SYSTEMATIC CONTROL OF LARGE COMPUTER PROGRAMS
CCL, UPDATE, AND FORTRAN PROCEDURES

by

J.P. Goedbloed and L. Klieb

Rijnhuizen Report 86-169

This work was performed as part of the research programme of the association agreement of Euratom and the 'Stichting voor Fundamenteel Onderzoek der Materie' (FOM) with financial support from the 'Nederlandse Organisatie voor Zuiver-Wetenschappelijk Onderzoek' (ZWO) and Euratom.
CONTENTS

I. Introduction 2
II. Survey of the package 6
III. Systematics of job and program control 11
IV. Index of files 20
V. Listing of the package 23/L.1
SYSTEMATIC CONTROL OF LARGE COMPUTER PROGRAMS

CCL, UPDATE, AND FORTRAN PROCEDURES

by

J.P. Goedbloed and L. Klieb
Association Euratom-FOM, FOM-Instituut voor Plasmafysica
Rijnhuizen, Nieuwegein, The Netherlands

ABSTRACT

A package of CCL, UPDATE, and FORTRAN procedures is described which facilitates the systematic control and development of large scientific computer programs. The package provides a general tool box for this purpose which contains many conveniences for the systematic administration of files, editing, reformatting of line printer output files, etc. In addition, a small number of procedures is devoted to the problem of structured development of a large computer program which is used by a group of scientists. The essence of the method is contained in three procedures N, R, and X for the creation of a new UPDATE program library, its revision, and execution, resp., and a procedure REVISE which provides a joint editor - UPDATE session which combines the advantages of the two systems, viz. speed and rigor.
I. INTRODUCTION

In this report a package of programs and procedures is described which facilitates the systematic handling of large scientific computer programs. These programs are usually written in standard FORTRAN 77 [1], which guarantees the portability of the codes, but the supporting control structures needed to operate these programs on a particular machine are inevitably system-dependent. In addition, these structures are usually extended with user-constructed procedures to suit the needs of a particular person. This rapidly leads to a situation where efforts are duplicated and systematic methods of control are not communicated because of the impossibility of exchanging working tools. The present report is an attempt to somewhat counterbalance this tendency, while at the same time exposing some methods which would be extremely effective when incorporated into a standard control language, if such a thing could ever be agreed upon.

We feel that, even though the package presented here is completely dependent on a particular computational environment, to be described below, yet it is useful to list all our tools (see Sec. V) in order to provide the full range of requisites involved in the operation of a large program written in FORTRAN. The typical user is a scientist whose main interest is in the results of his program, but who finds himself increasingly caught in what one of them has aptly phrased as "nursing my files" [2]. Since we have spent a considerable amount of time with this activity, it is hoped that some of the tools developed in the process will turn out to be useful for a larger public. This applies to two parts in particular: firstly, the exposition of the systematics of job and program control of Sec. III, and, secondly, those programs listed in Sec. V which are written in standard FORTRAN 77.

Of course, the package as a whole will be most useful for those computational scientists who live, happily or not, in the same or a similar computational environment as the authors. In order of decreasing generality this applies to the use of the following products of the Control Data Corporation (CDC) and the "Stichting Academisch Reken­centrum Amsterdam" (Foundation Computer Centre Amsterdam, to be abbreviated as SARA):

a) the CDC UPDATE utility [3] for maintaining and updating programs in compressed format on mass storage,

b) the CDC Network Operating System for Batch Environment NOS/BE [4] for the Cyber scalar computers, of which we employ a Cyber 750 (to be abbreviated as CY750 in the following),

c) the CDC Virtual Storage Operating System VSOS [5] for the Cyber vector
computers, of which we employ a Cyber 205 (to be abbreviated as CY205 in the following),
d) the SARA system of organization of files into what is called a Masterfile [6], i.e. a directory of many separate files stored on disk in one super-structure, and some other extensions to NOS/BE which were made at SARA,
e) the SARA editor SARED [7].
Facilities similar to those listed under a), d), and e) exist on most large computers, so that it should not present a major obstacle to translate the system-dependent parts in the package pertaining to updating, file organization, and editing for another system. However, the extensive use of the Cyber Control Language CCL [4] for the construction of job and program control procedures prevents an easy transfer from one system to the other (with the exception of the CDC NOS operating system which differs from the NOS/BE system, but which does incorporate CCL).

The specific configuration in which the present package has evolved is schematically depicted in Fig. 1.

![Fig. 1: Schematics of file handling](image)

Here, we have not hesitated to bring together entirely unequal partners in one picture in order to illustrate the basic situation. To the left three master-files (indicated by names starting with "MF") are shown which contain the sources of the main program HBT and of two auxiliary FORTRAN libraries HGOLIB and PPPLIB, whereas the two CDC computers at SARA are depicted on the right. The essential link between the two is established by means of a body of (mainly) CCL procedures contained in masterfile MFCCCL, the source of library CCLLIB. This controls the maintenance, updating, and execution of the main program HBT together with its libraries on either the CY750 or the CY205 computer (where it
should be remarked that the former serves as a front-end for the latter so that CCLLIB resides on the CY750).

From the scientific point of view our main interest is the program HBT (a program solving for equilibrium and stability of a High-beta Tokamak [8]), but for the purpose of the present report these three letters will merely indicate an arbitrary FORTRAN program which is to be controlled by means of the structure shown in the centre of Fig. 1. We note in passing that the CDC operating systems pose an upper limit of 7 characters for file names so that little freedom is left if one wishes to compose meaningful names having the name of the main program as a root (see Sec. III). Hence, the choice of very short program names. The FORTRAN library HGOLIB is a collection of subroutines, collected and partly written by Hans Goedbloed, which are called from HBT. It contains routines for printing arrays, conformal mapping of curves onto a circle, Fast Fourier transforms, eigenvalues of a real symmetric matrix, solving ordinary differential equations, etc. The FORTRAN library PPPLIB is the portable Plasma Physics Plotting Library which controls all plotting in HBT. PPPLIB has been fully documented in Ref [9], similar to the present report. Again, for the purpose of this report, the two names HGOLIB and PPPLIB will merely indicate arbitrary FORTRAN libraries which are needed for the operation of the main program.

The package of control procedures described in this report then consists of all files of MFCCL + those files of MFHBT, MFHGO, and MFPPP which belong to the controlling structure shown in Fig. 1. Evidently, the main sources of the latter three masterfiles do not belong to this structure so that their listing is omitted. Since the package has grown over a period of several years, the order of the files is somewhat arbitrary, although an attempt has been made to group files that logically belong together. Consequently, the very trivial procedures appear alongside the very substantial ones. In order to provide some guide here, Sec. II presents a survey of the different elements of the package. Section III then deals with the systematics of job and program control which constitutes the basic issue underlying the package. An index of all the files is provided in Sec. IV for the purpose of quick reference. Any further details desired can be extracted from the full listing of Sec. V, in particular from the COMMENT files which have been continually updated in the course of development of the package.

Finally, it is a pleasure to acknowledge the contribution of collaborators at SARA and our institute. In particular, we are indebted to Jacob Koot (SARA) who created the core of the REVISE package (files 72 - 77), to Uul Haan-
stra (SARA) who advised us on many questions concerning masterfiles and interactive procedures, to Hans Schrijver (Utrecht University) who initiated systematic file management at our institute and who wrote the program AUD (file 46), and to Dick Hogeweij who wrote the procedures 39 - 41 for communication with the local PDP 11/70 computer. The major motivation for the construction of the package has been the collaboration on the program HBT over many years with Jan Rem, Dick Hogeweij, and Rob Kleibergen of our institute and Ricardo Galvão (Instituto Pesquisas Espaciais, São José dos Campos, Brazil) and Paulo Sakanaka (University of Campinas, Brazil). The necessity of communicating changes of the program with a number of scientists, some of them located at a large distance, has posed the basic question about structured development of programs for which this report attempts to provide an answer.

REFERENCES

[1a] FORTRAN Version 5 Reference Manual, for use with the CDC NOS and NOS/BE systems (Control Data Corporation, Sunnyvale, California, 1983).
[1b] FORTRAN 2000 Version 1 Reference Manual, for use with the CDC VSOS system (Control Data Corporation, Sunnyvale, California, 1983).
II. SURVEY OF THE PACKAGE

In this section the overall structure of the package will be described. This structure is rather loose since files have been added in the course of time just when the need for a certain function arose. Broadly speaking, however, the procedures may be grouped into two categories which have been termed "conveniences" and "systematics". Here, "systematics" addresses the problem stated at the end of Sec. I (on how to systematically develop a program, while keeping it operational for a group of users), which will be described in detail in Sec. III, whereas "conveniences" is just anything else. A second division is connected with the particular computer on which a certain job is to be executed. Since at SARA the CY750 serves as a front-end for the CY205 this implies that the list of conveniences for the former is more extensive than that for the latter. Finally, a third distinction is concerned with the difference between non-interactive and interactive (i.e., having extended help facilities) CCL procedures. For purely accidental reasons (having to do with the later advent of both interactive CCL procedures and the CY205 computer at SARA), this division largely coincides with the second one. In the index of files of Sec. IV, the procedures within the range 1 - 48 are non-interactive, while those in the range 49 - 82 are interactive.

Based on these considerations, all the procedures are grouped in five categories A - E, which are again subdivided in smaller groups labelled with lower case letters. In brackets the index of the file in the list of Sec. IV is indicated so that the corresponding listing in Sec. V may be found easily.

A. Conveniences for use of the CY750

a) Systematic administration of masterfile contents (1, 83, 90, 93).

The contents of the four masterfiles MFCCCL, MFHT, MFHOG, and MFPPP shown in Fig. 1 have been systematically kept updated with a COMMENT file containing explanatory notes on each file added.

b) Maintaining the package itself (2-9, 31-33, 50-53, 82).

After creation of a new empty masterfile MFCCCL, the library CCLLIB is installed by means of the procedure NEWCCCL (2) which adds one procedure (e.g., NEWCCCL itself) to both MFCCCL and CCLLIB. Next, files may be added, replaced and deleted in both MFCCCL and CCLLIB by means of ADDCCCL (3), REPLCCCL (4), and DELCCCL (5). If this process of continuously adding and replacing files has created too much redundant space, the procedure COPYCCCL (6) is used to clear this. The procedures 7-9 exhibit information on attached libraries (using the library of system routines
PIASLIB), whereas the procedures 31-33 and 50-53 are used for purposes similar to the procedures 3-6 for masterfiles and libraries other than MFCCL and CCLLIB. The procedure ALIAS (82) has been used to remove all personal identifications from the present package. The procedure may be used in reverse on the present package to install new personal identifications by calling it with the desired parameters.

e) Editing aids (12 - 17, 19, 21).

