

A. IDENTIFICATION

TITLE: APLIB - A General Purpose User Library Maintenance Program
TYPE: COMPASS Main Program
CATALOG IDENTIFICATION: 000000445
PROGRAMMER: Bob Bedoll
ORGANIZATION: Computer Laboratory
 Michigan State University
DATE: January 19, 1970

TABLE OF CONTENTS

General Library Structure1
Usage1
Library Maintenance2
Updating The Library3
Security Code5
User Considerations5
Scope I/O Considerations6
Timing7
Special File Creation Considerations7
APLIB Examples	
Example I8
Example II9

1911

Received of the Treasurer of the University of California the sum of \$100.00

for the purchase of books

for the year 1911

for the purchase of books

for the year 1911

WILLIAM DE LAMAR

Special Agent in Charge

San Francisco

Sample 11

APLIB - A GENERAL PURPOSE USER LIBRARY MAINTENANCE PROGRAM

GENERAL LIBRARY STRUCTURE

The user library created by APLIB contains a special label record, followed by a directory of all files on the library. The directory entry for each file consists of the file reference name, its position on the library, and the date it was catalogued. The library files follow the directory in sequential order on the library tape.

The library resides permanently on a user defined magnetic tape. APLIB automatically maintains the library on a disk resident common file of name MTXXXX, where XXXXX is the tape reel number. On every call to APLIB the program determines whether the required common file is present. If it is, the library files are retrieved from it. If the common file is not present, APLIB will automatically request the operator to mount the user library tape, and then copy the tape to disk, creating the proper common file. In this manner, the operator need mount the library tape only once after each complete deadstart.

USAGE

File Retrieval

This section concerns retrieving files of a user library created by APLIB. The following section will discuss library creation and maintenance.

In order to retrieve any file in the user library only one call to APLIB is necessary:

APLIB(MTxxxxxx, *lfn1*, *lfn2*,...)

MTxxxxxx - xxxxxx is the visual tape reel identification (1 - 5 characters).
APLIB will ask for MT(xxxxxx).

lfn1, etc. - the library routines desired. This parameter can have any one of the following 4 forms:

1. *lfn1* - The library file of library reference name *lfn1* is copied onto LGO on back of any information already there, thus becoming part of the binary program input on LGO. LGO is not rewound, allowing additional programs to be added. This option is especially useful for pulling program subroutines off the user library.

Library Creation

To create a user library the N* parameter must be specified immediately following the library tape number:

APLIB (MT~~xxxxxx~~, N*, Zfn1, ...)

~~xxxxxx~~ specifies the visual reel identification of the user library tape (1-5 characters). APLIB will ask for tape MT(~~xxxxxx~~).

All parameters following the N* are optional. If present, they are acted upon after the library is created, and have the form and meaning described under the File Retrieval section.

When the N* option is specified, APLIB will ask the operator to mount tape XXXX. It will then create an 'empty' library. Files are then added to the library using the update control cards exactly as described in the Updating section.

(Once a user library is created, it should not be recreated on the same tape using the N* option without first ensuring that the corresponding common file MT~~xxxxxx~~ is not present.)

UPDATING THE LIBRARY

The library may be updated by adding, deleting, replacing, or renaming files on the library. To specify an update operation, the first parameter after the library tape number MT~~xxxxxx~~ must be U*:

APLIB(MT~~xxxxxx~~, U*, Zfn1...)

All additional parameters following the U* are optional. If present, they are acted upon after the updating operation is completed, and have the form and effect described under the File Retrieval section.

When an update operation is specified, APLIB requests the operator to mount the user library tape. It then reads update control cards from INPUT. As each control card is read, the proper updating operation is carried out on the disk resident user library. Update control cards are read until a 7-8-9 or a 6-7-8-9 card is encountered, then the partially updated library is copied back onto the library tape in its completed format. This tape is then copied back onto the disk resident common file.

