UR1986_50

1986/50. AID - auto-indexed documentation for the Perkin-Elmer 3200 minicomputer.

E. L. Martin

Abstract

AID is a help facility which makes use of existing system help facilities and extends them to provide documentation for user accounts. It is simple to use and provides different levels of help.

INTRODUCTION

Several help files already exist on the system account: some of these explain MTM commands, others provide help within programs such as EDIT and COPY. Althought the structure of these existing files is not identical, it is sufficiently similar to form the basis for a universal integrated system which can be expanded as required.

HELP FILE STRUCTURE

Help files consist of 80 character/record text files, in which individual help text messages are subdivided by some form of delimiter or convention, and larger blocks of text are split into screen-sized parts by another delimiter, thus enabling each part to be read in turn.

The help texts need not be in alphabetical order, but this may assist when updating files.

The help texts are preceded by header records (the index). The first of these contains the size (number of records) of the header and is followed by an indexed list of entries – each containing the position within the file of the first record of the entry and the number of records in the entry. All references to line numbers are in hexadecimal notation.

USING AID

AID operates at three levels in such a way that the user can either go directly to the information required, or if the file or keyword is unknown then a list of files or keywords can be obtained.

The first level is entered by typing AID (or simply A). A list of help files will be displayed, together with a note that some user accounts may contain additional files (PROGRAMS, JOBS, NOTES).

At the second level the user types AID (or A) followed by the name of the file required (which may be abbreviated to a single letter – e.g. A S). A list of keywords is then displayed. A user signed on to a private account who wishes to examine files in the group account should add /G immediately following the file name – e.g. A N/G .

At the third level the user adds the keyword, which may be abbreviated to the extent that it remains distinguishable from other keywords in the help file – i.e. A S N is equivalent to AID SYSCSS NONPRINT.

If the help text will not fit on one screen the user will be prompted to press RETURN (or ENTER) to get the next screen.

USER HELP FILES

At present provision is made for any user account to have up to three additional help files - PROGRAMS, NOTES, JOBS. There are programs to assist in the indexing and updating of user help files.

Creating a user help file

User help file descriptors must be PROGRAMS.HLP, JOBS.HLP or NOTES.HLP, no others will be recognised by this version of AID.

First allocate the help file, e.g. AL PROGRAMS.HLP,IN,80 and then create text using EDIT or LEX. If LEX is used, first delete any ruler before saving the file.

For each help entry begin by typing the keyword at the left-hand margin of the first line. Only the first 16 characters of the keyword are included in the index: they should be typed in capital letters. The following text should be indented by at least one character which enables the indexing program to distinguish between keywords and text. Repeat this procedure for each subsequent entry.

Indexing a user help file

Once the help file has been created (or amended) type MAKEHELP fd, where fd is the file descriptor of the help file. Once this is done AID will be able to access you file.

Amending a help file.

Before amending a help file the header records (index) must be removed as if they are listed to the screen during editing it is quite possible to lock the keyboard, affect the scrolling or produce other unwelcome effects.

To remove the index type REMINDEX fd, where fd is the file descriptor of the input file. If, in error, an attempt is made to remove an index from an un-indexed file then that file will be unchanged.

DISCUSSION

Adequate documentation is an important part of any computer system and is particularly important in a multi-user system with large numbers of programs. The AID system is a relatively painless way of providing such documentation in a controlled form so that users unfamiliar with a particular program or function can get on-line assistance. Examples of help files are included in Appendix 1 and program listings in Appendices 2-4.

[20 August 1986]

APPENDIX 1

Examples of AID displays:

Input by the user is preceded by an asterisk.

```
*A

Please specify which help file you require

System help files are:

BACKUP          COPY            DISPLAY       EDIT          FLIST
LINK            MTM             SYSCSS

SYCSS contains details of programs on the system account.

Private and group accounts may have their own help files -

PROGRAMS        JOBS            NOTES

CALL is: AID file keyword  -  where file is one of the above
         and keyword is the keyword or command you wish to list.
If signed on to a private account and want details from a group
help file add /G immediately after file, e.g. A P/G

AID may be abbreviated to A, files may be abbreviated to the
initial letter and keywords may be abbreviated to the extent that
they may be distinguished from one another. You may type in upper or
lower case.

*A M

 ADD             ALLOCATE        ASSIGN          BFILE           BIAS
 BREAK           BRECORD         BUILD           CAL             CANCEL
 CLOSE           COBOL           COMMAND         COMPILE         COMPLINK
 CONTINUE        DELETE          DISPLAY         EDIT            ENABLE
 ENDB            ENV             EXAMINE         EXEC            FFILE
 FILEDESC        FORT            FORTO           FORTZ           FRECORD
 HELP            INIT            INQUIRE         LINK            LIST
 LOAD            LOG             MACRO           MESSAGE         MODIFY
 OPTIONS         PASCAL          PASSWORD        PAUSE           PREVENT
 PRINT           PUNCH           PURGE           RELEASE         REMOVE
 RENAME          REPROTECT       REWIND          RPG             RUN
 RVOLUME         RW              SEND            SET             SIGNOFF
 SIGNON          SPOOLFILE       SSYS            START           SUBENV
 SUBMIT          TASK            TEMPFILE        VOLUME          WFILE
 XALLOCATE       XDELETE

*A P/G

 AID             DELTA           DIRSYS          GRAPH           INTERCAL
 LEX             OUP             PLOTDATA        PLOTHEAD        PPR
 PRINTT          SEARCH          SETUP           SSRCH           SORTBIB
 TTS             TXT             UNFOLD1
```