Calling EDGY (12) gives access to the SARA editor SARED on the CY750. EDDY (14) does the same, but keeps the edit file as a permanent file on disk. This is important when large files have to be edited over a period of several days, so that line numbers may be kept. The procedures 13, 15, 16, 17, 19 just provide some options missing in SARED. DIFFER (21) may be used to compare the contents of two files, e.g., to find out whether two files are identical or to check whether editing has not produced spurious effects.

d) Formatting output files (23 - 28).

Printing of files on a line printer is a powerful aid in the development of programs, in particular when the output is well-organized on labelled pages. This has been provided in the procedures 23-28, of which FOUT (26) and ROUT (28) probably have been called more than any other procedure of CCLLIB during the development of this package. An example of the output of FOUT (in a slightly modified form) is the listing of Sec. V.

e) Program execution aids (11, 18, 20, 22, 29 - 30).

Of this group, RUN5 (22) is frequently used to interactively run FORTRAN programs, whereas RIN (29) and LOC (30) are used to fire batch jobs and to collect the output produced. The remaining procedures 11, 18, 20 do not really have a common denominator. NOTE (18) is frequently called from other procedures to display an error condition.

f) Communication with a local computer (39 - 41).

These procedures are used to transport graph (39) and text (41) files from the CY750 to a local PDP 11/70 computer, equipped with a Versatec plotter and a Daisy-wheel printer. The corresponding programs to accept this on the PDP computer are not listed here. Examples of the plotted output may be found in Ref [9], whereas the listing of Sec. V is an example of the printed output.

g) System information (42 - 45, 98 - 99).

The procedure ZZSYSL (44) contains the most recent date of logging in. This procedure is automatically generated by a call of the public initialization procedure INIT on permfile PROCFIL,ID=PUBLIC (98) with "BEGIN,,,XXIDX,HF,..." at the beginning of a session. The latter call induces a call of the private procedure INIT on permfile PROCFIL, ID=XXIDX (99). In the latter procedure the user may insert all kinds of
convenient initialization statements like attaching the library CCLLIB and a specified masterfile MF,..., but also a call of ZZSYS1, which in turn calls ZZSYS2 (45), which then generates a new ZZSYS1 with the current date, so that the circle is closed. The purpose of this trick is to have ZZSYS2 exhibit only those system bulletins that have been changed since the previous session. SYS (43) has a similar purpose, except that the period is fixed to one week. DT (42) shows the date and time.

h) File information (10, 46 - 49).
This group of procedures informs the user about the presence of system files (10), about the permanent user files (46-48), and takes action to conserve the latter ones (49). It should be noted that the program DIR(47) exhibits the contents of masterfiles, but only when their names start with "MF".

B. Systematics for job control on the CY750

a) Creation of the supporting FORTRAN libraries (34, 91, 94, 96).
The procedure NEW (34) may be used to create the libraries HGOLIB and PPLLIB on the CY750. A call of NEW with the parameter HGO will launch the job NHGO (91) from MFHGO to create HGOLIB, whereas a call with PPP will launch the job NPPP (94) from MFPPP to create PPLLIB. The latter call also requires the UPDATE modification deck MPPP10 (96), which is included here since it is needed in addition to the source PPP10 listed in Ref. [9] to compile PPLLIB on the CY750.

b) Control of the main program (35 - 38, 84 - 86).
Similar to the procedure NEW for libraries, the procedure N (35) creates a new UPDATE program library for the main program HBT and compiles an executable binary by launching the job NHBT (84) from MFHBT. The procedure R (36) revises the UPDATE program library and compiles a corresponding binary by means of the job RHBT (85) from MFHBT. Finally, the procedure X (37) executes the binary by means of the job XHBT (86) from MFHBT. This method of working on the three levels embodied in the procedures N, R, and X constitutes the core of our systematic job control, which is discussed in detail in Sec. III.C.

C. Conveniences for computing on the CY205.
a) File management (54 - 61).
The procedures ADD205 (54) and GET205 (56) control the transport of files from the CY750 to the CY205, and vice versa, whereas DEL205 (55), RNM205 (57), and PER205 (61) change the status of existing files on the CY205. AUD205 (58) and ATT205 (59) serve a similar purpose as the corresponding procedures 46 and 49 for the CY750, viz. to provide information on the permanent user files on the CY205 and to take action to conserve them.
b) Program execution (62–64, 80–81).

The procedure RUN205 (80) is used to compile and execute an arbitrary FORTRAN program. If a plot file is produced (through PPPLIB) this has to be converted with PLT205 (64) to a graph file that can be visualized by means of the system program GRIMAS on the CY750. Jobs for the CY205 can be submitted with RIN205 (63), their fate may then be followed with Q205 (62), whereas the resulting output on the CY750 front-end can be collected again with LOC (30). The procedure VAST205 (81) serves to assist in the vectorization of FORTRAN 200 programs.

D. Systematics for job control on the CY205

a) Creation of the supporting FORTRAN libraries (65, 92, 95, 97).

The procedure NNEW (65), the job NNEWG (92) of NNEWG, and the job NNP (95) and the modification deck NNP10A (97) of NNP are used to create the libraries HGOLIB and PPPLIB on the CY205. This process is analogous to the one discussed under B.a), except that it takes place in two steps: an interactive UPDATE part on the CY750 front-end and a batch job on the CY205. Notice the use of double initial letters to distinguish CY205 procedures from the corresponding CY750 ones.

b) Control of the main program (66–68, 87–89).

Job control for the CY205 is completely analogous to that for the CY750 discussed under B.b), except for the split in interactive UPDATE part on the CY750 and batch job on the CY205, mentioned above. Hence, again three procedures NN (66), RR (67), XX (68) and three corresponding jobs NNHBT (87), RRHBT (88), XXHBT (89) for the creation, revision and execution of the main program.

E. Systematics for program updating

a) UPDATE procedures (69–71).

The three interactive UPDATE procedures NU (69), RU (70), and SU (71) may be exploited to create a new UPDATE program library, to revise an existing one, and to retrieve the source. These three functions are all that is needed for the systematic control of program development by means of UPDATE. Clearly, NU and RU just correspond to the UPDATE parts of the composite procedures NN (66) and RR (67) discussed above whereas RUN205 (80) corresponds to the batch job part.

b) The REVISE package (72–79).

This package, which may be installed as a separate library by means of the procedure INSTAL (72), has been developed in order to resolve the following dilemma (to be discussed in Sec. III.A): program development by means of a modern editor is fast but risky, program development by means of UPDATE is rigorous but time-consuming. The procedure REVISE (73) combines the positive features of speed and rigor, while avoiding the negative ones. This is accomplished by a joint editor – UPDATE
session in which the editor is used to create program changes, whereas
the corresponding UPDATE modification deck is automatically created
afterwards by means of the program MODGEN (77). This closing piece of
our systematics will be discussed in detail in Sec. III.D. The
programs USL (78) and UML (79) serve to rearrange the output of UPDATE
creation and correction runs, respectively, to a more compact format
with editor line numbers added. An example of the output produced by
USL is the listing of PPP10 in Ref. [9].
III. SYSTEMATICS OF JOB AND PROGRAM CONTROL

A. Editing or updating?

The package of procedures presented in this report may be considered as a general tool box which is convenient for operating jobs and programs on a large computer system. In addition, this package also contains an inner core consisting of a restricted number of procedures which embody a systematic method of working with a large computer by a group of scientists. This method is the subject of the present section.

Consider the basic problem: Over a number of years a large computer program has been developed (in our case: HBT) which is used and further developed by a group of scientists, possibly located at different places. Let the present version of the program be indicated by the number 45 so that this source will be called HBT45. Suppose one member of the group is interested in investigating a specific physical phenomenon which requires a modification of the source code. The most straightforward manner for him to proceed is to create a new source HBT46 with the required properties by means of the editor (upper part of Fig. 2). This method is fast and permits one to create many changes of the same kind at one stroke. In addition, attention may be paid to layout, comments, and other cosmetic features. However, there are distinct disadvantages associated with this way of proceeding: 1) one quickly loses the possibility of backtracking so that the communication between the members of the group (which is defined by the fact that they are working with the same

![Diagram](image)

Fig. 2: Editing and updating programs
program) is impeded, 2) inadvertently, alongside the intended improvements of
the code, trivial errors may have been introduced with the result that the new
version of the code should be considered inferior to the old one. Consequently,
the speed of this approach usually turns out to be illusory: much of the time
gained originally is lost afterwards in tracing errors, the least exciting part
of computing!

A solution to the above problem is to systematically keep existing
code, which has been sufficiently tested and agreed upon by the members of the
group, separate from modifications, which may be tentative or of individual
interest only. A nice way of implementing this idea is through the use of the
CDC UPDATE utility (Ref. [3] of Sec. I), which is schematically indicated in
the lower part of Fig. 2. Rather than modifying the source of the program
directly, as in the upper part of Fig. 2, UPDATE operates on an image of the
source which is called the UPDATE program library (which we indicate with the
prefix U). Such a program library consists of images of the source lines in
compressed format, supplemented with sequence number and correction history
information for each line. A program library is created by an UPDATE creation
run (HBT45 ---> UHBT45); its contents may be changed in a subsequent correction
run (UHBT45 + MHBT46 ---> UHBT46). In the latter, the old program library
(UHBT45) is updated by means of a modification deck (MHBT46) consisting of
UPDATE directives which, in essence, only delete or insert full lines. The
resulting new program library (UHBT46) will then be equivalent to a new source
(HBT46), plus information enabling one to undo the modifications so that the
old source(s) may be recovered again.

Fig. 3 provides an illustration of the basic steps:

a) In order to obtain a source for the UPDATE program library UHBT45, the
original code is divided into separate decks (usually corresponding to the
different subroutines) preceded by a *DECK directive. This induces the UPDATE
line identification, which is shown for the particular subroutine CYL: CYL.1
(the added *DECK line), CYL.2, ..., CYL.3. [We note in passing that a similar
UPDATE directive exists for labeling COMMON blocks, viz. the *COMDECK direc-
tive. Together with the *CALL directive, which inserts the contents of the
specified comdeck at a particular place, this cures one of the deficiencies of
FORTRAN 77, viz. the need to repeat COMMON blocks in full for every pertinent
subroutine.]

b) The subroutine CYL (which solves two ordinary first order differential
equations for the quantities \( Y_1 \) and \( Y_2 \)) is subsequently modified to investigate
the effects of a better estimate of the initial data for \( Y_1 \) and \( Y_2 \). The
required modification deck MHBT46 consists of a *IDENT directive which provides the name (MOD46) for the modification set, followed by optional comment lines (for mnemonic purposes only) preceded by the */ directive, and a number of delete (*D) and insert (*I) instructions with the new FORTRAN statements to be included in the new program.

c) The resulting modified code on the new program library UHBT46 will then consist of a mixture of original lines (labeled CYL...) and modified lines (labeled MOD46...), which are clearly distinguished by the UPDATE line numbers. In addition, the deleted lines (CYL.19,20) are still present as inactive lines within the program library so that they may be recovered, if desired.
Usually, this process of modifying the code continues until a definite new level has been reached or until the modification deck has grown to an awkward size. At that moment, an exchange between the members of the group takes place, in which a new standard for the program is agreed upon which incorporates all those changes which constitute well-tested improvements. After that, the process may start all over again.

