

---

Send all comments, criticisms and contributions to the editor  
T.W. Lanzatella

---

#### NOTICE OF CHANGES TO THE OPERATING SYSTEM

With this tape, the entire operating system has been updated through PSR level 388/387 (level 7). Modifications supplied by Control Data with this release include only corrections to existing programs, no new features were released. A large number of these corrections were installed prior to the official release of PSR 388/387. Anyone who is interested in reading through the modifications supplied by Control Data should contact T.W. Lanzatella.

Concurrent with PSR 388/387 installation, nearly 6000 cards of corrections and new features were added by UCC Systems' staff.

Bruce Johnson converted the old MOMS DMPCOR to run under KRONOS. Documentation is available via WRITEUP.

Don Hammes changed CPCOM so that the status plotter queue is named PLOTS instead of STATOS. Also, Don changed LFM so that no one can dispose files without dispose permission, COSE in the access word.

Tim Salo added some badly needed enhancements to the LIMITS utility output format. Aside from a couple misspellings, the new format is a definite improvement.

N.L. Reddy added an entry to COMSBIO for the status plotter. The system now treats the plotter as a regular BATCHIO type equipment.

James Mundstock added modifications to CALLPRG which:

1. fix the security leak caused by the ability to insert a card into the CALLPRG directory referencing VALIDUX.
2. sends a message to a TELEX user when a tape resident program is called.
3. allows the parameter, INDEX, in any position on the WRITEUP directive.

EJM also added a new common deck, COMCTID, which performs analogously to COMPTID.

Alan Johnston added a new DSD command, MSG, which enables the operator to send a message to either EXPORT or SUPIO terminals.

Kevin Matthews added a program called PDUMP which aids the operators in permanent file dumping by automatically setting the myriad of PFS arguments necessary to initiate dumping.

Bill Elliott, by far the most prolific contributor to this tape, added the following changes.

1. A new control card, DENSITY (lfn,n), enables a user to change the density of a tape to a value other than that specified by the LABEL card. This directive

is presently in the system but will most likely be subject to further system staff scrutiny because of the awkwardness of the \*n\* parameter which indicates the desired density in the following manner:

2 = HI (556)  
1 = LO (200)  
3 = HY (800)  
4 = PE (1600)

The DENSITY card was implemented as a consequence of the addition of the DENSITY macro to COMPMAC which has the format:

DENSITY lfn, density, recall.

2. A new control card, NEXTVSN (lfn), which, when executed, invokes end-of-tape processing on multi-reel file lfn by executing a CLOSER macro. The utility can save a multi-reel file user an immense amount of time and PRU's transferred.
3. Three new control cards which enhance permanent file processing; FDEFINE, FSAVE and ACQUIRE (or A). FDEFINE performs the following function:

```
PURGE(lfn/NA)
DEFINE(lfn/NA)
```

FSAVE performs the following function:

```
ATTACH (lfnx = lfn/NA)
IF(FILE(lfnx,AS))GOTO,1.
IF(SIZE(lfn).LE.320B)GOTO,2.
DEFINE(lfnx = lfn)
GOTO,3.
1,IF(SIZE(lfn).GT.320B)GOTO,3.
PURGE(lfnx)
2,REPLACE(lfn)
GOTO,4.
3,COPY(lfn,lfnx,v)
4,*.END OF PROCEDURE
```

ACQUIRE performs the following function:

```
ATTACH(lfn/NA)
IF(FILE(lfn,AS))GOTO,1.
GET(lfn)
1,*.END OF PROCEDURE
```

A new parameter was also added to the PFILES control cards, PL, which works identically to the PN parameter except that LIBRARY user number is automatically defaulted, i.e., PL = PACKNAME = PN = PACKNAME, UN = LIBRARY.

These extensive changes to PFILES, implemented without prior warning to the general staff, have come under considerable verbal criticism. At least

two of these control cards, FDEFINE and FSAVE will be deleted from the system. Further disposition of these changes is still uncertain. Unless further criticism is received regarding the remaining changes to PFILES, they will remain in the system. Staff members are encouraged to publicize their comments either through this newsletter or through any other form of written notification.

4. A new PP program, UFM (U of M File Manager). Only two functions are presently performed by UFM and they are closely related to MAGNET operations. UFM was written to avoid making extensive changes to CPM and LFM.
5. A new macro named = 99 was added to COMCMAC. This macro forms the RA + 1 calls to UFM. Presently it is only being used for the new control card, DENSITY.
6. MAGNET has been given the capability of forcing at least one tape job to a control point when it finds that all tape units are occupied and all have rolled status. UFM function 0 selects the tape job needing the least field length to execute.
7. Program PREVIEW was modified so that when any parameter is specified on the directive, PREVIEW simply counts the number of jobs waiting for tapes rather than listing the job names. In addition, when run as a system origin job, PREVIEW shows the jobs using the currently mounted private disk pack.
8. A new DSD command, DISMOUNT IDLE UNITS. This command direct MAGNET to unload all tapes not assigned to a job.
9. A special request processor has been added to MAGNET. Its first job is to handle DENSITY requests.