*A P GRAPH

GRAPH

Used to produce X-Y plot file.

Format: GRAPH fd[,dev:]

- where fd is the GRAPH-format input file, and
dev: is the listing device (default = CON:).

Example:

GRAPH AFM.GRA Input file AFM.GRA, listing to CON:

Notes: Output file is TEMP.PLT - use a plotting utility (e.g. A3PLOT to produce plot. If not plotting immediately then rename TEMP.PLT (it will be rewritten when GRAPH or GRAPH is next run.

For documentation see file SYS3:HLPGRA.DOC/70. IF ON publications account type HELP GRAPH

APPENDIX 2

AID program listing.

```
*  AID.CSS
$IFNULL
ease specify which help file you require; $WR
$WR System help files are:; $WR
$WR BACKUP          COPY           DISPLAY      EDIT         FLIST
$WR LINK            MTM            SYSCSS;$WR

$WR SYCSS contains details of programs on the system account; $WR
$WR Private and group accounts may have their own help files -
$WR; $WR PROGRAMS       JOBS           NOTES; $WR
$WR CALL is: AID file keyword  -  where file is one of the above
$WR .        and keyword is the keyword or command you wish to list.
$WR If signed on to a private account and want details from a group
$WR help file add /G immediately after file, e.g. A P/G; $WR
$WR AID may be abbreviated to A, files may be abbreviated to the
$WR initial letter and keywords may be abbreviated to the extent that
$WR they may be distinguished from one another. You may type in upper or
$WR lower case.
$EXIT; $ENDC
$BUILD AID.TMP
@1 @2
$ENDB
$IFX PROGRAMS.HLP
$BUILD AID.TMP,A
P
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
$IFX PROGRAMS.HLP/G
$BUILD AID.TMP,A
P
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
$IFX JOBS.HLP
$BUILD AID.TMP,A
J
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
$IFX JOBS.HLP/G
$BUILD AID.TMP,A
J
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
```

```
$IFX NOTES.HLP
$BUILD AID.TMP,A
N
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
$IFX NOTES.HLP/G
$BUILD AID.TMP,A
N
$ENDB
$ELSE
$BUILD AID.TMP,A

$ENDB; $ENDC
$WR; PRE ETM
L AID/P,5; AS 2, AID.TMP; AS 3,CON:; AS 5, CON:; ST
             ENA ETM; $EXIT

$TITL AID.FTN   02-APR-1986  (REVD 20-AUG-86)  E.L.MARTIN
C     A general purpose HELP program which can read several help files
C     on SYSF:-/S or as enabled by AID.CSS
      CHARACTER*4096 CONTENTS
      CHARACTER*80 ALINE,KLINE
      CHARACTER*32 FILE,FNAME
      CHARACTER*16 KEYWORD
      CHARACTER*1 P1,P2,P3,P4,P5,P6
      LOGICAL FOUND
      CONTENTS=' '
      FOUND=.FALSE.
      READ(2,901)KLINE
  901 FORMAT(A80)
C     Make sure keyword is in UPPER CASE
      DO 100, I=1,16
      IF(KLINE(I:I) .GE. CHAR(97) .AND. KLINE(I:I) .LE. CHAR(122))
     .KLINE(I:I)=CHAR(ICHAR(KLINE(I:I))-32)
  100 CONTINUE
      FILE=' '
      KEYWORD=' '
C     Get name of help file input as @1
      N=INDEX(KLINE,' ')
      M=INDEX(KLINE,'/')
      FILE=KLINE(1:N-1)
C     Now move on to get keyword input as @2
      KEYWORD=KLINE(N+1:16)
C     Get length of keyword
      KL=INDEX(KEYWORD,' ')-1
      READ(2,902)P1
      READ(2,902)P2
      READ(2,902)P3
      READ(2,902)P4
      READ(2,902)P5
      READ(2,902)P6
C     Check file descriptor
      IF(FILE(1:1) .EQ. 'B') THEN
        FNAME='SYSF:BACKUP.HLP/S'
        FOUND=.TRUE.
```

```
      ENDIF
      IF(FILE(1:1) .EQ. 'C') THEN