An obvious objection against the use of UPDATE is that it is much slower in human resource time than directly producing the changes desired by means of an editor. In addition, it is much less flexible since it allows operations on full lines only (dating back to the time when the use of punched cards made the operations of deletion and insertion of cards the fundamental ones). This objection would appear to be prohibitive for the use of UPDATE at the present time. However, this defect has been cured in the procedure REVISE which enables one to use a fast editor and, yet, to conserve the advantages of a rigorous bookkeeping of modifications by means of UPDATE. This will be discussed in Sec. III.D.

In conclusion: The use of REVISE provides the necessary tools to overcome the dilemma stated at the beginning of this section and illustrated in Fig. 2. The editor is used to create program changes (HBT45 → HBT46), but a background system generates the corresponding UPDATE modification deck (MHBT46) needed to transform the UPDATE program library (UHBT45 → UHBT46). Hence, although the evolving program itself is employed in the editor, the UPDATE modification deck is considered to be the carrier of the evolution.

B. Naming conventions

Since working with large programs usually induces an avalanche of secondary files, it is useful to pay some attention to file names in order not to be lost in a mass of unintelligible names. Here, too, it pays off to adopt a systematic method of working.

It is logical to start from the name of the main program (HBT, in our case) and to use this as a root for the composition of names for associated files. We will exploit one-letter prefixes to distinguish the different kinds of files and suffixes of three characters to label the different files of one kind. E.g., BHBT46 will be the compiled file nr. 46 of the program HBT. In a CCL procedure, such a composite file name can be produced from the expression #B_FN_B, if FN and B are keywords of the procedure which are substituted by specified values FN=HBT and B=46 upon execution. (Here, #B signifies that this B is not to be substituted and _ signifies that the symbols to the left and to
the right are to be concatenated.] Notice that we use the letter B here for two purposes, viz. to indicate the constant prefix and the variable keyword.

The following prefixes have been exploited for the composition of meaningful file names:

- **B**: executable binary file (produced by the FORTRAN compiler),
- **C**: UPDATE COMPILE file (input for the FORTRAN compiler),
- **D**: data file (to be processed by auxiliary programs),
- **I**: input file (in NAMELIST format),
- **M**: UPDATE modification deck,
- **O**: output file (permanent file, to be distinguished from OUTPUT),
- **P**: plot file (to be processed by plot facilities),
- **S**: source file (suppressed when used as a prefix),
- **U**: UPDATE program library.

In addition, the following prefixes have been used to indicate batch jobs:

- **N**: creation of a new UPDATE program library and compilation of an associated binary,
- **R**: revision of an old UPDATE program library and compilation of an associated binary,
- **X**: execution of a binary with given input.

These letters are also used as names for the three corresponding procedures which will be discussed below.

It is to be noted that, according to our conventions, the job executing the binary file BHBT46 with input file IHBT60 would have to be named XHBT4660 (obtainable from a CCL expression X_FN_B_I). However, this file name exceeds the 7 characters permitted by the NOS/BE operating system (Ref. [4] of Sec.I). We have resolved this conflict by dropping the root file name (HBT) for job names, so that this job is now called X4660 (from X_B_I). For consistency, the UPDATE creation and correction runs preceding this job also have contracted names like N45 (from N_S) and R46 (from R_M). Thus, we have obtained a consistent way of labeling all occurring files with unique names not exceeding 7 characters.

**C. Job Control**

From the discussions of Secs. III.A and B it follows that a systematic way of developing and executing a large computer program leads to a distinction of three basic steps, embodied by the three CCL procedures N, R, and X. (See listings on pages L.22-24). Starting from a new source, a new UPDATE program
library is created through the procedure $N$. This initial step is then followed by many steps of alternating revisions by means of the procedure $R$ and executions by means of the procedure $X$. Of course, the relative use of $R$ and $X$ depends on whether the program is in a phase of development (more use of $R$) or production (more use of $X$).

The operation of the job control procedures is illustrated in Fig. 4 which shows the sequence of steps resulting from the three following calls, resp.:

"$N$, HBT, S=45." (producing the job N45),

"$R$, HBT, U=45, M=46." (producing the job R46),

"X, HBT, B=46, I=60, O, P." (producing the job X4660).

The first result of a call of the procedure $N$, $R$, or $X$ is the extraction from the masterfile $MFHBT$ of the pertinent file $NHBT$, $RHBT$, or $XHBT$, containing a framework for the job which is turned into the actual job $N45$, $R46$, or $X4660$ by automatic substitution of the specified parameters. Here, the job $N45$ collects the source $HBT45$ from the masterfile $MFHBT$, starts an UPDATE creation run to create a new program library $UHBT45$ and COMPILE file $CHBT45$ (having the COMMON blocks inserted at the proper positions), and instructs the FORTRAN compiler to compile an executable binary $BHBT45$. Next, the job $R46$ collects the modification file $MHBT46$ from the masterfile $MFHBT$, starts an UPDATE correction run to revise the old program library $UHBT45$ resulting in a new program library

---

Fig. 4: Schematics of job control (procedures $N$, $R$, and $X$)
UHBT46 and COMPILE file CHBT46, which is processed by the compiler to produce a new binary BHBT46. Finally, the job X4660 collects the input file IHBT60 from masterfile MHBT and executes the binary BHBT46, which produces the permanent output file O4660 and the plot file P4660 on disk.

Having introduced the necessary semantics, we may now sketch the sequence of steps during the actual development of a program. After a redefinition of the source of the program, one initializing run X465 is made. This run may be followed by a series of revisions R46, R46A, R46B, ..., where REVISE is the appropriate tool for the creation of the necessary modification decks MHBT46, MHBT46A, MHBT46B, .... Of course, all these revisions are to be tested, e.g., by means of a test input file IHBT1, so that the revisions are accompanied by testruns X461, X46A1, X46B1, .... When this period of development has led to a satisfactory revision, say R46F, a production period may follow during which the actual physical problem is investigated by means of runs X46F60, X46F61, ....

With respect to file organization, it is to be noted that we have chosen to store the sources of the program, the modification decks and the input files in one masterfile (MHBT), whereas the UPDATE program libraries and the binaries are kept as separate permanent files on disk. The reason for this distinction is the easy protection of the contents of masterfiles by means of tape dump facilities (see Ref. [6] of Sec. I). Thus, a regular back up on tape is maintained for the contents of masterfiles only. If all disk files would get lost, it would be an easy matter to reconstruct them from the masterfiles residing on tape.

Finally, similar methods have been employed for the creation and revision of libraries by means of the procedure NEW, as mentioned in Sec. II.B.a). Also, analogous procedures NNEW, NN, RR, and XX have been developed for the CY205, as mentioned in Sec. II.D. The use of double prefixes to distinguish files for the CY205 from files for the CY750 have posed some additional problems with respect to file names, which have been solved by some further tricks. For details the reader is referred to the listings of Sec. V.

D. Program control

Let us now return to the editor - UPDATE dilemma of Sec. III.A. It is clear that the procedures N and R constitute the essence of our method of systematic program development, where the procedure N is used only to redefine the standard of the source of the program, whereas the procedure R is used all
the time. This is illustrated once more in the upper part of Fig. 5, where the prefixes defined in Sec. III.B are now used as abbreviations for the file names. Here, every "box" indicates a basic step in program development by means of either the procedure N or the procedure R. Program evolution is controlled in this case by the modification deck M which transforms the UPDATE program library U into U'. On the other hand, if we were to consider program development through the evolution $S \rightarrow S'$ of the source, preferentially using the editor (vertical box of the middle part of Fig. 5), the use of UPDATE would have to be restricted to the procedure N alone (horizontal boxes). However,
this would exclude a meaningful use of UPDATE (apart from the facility of COMMON block inclusion) since no method would be built in to keep track of the changes of the source. Evidently, the essential missing link here is a program which compares the two different sources $S$ and $S'$ and creates the corresponding UPDATE modification deck $M$ after the editing session. This link is provided by the FORTRAN program MODGEN which is the central constituent of the REVISE package (lower part of Fig. 5, where the box includes the editing session and the UPDATE revision $R$ without compilation of the binary $B'$).

As indicated in the lower part of Fig. 5, REVISE retrieves the complete source $S$ or some decks from the old UPDATE program library $U$ (step 1) after which the editor is entered (step 2). The editing session proceeds as usual, except that an exit automatically starts the program MODGEN to generate the UPDATE modification set $M$ (step 3). Finally, a second UPDATE run produces the new program library $U'$ (from $U$ and $M$) and the new COMPnle file $C'$ (step 5). The latter file may then be presented to either one of the compilers (FTN5 for the CY750 or FTN200 for the CY205) to complete the cycle corresponding to the procedures $R$ or RR.

REVISE may also be used after the completion of many editing sessions, possibly stretching over a period of months or even years, as long as the original UPDATE program library $U$ has been kept. In step 2 one then simply inserts the new source $S'$ directly into the edit file and immediately leaves the editor again. The rest of the procedure is identical to that described above so that a modification deck $M$ is now obtained which reflects the evolution of the program over the past years.

Hence, by means of REVISE we may exploit all the advantages of using a fast editor, without losing the systematics provided by UPDATE. It is our experience in the operation of large computer programs that this systematics is imperative to prevent mistakes and to communicate changes to the other users of the program. It should be stressed, however, that this method is useful also in the absence of a group of other users, since it allows one to backtrack to an older version of the code or to keep several similar versions simultaneously operational.
IV. INDEX OF FILES

A complete list of the names of the different procedures and programs is presented here together with a short description of their use. For more extensive explanations the reader is referred to the respective COMMENT files and the commentary parts of the files themselves.