#### PROPOSED CHANGES TO THE OPERATING SYSTEM

Bill Elliott has submitted the following proposal change:

Addition of a C option to the TDUMP directive. The C option would permit dumping of even parity SI, S, L and F format tapes.

Bruce Johnson proposes the following change to PPTTEXT. The micro NMOT in PPTTEXT should be changed so that BC becomes LD. (Several complaints have already been received regarding this change. They will be printed in the next issue of DEAD-START SYSTEMS NEWSLETTER.)

Jim Mundstock has proposed that the CALLPRG index be changed from an indirect access permanent file to a fast attach permanent file. Preliminary estimates of the time savings involved range from 15 to 30 minutes a day of disk channel time - a precious commodity at this time. Unless any complaints about this change are received before the next tape, the change will be installed at that time.

Proposed proposal: scuttlebutt at Lauderdale has it that a war is in the offing over whether the operating system should be warped to accommodate internal scheduling of maintenance routine or to leave the scheduler alone and allow the routines

to float along with all jobs in the system (TWL's choice of words). The topic should be brought up in a systems' meeting. In either case, a written proposal must be submitted before any change of this nature is installed. Warring factions: KCM vs. NLR.

#### SYSTEM MAINTENANCE: PROCEDURES AND PEOPLE

One of the main functions of this newsletter is to act as a guide for UCC programming staff to the diverse area of system maintenance procedures. In the past, this important area has rarely been documented. Word of mouth was the chief method of describing maintenance procedures. As the size of UCC grows and the number of people directly involved with system maintenance increases, word of mouth communication becomes more and more inefficient.

The hope is that every effort put into this section of the newsletter will help to stave off a communication breakdown in the future.

Betty Stahl is now maintaining a record of all Program Summary Reports (PSRs) sent to CDC and their replies (if any). Please be sure that a copy of all PSRs is sent to Betty in 217 ExpEng.

Two documents describing the CALLPRG utility are included below. The first, by Andrew Mickel, describes the CALLPRG utility and the motivation behind its use. The second, by Marisa Riviere, is the CALLPRG report. The report is composed of four parts.

1. CALLPRG procedures
2. RFM documentation
3. CALLPRG index maintenance
4. Classification of CALLPRG programs

Items 1 through 3 are included in this version of the report. The fourth item is omitted due to its size and lack of relevance to people outside the CALLPRG group. Anyone interested in obtaining a copy of the fourth item should contact M. Riviere.

An Introduction to CALLPRG for UCC Programmers  
A. B. Mickel 4/9/75

(In part, this introduction is based on memos in the past by Marisa Riviere and Mike Frisch.)

The running KRONOS operating system is initialized through a procedure called "deadstarting." The console operator can initiate a deadstart by pressing a button and causing the machine to read a small PP program from a panel of switches. Eventually a magnetic tape (called the DEADSTART Tape) is read which not only allows the operator to configure the system and enter time and date parameters, but also contains the binaries of control cards which will become part of the locked "Common" file named SYSTEM. Additionally a LIBRARY Tape is read containing other files which becomes part of the running system. Because we at UCC maintain a KRONOS operating system augmented with many additional control card callable compilers, libraries, and "system" routines and other non-control card callable user software as well, an additional mechanism was created to prevent filling up the LIBRARY and primarily the DEADSTART Tapes, too large a DEADSTART Tape can slow down the deadstart procedure!

Control cards are normally processed by a routine called 1AJ (Advance Job), a PP program. In the event 1AJ cannot find the program called for either in the local user file space, or in SYSTEM, instead of returning the message: "ILLEGAL CONTROL CARD", it now turns over control to CALLPRG (pronounced Call-prog). CALLPRG is an "extension" to 1AJ which can access programs stored outside SYSTEM and other system-defined files - namely on disk (permanent files) or on tape by using a table or index, which is kept on disk. Under MOMS (the old U of M operating system from 1970-1974), CALLPRG was developed to:

- 1) Allow low usage software to be kept on tape.
- 2) Allow simpler access with fewer control cards of user software no longer requiring the user to know where the programs are kept (disk file or tape).
- 3) Allow systems staff to easily move programs and add new ones without detailed bother of informing users.

An interesting fact: CALLPRG was called in 100,000 times in the first 6 months of running KRONOS - from Sept., '74 to Feb., '75.

CALLPRG passes all parameters specified by users on to the program sought. CALLPRG has subsequently been extended to allow FUTURE and PAST (New and Old) versions of programs to be included under the same name as the "Current" version. Also non-control card callable files such as user libraries or relocatable binary programs can be accessed by a "FETCH" and text files for documentation purposes via "WRITEUP."

