PACKAGE : APLIO

DESCRIPTION : GENERALIZED APL-OS INTERFACE UTILITIES

DATE : JANUARY 1975

IDENTIFICATION:

1) TITLE: APLIO
2) SOURCE LANGUAGE: 360/370 ASSEMBLER
3) CONTRIBUTOR: JAMES H. WHITE
4) DATE JANUARY 1975
5) MEMORY REQUIREMENTS: 64K FOR MOST JOBS
6) CONTACT: JAMES H. WHITE, UTA COMPUTER CENTER.

PURPOSE:

APLIO IS A COLLECTION OF FOUR UTILITIES WHICH ALLOW THE BATCH USER TO ACCESS APL FILES IN A VARIETY OF WAYS. HERE IS A LIST OF THE AVAILABLE UTILITIES AND WHAT THEY DO :

APLINPUT ALLOWS ONE TO CREATE AND APPEND DATA TO AN APL FILE.
APLCOPY ALLOWS ONE TO COPY AN APL FILE INTO AN OS DATASET.
APLPRINT ALLOWS ONE TO PRINT AN APL FILE.
APLPUNCH ALLOWS ONE TO PUNCH AN APL FILE.

THESE UTILITIES WILL HANDLE ANY SIZE ARRAY, VECTOR, OR MATRIX, AND ANY DATA TYPE(DATA TYPE CONVERSION DOES NOT OCCUR, SO NUMERIC FILES WILL NEED TO BE CONVERTED TO CHARACTER IF THEY ARE TO BE PRINTED OR PUNCHED). IF THE LENGTHS OF THE RECORDS ARE TOO SHORT, PADDING WITH SPACES OR ZEROS WILL OCCUR, DEPENDING ON DATA TYPE. IF THE RECORDS ARE TOO LONG, TRUNCATION WILL OCCUR AT THE LRECL OF THE DATA SET AND A WARNING THAT POSSIBLE DATA LOSS MAY HAVE OCCURRED WILL BE GIVEN. IF APL FILES ARE BEING CREATED OR MODIFIED, THE DATA CAN BE APPENDED IN RECORD FORM, I.E. ONE RECORD (OR CARD) PER COMPONENT, OR IT CAN BE APPENDED IN MATRIX FORM BEING COMPATIBLE WITH THE OS BLOCKSIZE.

USAGE:

BEFORE USING THE BATCH PROGRAM, THE APL FILE ACCESS MATRIX NEEDS TO BE MODIFIED TO ALLOW BATCH FUNCTIONS TO USE IT. THE ACCOUNT 1000 NEEDS TO BE GIVEN ACCESS DEPENDING UPON WHAT IS TO BE REQUESTED. IF ONLY THE OWNER IS TO BE RUNNING THE BATCH JOB, THEN ONLY READ ACCESS NEED BE GIVEN. IF SOMEONE OTHER THAN THE FILE OWNER IS TO RUN THE JOB, THEN ACNT. 1000 WILL NEED TO HAVE ACCESS TO READ THE ACCESS MATRIX AND THE OTHER ACNT. WILL NEED TO HAVE ACCESS TO THE FILE.

NOTE: INFORMATION CONCERNING ACCESS MATRICES CAN BE FOUND IN THE APL*PLUS FILE SUBSYSTEM MANUAL. IT IS SUGGESTED THAT TO AVOID CONFUSION, ACCOUNT 1000 BE GIVEN ACCESS OF -1 (AND NEVER A PASSNUMBER). AN EXAMPLE WOULD BE LIKE THIS:

```
(2 3(RHO) 1000  -1  0    1999  1  0) FA 17 1
```

WHERE (RHO) IS THE RHO OPERATOR. THIS EXAMPLE ALLOWS BATCH USER A999 AND THE OWNER ACCESS TO READ THE FILE. (THE OWNER ACNT. IS DETERMINED BY THE FIRST FOUR DIGITS OF THE ACCOUNT NUMBER OF THE FILE, AND ALL 1'S CHANGED TO A'S, 2'S TO B'S ETC.
I.E. ACNT 1999 = A999

INDIVIDUAL UTILITIES :

****

APLPRINT WILL PRINT AN APL FILE

SYNTAX IS AS FOLLOWS:

```
//JOBNAME JOB A999,'YOUR NAME',MSGLEVEL=(0,0)
/*LOCK          YOURLOCK
/*BOX           BOX#
/*MAXREG        64K
/*TIME          30S
//  EXEC APLPRINT,FILE='APL FILE NAME'
//
```

IF THE FILE ALREADY CONTAINS CARRAIGE CONTROL AND YOU WISH TO USE IT, CODE A ',CC' FOLLOWING THE FILE NAME INSIDE THE QUOTES.