Masterfile MFCCL:

1. COMMENT - explanatory notes on the files of MFCCL  
   page L. 1
2. NEWCCL - create library CCLLIB  
   L. 6
3. ADDCCL - add file to masterfile MFCCL and library CCLLIB  
   L. 7
4. REPCCL - replace file in masterfile MFCCL and library CCLLIB  
   L. 8
5. DELCCL - delete file from masterfile MFCCL and library CCLLIB  
   L. 8
6. COPYCCL - copy masterfile MFCCL and library CCLLIB  
   L. 9
7. LISTLIB - list contents of an attached library  
   L. 9
8. RELIST - reformat output of LISTLIB (auxiliary program for LISTLIB)  
   L. 9
9. LIBLIST - exhibit attached libraries  
   L.10
10. ZZFILES - exhibit attached system files  
    L.10
11. COST - exhibit system seconds used since last call of COST  
    L.10
12. EDCY - edit with SARA editor SARED on the CY750  
    L.10
13. HEAR - rearrange lengthy comment lines (auxiliary program for EDCY)  
    L.11
14. EDDY - edit with SARED on the CY750, keeping permanent edit file  
    L.11
15. FED - produce format files for use in SARED  
    L.12
16. STRIP - strip last columns from a file (using SARED)  
    L.12
17. DCL - delete columns from a file (FORTRAN program)  
    L.12
18. NOTE - exhibit message  
    L.13
19. COUNT - exhibit column numbers  
    L.13
20. CALC - FORTRAN pocket calculator  
    L.14
21. DIFFER - compare two files  
    L.14
22. RUN5 - FTN5 compilation and execution of a program  
    L.14
23. WPD - write file on OUTPUT, adding line numbers (using SARED)  
    L.15
24. FPRINT - print file at line printer  
    L.15
25. PAG - write file in labelled pages on OUTPUT (auxiliary program for FPRINT)  
    L.15
26. FOUT - write files of attached masterfile on OUTPUT  
    L.16
27. UFOUT - write UPDATE source files of attached masterfile on OUTPUT  
    L.17
28. ROUT - route file OUTPUT to line printer  
    L.19
29. RIN - route job to the input queue of the CY750  
    L.19
30. LOC - make remote output file local  
    L.19
31. COPYMF - copy masterfile to eliminate redundant space  
    L.20
32. UPCOM - update COMMENT file with new CY and date (auxiliary program for COPYMF)  
    L.20
33. DU - duplicate permanent file to new ID  
    L.21
34. NEW - create a new FORTRAN library on the CY750  
    L.21
35. N - create a new UPDATE program library and binary
36. R - create a revised UPDATE program library and binary
37. X - execute binary with given input file
38. JOBCRD - rewrite job card (auxiliary program for X)
39. PLOUT - send graph file from CY750 to local plotter
40. SPLIT - convert G-coded files (auxiliary program for PLOUT)
41. TOUT - send text file from CY750 to local computer
42. DT - display date and time
43. SYS - exhibit recent changes of the system and user bulletins
44. ZZSYS1 - exhibit changes of system bulletins since previous log in
45. ZZSYS2 - auxiliary procedure for ZZSYS1
46. AUD - exhibit compact AUDIT of all permanent user files
47. DIR - exhibit the contents of the masterfiles shown in AUDIT
48. PASAUD - reformat system AUDIT (auxiliary PASCAL program for AUD)
49. ATT750 - attach all CY750 user files (auxiliary program: KEEP)
50. ADDP - add a procedure to a library
51. DELP - delete a procedure from a library
52. GETP - get a procedure from a library
53. REPPI - replace a procedure from a library
54. ADD205 - add permanent file on the CY205
55. DEL205 - delete permanent file from the CY205
56. GET205 - get permanent file from the CY205
57. RNM205 - rename permanent file on the CY205
58. AUDP205 - exhibit AUDIT of all permanent user files on the CY205
59. ATT205 - attach all CY205 user files (auxiliary program: SAVE)
60. BUD205 - exhibit the budget left on the CY205
61. PER205 - permit a second user to access CY205 files
62. Q205 - exhibit user jobs in the queues of the CY205
63. RIN205 - route job to the input queue of the CY205
64. PLT205 - convert plot file from the CY205 to graph file for the CY750
65. NNEW - create a new FORTRAN library on the CY205
66. NN - create a new UPDATE program library on the CY750 and compile a corresponding binary on the CY205
67. RR - revise an existing UPDATE program library on the CY750 and compile the corresponding binary on the CY205
68. XX - execute a binary on the CY205 with given input
69. NU - create new UPDATE program library on the CY750
70. RU - revise existing UPDATE program library on the CY750
71. SU - retrieve source from an existing UPDATE program library
72. INSTAL - install the package of UPDATE procedures 73-79 as a separate library
73. REVISE - edit decks of an UPDATE program library and automatically produce the resulting modification deck
74. ASXDOE - ask permission to delete old edit file (auxiliary procedure for REVISE)
75. ASKDECK - ask deck names to be edited in REVISE
76. MAKEFIL - auxiliary program for REVISE
77. MODGEN - compare two files and generate the corresponding UPDATE modification deck
78. USL - reformat UPDATE source listing of a creation run
79. UML - reformat UPDATE modification listing of a correction run
80. RUN205 - FTN200 compilation and execution of a program
81. VAST205 - run program VAST (Vector and Array Syntax Translator) on the CY205
82. ALIAS - transfer files from one master file to another while changing all personal identifications

Masterfile MFHBT:
83. COMMENT - explanatory notes on the files of MFHBT
84. NHBT - framework for a job fired by procedure N of MFCCCL
85. RHBT - framework for a job fired by procedure R of MFCCCL
86. XHBT - framework for a job fired by procedure X of MFCCCL
87. NNHBT - framework for a job fired by procedure NN of MFCCCL
88. RRHBT - framework for a job fired by procedure RR of MFCCCL
89. XXHBT - framework for a job fired by procedure XX of MFCCCL

Masterfile MFHGQ:
90. COMMENT - explanatory notes on the files of MFHGQ
91. NHGQ - framework for a job fired by procedure NEW of MFCCCL
92. NNHGQ - framework for a job fired by procedure NNEW of MFCCCL

Masterfile MFPPP:
93. COMMENT - explanatory notes on the files of MFPPP
94. NPPP - framework for a job fired by procedure NEW of MFCCCL
95. NNPFP - framework for a job fired by procedure NNEW of MFCCCL
96. MPPPLIB - UPDATE modification deck to produce plotting library PPPLIB on the CY750
97. MPPPLIBA - UPDATE modification deck to produce plotting library PPPLIB on the CY205

Permanent files:
98. PROCFLD.PUBLIC - public initialization procedure INIT
99. PROCFLD.XXXIDX - private initialization procedure INIT
V. LISTING OF THE PACKAGE

A complete listing is provided of all the source files of the library CCLLIB, which are kept in the masterfile MFCCL (pages L.1 - 70) + those files of the masterfiles MFHBT (pages L.71 - 73), MFHGO (page L.74), and MFPPP (pages L.75 - 76) which belong to the controlling structure shown in Fig. 1. In addition, the initialization procedures on the permanent files PROCFIL (page L.77) are also listed. Notice that all personal identifications have been removed from the package by means of the procedure ALIAS (page L.69). To obtain a working package, this procedure should be used in reverse to install the pertinent identifications.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. -------- COMMENT, 1 --------

COMMENT.
MASTERFILE MFCCL CY=30
01/07/86
********
MASTERFILE MFCCL CONTAINS THE SOURCES OF THE CCL PROCEDURES AND
FORTRAN PROGRAMS CORRESPONDING TO LIBRARY CCLLIB FACILITATING
INTERACTIVE WORK ON THE SARA CYBERS.
********
FILES:
COMMENT - THIS FILE.
NEWCCL - CREATING NEW LIBRARY CCLLIB, ADDING A FIRST FILE FN WHICH
SHOULD BE A CCL PROCEDURE.
ADDCCM - ADD FILE FN TO LIBRARY CCLLIB.
REPCCL - REPLACE OLD FILE FN BY NEW ONE IN LIBRARY CCLLIB.
DELCCL - DELETE FILE FN FROM LIBRARY CCLLIB.
COPYCCL - COPY MFCCL WITH MFCOPY AND LIBRARY CCLLIB WITH COPYLIB
LISTLIB - LIST CONTENTS OF LIBRARY LIB WITH LISTLIB OF PIASLIB USING
PROGRAM RELIST.
LIFILES - LIST ATTACHED LIBRARIES WITH LIFILES OF PIASLIB.
REAR - PROGRAM REARRANGING LENGTHY COMMENT LINES TO
72 CHARACTERS, CALLED FROM FILE B, WHICH IS PRODUCED FOR USE IN AN
INTERACTIVE SESSION OF SAURED STARTED BY PROCEDURE EDCY.
--- MASTERFILE MFCCY-30 --- 02/07/86 - 00:10:44. ------- COMMENT, 2 -------

EDDY - EDIT WITH SARED ON THE CYBER INTERACTIVELY.

EDFILE IS KEPT AS PERMANENT FILE EDFILE.
CALL: "EDDY,EDFILE,(FIRST)."

EDFILE IS CREATED BY SPECIFYING "FIRST" IN THE FIRST CALL OF
EDDY. THE SARED COMMANDS "E", "C", "B,D" DO NOT AFFECT THE
CONTENTS OF EDFILE, UNLESS IT HAS BEEN CLEARED COMPLETELY
WITH "EDL". HENCE: AVOID THE USE OF THESE FOUR COMMANDS
AND ONLY USE "ADD", "INSERT", AND THE USUAL STRING
REPLACEMENT COMMANDS.

FED - PRODUCE FORMAT FILE FOR USE IN SARED.
CALL FROM SARED:
"!FED,FF,", "USE,FF, " FOR EDITING FORTRAN FILES,
"!FED,FN,", "USE,FN, " FOR EDITING OTHER FILES.

STRIP - STRIP LAST COLUMNS (DEFAULT: LINE NUMBERS)
AND TRAILING BLANKS FROM A FILE.
CALL: "STRIP,FN,(L,...)."

DCL - DELETE COLUMNS 11 TO 12 OF FILE FN1 AND WRITE THE RESULTING
FILE ON FN2.
CALL: "DCL,FN1,FN2,(I1,I2)."

IF FN1=FN2 THE ORIGINAL FILE FN1 IS OVERWRITTEN WITH FN2.
IF I2 IS OMITTED ONLY COLUMN 11 WILL BE DELETED.

NOTE - NOTE "MESSAGE" ON FILE L (DEFAULT CONNECTED).
CALL: "NOTE,FN,MESSAGE(L=,...)."

DEFAULT: "MESSAGE" APPEARS ON FILE OUTPUT (UNLESS L=...).
WHICH IS CONNECTED (UNLESS DISCONNECT).