For disk files, CALLPRG does a simple Attach for direct access files; a Get is performed for indirect access files thus making a local file copy of the program for the user program. For tapes, CALLPRG makes an internal tape request and searches for a file binary of the appropriate name on the tape. CALLPRG checks to see if the binary already exists in the local file space of the user before making an attempt to access disk or tape. Tape files are copied to local disk files. Therefore subsequent calls to the same control card in a user program (without an intervening RETURN) load the original CALLPRG provided program. For the most part the scheme is entirely transparent to the user.

Lower usage software is stored on tape, probably sharing the tape with other programs. The rule of thumb in where a program should reside (disk or tape) is determined by usage, but so as to not infuriate the UCC computer operators, the sum of monthly usage of all the CALLPRG callable programs on a single tape should not exceed 25 or 30 (which is about one use of the tape per day).

#### Procedures:

Just for the record, one often may need to refer to instructions on "how to get a program under CALLPRG." For each program, there must be an index card. There are 4 types of index cards: one for disk files, one for tape files, one for comments, and one for documentation (WRITEUP) files (which are currently on a separate index). Programmers wishing to find out the format for valid index cards can do a

WRITEUP,CALLPRG.

All changes (additions, deletions, modifications) of programs under CALLPRG must be communicated to Marisa Riviere, the current CALLPRG maintainer. At the present time this is done by memo or phone, but in the future, changes may be communicated via computer program.

For current, FUTURE, PAST AND FETCH type CALLPRG files, the index currently resides on a text file accessed by:

GET,CALLPRG/UN=YZE6008,PN=STF.  
CS,CALLPRG,OUTPUT.

Most disk files for this index currently reside under UN=YZE6008,PN=SP. Some are under UN=LIBRARY. All disk files are CT=PUBLIC,M=READ,FS=SC.

All tape files have VSNs of the form LB00nm, and are default at HI density, I format, KRONOS Labelled, read only (PO=R) and are on 1200 foot reels.

For WRITEUP type files, the index is currently accessed by:

GET,WRITEUP/UN=YZE6000,PN=STF.  
CS,WRITEUP,OUTPUT.

Most WRITEUP type disk files all currently reside under UN=YZE6000,PN=SP, and are all CT=PUBLIC,M=READ,FS=SC. Others are under UN=LIBRARY.

As yet there are no known WRITEUP type tape files.

You may wonder why we did not use UN=LIBRARY as the place to keep all of these files. After all KRONOS has such a user number and it is convenient to use under timesharing (TELEX), as it is under MERITSS. One reason might be that to affect changes in the permanent file catalog of UN=LIBRARY one needs system origin privileges or access to the console. This is not in the spirit of the third point mentioned earlier motivating the creation of CALLPRG, namely making changes easy.

For further CALLPRG information and procedures, see news from Marisa Riviere.

## CALLPRG PROCEDURES

Due to the extended number of programs that are now on the CALLPRG account number and our intention of keeping information available about changes implemented on our system, I found it necessary to review the organization of the CALLPRG programs.

I would like to divide the CALLPRG programs into two different categories: Programs that can be considered an extension of the DEAD START tape or the LIBRARY TAPE and programs that can be considered as private packages. I will consider the first type an extension of the SYSTEM and I would like to keep the programs of this type on the CALLPRG account number. From the SYSTEM point of view, I will be specially concerned about having up-to-date information on changes implemented on these types of programs. I classify into a different type the CALLPRG packages that are not so related with the SYSTEM. These other types of programs may be kept under the account numbers belonging to the programmers in charge of them. A report about the changes to this type of program will be up to the individual programmer. There will not be a rush to remove these programs from the CALLPRG account number as long as they do not need to be modified. All the programmers that want to leave their programs on the CALLPRG account number are welcome to do so. However, they will have to follow the procedures that I establish for the CALLPRG account to set up new versions of their programs.

A decision must be made about the classification of the programs into the two different types. As a first approach, I consider that COMPILERS and LIBRARIES are one extension of the LIBRARY TAPE.

Utility packages may be considered an extension of the DEAD START TAPE. Other packages could be transferred to the category of private type of CALLPRG packages. I do not think that the division has to be very strict. I would appreciate it if the programmers in charge of each set of programs would review my classification and correct or fill in the fields provided for a short description of the programs and information about documentation.

The CALLPRG account number will have an information file that will show each time a CALLPRG account number file has been modified. To implement the creation of this information file, the CALLPRG account number and password will no longer be used for jobs that save, purge or define files on it. The programmers that will be handling their own files on the CALLPRG account number will be able to do so through the use of RFM, a resident files manipulator, which use I will explain below, do not panic, RFM is as easy to use as the PURGE, SAVE, and DEFINE control cards.

(e.g.: The two control cards  
ACCOUNT,YZE6008,PW.  
SAVE,FILES=SC, CT=PU,M=R.  
will be replaced by the card  
RFSAV(FN=FILE,AN=YZE6008)

With the introduction of RFM we will be able to have easy access to information about modifications to the programs kept on the CALLPRG account number. At the start of this procedure RFM will be available but not enforced. If you have problems with RFM you can go back and use the account number and password of CALLPRG to change your files. I will wait to enforce RFM until

everyone becomes familiar with its use. During that time I would appreciate receiving suggestions about possible changes to RFM. For the implementation of RFM I need to have the account numbers that will be validated to handle CALLPRG files on the CALLPRG account. All of those that will need to be validated, please send me a note with your name and account number. [RFM documentation will be available soon.]