IF THE FILE IS TO BE ERASED AT THE END OF THE JOB, CODE A ',ERASE' FOLLOWING THE FILE NAME, INSIDE THE QUOTES.

****

APLPUNCH WILL PUNCH AN APL FILE
SYNTAX IS AS FOLLOWS:

```
//JOBNAME JOB A999,'YOUR NAME',MSGLEVEL=(0,0)
/*LOCK          YOURLOCK
/*BOX           BOX#
/*MAXREG        64K
/*TIME          30S
//  EXEC  APLPUNCH,FILE='APL FILE NAME'
//
```

IF THE FILE IS TO BE ERASED AT THE END OF THE JOB, CODE A ',ERASE' FOLLOWING THE FILE NAME, INSIDE THE QUOTES.

****

APLCOPY WILL COPY AN APL FILE TO AN OS DATA SET
SYNTAX IS AS FOLLOWS:

```
//JOBNAME JOB A999,'YOUR NAME',MSGLEVEL=(0,0)
/*LOCK          YOURLOCK
/*BOX           BOX#
/*TIME          30S
//  EXEC APLCOPY,FILE='APL FILE NAME'
//SYSUT2 DD DSN=A999.DATA.SET,DISP=OLD
//
```

IF THE FILE IS IN CHARACTER FORM AND YOU WANT THE PROGRAM TO SET UP THE DATA SET FOR PRINTING, PLACE A ',PRINT' FOLLOWING THE FILE NAME, WITHIN THE QUOTES.

IF THE FILE IS TO BE SET UP FOR PUNCHING, CODE A ',PUNCH' FOLLOWING THE FILE NAME, WITHIN THE QUOTES.

****

APLINPUT WILL CREATE AND/OR BUILD AN APL FILE.

SYNTAX IS AS FOLLOWS:

```
//JOBNAME JOB A999,'YOUR NAME',MSGLEVEL=(0,0)
/*LOCK        YOURLOCK
/*BOX         BOX#
/*TIME        30S
/*MAXREG      64K
// EXEC  APLINPUT,FILE='APL FILE NAME'
   .
   .
   YOUR DATA CARDS TO BE PLACED INTO AN APL FILE GO HERE
   .
//
```

OTHER OPTIONS INCLUDE:

TYPE=XXX WHERE XXX IS EITHER 'CHARACTER', INTEGER', 'BINARY', OR 'FLOATING'. THIS DENOTES THE DATA TYPE (CHAR. DEFAULT).

DSN='A999.DATA.SET.NAME' SUPPLIES THE DATA SET NAME IF INPUT DATA NOT COMING FROM CARDS.

BLOCK=YES WILL CAUSE MATRIX APL COMPONENTS TO BE CREATED, WHERE THE BLOCKSIZE OF THE DATASET IS USED TO DETERMINE THE BLOCKING FACTOR. I.E. IF THE DATASET IS CARD IMAGE WITH A BLOCKSIZE OF 1600, THEN CHARACTER MATRICES RANKED 20 BY 80 WILL BE PRODUCED.

BLOCK=NO SPECIFIES ALL DATA IS TO BE APPENDED IN VECTOR FORM

BLOCK=XX WHERE XX IS AN INTEGER, WILL FORCE MATRICES WITH XX ROWS REGARDLESS OF THE BLOCKSIZE.

PURGE=YES SPECIFIES THAT IF THERE IS ANY DATA IN THE APL FILE, PURGE IT PRIOR TO ADDITION OF THE NEW DATA, OTHERWISE THE NEW DATA IS APPENDED TO THE END OF THE EXISTING INFORMATION.

NOTE: IN ALL CASES, THE APL FILE IS TIED IF IT EXISTS, IF IT DOES NOT EXIST, IT WILL BE CREATED IF THE ACNT RUNNING THE JOB MATCHES THE FIRST 4 DIGITS OF THE FILE ACNT NUMBER. ALSO, NO DATA TYPE CONVERSION CAN BE ACCOMPLISHED, SO IF NUMBERS ARE PUNCHED ON CARDS AND WISH TO BE CONVERTED TO ACTUAL NUMERIC VALUES, THEY NEED TO BE MOVED TO APL AS CHARACTERS FIRST, THEN CONVERTED TO NUMBERS BY USE OF THE FUNCTION DELTA-FI IN LIB 905 PLUSFNS.