COUNT - COUNT COLUMN NUMBERS ON THE SCREEN.

CALC - FORTRAN POCKET CALCULATOR.
CALL: "CALC,SANY FORTRAN EXPRESSIONS."

DIFFER - COMPARE FN1 AND FN2 WITH PROGRAM Diff OF LIBRARY PROGS.
DICK WINTER (MC). DIFF IS WRITTEN IN A SPECIAL ASSEMBLER
LANGUAGE THAT MAY NOT BE SUPPORTED BY THE SYSTEM
SOME FUTURE DATE.
CALL: "DIFFER,FN1,FN2,(L,...)."

DEFAULT: LISTING APPEARS ON CONNECTED FILE OUTPUT
(UNLESS L=...).

RUNS - FTNS COMPILATION AND EXECUTION OF FILE.
CALL: "RUNS,FN,(B=,...).OPT=,...,L=,...,KEEP,PMO,NODEX)."

DEFAULT: OPT=2 (UNLESS OPT=...), NO LISTING (UNLESS L=...).
WHEN OUTPUT APPEARS ON FILE LIST, PREVIOUS OUTPUT IS
NOT KEPT (UNLESS KEEP), NO POST MORTEM DUMP (UNLESS
PMO), BINARY IS IMMEDIATELY EXECUTED (UNLESS NOEX).

WFO - WRITE FILE FN ON OUTPUT USING SARED.
CALL: "WFO,FN,(NOL)."

DEFAULT: ADDING LINENRS (UNLESS NOL).

FPRINT - WRITE FILE FN ON OUTPUT WITH PROGRAM PAG.
CALL: "FPRINT,FN,(CY=,...)."
CY MAY BE OMITTED FOR LOCAL FILES.

PAG - PROGRAM WRITING ATTACHED FILE FN IN NUMBERED PAGES ON
OUTPUT.
CALL: "PAG,FN,(LEFT TEXTS,RIGHT TEXTS,N1=,...,N2=,...)."
LEFT TEXTS AND RIGHT TEXTS APPEAR IN THE HEADER.
ALL LINES OF FN ARE PRINTED (UNLESS N1=,...,N2=,...).

FOUT - PROGRAM WRITING FILES OF ATTACHED MASTERFILE ON OUTPUT.
CALL: "FOUT,FN1/FN2/..."

UFOAT - WRITE UPDATE SOURCES FROM ATTACHED MASTERFILE ON OUTPUT.
CALL: "UFOAT,FN1/FN2/..."

ROUT - ROUTE FILE OUTPUT TO LINEPRINTER.
CALL: "ROUT,(OUTPUT),(SHIFT,T10,FID,IC)."

DEFAULT: FILE IS OUTPUT (UNLESS "OUTPUT" IS SPECIFIED
DIFFERENTLY) WITH CONTROL CHARACTERS IN COLUMNS (UNLESS
SHIFT) AND IS WRITTEN IN DISPLAY CODE (UNLESS IC).
BANNER IS LABELLED WITH FID=XXIDX/XXIIX.

RIN - ROUTE JOB TO INPUT QUEUE.
CALL: "RIN,JOB,(TID=,...,FID=,...)."

OUTPUT APPEARS AT TERMINAL (UNLESS TID).

LOC - MAKE LOCAL OUTPUT FILE ZZ LOCAL UNDER THE NAME ZZ.
PAGE, AND ROUTE TO THE LINEPRINTER (WHEN GIVEN PERMISSION).
CALL: "LOC,ZZ."

COPYMF - COPY MASTERFILE MF, CALLING PROGRAM UPCOM.
CALL: "COPYMF,MF,EX=,...."

ALL FILES OF MF ARE COPIED (EXCEPT EX=...).

UPCOM - PROGRAM UPDATING COMMENT FILE OF A MASTERFILE WITH A NEW
CY AND DATE.

DU - DUPLICATE PERMFILE FN OF ID=XXIDX TO ID=XXID.
CALL: "DU,FN,(CY=,...,NID=,...)."

DEFAULT: NID=XXXXX.

NEW - PRODUCE JOB NFN_S (OR NFN_M) FROM FILE NFN OF MF.

DEFAULT: NID=XXXXX.
--- MASTERFILE MFCC CY=30 --- 02/07/86 - 00:10:44. -------- COMMENT, 3 --------

**PRODUCE JOBS FROM PROGRAM WRITING LOCAL PROCEDURE CALLED FROM PROCEDURE ZZSYS1 WHICH COMPACT AUDIT OF IO::: ••• • • •**

**PROGRAM CONSTRUCTING DIRECTORY OF PASAUD - PASCAL3 PROGRAM, WRITTEN BY HANS SCHRIJVER, REFORMATING ZZSYS2 ATT750 TO SAME NAME, SHOWING UPON CALLING WHICH PROGRAM WILL OVERWRITE CALLING PROGRAM**

CALL: "JOBCRD,FN,T= •• ,IO= •• ,LP=NP."  
**PARAMETERS NOR,TID: SEE UNDER PROCEDURE NEW.**

**PROCEDURE CREATING JOBS FILE FROM FILE FN RESIDING IN MFFN. THIS JOB EXECUTES BFN_B WITH INPUT IFN_I.**

CALL: "JOBCRD,FN,I=... ,T=... ,LP=NP."  

**CONVERSE LOCAL G-GRAPH FILE "PFN" TO G-CODE FILE, INSERT FIRST LINE "\\LFN\,P,NOVERSA,NODELETE", AND ROUTE FILE TO DESTINATION XXA (11/70), WHERE IT WILL BE CONVERTED BACK AGAIN TO A GRAPH FILE AND PLOTTED.**

CALL: "PLOUT,LFN,(NOVERSA,NODELETE)."  
**PARAMETERS NODELETE: COPY OF THE FILE KEPT ON THE VERSATEC.**

**THIS PROGRAM MAKES A LITTLE CONVERSION OF G-CODE FILES WHICH FACILITATE THE CONVERSION BACK TO A GRAPH FILE AGAIN.**

**THIS PROGRAM IS CALLED BY PROCEDURE PLOUT.**

CALL: "TOUT,LFN,(NOVERSA,NODELETE)."  

**PRINT DATE AND TIME.**

**PROGRAM WRITING LOCAL FILE ZZSYS WITH PROCEDURE OF THE SAME NAME, SHOWING UPON CALLING WHICH FILES OF SYSSWILL AND USERGL CHANGED OVER THE PAST 7 DAYS.**

CALL: "SYS","ZZSYS." (SEE PROCEDURE INIT IN PROCFIL).  

**PROCEDURE CALLING ZZSYS2 WITH THE DATE OF THE PREVIOUS FILE. THIS FILE HAS NO COUNTERPART IN CCLLIB.**

**PROCEDURE CALLED FROM PROCEDURE ZZSYS WHICH IN TURN IS CALLED FROM PROCEDURE INIT ON PROCFIL TO DISPLAY THE FILES OF SYSSWILL AND USERGL WHICH HAVE CHANGED SINCE THE LAST INTERACTIVE SESSION. THIS PROCEDURE ALSO UPDATES ZZSYS1 WITH THE PRESENT DATE FOR USE NEXT TIME.**

**UPDATING DATE RESIDES IN THE RT REGISTER WHICH IS ASSIGNED TO THE D-PARAMETER OF SYSSWILL, SO That A CALL TO SYSSWILL CHANGES THE CONTENT OF RT FROM THE PREVIOUS TO THE PRESENT DATE.**

**COMACT AUDIT OF ID=... (DEFAULT: AT TERMINAL) BY CALLING PROGRAM PASAUD, AND DIRECTORY OF MASTERFILES BY CALLING PROGRAM DIR IF ID IS SPECIFIED.**

**PROGRAM CONSTRUCTING DIRECTORY OF THE MASTERFILES SHOWN IN AUDIT OF ID=...**

**PASCAL PROGRAM, WRITTEN BY HANS SCHRIJVER, REFORMATING OUTPUT OF SYSTEM AUDIT.**

**ATT750 - ATTACHES AND RETURNS ALL PERFILES OF ID=... ON THE 750 BY RUNNING PROGRAM KEEP AND PROCEDURE ZZATT WHICH IT PRODUCES. THIS PROGRAM READS THE OUTPUT FILE "ZZAUD" PRODUCED BY "AUDIT,A=P,LF=IZAUD,ID=..." AND WRITES THE PERFILES FOUND ON A PROCEDURE FILE "ZZATT", WHICH WILL ATTACH (AND SUBSEQUENTLY RETURN) ALL THESE FILES.**

CALL: "ATT750(?,ID=...)".  
**DEFAULT: ID=XXID."**
A00205  ADD PERMFILE TO THE 205.
CALL: "A00205(?,?,FILENAME,CODE)."
2230

B0205  ATTACH ALL PERMFILES ON THE 205, AUDIT, AND RUN PROGRAM SAVE TO RESET THE DATE OF LAST ACCESS TO TODAY.
CALL: "B0205(?,?,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2350

DIFF  REWRITE THE DIFFERENCE BETWEEN OLDPL UFN U AND NEWPL UFN U FROM FILE FNNF ON THE 205.
CALL: "DIFF(?,?,FILENAME,NEWPL,NEW)."
2590

DUP205  GIVES THE 205-DATED LEFT FOR U=XXU1XX.
CALL: "DUP205(?,?,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2370

GET205  GET PERMFILE FROM THE 205.
CALL: "GET205(?,?,FILENAME,CODE)."
2270

RNM205  REPLACE IDENTIFICATION OF SOURCE.
CALL: "RNM205(?,?,FILENAME,NEW)."
2290

AUD205  AUDIT OF PERMFILES ON THE 205.
CALL: "AUD205(?,?,LOG,OUT)."
2320

ATD205  ATTACH ALL PERMFILES ON THE 205, AUDIT, AND RUN PROGRAM SAVE TO RESET THE DATE OF LAST ACCESS TO TODAY.
CALL: "ATD205(?,?,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2340

DUP205  GIVES THE 205-DATED LEFT FOR U=XXU1XX.
CALL: "DUP205(?,?,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2370

PER205  GRANTS PERMISSION TO USER TO ACCESS U=XXU1XX PERMFILE ON THE 205.
CALL: "PER205(?,?,FILENAME,USER=AC,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2420

Q205  SHOWS THE QUEUES ON THE 205 FOR U=XXU1XX.
CALL: "Q205(?,?,U=XXU1XX,A=AC,P=PA,T=ID,TID,FID=FID)"
2470

RIN205  ROUTE JOB TO INPUT QUEUE OF THE 205, WHERE AN INPUT RECORD "QM" AND ANOTHER SECOND RECORD "Q2M" MAY BE INCLUDED.
CALL: "RIN205(?,?,JOB,IN,INF,TID,FID)"
2440