There are four update control cards:

ADD lzm , zfn , NR

The contents of the Scope file zfn are added to the library and given the library reference name lzm . The file zfn is rewound first unless NR, the no rewind parameter, is specified. zfn is optional. If absent, lzm is taken to be both the library reference name and the name of the Scope logical file to be added. The file to be added is copied onto the library until an EOF or EOI (end-of-information) is encountered. (See User Consideration #2.)

REPLACE lzm , zfn , NR

This control card is identical in format to the ADD control card. If the library file lzm is not on the library, the dayfile message "FILE NOT IN LIBRARY" appears but the contents of the file zfn is added. If lzm is on the library, the contents of the file zfn replace on the library the contents of library file lzm . zfn and NR are optional as on the ADD card.

In all cases, the file on the library will have library reference name lzm .

DELETE lzm

The library file of library reference name lzm is deleted from the library. (A REPLACE is identical to a DELETE followed by an ADD.)

RENAME lzm , lzm

The library reference name lzm is changed to lzm . The contents of the library file remain unchanged.

The control words ADD, REPLACE, DELETE, and RENAME must start in card column 1, with no imbedded blanks. After that, the control cards are free field. Parameters are delimited by 1 or more blanks or characters other than a number or letter. A separate control card is required for each file the User wishes to update.

If the User tries to add a file with library reference name identical to a file already on the library the file will still be added, however, the dayfile message "DUPLICATE FILE ADD" will appear and the duplicate file name will be prefixed with a "D" to make it unique.

If the user attempts to add a non-existent file, the dayfile message "NO SUCH FILE FOR ADD *Zfn*" will appear and the request will be ignored.

An update or library creation operation always produces a listing of the library directory as described under the L* option.

SECURITY CODE

Update access to the user library tape may be controlled by an optional security code. To initiate security code protection, the User must include the control card:

SC XXXXXXXXXXXX

SC starts in card column 1. XXXXXXXXXXXX starts in card column 4 and is the 10 character (any character) security code.

The initial control card may be specified when creating or updating the library, its order is immaterial. The security code is entered in special coded form into the tape label. On ALL subsequent UPDATE runs, the IDENTICAL security code card must be specified as the FIRST card in the set of update control cards. If not present, or incorrect, the job will abort with an appropriate dayfile message.

Once the update operation has begun, the security code may be changed in a manner identical to the way it was first specified. However, once initially specified, security code protection may not be deleted.

USER CONSIDERATIONS

1. Files added to a user library must be disk resident. Because coded and binary files have identical representation on the disk, files to be added may be either coded or binary.

Files should always be retrieved onto the disk. If retrieved onto tape, the tape will be written in binary mode.

2. Files are added to the library until an EOF or EOI is encountered. Files should not be added from INPUT.

3. Because of the high percentage of parity errors on 6500 tape operations, APLIB will attempt recovery on all unrecovered write parity errors. The user

00000445
APLIB
1/19/70
Page 6

Library tape will be rewound and the update copy to tape operation repeated, up to a maximum of 20 times.

4. APLIB always checks the tape label to ensure that the correct tape is present. If the tape has been wiped out, or the wrong tape assigned, the job will abort with a dayfile message.
5. A request to retrieve a file not on the library will generate the dayfile message "FILE NOT IN LIB - zfn " and the request will be ignored.
6. Required field length - 6100 words. Field length is always automatically reduced to 6100 words when entered and returned to original length at end.
7. Because of the current status of the 6500 system, the User is advised to keep a backup of his library tape. The tape may be copied to a backup tape by:

COPYBF(*tape1*, *tape2*, *n*)

where *n* is greater than or equal to the number of library files + 2.

SCOPE I/O CONSIDERATIONS

If the User wishes to add an existing card file (binary or coded) to the library he is advised to make two separate runs. The first run should merely copy the card deck to a common file, which may then be added to the library on a subsequent run. In this manner, the conflict between End-of-Files and End-of-Records does not arise.