EXAMPLES:

```
//JOBNAME JOB A999,'YOUR NAME',MSGLEVEL=(0,0) ---|
/*LOCK         YOURLOCK                           |
/*BOX          BOX#                                >--|
/*TIME         30S                                |    |
/*CARDS        X(HUNDREDS) FOR PUNCHING ONLY   ---|    |
                                                       |
                   IN THE FOLLOWING EXAMPLES.          |
         THESE CARDS WILL BE REFERRED TO AS HASP-------/


<HASP GOES HERE>
//  EXEC  APLPRINT,FILE='1234 MYFILE'
//
```

THIS EXAMPLE WILL PRINT THE APL FILE '1234 MYFILE' AS IS.

```
<HASP GOES HERE>
//  EXEC  APLPUNCH,FILE='4321 HISFILE'
//
```

THIS EXAMPLE WILL PUNCH THE APL FILE '4321 HISFILE' AS IS

```
<HASP GOES HERE>
//  EXEC  APLCOPY,FILE='987654 TEMPFILE'
//SYSUT2  DD  DSN=A999.TEMP.FILE,DISP=OLD
//
```

THIS EXAMPLE WILL COPY THE APL FILE '987654 TEMPFILE' INTO THE OS DATA SET A999.TEMP.FILE

```
<HASP GOES HERE>
//  EXEC  APLPRINT,FILE='1888 PRINTFILE,CC'
//
```

IN THIS EXAMPLE, THE APL FILE '1888 PRINTFILE' WILL BE PRINTED, USING THE FIRST CHARACTER OF EACH LINE AS CARRAIGE CONTROL.

```
<HASP GOES HERE>
//  EXEC APLPRINT,FILE='1079001 FROG,CC,ERASE'
//
```

THIS EXAMPLE WILL PRINT THE APL FILE '1079001 FROG', USING THE FIRST CHARACTER OF EACH LINE AS CARRAIGE CONTROL AND ERASING THE FILE IF THE JOB FINISHES WITHOUT ERROR.

```
<HASP GOES HERE>
//  EXEC  APLCOPY,FILE='666 DATA-FILE,ERASE,PRINT'
//SYSUT2   DD   DSN=A999.DATA.SET,DISP=(NEW,CATLG),
//              SPACE=(TRK,(5,2)),UNIT=SYSDA
//
```

THIS EXAMPLE WILL COPY THE APL FILE '666 DATA(DELTA)FILE', SETTING IT UP FOR PRINTING, I.E. BLOCKSIZE=1330,LRECL=133, AND ERASING THE APL FILE IF THE JOB FINISHES NORMALLY.

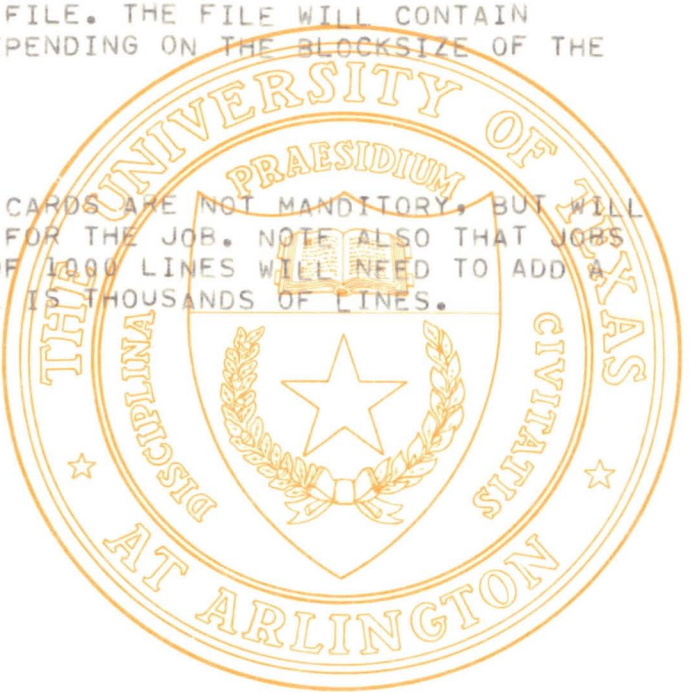
```
<HASP GOES HERE>
// EXEC APLINPUT,FILE='12345 NEWFILE'
  .
  DATA CARDS
  .
  .
//
```

THIS EXAMPLE WILL CREATE THE APL FILE '12345 NEWFILE' (OR TIE TO IT IF IT EXISTS) AND PLACE THE CARD DECK SUPPLIED INTO IT IN VECTOR FORM (1 COMPONENT PER CARD SUPPLIED).

```
<HASP GOES HERE>
// EXEC APLINPUT,FILE='12345 NEWFILE',DSN='A234.DATA',
                 TYPE='INTEGER',BLOCK=YES
//
```

THIS EXAMPLE WILL CREATE THE APL FILE '12345 NEWFILE' UNLESS IT EXISTS ALREADY, THEN IT WILL TIE TO IT; THEN IT WILL TRANSFER THE DATA FROM THE DATASET 'A234.DATA' WHICH IS IN INTEGER FORM INTO THE APL FILE. THE FILE WILL CONTAIN MATRIX COMPONENTS, STRUCTURED DEPENDING ON THE BLOCKSIZE OF THE DATA SET.