PLT205  CONVERTS THE BINARY PRINTFILE "NAME" FROM THE 205 TO A GRAPHFILE TO BE VISUALIZED WITH BRINAS.
CALL: "PLT205(?,?,NAME)."
2470

NNEW  PRODUCES NEWPL UFN U FROM THE SOURCE FNNF (OF MFFN), OR UFN M FROM UFN S + MODIFICATION DECK MFFN M (OF MFFN), AND CREATE A JOB FILE FN S (OR NN.FN S) FROM FILE FNNF ON THE 205.
CALL: "NNEW(?,?,FILENAME,NEWPL,NEW,PREMFILE,LIB=LIBRARY)"
2490

N  ROOT FILE NAME OF THE PROGRAM
S  IDENTIFICATION OF SOURCE FILE FNNF
M  NEWPL IS MODIFIED WITH MFFN M (FROM MFFN)
NOUN  NO UPDATE LISTING IS MADE ON THE 750
NEWPL NOT CATALOGED ON THE 750
FILENAME FROM THE 205

NH  PRODUCES NEWPL UFN U FROM THE SOURCE FNNF (OF MFFN), AND CREATE A JOB FILE FN S (OR NN.FN S) FROM FILE FNNF (OF MFFN), WHICH COMPILES BFN B ON THE 205.
CALL: "NH(?,?,FILENAME,NEWPL,NEW,PREMFILE,LIB=LIBRARY)"
2610

N  ROOT FILE NAME OF THE PROGRAM
S  IDENTIFICATION OF SOURCE FILE FNNF
U  IDENTIFICATION OF NEWPL UFN U (ON 750)
B  IDENTIFICATION OF BINARY FILE B (ON 205)
NOUN  NO UPDATE LISTING IS MADE ON THE 750
NEWPL NOT CATALOGED ON THE 750
FILENAME FROM THE 205

RR  PROCEDURE CREATING JOB RR M FROM FILE XFN ON THE 205.
CALL: "RR(?,?,FILENAME,M=LIBRARY)"
2740

N  ROOT FILE NAME OF THE PROGRAM
U  IDENTIFICATION OF ODLPL UFN U (ON 750)
M  NEWPL IS MODIFIED WITH MFFN M (OF MFFN)
V  IDENTIFICATION OF NEWPL UFN V (ON 750)
B  IDENTIFICATION OF BINARY FILE B (ON 205)
ULIST  UPDATE LISTING OF THE CHANGES IS MADE
NEWPL NOT CATALOGED ON THE 750
FILENAME FROM THE 205

XX  PROCEDURE CREATING JOB XX_B_I FROM FILE XFN RESOLVING IN MFFN. THIS JOB EXECUTES BFN B WITH INPUT FNN.
2870
IDENTIFICATION OF INPUT
RETRIEVES ONE OR MORE DECKS FROM
DECKS TO
READS OUTPUT LISTFILE OF
ASK
REVISES OLD DECK; "M:;:-Q"
COMPILE
NEWPL; DEFAULTS: 0/NEWPL
JOB
OLD SOURCE
1ST UPDATE LISTING
INSTALS
PREPARES INPUT
COMPILE
2NO
COMPILE
INSTRUCTS INSTAL
COMPARES TWO FILES
OLD UPDATE PROGRAM LIBRARY
NO UPDATE LISTING OF

RU
- PRODUCES NEWPL U AND COMPILE FILE C FROM THE SOURCE S.
CALL: "N(U?,I,S,?U,C,FID,?IDID)."
2970
S = SOURCE FILE
U = NEW UPDATE PROGRAM LIBRARY; DEFAULT: "NEWPL"
C = COMPILE FILE; DEFAULT: "COMPILE"; SUPPRESS:"C=O"
FID = FID FOR UPDATE OUTPUT.
3020
RU
- REVISES OLDPL U WITH MODIFICATION DECK M TO PRODUCE NEWPL V
AND COMPILE FILE C.
3050
U = OLD UPDATE PROGRAM LIBRARY
M = MODIFICATION DECK; "M=O" GIVES COMPILE OF OLDPL
V = NEW UPDATE PL; DEFAULT: "NEWPL"; SUPPRESS:"V=O"
C = COMPILE FILE; DEFAULT: "COMPILE"; SUPPRESS:"C=O"
ULIST = UPDATE LISTING OF THE CHANGES IS MADE
3100
FID = FID FOR UPDATE OUTPUT.
3110
SU
- RECOVERS THE SOURCE S FROM AN OLD UPDATE PL U.
CALL: "N(S?,I,S,?U,FID,?IDID),"
3130
U = OLD UPDATE PROGRAM LIBRARY
S = SOURCE FILE
FID = FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".
3170
INSTAL
- INSTALS A LIBRARY FOR UPDATE PROCEDURES AND PROGRAMS WRITTEN
BY JOSE KOOT AND AMPLIFIED BY HANS GOEDBLOED.
IT CONTAINS THE CCL PROCEDURES REWRITE/ASKDOE AND THE FORTRAN
PROGRAMS ASKDECK/MAKEFIL/MOGEN/USL/UML.
CALL: "INSTAL(I,LIB,PROFFILE)."
3220
LIB = LFN OF THE LIBRARY; DEFAULT:"LIBRARY"
PROFFILE = INSTRUCTS INSTAL TO READ FILES TO BE INSTALLED
FROM FILE INSTAL ITSELF. NU OPTIONS FOR THE USER.
3250
TO USE INSTAL WHICH IS SUPERFLUOUS HERE SINCE THE LIBRARY
CALLLIB ALREADY CONTAINS THE FILES INSTALLED BY INSTAL,
3270
INSERT:
3280
AFTER 1ST EOR: PROCEDURE REVISE
3290
AFTER 2ND EOR: PROCEDURE ASKDOE
3300
AFTER 3RD EOR: FORTRAN PROGRAMS ASKDECK/MAKEFIL/MOGEN/USL/UML
NOT SEPARATED BY COMMENTS OR EOR'S!.
3320
REVISE
- RETRIEVES ONE OR MORE DECKS FROM AN UPDATE LIBRARY L. THE
RETRIEVED DECKS ARE PUT INTO AN EDITFILE S AND THE EDITOR
IS CALLED. AFTER THE USER HAS FINISHED EDITING, THE NEW
VERSION T IS COMARED WITH THE OLD ONE AND A MODIFICATION
DECK M IS MADE. IF WANTED, THIS DECK IS PRESENTED TO UPDATE
WHICH PRODUCES A NEW PROGRAM LIBRARY V AND A COMPILE FILE C.
CALL: "REVISE(T?,U(S,?O,M,C,T,V,C,ULIST,FID,?IDID),"
3390
U = LFN OF THE OLDPL
S = DECKS TO BE UPDATED; DEFAULTS: $ALL&?$ASK.$
V = MODIFICATION DECK; DEFAULT: "MODFILE"
C = CORRECTION SET IDENTIFIER; DEFAULT: "MOD"
S = OLD SOURCE DERIVED FROM OLDPL; DEFAULTS: 0/OLDSRC
T = NEW SOURCE AFTER EDITING; DEFAULTS: 0/NEWSRC
V = NEWPL; DEFAULTS: U/NEWPL
C = COMPILE FILE; DEFAULTS: U/COMPILE
FID = FID FOR UPDATE OUTPUT; DEFAULTS: 0
3460
ULIST = 1ST UPDATE LISTING (RETRIAL OF DECKS); DEFAULT: 0
UL2 = 2ND UPDATE LISTING (CORRECTION RUN); DEFAULT: 0/1.
3490
ASKDOE
- ASK FOR PERMISSION TO DELETE THE OLD EDITFILE. IT SHOULD
BE CALLED BEFORE ENTERING ED IN ORDER TO PREVENT MIXING WITH
THE CONTENTS OF AN EXISTING EDITFILE.
3500
TYPICAL USE: ".IF(FILE(LIZZITZ,ASS) ASKDOE)."
3520
ASKDECK
- PREPARES INPUT FILE FOR UPDATE RUN WHICH RETRIEVES SPECIFIED
DECKS FROM OLDPL. THE DECKNAMES ARE OBTAINED INTERACTIVELY.
3550
CALLED FROM REVISE: "ASKDECK,INPUTFILE".
3560
MAKEFIL
- READS OUTPUT LISTFILE OF AN UPDATE RUN (WITH LIST OPTION L=7)
WHICH HAS RETRIEVED SPECIFIED DECKS FROM OLDPL AND PRODUCES
TWO FILES: WITHERS AND WITHOUT. WITHERS CONTAINS LINE IMAGE
3590
AND SEQUENCE INFORMATION, WITHOUT CONTAINS LINE IMAGES ONLY.
3600
UNWANTED COMMON DECKS OBTAINED BY SELECTIVE UPDATE MODE ARE
REMOVED BY COMPARING THE CONTENTS OF LISTFILE WITH THE
3610
DECK LIST REQUESTED ON THE UPDATE INPUT FILE IMPFILE.
3630
CALLED FROM REVISE:
"MAKEFIL,LISTFILE,WITHSEQ,WITHEQ,(INPUTFILE)."
3640
MOGEN
- COMPARES TWO FILES (NEWFILE AND OLDFILE) AND PREPARES AN
UPDATE CORRECTION SET IN MODFILE.
3660
CALLED FROM REVISE: "MOGEN,OLDFILE,NEWFILE,MODFILE,CI.".
3680
**USL** - PROGRAM REFORMATING OUTPUT OF UPDATE SOURCE LISTING OBTAINED FROM A CREATION RUN OR FROM AN AUDIT RUN OF OLDPL.

AFTER "UPDATE,F,I=SOURCE,N,L=A1240,FN.",

OR "UPDATE,F,P=OLDPL,L=A1240,FN.",

CALL: "USL,FN,(NOLIST),".

THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL.

**FTN200** - RELEVELMENTS OF A CORRECTION RUN.

AFTER "UPDATE,F,P=OLDPL,L=MOD,N,L=A1240,FN.",

CALL: "FTN200,FN,(LIST),".

PARAMETER "LIST" SWITCHES LISTING OF EFFECTED CHANGES ON.

**RUN205** - CREATES A JOB FOR THE 205 WHICH PERFORMS A COMPLETE COMPILE, LOAD, AND EXECUTE SEQUENCE OF THE FORTRAN PROGRAM "NAME", OR PART OF IT RESULTING IN PERMANENT BINARY "B" OR GOFILE "G".