The CALLPRG index will be maintained in a MODIFY OPL format (CPOPL). I will request the modifications for the index to be sent to me in the format of MODIFY input cards. I have attached a list of CPOPL for a description of the format of the index modification cards. As you can read on CPOPL documentation, CPOPL format is designed to keep track of the reasons and dates of the changes implemented on the CALLPRG index.

I review and extend here the procedures that I set up initially for the modifications of the CALLPRG files and the index cards. I strongly recommend not changing the contents of the CALLPRG files themselves as a way to avoid submitting modifications for the index. I have no way of enforcing this request for the CALLPRG programs that are going to be under private account numbers. That does not mean that I would not appreciate being notified of the change and the reason for it. For the CALLPRG programs that will remain on the CALLPRG account number, even when the index card is not changed, the information file created by RFM will show the changes, besides, RFM will not allow replace operations. A file, even an indirect access file, will have to be purged first to create another with the same name. In other words, there will be a certain period of time where files may not be available to users. I want to have the procedure of not changing the contents of an active CALLPRG file at least for the programs that are going to be left on the CALLPRG account number. I also suggest it for the private CALLPRG files. This will help to keep information on modifications on all the CALLPRG packages.

To set up new programs on CALLPRG follow the following steps:

1. Save or define your file either on the CALLPRG account number or on your own account number. Make the file public, read only and secured.
2. Send me the deck of MODIFY input to apply to CPOPL to produce the new index.
3. Inform me, if possible, where documentation for the new program is available.

To change already available CALLPRG programs follow the following steps:

1. Save or define your file with the new version of the program on the CALLPRG account number or on your own account number. For programs on the CALLPRG account number this file has to have a different name than that of the file that already contains the version of the program in use. For programs that are not on the CALLPRG account number the following steps may be omitted.
2. Send me a deck of MODIFY input to apply to CPOPL to produce a new index.

3. Wait until I send you a note informing you that the index has been modified to purge the file that was in use with the old version of the program. You can also notice the change by listing the CALLPRG index yourself, and then proceed to purge the old file. Note that the CALLPRG files are property and responsibility of each one of the programmers in charge of them. I suggest that you wait two or three days to purge the old file to be protected in the event that the index may need to be replaced by its older version.
4. If there is documentation of your program on WRITEUP and the documentation is affected by the change, do not forget to update your WRITEUP file. (You can replace the WRITEUP file yourself. If you want the documentation to be changed at the very same time of the program, contact me to arrange it.) Do not forget to update any other documentation as well.

To place new programs or to change old versions of CALLPRG programs on the CALLPRG TAPE LIBRARY, follow the following steps:

1. Request me to assign you a tape from the CALLPRG TAPE LIBRARY. I will send you a CALLPRG tape form for the tape that you will be able to use. At the time that you receive the tape form, the tape will be ready for you to write on it (write ring on, no ring label off).
2. Write the tape. The CALLPRG tapes are KRONOS internal labeled tapes. The LABEL card should be the following:
 

```
LABEL(T,VSN=LBXXXX,PO=W)
XXXX=library tape number as shows on the tape form
```
3. Request the tape librarian to take the write ring off the tape and to place a "NO RING" label on the tape.
4. Fill in the fields on the tape form under "PROGRAMMER," "NO RING LABEL ON (date)," and "CONTENTS."
5. Send me the form and the modification deck for the index.
6. Inform me which library tape, if any, containing the old version, can be released after the new index goes in. I will fill in the "READY TO USE" and "WRITE RING ON" fields on the old tape form, no sooner than a week after the new index is on. If you want to make sure the old version remains on a library tape longer than that, let me know about it.
7. Do not forget to update documentation if needed.

A decision about when a program should be on a tape rather than on a permanent file may be based on the size of the program and how frequently it is used.

For the format of the CALLPRG cards see the documentation of CALLPRG, which is available on WRITEUP.

I change the CALLPRG index once a week, but I may do it at any other time upon request.

I want to include in this description procedures for the handling of private user owned packages that may be included on the CALLPRG index.

So far, the number of privately owned programs on CALLPRG is very small and I have not yet developed a well established procedure to handle this type of program. I request that anyone who wants to introduce a private user program on CALLPRG contact me for arrangements. The following description, that eventually may be modified, covers the preliminary information on set up of private user programs on CALLPRG.