THE /*TIME AND /*MAXREG CARDS ARE NOT MANDITORY, BUT WILL IMPROVE THE TURNAROUND TIME FOR THE JOB. NOTE ALSO THAT JOBS REQUIRING OUTPUT IN EXCESS OF 1000 LINES WILL NEED TO ADD A /*LINES X CARD, WHERE X IS THOUSANDS OF LINES.



ERRORS:

WHEN AN ERROR IS ENCOUNTERED, THE FOLLOWING WILL BE PRINTED.

```
**************************************************************
**************************************************************
**************************************************************
***                                                        ***
***                        E R R O R                       ***
***                                                        ***
***                                                        ***
***    ( MESSAGE DESCRIBING ERROR)                         ***
***                                                        ***
***    REQUEST CANCELLED                                   ***
***                                                        ***
**************************************************************
**************************************************************
**************************************************************
```

POSSIBLE ERRORS ARE:

//SYSUT2 DD CARD MISSING
IF USING APLCOPY, THEN YOU NEED TO USE A //SYSUT2 DD CARD DEFINING THE DATASET TO COPY THE DATA INTO.

NO PARAMETER GIVEN
YOU HERE NEED TO GIVE IT AT LEAST THE FILE NAME.

CANNOT UNDERSTAND THE PARAMETER XXXXXX
WHERE XXXXXX WAS SOME PARAMETER SENT TO THE PROGRAM WHICH IT DOES NOT RECOGNIZE, CHECK SPELLING.

APL PLUS AAAAA -- FILE BBBBBB ERROR
WHERE AAAAA IS THE OPERATION ATTEMPTED, SUCH AS FSTIE OR FERASE AND BBBBBB IS THE APL ERROR, SUCH AS ACCESS ERROR, OR NAME ERROR.

FILE HAS NO COMPONENTS
CANNOT COPY AN APL FILE WHICH IS EMPTY.

THE ACNT XXXX DOES NOT HAVE ACCESS TO THE FILE
THE ACNT XXXX HAS NOT BEEN PLACED IN THE ACCESS MATRIX. WHERE ACNT XXXX IS THE PRESENT ACCOUNT RUNNING THIS JOB.

PRINT AND PUNCH CANNOT BOTH BE SPECIFIED
ALL OUTPUT MUST GO TO SYSUT2 AND THAT CAN BE ASSIGNED TO ONLY ONE DEVICE, IF BOTH PRINT AND PUNCH ARE GIVEN THE PROGRAM IS FORCED TO DECIDE FOR ITSELF, SO IT GIVES AN ERROR.

PRINT OR PUNCH NOT SPECIFIED FOR OUTPUT DATASET OR ZERO BLOCK SIZE ON DATA SET
IF SYSUT2 DEFINES A DSN, THEN IT MUST SUPPLY A NON ZERO BLOCK SIZE.

SYSTEM REQUIREMENTS:

THE PROGRAMS FROM#APL AND TO#APL

APL MUST BE UP AND RUNNING AT THE TIME OF THE SUBMITTAL.

APL FILE SUBSYSTEM MUST BE OPERATIONAL.

OTHER COMMENTS:

WHEN CHARACTER INFORMATION IS PASSED, TRANSLATION FROM APL TO EBCDIC OCCURS FOR AS MANY APL CHARACTERS AS POSSIBLE. ANY CHARACTERS WHICH CANNOT POSSIBLE BE CONVERTED ARE TRANSLATED TO SPACES. SOME SPECIAL CHARACTER TRANSLATION INCLUDES:

APL	EBCDIC
EPSILON	&
DIVIDE	%
ALPHA	@
DELTA	-
DELTA U.S.	=
NOT EQUAL	#

WHEN USING FLOATING POINT DATA, APL DEALS WITH DOUBLE PRECISION FORMS ONLY. I.E. 8 BYTES PER VALUE.

ALL BOOLEAN DATA IS TRANSFERRED IN RAVELLED FORM, SO ANY MATRICES AND ARRAYS WILL BE COPIED AS VECTORS. ALSO NOTE THAT SINCE O.S. OPERATES ON NOTHING LESS THAN A BYTE, IF A BOOLEAN MATRIX RANKED 5 BY 5, 7 ZERO'S WOULD BE ADDED AND 4 BYTES WOULD BE THE RESULT.

THESE PROGRAMS WILL WORK WITH DATASETS WHICH ARE SEQUENTIALLY ORGANIZED. THAT IS, PS, OR PO, WHERE A MEMBER NAME WAS SPECIFIED ON THE DD CARD FOR THE PO DATASET. THIS UTILITY WILL NOT WORK ON INDEXED SEQUENTIAL OR VARIABLE LENGTH DATASETS.