ALTERNATIVELY, IF "NAME" IS NOT SPECIFIED, A RUN IS PREPARED STARTING FROM EITHER "B" OR "G".

CALL:

"UNO205,(T),NAME,B,G,NOEX,1,P,OPT,UNS,L,LD,TL,WS,LP,NOR(1)."

**VAST205** - TRANSLATE PROGRAM NAME WITH VAST ON THE 205 AND CATALOG OUTPUT ON THE 750.

CALL: "VAST,(T),NAME,(OUTPUTFILE)."

**ALIAS** - TRANSFERS FILES FROM AN ATTACHED MASTERFILE PFN1 TO A SECOND ATTACHED MASTERFILE PFN2 WHILE CHANGING ALL PERSONAL ID'S, ACCOUNTS, AND PASSWORDS INTO THE SPECIFIED ONES.

CALLS:

"MFUSE,PFN1,(H=MF1),ID=XXIDX."

"MFUSE,PFN2,(H=SECOND),ID=....."

"ALIAS,(T),FLIST,(MF1,MF2,ID,ACC,UN,12,AC,U1,U2,PA,TA,TA)."

PARAMETERS:

FLIST - LIST OF FILES TO BE TRANSFERRED.

"PFN" - ONE FILE

"PFN1/PFN2,..." - A FEW FILES (STRING <= 40 CHARSW)

"S $" - ALL FILES

**VAST** - PROGRAM REFORMATING OUTPUT OF UPDATE SOURCE LISTING OBTAINED FROM A CREATION RUN OR FROM AN AUDIT RUN OF OLDPL.

AFTER "UPDATE,F,I=SOURCE,N,L=A1240,FN.",

OR "UPDATE,F,P=OLDPL,L=A1240,FN.",

CALL: "USL,FN,(NOLIST),".

THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL.

**FTN200** - RELEVELMENTS OF A CORRECTION RUN.

AFTER "UPDATE,F,P=OLDPL,L=MOD,N,L=A1240,FN.",

CALL: "FTN200,FN,(LIST),".

PARAMETER "LIST" SWITCHES LISTING OF EFFECTED CHANGES ON.

**RUN205** - CREATES A JOB FOR THE 205 WHICH PERFORMS A COMPLETE COMPILE, LOAD, AND EXECUTE SEQUENCE OF THE FORTRAN PROGRAM "NAME", OR PART OF IT RESULTING IN PERMANENT BINARY "B" OR GOFILE "G".

ALTERNATIVELY, IF "NAME" IS NOT SPECIFIED, A RUN IS PREPARED STARTING FROM EITHER "B" OR "G".

CALL:

"UNO205,(T),NAME,B,G,NOEX,1,P,OPT,UNS,L,LD,TL,WS,LP,NOR(1)."

**VAST205** - TRANSLATE PROGRAM NAME WITH VAST ON THE 205 AND CATALOG OUTPUT ON THE 750.

CALL: "VAST,(T),NAME,(OUTPUTFILE)."

**ALIAS** - TRANSFERS FILES FROM AN ATTACHED MASTERFILE PFN1 TO A SECOND ATTACHED MASTERFILE PFN2 WHILE CHANGING ALL PERSONAL ID'S, ACCOUNTS, AND PASSWORDS INTO THE SPECIFIED ONES.

CALLS:

"MFUSE,PFN1,(H=MF1),ID=XXIDX."

"MFUSE,PFN2,(H=SECOND),ID=....."

"ALIAS,(T),FLIST,(MF1,MF2,ID,ACC,UN,12,AC,U1,U2,PA,TA,TA)."

PARAMETERS:

FLIST - LIST OF FILES TO BE TRANSFERRED.

"PFN" - ONE FILE

"PFN1/PFN2,..." - A FEW FILES (STRING <= 40 CHARSW)

"S $" - ALL FILES

**VAST** - PROGRAM REFORMATING OUTPUT OF UPDATE SOURCE LISTING OBTAINED FROM A CREATION RUN OR FROM AN AUDIT RUN OF OLDPL.

AFTER "UPDATE,F,I=SOURCE,N,L=A1240,FN.",

OR "UPDATE,F,P=OLDPL,L=A1240,FN.",

CALL: "USL,FN,(NOLIST),".

THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL.

**FTN200** - RELEVELMENTS OF A CORRECTION RUN.

AFTER "UPDATE,F,P=OLDPL,L=MOD,N,L=A1240,FN.",

CALL: "FTN200,FN,(LIST),".

PARAMETER "LIST" SWITCHES LISTING OF EFFECTED CHANGES ON.

**RUN205** - CREATES A JOB FOR THE 205 WHICH PERFORMS A COMPLETE COMPILE, LOAD, AND EXECUTE SEQUENCE OF THE FORTRAN PROGRAM "NAME", OR PART OF IT RESULTING IN PERMANENT BINARY "B" OR GOFILE "G".

ALTERNATIVELY, IF "NAME" IS NOT SPECIFIED, A RUN IS PREPARED STARTING FROM EITHER "B" OR "G".

CALL:

"UNO205,(T),NAME,B,G,NOEX,1,P,OPT,UNS,L,LD,TL,WS,LP,NOR(1)."
-L.7-  

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------ NEWCCL, 2 ------  

LIBRARY(CCLLIB).  
REVERT.  
*  
EXIT,S.  
COMMENT,** ERROR **  
RETURN,ZZINP,OUTPUT,LIB.  
REVERT,ABORT.  
*  
*DATA,ZZINP.  
LIBRARY(LIB,NEW)  
ADD(*,FN)  
FINISH.  
ENDRUN.  

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------ AODCCL, 1 ------  

.PROC,ADOCCL,FN,TYPE.  
*  
ADOCCL FILE FN TO LIBRARY CCLLIB (TYPE: PROC/REL/ABS/PAS).  
*IF, $TYPE$=$PROC$.OR.$TYPE$=$REL$.OR.$TYPE$=$ABS$.OR.$TYPE$=$PAS$, OK.  
MFUSE,MFCCL,ID=XXIOX.  
FAOD,fN.  
NOTE,$FN ADDED TO MFCCL$.  
RETURN,CCLLIB.  
ATTACH,CCLLIB,ID=XXIOX.  
CONNECT,OUTPUT.  
*IF, $TYPE$==$PROC$, PROCEDURE.  
EDITLIB,1=ZPROC.  
.ENDIF, PROCEDURE.  
*IF, $TYPE$==$REL$, RELOCATABLE.  
FN5,1=FN, $B=REL, L=0, DPT=2, PL=10000$.  
EDITLIB,1=ZREL.  
.ENDIF, RELOCATABLE.  
*IF, $TYPE$==$ABS$.OR.$TYPE$==$PAS$, ABSOLUTE.  
*IF($TYPE$==$ABS$) FN5,1=FN, $B=REL, L=0, DPT=2, PL=10000$.  
*IF($TYPE$==$PAS$) PAS3,FN,,REL,L-.  
LOAD,REL.  
NOGO,ABS.  
ETOITLIB,I=ZZABS.  
.ENDIF, ABSOLUTE.  
EXIT,S.  
NOTE,$ERROR: TYPE SHOULD BE PROC/REL/ABS/PAS$.  
RETURN,ZZPROC,ZZREL,ZZABS.  
ATTACH,CCLLIB,IO=XXIDX,MR=1.  
REVERT.  
*  
FINISH.  
ENDRUN.  
*  
DATA,ZZPROC.  
LIBRARY(CCLLIB,OLD)  
REWIND(OLD)  
ADD(*,FN)  
FINISH.  
ENDRUN.  
*  
DATA,ZZREL.  
LIBRARY(CCLLIB,OLD)  
REWIND(REL)  
ADD(*,REL)  
SETAL(FN,1)  
FINISH.  
ENDRUN.  
*  
DATA,ZZABS.  
LIBRARY(CCLLIB,OLD)  
REWIND(ABS)  
ADD(*,ABS,AL=1)  
FINISH.  
ENDRUN.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. -------- REPCCL, 1 -------

.* REPCCL,FN,TY.
.* REPLACE FILE FN IN LIBRARY CCLLIB (TYPE: PROC/REL/ABS/PAS).
.* IF $TYPE$=PROC,OK.
.* IF $TYPE$=REL,OK.
.* IF $TYPE$=ABS,OK.
.* IF $TYPE$=PAS,OK.
.* MFUSE,MFCCL,ID=XXIDX.
.* FREP,FN.
.* NOTE, FN REPLACED IN MFCCL.
.* RETURN,CCLLIB.
.* ATTACH,CCLLIB,ID=XXIDX.
.* CONNECT,OUTPUT.
.* .IF $TYPE$=PROC,PROCEDURE.
.* .ENDIF,PROCEDURE.
.* .IF $TYPE$=REL,RELOCATABLE.
.* .FINS,I=FN,B=REL,L=0,OPT=2,PL=10000.
.* .ENDIF,RELOCATABLE.
.* .IF $TYPE$=ABS,ABSOLUTE.
.* .IF $TYPE$=PAS,PAS.
.* .LOAD,REL.
.* .NOGO,ABS.
.* .EDITLIB,I=ZZPROC.
.* .EDITLIB,I=ZZREL.
.* .EDITLIB,I=ZZABS.
.* .LOAD,REL.
.* .NOGO,ABS.
.* .EDITLIB,I=ZZPROC.
.* .EDITLIB,I=ZZREL.
.* .EDITLIB,I=ZZABS.
.* .EXTEND,CCLLIB.
.* .RETURN,CCLLIB.
.* .ATTACH,CCLLIB,ID=XXIDX,MR=1.
.* .CONNECT,OUTPUT.
.* .EDITLIB,I=ZZPROC.
.* .EDITLIB,I=ZZREL.
.* .EDITLIB,I=ZZABS.
.* .EXTEND,CCLLIB.
.* .RETURN,CCLLIB.
.* .ATTACH,CCLLIB,ID=XXIDX,MR=1.
.* .CONNECT,OUTPUT.
.* .EDITLIB,I=ZZPROC.
.* .EDITLIB,I=ZZREL.
.* .EDITLIB,I=ZZABS.
.* .EXTEND,CCLLIB.
.* .RETURN,CCLLIB.
.* .ATTACH,CCLLIB,ID=XXIDX,MR=1.
.* .CONNECT,OUTPUT.
.* .DATA,ZZPROC.
.* DATA,ZZREL.
.* DATA,ZZABS.
.* END.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. -------- DELCCL, 1 -------