        FNAME='SYSF:COPY32.HLP/S'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'D') THEN
        FNAME='SYSF:DISPMTM.HLP/S'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'E') THEN
        FNAME='SYSF:EDIT32.HLP/S'
        FOUND = .TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'F') THEN
        FNAME='SYSF:FLIST.HLP/S'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'L') THEN
        FNAME='SYSF:LINK0103.HLP/S'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'M') THEN
        FNAME='SYSF:SINGMTM.HLP/S'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'S') THEN
        FNAME='SYSF:SYSCSS.HLP/S'
        FOUND=.TRUE.
      ENDIF
C     Check for files in private/group account
      IF(FILE(1:1) .EQ. 'P' .AND. P1 .EQ. 'P' .AND. M .EQ. 0)THEN
        FNAME='PROGRAMS.HLP'
        FOUND=.TRUE.
      ELSE IF(FILE(1:1) .EQ. 'P' .AND. P2 .EQ. 'P' .AND. M .NE. 0)THEN
        FNAME='PROGRAMS.HLP/G'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'J' .AND. P3 .EQ. 'J' .AND. M .EQ. 0) THEN
        FNAME='JOBS.HLP'
        FOUND=.TRUE.
      ELSE IF(FILE(1:1) .EQ. 'J' .AND. P4 .EQ. 'J' .AND. M .NE. 0)THEN
        FNAME='JOBS.HLP/G'
        FOUND=.TRUE.
      ENDIF
      IF(FILE(1:1) .EQ. 'N' .AND. P5 .EQ. 'N' .AND. M .EQ. 0) THEN
        FNAME='NOTES.HLP'
        FOUND=.TRUE.
      ELSE IF(FILE(1:1) .EQ. 'N' .AND. P6 .EQ. 'N' .AND. M .NE. 0)THEN
        FNAME='NOTES.HLP/G'
        FOUND=.TRUE.
      ENDIF
      IF(FOUND) THEN
C     Add file extension if in a group account
      OPEN(UNIT=1,FILE=FNAME,FORM='BINARY',ACCESS='DIRECT',RECL=80)
      ELSE
C     If help file not found display message
        WRITE(5,803)
  803   FORMAT(/' Help file not found - type AID to list available files
```

```
     .'//)
        GOTO 999
        ENDIF
C     Now ready to read help file
C     First check for number of records in index header
      READ(1)ALINE
      NRI=ICHAR(ALINE(3:3))*256
      NRI=NRI+ICHAR(ALINE(4:4))-1
C     If no keyword then display available keywords
      IF(KL .LE. 0) THEN
        J=1
C     Now get list of keywords from index and display them
        DO 130 I=1,NRI
        READ(1)ALINE
        CONTENTS(J:J+15)=ALINE(1:16)
        J=J+16
  130   CONTINUE
        LL=NRI/5+1
        J=1
        DO 135, I=1,LL
        WRITE(5,905)CONTENTS(J:J+79)
        J=J+80
  135   CONTINUE
        GOTO 999
      ENDIF
C     If keyword to be found then check index
      DO 140 I=1,NRI
      READ(1)ALINE
C     If keyword found, then get start record and number of records
C     otherwise display message
      IF(INDEX(ALINE,KEYWORD(1:KL)) .EQ. 1)THEN
        SLN=256*ICHAR(ALINE(23:23))+ICHAR(ALINE(24:24))+1
        NL=256*ICHAR(ALINE(27:27))+ICHAR(ALINE(28:28))-1
        GOTO 150
      ENDIF
  140 CONTINUE
      WRITE(5,800)
  800 FORMAT(' Keyword not found')
      STOP
  150 DO 160 I=SLN,SLN+NL
      READ(1,REC=I)ALINE
C     Unless end of screen, write info; otherwise pause until CR
      IF(ALINE(1:2) .NE. '/?')THEN
        WRITE(5,905)ALINE
  905   FORMAT(1X,A80)
      ELSE
        WRITE(5,805)
  805   FORMAT('              *** Press RETURN for more information ***')
        READ(3,902)
      ENDIF
  160 CONTINUE
      WRITE(5,902)
  902 FORMAT(A1)
  999 CONTINUE
      END
```