1. Save or define the file with the program under your own account number. Make this file public, read only and secured. Programs on tape should be written on KRONOS internal labeled tapes, stored at the UCC Lauderdale site. To protect the information written on the tape request the write ring to be taken off the tape and a "NO RING" label to be placed on the tape.
2. Send me the cards to modify the index in MODIFY format to apply to CPOPL. If you submit new cards each time that you modify the program, you may omit the reasons for the modifications.
3. Inform me where documentation for each newly introduced private CALLPRG program is available. Note that the CALLPRG index is a public file and that your index cards are going to be listed under your name, telephone number and address. Users of your packages will refer to you with any problems and information about the packages.

M. Riviere  
April, 1975

RFM HANDLES SAVE, DEFINE AND PURGE REQUESTS ON FILES THAT ARE FILE RESIDENTS IN A RESIDENT FILES ACCOUNT. RFM IS A FIRST APPROACH TO PROVIDE FILE SECURITY FOR FILES THAT CAN BE CREATED AND PURGED UNDER A SINGLE RESIDENT ACCOUNT BY SEVERAL VALIDATED USERS, RUNNING JOBS UNDER THEIR OWN ACCOUNT NUMBERS. THROUGH RFM, A VALIDATED USER DOES NOT NEED TO KNOW THE PASSWORD OF THE RESIDENTS FILE ACCOUNT TO HANDLE FILES ON IT. ANY VALIDATED USER CAN CREATE FILES ON THE RESIDENT FILES ACCOUNT. ANY VALIDATED USER CAN PURGE THE FILES THAT HE HAD PREVIOUSLY CREATED, BUT NOT FILES CREATED BY OTHER USERS. RFM USES AS FILE PROTECTION THE USERS ACCOUNT NUMBER, AND THE FILE PASSWORD, IF USED. RFM TRIES TO SIMULATE THE SCOPE PROCEDURE FOR PERMANENT FILES HANDLING. A RESIDENT FILES ACCOUNT IS EQUIVALENT TO A SUBDIRECTORY, WHERE EACH FILE IS PROTECTED BY ITS PASSWORD AND THE ACCOUNT NUMBER OF THE USER THAT CREATED IT. RFM READS A VALIDATION FILE OF LEGAL RESIDENT FILES ACCOUNTS (RFMVAL). RFMVAL IS A PRIVATE SYSTEM FILE THAT CONTAINS A LIST OF LEGAL RESIDENT FILE ACCOUNTS AND THEIR USER INDEXES. RFM DOES NOT HONOR REQUESTS ON ACCOUNTS THAT ARE NOT INCLUDED ON RFMVAL. RFM READS A VALIDATION FILE ON THE RESIDENT FILE ACCOUNT TO SEE IF THE CALLING USER IS VALIDATED FOR IT (RAVAL). AFTER RFM COMPLETES A REQUESTED TASK OF SAVING, DEFINING OR PURGING A FILE, IT UPDATES AN INFORMATION FILE DESCRIBING THE ACTION TAKEN AND THE USER THAT REQUESTED IT (RAINF). RAVAL AND RAINF ARE PUBLIC FILES ON THE RESIDENT FILES ACCOUNT, BUT THEY CAN NOT BE CHANGED BY VALIDATED USERS BY RFM REQUESTS. RAINF IS UPDATED BY RFM AND ONLY THE MASTER USER OF THE ACCOUNT CAN PURGE IT. RAVAL IS UPDATED ONLY BY THE MASTER USER WITH INFORMATION ABOUT NEW VALIDATED USERS. RFM SAVES AND DEFINES FILES OF ONLY PUBLIC TYPE, WITH ONLY READ PERMISSION AND MAKES THEM SECURED FILES. (THIS CAN BE EXTENDED TO OTHER MODES IN A GENERALIZATION OF RFM) RFM WAS WRITTEN MAINLY TO HANDLE THE FILES OF THE CALLPRG RESIDENTS FILES ACCOUNT. AN EXTENSION OF RFM TO HANDLE FILES ON OTHERS RESIDENT FILES ACCOUNTS MAY BE POSSIBLE IF THE CONCEPT OF RESIDENT FILES ACCOUNTS IS INTRODUCED IN KRONOS. TO INTRODUCE THIS CONCEPT THE ACCOUNTS DEFINED AS RESIDENT FILES ACCOUNTS SHOULD ALLOW A SYSTEM JOB TO RUN UNDER ITS USER INDEX (NOTE THAT THIS IS NOT THE USER INDEX OF THE CALLING USER) RFM USES THE USER CONTROL WORD OF THE FILES THAT IT CREATES ON THE RESIDENT FILES ACCOUNT TO KEEP THE ACCOUNT NUMBER OF THE JOB THAT REQUESTED THE CREATION TASK. TO VALIDATE A PURGE REQUEST, RFM MAKES SURE THAT THE USER CONTROL WORD OF THE FILE TO BE PURGED AND THE ACCOUNT NUMBER OF THE JOB THAT IS REQUESTING THE PURGE TASK ARE THE SAME.

