the Army

Commander
US Army Electronic Command
Night Vision Laboratory
ATTN: Mr. John Dehne (1)
ATTN: Mr. R. A. Oswalt (1)
Ft. Monmouth, NJ 07703

Commander
US Army Materiel and Mechanics Research Center
ATTN: AMXMR-S, Mr. R. J. Geromini (1)
ATTN: AMXMR-S, Mr. D. M. Gracia (1)
Watertown, MA 02172

Commander
US Army Ames Research Center
Air Mobility Research & Development Laboratory
ATTN: SAVDL-AS, Dr. J. Hwang (1)
ATTN: SAVDL-D, Mr. Mike Kodani (1)
Moffett Field, CA 94035

Commander
US Army Test & Evaluation Command
ATTN: AMSTE, Mr. Leon Brill (1)
Aberdeen Proving Ground, MD 21005

Commander
US Army Electronic Command
ATTN: AMSEL-SA-L, Mr. Richard Caccamise (1)
ATTN: AMSEL-IO, Mr. Henry Chambers (1)
ATTN: AMSEL-IO, Mr. Thomas Dames (2)
ATTN: AMSEL-IO, Mr. W. R. King (1)
ATTN: AMSEL-IO, Mr. E. Oltarzewski (1)
Ft. Monmouth, NJ 07703

Commander
US Army Missile Command
ATTN: AMCMI, Mr. Jim Collins (2)
ATTN: AMCMI, Mr. James Williams (1)
Redstone Arsenal, AL 35809

US Army Aberdeen Proving Ground
ATTN: Mr. J. Whallon (1)
Aberdeen Proving Ground, MD 21005

Commander
Picatinny Arsenal
ATTN: SARPA, Mr. A.G. Edwards (1)
ATTN: SARPA-MIS, Mr. David
Grobstein (2)
Dover, NJ 07801

Commander
Edgewood Arsenal
ATTN: SAREA, Mr. S. Goldberg (2)
Edgewood, MD 21005

Commander
US Army Aviation Systems Command
ATTN: Mr. Paul Lascala (2)
ATTN: Dr. I. Peterson (1)
St. Louis, MO 63166

Commander
Watervliet Arsenal
ATTN: SARWV, Mr. James Pascale (2)
Watervliet, NY 12189

Commander
Yuma Proving Ground
ATTN: Mr. Fred Jones (1)
Yuma, AZ 85364

Director
US Army Mobility Research &
Development Center
ATTN: STSFB, Mr. H. Bero (1)
ATTN: STSFB, Mr. Sam McCutchen (2)
Fort Belvoir, VA 22060

Commander

US Army Harry Diamond Laboratories

ATTN: AMXDO, Mr. Hubert Matthews (1)

ATTN: AMXDO, Ms. L. Jean Marroletti (1)

Washington, DC 20438

Director

US Army Ballistic Research Laboratory

ATTN: Mr. Sam Taylor (1)

Aberdeen, MD 21005

Commander

US Army Materiel Development and Readiness Command

ATTN: DRCMS, Mr. John Cianflone (1)

ATTN: DRCMS, Mr. John Gilbert (1)

ATTN: DRCMS, Dr. Ronald Uhlig (1)

5001 Eisenhower Avenue

Alexandria, VA 22333

Commander

US Army Armament Command

ATTN: DRSAR-SA, Mr. Stu Olson (1)

ATTN: DRSAR-MSE (1)

ATTN: DRSAR-MST (1)

ATTN: DRSAR-JCAP-E (1)

ATTN: DRSAR-JCAP-M (1)

Rock Island, IL 61201

Commander

Rock Island Arsenal

ATTN: SARRI-L (1)

ATTN: SARRI-LPL (2)

ATTN: SARRI-LA (5)

ATTN: SARRI-LE (5)

ATTN: SARRI-LR (10)

ATTN: SARRI-LS (5)

ATTN: SARRI-LW (5)

ATTN: SARRI-LR-S, Dr. Hurt (10)

ATTN: SARRI-R (1)

Rock Island, IL 61201

Commander

White Sands Missile Range

ATTN: Mr. James Field (1)

White Sands Missile Range, NM 88002

DISTRIBUTION LIST UPDATE

- - - FOR YOUR CONVENIENCE - - -

Government regulations require the maintenance of up-to-date distribution lists for technical reports. This form is provided for your convenience to indicate necessary changes or corrections.