APPENDIX 3

MAKEHELP program listing.

```
* MAKEHELP.CSS - INDEXES A HELP FILE
XAL HLP.TMP,IN,80
REP @1,0
L MAKEHELP,5
AS 1,@1; AS 2,HLP.TMP; ST
SWAP HLP.TMP,@1
REP @1,FF00
$EXIT

$TITL MAKEHELP.FTN   27-MAR-1986  (REVISED 18-AUG-86)
C     Indexes a .HLP file. Any existing header must be removed before
C     running MAKEHELP.
      CHARACTER*80 ALINE
      CHARACTER*16 KEYWORD(1024)
      INTEGER*4 KLN(1024),KLR(1024)
C     LU 1 = Help file     LU 2 = Temporary file (HLP.TMP)
C     After program is finished these files are swapped.
      OPEN(UNIT=1,FORM='BINARY',ACCESS='DIRECT',RECL=80)
      OPEN(UNIT=2,FORM='BINARY',ACCESS='DIRECT',RECL=80)
C     LN = Line number    NRI = Number of records in index
      LN=1
      NRI=1
C     Input file is read line by line and keywords sought, when found
C     lines are read until end of entry to get number of lines in entry.
C     KLN = start line number  KLR = number of lines in record
  100 READ(1,END=150)ALINE
      IF(ALINE(1:1) .NE. ' ' .AND. ALINE(1:1) .NE. '/' .AND. ALINE(1:1)
     . .NE. '!')THEN
      KEYWORD(NRI)=ALINE(1:16)
      KLN(NRI)=LN
      GOTO 110
      ELSE
      LN=LN+1
      GOTO 100
      ENDIF
  110 READ(1,END=115)ALINE
      LN=LN+1
      IF(ALINE(1:1) .EQ. ' ' .OR. ALINE(1:1) .EQ. '/' .OR. ALINE(1:1)
     . .EQ. '!')GOTO 110
  115 KLR(NRI)=LN-KLN(NRI)
      NRI=NRI+1
C     Found first line of next entry so backspace and return.
      BACKSPACE (1)
      GOTO 100
C     End of input file so length of index now known.
C     Add offset to start of entries
  150 DO 160 I=1,NRI-1
      KLN(I)=KLN(I)+NRI-1
      KLR(I)=KLR(I)-1
  160 CONTINUE
C     Construct header record
      ALINE(1:1)=CHAR(0)
      ALINE(2:2)=CHAR(0)
      IF(NRI .GT. 256)THEN
```

```
          J=NRI/256
          ALINE(3:3)=CHAR(J)
          ALINE(4:4)=CHAR(NRI-J*256)
        ELSE
          ALINE(3:3)=CHAR(0)
          ALINE(4:4)=CHAR(NRI)
        ENDIF
        WRITE(2)ALINE
C       Construct index
        DO 180 I=1,NRI-1
        ALINE(1:16)=KEYWORD(I)
        IF(KLN(I) .GT. 256)THEN
          J=KLN(I)/256
          ALINE(23:23)=CHAR(J)
          ALINE(24:24)=CHAR(KLN(I)-J*256)
        ELSE
          ALINE(23:23)=CHAR(0)
          ALINE(24:24)=CHAR(KLN(I))
        ENDIF
        ALINE(27:27)=CHAR(0)
        ALINE(28:28)=CHAR(KLR(I))
        WRITE(2)ALINE
  180 CONTINUE
C       Rewind input file ready to write to end of HLP.TMP
        REWIND 1
C       Now write main help file text - stop at end of file.
  200 READ(1,END=999)ALINE
        WRITE(2)ALINE
        GOTO 200
  999 STOP
        END
```

APPENDIX 4

REMINDEX program listing.

```
* REMINDEX.CSS - REMOVES INDEX FROM A HELP FILE
XAL HLP.TMP,IN,80
L REMINDEX,5
REP @1,0
AS 1,@1; AS 2,HLP.TMP; ST
SWAP HLP.TMP,@1
REP @1,FF00
$EXIT
```

```
$TITL REMINDEX.FTN   18-AUG-86    E.L.MARTIN
C     Removes header from indexed help file prior to editing.
      CHARACTER*80 ALINE
      OPEN(UNIT=1,FORM='BINARY')
      OPEN(UNIT=2,FORM='BINARY')
      READ(1)ALINE
      IF(ICHAR(ALINE(1:1)) .EQ. 0)THEN
      NRI=256*ICHAR(ALINE(3:3))+ICHAR(ALINE(4:4))
      DO 100 I=1,NRI-1
      READ(1)
  100 CONTINUE
      ELSE
      BACKSPACE 1
      ENDIF
  200 READ(1,END=500)ALINE
      WRITE(2)ALINE
      GOTO 200
  500 STOP
      END
```