946623

CALLING SEQUENCE

THE CONTROL CARDS TO USE RFM ARE:

- RFSAV(P1,P2,P3,P4) - TO SAVE A FILE ON THE RESIDENT FILES ACCOUNT
- RFDEF(P1,P2,P3,P4) - TO DEFINE A FILE ON THE RESIDENT FILES ACCOUNT. (THIS INCLUDES THE COPY OF THE LOCAL FILE TO THE PERMANENT FILE)
- RFPUR(P1,P2,P3,P4) - TO PURGE A FILE FROM THE RESIDENT FILES ACCOUNT

WHERE:

- FN=FNAME            NAME OF THE FILE TO SAVE, PURGE OR DEFINE.  
<>                    (NO FILE NAME CAUSES RFM ERROR)
- PN=PACKNAME        NAME OF THE PACK WHERE THE RESIDENT FILES ARE.  
<SP>
- AN=RESACC           ACCOUNT NUMBER FOR THE RESIDENT FILES.  
<>                    (NO ACCOUNT NUMBER CAUSES RFM ERROR)
- PW=PASSWORD        READ PASSWORD FOR THE FILE TO DEFINE, SAVE OR  
PURGE.  
<>

FILES FORMATS

RFM HANDLES ONLY DISK RESIDENT FILES.  
FOR DEFINE OPERATIONS THE USER SHOULD ALLOW ENOUGH TIME TO COPY THE LOCAL FILE TO THE DEFINED PERMANENT FILE.  
THE SPECIAL RFM FILES, RAVAL AND RAINF HAVE THE FOLLOWINGS FORMATS:

RAVAL - RESIDENT ACCOUNT VALIDATION FILE  
(RESIDENT ACCOUNT FILE, INDIRECT ACCESS)

COLUMNS:

- 1 TO 10    - NOT USED
- 11 TO 20   - ACCOUNT NUMBER, LEFT ADJUSTED
- 21 TO 40   - NOT USED
- 41 TO 60   - USERS NAME
- 61 TO 70   - USERS UCC MAILING ADDRESS
- 71 TO 80   - USERS TELEPHONE (OFFICE)

COLUMNS 11 TO 20 AND 41 TO 60 ARE THE ONLY ONES USED BY RFM.  
THE OTHER ONES ARE USED FOR MAILING LIST AND PHONE DIRECTORY OF THE RESIDENT ACCOUNT USERS.

RAINF - RESIDENT ACCOUNT INFORMATION FILE  
(RESIDENT ACCOUNT FILE, INDIRECT ACCESS)

COLUMNS:

- 1 TO 10    - USER ACCOUNT NUMBER
- 11 TO 20   - MODIFIED FILE NAME
- 21 TO 30   - DATE WHEN RFM WAS REQUESTED A TASK
- 31 TO 40   - TIME WHEN RFM WAS REQUESTED A TASK
- 41 TO 50   - DESCRIPTION OF THE ACTION REQUESTED
- 51 TO 60   - FIELD SHOWING IF FILE WAS CREATED WITH A  
READ PASSWORD

9.0624

61 TO 80 - USERS NAME

RAINF IS NOT UPDATED IN CASE OF ERRORS, SUCH AS ILLEGAL ACTIONS REQUESTED ON THE RESIDENT ACCOUNT FILES. IT IS UPDATED, HOWEVER, WHEN A DIRECT ACCESS PERMANENT FILE IS DEFINED AND THE JOB DOES NOT HAVE ENOUGH TIME TO DO THE COPY TASK. IT IS UP TO THE USER TO RE-CREATE THE FILE, AFTER PURGING THE INCOMPLETED ONE. THIS PURGE HAS TO BE DONE ALSO BY CALLING RFM.  
RAINF HAS TO BE MAINTAINED BY RESIDENT ACCOUNT MAINTENANCE JOBS TO KEEP IT OF A SIZE SUCH THAT FITS AS A INDIRECT ACCES FILE.

RFMVAL - LEGAL RESIDENTS ACCOUNT FILE  
(SYSTEM FILE, INDIRECT ACCESS)  
COLUMNS

- 1-7 ACCOUNT NUMBER
- 21-30 USER INDEX, 0 FILLED, RIGHT UDJUSTED.

RFMVAL IS MAINTAINED BY THE SYSTEM PROGRAMMER IN CHARGE OF THE RESIDENT FILES ACCOUNTS.

MESSAGES

\$RFM ERRORS

FIRST LINE OF ANY RFM DAYFILE MESSAGE.  
IT INDICATES THAT THE FOLLOWING LINE IS AN RFM ERROR MESSAGE.

\$ACCOUNT NOT VALIDATEDS

INDICATES THAT THE USER DOES NOT HAVE HIS ACCOUNT NUMBER INCLUDED IN THE RESIDENT ACCOUNT VALIDATION FILE.

\$INVALID PURGE REQUESTS

INDICATES THAT THE USER IS ATTEMPTING TO PURGE A RESIDENT FILE CREATED BY ANOTHER USER.