Naturally, it is possible to add card files to the library all in one run, by first copying from input onto a local file. The User should be aware, however, that APLIB will read its update control cards from the current position of the INPUT file. Note that an End-of-File on INPUT can only be achieved by a 6-7-8-9 End-of-Information card. Thus, a COPYBF or COPYCF will always copy to a 6-7-8-9 card. COPYBR or COPYCR will only copy to a 7-8-9 End-of-Record card. However, each COMPASS or FTN subprogram is 1 logical record, and 1 COBOL program consists of many records.

Though a 7-8-9 is defined by SCOPE as End-of-Record, FTN will recognize it as an End-of-File. The FTN "END FILE" statement, however, will write a standard SCOPE End-of-File.

In addition, each FTN "BUFFER OUT" statement is 1 logical record, but all output generated by FTN coded WRITE together comprise 1 logical record unless separated by FTN "END FILE".

TIMING

These are average times and necessarily approximate.

Scope 3.1.2

CPU time for file retrieval - .05 sec

CPU time for library updating or creation - .5 sec

Scope 3.1.6

File retrieval time is the same.

CPU time for library updating or creation - 15 seconds; however,

PPU time is greatly reduced from 3.1.2.

SPECIAL FILE CREATION CONSIDERATIONS

* $\mathcal{L}f_n$ option - if the local file $\mathcal{L}f_n$ is already present at the control point, APLIB will overwrite the contents of that file i.e., it will rewind it and replace its contents with the contents of the requested file on APLIB.

R* $\mathcal{L}f_n$ = $\mathcal{L}f_n$ - identical consideration as in the * $\mathcal{L}f_n$ option.

C* $\mathcal{L}f_n$ - if the local file $\mathcal{L}f_n$ is already present or if the common file $\mathcal{L}f_n$ is already present AT THE CONTROL POINT, APLIB will rewind it and replace its contents with the contents of the requested file on APLIB. (If local, $\mathcal{L}f_n$ will then be made common.) If the common file $\mathcal{L}f_n$ is present in the system but NOT at the control point, APLIB will create a local file as in the * $\mathcal{L}f_n$ option.

The common file $\mathcal{L}f_n$ not at the control point will not be altered. This is to protect existent common files. In this event, the dayfile message "FILE

ALREADY COMMON - LOCAL FILE ONLY CREATED" will appear.

00000445
APLIB
1/19/70
Page 8

APPENDIX

APLIB EXAMPLES

Example I. Create an update file and catalogue it onto a new APLIB tape along with a test data deck.

COPYCR(INPUT,TEST1)

UPDATE(N)

COMMON APLIB.

APLIB(MT201,N*)

7 8 9

(TEST1 DATA DECK)

7 8 9

*DECK PROG

(UPDATE DECK)

7 8 9

ADD UPDAT1,NEWPL

ADD TEST1

6 7 8 9

00000445
APLIB
3/2/70
Page 10

APLIB - ADDENDUM

This supplement reflects changes made to the APLIB system effective February 1, 1970.

APLIB now maintains the user library on a permanent file rather than a common file. The permanent file name remains MT~~xxxxxxxx~~, where ~~xxxxxxxx~~ is the User's tape number (physical reel number). All permanent file maintenance is handled by APLIB, which will automatically attach, create, or recreate the User's permanent file resident library.

APLIB creates the permanent file with Control, Modify, and Extend passwords all specified, in order to allow multi-read access. To purge an APLIB file, the User must specify a password of PURGE

ATTACH, $\mathcal{L}f_i$, MT~~xxxxxxxx~~, PW=PURGE.

PURGE, $\mathcal{L}f_i$.

The updating procedure has been redesigned internally to provide greater reliability. The entire user library is now completely updated on disk, and then copied to the User's library tape. In the event of an unrecoverable tape parity error, the user library will still remain intact as a permanent file.

All other specifications remain the same.

Example II. Make corrections to the update program deck. Compile and run it using the TEST1 data deck as input.

```
COMMON APLIB.  
APLIB(MT201,*TEST1,R*UPDAT1=OLDPL)  
UPDATE(Q)  
FTN(1=COMPILE)  
LGO.  
78  
9  
*IDENT  
(UPDATE CORRECTIONS)
```

319
316
315
313
~~312~~
284
281
273

everything below except 144