If a change in our mailing lists should be made, please check the appropriate boxes below. For changes or corrections, show old address *exactly* as it appeared on the mailing label. Fold on dotted lines, tape or staple the lower edge together, and mail.

☐ Remove Name From List

☐ Change or Correct Address

Old Address:

Corrected or New Address:

COMMENTS

Date: _____ Signature: _____

Technical Report # R-TR-76-010

B-3

FOLD HERE

Return Address:

POSTAGE AND FEES PAID
DEPARTMENT OF THE ARMY
DOD 314



OFFICIAL BUSINESS
Penalty for Private Use \$300

Commander
Rock Island Arsenal
Attn: SARRI-LR-S
Rock Island, Illinois 61201

FOLD HERE

AD
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TR-76-010
27 Pages, Incl Figures

UNCLASSIFIED
1. Software Maintenance
2. ADP Utilities
I. James J. Hurt
II. Rock Island Arsenal
III. Research Directorate
General Thomas J. Rodman Laboratory
Rock Island Arsenal

DISTRIBUTION
Approved for public release;
distribution unlimited.

A portable and easily used source text editor is described.
This computer utility provides a method to maintain and
modify computer source text in a manner that is very easy to
use, provides a complete audit trail of changes, and can
be used on a wide variety of computers, eg IBM-360-370,
CDC-6000, UNIVAC-1100, et al.

AD
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TR-76-010
27 Pages, Incl Figures

UNCLASSIFIED
1. Software Maintenance
2. ADP Utilities
I. James J. Hurt
II. Rock Island Arsenal
III. Research Directorate
General Thomas J. Rodman Laboratory
Rock Island Arsenal

DISTRIBUTION
Approved for public release;
distribution unlimited.

A portable and easily used source text editor is described.
This computer utility provides a method to maintain and
modify computer source text in a manner that is very easy to
use, provides a complete audit trail of changes, and can
be used on a wide variety of computers, eg IBM-360-370,
CDC-6000, UNIVAC-1100, et al.

AD
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TR-76-010
27 Pages, Incl Figures

UNCLASSIFIED
1. Software Maintenance
2. ADP Utilities
I. James J. Hurt
II. Rock Island Arsenal
III. Research Directorate
General Thomas J. Rodman Laboratory
Rock Island Arsenal

DISTRIBUTION
Approved for public release;
distribution unlimited.

A portable and easily used source text editor is described.
This computer utility provides a method to maintain and
modify computer source text in a manner that is very easy to
use, provides a complete audit trail of changes, and can
be used on a wide variety of computers, eg IBM-360-370,
CDC-6000, UNIVAC-1100, et al.

AD
Research Directorate, General Thomas J. Rodman Laboratory
Rock Island Arsenal, Rock Island, Illinois 61201
MEDIT - A Program to Edit Computer Source Programs
Prepared by: James J. Hurt
Security Class. (of this report): Unclassified
Technical Report R-TR-76-010
27 Pages, Incl Figures

UNCLASSIFIED
1. Software Maintenance
2. ADP Utilities
I. James J. Hurt
II. Rock Island Arsenal
III. Research Directorate
General Thomas J. Rodman Laboratory
Rock Island Arsenal

DISTRIBUTION
Approved for public release;
distribution unlimited.

A portable and easily used source text editor is described.
This computer utility provides a method to maintain and
modify computer source text in a manner that is very easy to
use, provides a complete audit trail of changes, and can
be used on a wide variety of computers, eg IBM-360-370,
CDC-6000, UNIVAC-1100, et al.