\$ERROR IN FILE PASSWORDS

INDICATES THAT THE USER DOES NOT PROVIDE THE RIGHT PASSWORD TO PURGE THE REQUESTED FILE.

\$ARGUMENT ERRORS

INDICATES ERROR IN THE CALLING OF RFM.

\$ILLEGAL FILE NAMES

INDICATES THAT THE USER IS PEQUESTING RFM ACCTION ON A RESERVED FILE NAME. (RAVAL AND/OR RAINF)

\$AN NOT A LEGAL RESIDENT FILES ACCOUNTS

INDICATES THAT AN IS NOT VALIDATED AS A RESIDENT FILES ACCOUNT.

\$RFM VALIDATION ERRORS

INDICATES THAT THERE IS AN ERROR ON PFMVAL. REQUEST THE SYSTEM PROGRAMMER IN CHARGE OF RESIDENT FILE ACCOUNTS TO FIX IT.

NO 1434

```

* CALLPRG ON CPOPL CONTAINS THE SOURCE OF THE CALLPRG INDEX CARDS. EACH MODI-
* CATION TO DECK CALLPRG IS NAMED BY THE DATE ON WHICH THE MODIFICATION IS
* PERFORMED. DECK CALLPRG ALSO CONTAINS A CHRONOLOGICAL LIST OF REASONS FOR
* MODIFICATIONS TO THE CALLPRG INDEX CARDS. THIS LIST REFLECTS THE MODIFICA-
* TIONS TO CALLPRG PROGRAMS.

* TO OBTAIN A LIST OF CPOPL RUN THE FOLLOWING JOB:
* FETCH,CPOPL.
* MODIFY(P=CPOPL,C,Z)+*EDIT CALLPRG
* CS(COMPILE,OUTPUT)

* TO OBTAIN THE CALLPRG INDEX RUN THE FOLLOWING JOB:
* MODIFY(P=CPOPL,S,U,Z)+*DECK CALLPRG+*D 1,5+*D 6,7
* COPYU(SOURCE/R,CALLPRG/PR,0,1,X,*)
* CS(CALLPRG,OUTPUT)

* TO CREATE A NEW CPOPL RUN THE FOLLOWING JOB:
* FETCH,CPOPL.
* MODIFY(P=CPOPL,N=NCPOPL)
* 7/8/9
* /* THIS DECK MODIFIES CPOPL
* /*
* *INWIDTH 80
* /* CALLPRG INDEX CARDS CAN BE UP TO 80 CHARACTERS LONG
* *DECK CALLPRG
* *IDENT YYMMDD
* /*NOTE THAT THE IDENT OF EACH MODIFICATION IS THE DATE ON WHICH THE
* /* MODIFICATION IS MADE.
* /*
* *DELETE UUUVVW.W
*         YY/MM/DD.
* /* CHANGE INDEX DATE.
* /* SET DATE FOR WHEN THE NEW INDEX IS SCHEDULED IN PIA&SPF FORMAT.
* /* NOTE THAT THE IDENT OF THE DATE WILL CHANGE WITH EACH NEW INDEX.
* /*
* /* INSERT SUBMITTED DECKS WITH MODIFICATIONS AFTER THIS CARD.
* /*
* /* SAMPLE DECK 1
* *INSERT CALLPRG.6
* *.*
*     PRG1      YY/MM/DD.  NNNN
* *
*     SHORT DESCRIPTION OF PRG1
* *INSERT CALLPRG.7
* *
*     N. NNN
* *PRG1,UN=.....
* /*
* /* SAMPLE DECK 2
* *INSERT CALLPRG.6
* *.*
*     PRG2      YY/MM/DD.  MMMM
* *
*     REASON FOR CHANGING PR2
* *.*
*     PRG3      YY/MM/DD.  MMMM
* *
*     REASON FOR REMOVING PRG3
* *.*
*     PRG4      YY/MM/DD.  MMMM
* *
*     SHORT DESCRIPTION OF PRG4
* *DELETE XXYYZZ.I
* *PRG2,UN=.....
* *DELETE AABCC.J
* *INSERT PQQRR.K
* *PRG4,UN=.....
* /*
* *EDIT CALLPRG

* PROGRAMMER NNNN, THAT DOES NOT HAVE AN ENTRY ON THE CALLPRG INDEX IS OPEN
* ONE OF INSERTING HIS NAME AND THE

```

651495

\* PROGRAMER MMMM, THAT ALREADY HAS AN ENTRY WITH HIS NAME ON THE CALLPRG INDEX.  
 \* IS CHANGING PRG2, REMOVING PRG3 AND ADDING PRG4. HE ALSO INSERTS THE  
 \* COMMENT CARDS DESCRIBING THE CHANGES AFTER CALLPRG.6.  
 \* TO REQUEST MODIFICATIONS FOR THE CALLPRG INDEX SEND THE MODIFICATION  
 \* M. RIVIERE. CONSTRUCT THE MODIFICATION DECKS FOLLOWING THE DESCRIPTION OF  
 \* SAMPLE DECK 1 AND SAMPLE DECK 2 ON THE ABOVE EXAMPLE.  
 \* THE IDENTS CALLPRG.6 AND CALLPRG.7 WILL REMAIN THE SAME IN EVERY COPY

75/33/10.

\* M. MMMM  
 PRG2,UN=... (OLD VERSION)  
 UNPAGE,UN=YZE6008,PN=STF,IA=UNPAGE/UNPAGE.  
 LPKODE,UN=YZE6008,PN=STF,PW=EGGERS,IA=LPKODE.  
 DMPECS,UN=YZE6008,PN=STF,IA=DMPECS/DMPECS.  
 PSCLIB,TY=FETCH,UN=YZE6008,PN=STF,DA=PSCLIB.  
 LOADSD,UN=YZE6008,PN=STF,IA=LOADSD/LOADSD.  
 SMTEXT,TY=FETCH,UN=YZE6008,PN=STF,PW=CPTEXT,IA=SMTEXT/SMTEXT.  
 PRG3,UN=... (NO LONGER IN USE)  
 APEX,UN=YZE6003,PN=STF,PW=CLEPE,DA=APEX/APX1.  
 EISINFO,TY=FETCH,MT=L8007+EISINFO/RAKVEC.  
 EISPACK,TY=FETCH,UN=YZE6009,PN=STF,PW=CLEPE,UL=FT3LIB,DA=EISPACK/EIS1.

MODIFY(P=CPOPL,N=N,C=C)

MODIF

-IDENT- 750321

- 1 \*IDENT 750321
- 2 \*DELETE 750310.1
- 3 . 75/03/21.
- 4 \*INSERT CALLPRG.6
- 5 \*.\* PRG1 75/03/21. MNNN
- 6 \*. SHORT DESCRIPTION OF PRG1
- 7 \*INSERT CALLPRG.7
- 8 . N. MNNN
- 9 PRG1,UN=.... (NEW)
- 10 \*/
- 11 \*INSERT CALLPRG.6
- 12 \*.\* PRG2 74/03/21. MMMM
- 13 \*.\* REASON FOR CHANGING PRG2
- 14 \*.\* PRG3 75/03/21. MMMM
- 15 \*. REASON FOR REMOVING PRG3
- 16 \*. PRG4 75/03/21. MMMM
- 17 \*. SHORT DESCRIPTIN OF PRG4
- 18 \*DELETE CALLPRG.9
- 19 PRG2,UN=.... (NEW VERSION)
- 20 \*DELETE CALLPRG.16
- 21 \*INSERT CALLPRG.18
- 22 PRG4,UN=... (NEW PROGRAM)
- 23 \*/
- 24 \*EDIT CALLPRG

DECK STATUS AND MODIFICATIONS.  
DECK - CALLPRG

MODIF

CALLPRG MODIFIERS.

750310 CPOPL 750321

- \*WIDTH 80
- D . 75/03/10.
- A . 75/03/21.
- A \*.\* PRG2 74/03/21. MMMM
- A \*.\* REASON FOR CHANGING PRG2
- A \*.\* PRG3 75/03/21. MMMM
- A \*. REASON FOR REMOVING PRG3
- A \*. PRG4 75/03/21. MMMM
- A \*. SHORT DESCRIPTIN OF PRG4
- A \*.\* PRG1 75/03/21. MNNN
- A \*. SHORT DESCRIPTION OF PRG1
- A . N. MNNN
- A PRG1,UN=.... (NEW)
- D PRG2,UN=... (OLD VERSION)
- A PRG2,UN=.... (NEW VERSION)
- D PRG3,UN=.... (NO LONGER IN USE)
- A PRG4,UN=... (NEW PROGRAM)

104 ACTIVE CARD(S).

3 INACTIVE CARD(S).

661 99

. N. NNNN

PRG1,UN=.... (NEW)

. M. MMMM

PRG2,UN=.... (NEW VERSION)

UNPAGE,UN=YZE6008,PN=STF,IA=UNPAGE/UNPAGE.

LPKODE,UN=YZE6008,PN=STF,PW=EGGERS,IA=LPKODE.

DMPECS,UN=YZE6008,PN=STF,IA=DMPECS/DMPECS.

PSCLIB,TY=FETCH,UN=YZE6008,PN=STF,DA=PSCLIB.

LOADSD,UN=YZE6008,PN=STF,IA=LOADSD/LOADSD.

SMTEXT,TY=FETCH,UN=YZE6008,PN=STF,PW=CPTEXT,IA=SMTEXT/SMTEXT.

APEX,UN=YZE6008,PN=STF,PW=CLEPE,DA=APEX/APX1.

EISINFO,TY=FETCH,MT=LBC007+EISINFO/RAKVEC.

PRG4,UN=... (NEW PROGRAM)

EISPACK,TY=FETCH,UN=YZE6008,PN=STF,PW=CLEPE,UL=FT3LIB,DA=EISPACK/EIS1.

651506