.* DELCCL,FN.
.* DELETE FILE FN FROM LIBRARY CCLLIB.
.* MFUSE,MFCCL,ID=XXIDX.
.* FDEL,FN.
.* NOTE,$FN DELETED FROM MFCCL.
.* RETURN,CCLLIB.
.* ATTACH,CCLLIB,ID=XXIDX.
.* CONNECT,OUTPUT.
.* EDITLIB,I=ZZINP.
.* EXTEND,CCLLIB.
.* RETURN,ZZINP,CCLLIB,OUTPUT.
.* ATTACH,CCLLIB,ID=XXIDX,MR=1.
.* .CONNECT,OUTPUT.
.* .DATA,ZZPROC.
.* DATA,ZZREL.
.* DATA,ZZABS.
.* END.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- DELCCL, 2 -------

RETURN,ZZINP,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
REVERT,ABORT.
.* DATA,ZZINP.
LIBRARY(CCLLIB,OLD)
DELETE(FN)
FINISH.
ENDRUN.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- COPYCCL, 1 -------

.* PROC,COPYCCL.
.* COPY MFCCL WITH MFCOPY AND LIBRARY CCLLIB WITH LIBRARY CCLLIB.
.* PROCEDURE COPYL COLUMN OF PIASLIB.
MFUSE.
MFUSE,MFCCL,M=OLD,ID=XXIDX,MR=1.
MFNEW,MFCCL,M=NEW,ID=XXIDX.
FGET,COMMENT,M=OLD.
UPCOM.
FAEEI,COMMENT,M=NEW.
MFSET,MSG=F,ABT=F.
MFCOPY,M=OLD,N=NEW,REP.
MFSET,MSG=PREV.
RETURN,OLD,NEW,COMMENT.
MFUSE,MFCCL,ID=XXIDX.
COMMENT.** NEW CYCLE OF MFCCL ATACHED **
.* RETURN,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LISTLOAD(PIASLIB,COPYL)
EXECUTE(COPYL,CCLLIB,NEWLIB)
RETURN,PIASLIB.
CATALOG,NEWLIB,CCLLIB,ID=XXIDX.
RETURN,CCLLIB,NEWLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
COMMENT.** NEW CYCLE OF CCLLIB ATACHED **
REVERT.
.* EXIT,S.
NOTE,$ERROR$.
REVERT,ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- LISTLIB, 1 -------

.* PROC,LIS TLIB,LIB,PR=N/Y.
.* LIST CONTENTS OF LIBRARY LIB WITH LISTLIB OF PIASLIB.
RETURN,OUTPUT.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LISTLOAD(PIASLIB,LISTLIB)
.REL,L,PR=SN$, PRNY.
EXECUTE(LISTLIB,LIB)
.END, PRNY.
EXECUTE(LISTLIB,LIB,OUTPUT)
RELIST.
ROUT.
.REWIT, PRNY.
RETURN,PIASLIB.
REVERT.
.* EXIT,S.
NOTE,$ERROR$.
RETURN,PIASLIB.
REVERT,ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- RELIST, 1 -------

PROGRAM RELIST(OUTPUT,TAPES=OUTPUT)
--- MASTERFILE MFCCL CY=30 --- 02/07/86 - 00.10.44. RELIST, 2 ---

IMAX=1000
IPMAX=20
ILMAX=60

C
REWIND 6
OPEN(7,FILE='STORE')
DO 10 I=1,IMAX
READ(6,1,ERR=100,END=20) LINE
IF(LINE(1:1).EQ.'Q') WRITE (7,2)
WRITE(7,1) '/LINE(2;73)
10 CONTINUE
20 REWIND 7
REWIND 6

C
READ(7,'(A/)'),ERR=100,END=200) HEADER
DO 30 IP=1,IPMAX
WRITE(6,3) HEADER(J=3:30),HEADER(31:40),HEADER(41:50),IP
IL1=IP+ILMAX
IL2=1+ILMAX
READ(7,1,ERR=100,END=200) LINE
WRITE(6,1) LINE
30 CONTINUE

100 STOP 'ERROR'
200 CLOSE(7,STATUS='DELETE')

--- MASTERFILE MFCCL CY=30 --- 02/07/86 - 00.10.44. LIBLIST, 1 ---

.* LIST ATTACHED LIBRARIES WITH LIBLIST OF PIASLIB.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LIBLOAD(PIASLIB,LIBLIST)
EXECUTE(LIBLIST)
RETURN,PIASLIB.

--- MASTERFILE MFCCL CY=30 --- 02/07/86 - 00.12.09. ZZFILES, 1 ---

.* LIST SYSTEM FILES STARTING WITH ZZZZ BY
.* CALLING PROGRAM ZZFILES OF LIBRARY PIASLIB.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LIBLOAD(PIASLIB,ZZFILES)
EXECUTE(ZZFILES)
RETURN,PIASLIB.

--- MASTERFILE MFCCL CY=30 --- 02/07/86 - 00.12.09. COST, 1 ---

.* SHOW SYSTEM SECONDS USED SINCE LAST CALL OF COST BY.
.* CALLING PROGRAM COST OF LIBRARY PIASLIB.
ATTACH,PIASLIB,ID=PIASS,SN=S.
COMMENT,/* USED SINCE LAST CALL OF COST:
LIBLOAD(PIASLIB,COST)
EXECUTE(COST)
IF($X$=$Y$) RETURN,ZZZZZUC.
RETURN,PIASLIB.

--- MASTERFILE MFCCL CY=30 --- 02/07/86 - 00.12.09. EDCY, 1 ---

.* EDIT WITH SARED ON THE CYBER INTERACTIVELY.
.* CALL: "EDCY(U)."
.* PARAMETER U PREPARES FILE 8 TO BE USED FOR REARRANGING
.* COMMENT LINES WITH PROGRAM REAR, IN SARED, THE SET OF
.* COMMANDS "W,A,L1 L2", "DELL1 LZ" REPLACES
.* COMMENT LINES L1-L2 WITH LINES OF 72 CHARACTERS.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- EDCY, 2 -------

.CP,EDCY,EDFILE,FIRST=N/Y.
.RETURN,B.
.COPY,ZZB,B.
.REWIND,B.
.ENDIF, USEB.
.RETURN,ZZB.

EDCY, 2
02/07/86 - 00.12.09.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- REAR, 1 -------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- EDDY, 1 -------

.PROC,EDDY,EDFILE,FIRST=N/Y.
* EDIT WITH SARED ON THE CYBER, KEEPING PERMANENT FILE EDFILE.
* EDFILE IS CREATED BY SPECIFYING "FIRST" IN THE FIRST CALL OF EDDY.
* THE SARED COMMANDS "E", "C", "B", "Q" DO NOT AFFECT THE CONTENTS
* OF EDFILE, UNLESS IT IS CLEARED COMPLETELY WITH "DEL".
* HENCE: AVOID THE USE OF THESE FOUR COMMANDS, AND ONLY USE
* "ADD", "INSERT", AND THE USUAL STRING REPLACEMENT COMMANDS.
.RETURN,ZZZZZ1Z,ZZZZZ3Z.
.IF,$FIRST=$N,$FIRSTNY.
SC,A,EDFILE,ID=XXIDX.
.ELSE, FIRSTNY.
SC,C,EDFILE,ID=XXIDX.
.ENDIF, FIRSTNY.
REVERT.

- L.11 -
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- FED, 1 ---------

.PROC,FED,FILE.
** PRODUCING FORMAT FILE FOR SARED.
** FILE=FN: FORMAT FOR EDITING FORTRAN FILES.
** IF, $FILE$=FF$ OR $FILE$=FN$, OK.
COPY,ZIFF,FF.
REWIND,FF.
.ENDIF, FORTRAN.
COPY,ZFF,FF.
REWIND,FF.
.ENDIF, COMMENT.
RETURN,ZFF,ZIFF.
REVERT.
** ENDIF, OK.
NOTE, $ERROR$: SPECIFY "FED,FF" OR "FED,FN$.
RETURN,ZFF,ZIFF.
REVERT,ABORT.
** EXIT,S.
NOTE, $ERROR$: SPECIFY "FED,FF" OR "FED,FN$.
RETURN,ZFF,ZIFF.
REVERT,ABORT.
**
.DATA,ZIFF.
FORMAT \ FORTRAN
SET COUNT=1
SET LINE=19
SHOW FORMAT SHOW
REWIND FF
**
.DATA,ZIFF.
FORMAT \ FORTRAN
SET COUNT=1
SET LINE=19
SHOW FORMAT SHOW
REWIND FN
**
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- STRIP, 1 ---------

.PROC,STRIP,FN,L=72./
** STRIP LAST COLUMNS (DEFAULT: LINES RS AFTER COLUMN 72)
** AND TRAILING BLANKS FROM A FILE.
ED,USE,INSTR.
COMMENT,** FN STRIPPED (L CHARS) **
RETURN,INSTR,ELOG.
REVERT.
**
.EXIT,S.
COMMENT,** ERROR **
REVERT,ABORT.
**
.DATA,INSTR.
E,FN
F,Bl,H
W,FN,O
E,FN
W,FN,O
SC,INIT
B,G

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- DCL, 1 ---------

PROGRAM DCL
C **************************** ***********************************************
C * DELETE COLUMNS 11 TO 12 OF FILE FN1 AND WRITE THE RESULTING *
C * FILE ON FN2. *
C * CALL: "DCL,FN1,FN2,11,12." *
C * IF FN1=FN2 THE ORIGINAL FILE FN1 IS OVERWRITTEN WITH FN2. *
C * IF 12 IS OMITTED ONLY COLUMN 11 WILL BE DELETED. *
C **************************** ***********************************************

C
IMPLICIT INTEGER(A-Z)
PARAMETER(MXLINES=10000)

CALL GETPARM(FN1,DUM,PARSTAT)
CALL GETPARM(FN2,DUM,PARSTAT)
IF (FN2.EQ.FN1) CALL GETPARM(C1,DUM,PARSTAT)
CALL GETPARM(C2,DUM,PARSTAT)
IF (PARSTAT.LT.0) CALL GETPARM(C1,DUM,PARSTAT)
CALL GETPARM(C2,DUM,PARSTAT)
OPEN(10,FILE=FN1)
OPEN(20,FILE=FN2)
READ(C1,'(15)')
READ(C2,'(15)')
IF(I1.EQ.1) THEN
  DO 10 L=1,MXLINES
    READ(10,'(A)',ERR=100,END=30) LINE
    WRITE(20,'(A)',LINE=I1-1//LINE(12+1:133))
  CONTINUE
ELSE
  DO 20 L=1,MXLINES
    READ(10,'(A)',ERR=100,END=30) LINE
    WRITE(20,'(A)',LINE=I1-1//LINE(12+1:133))
  CONTINUE
ENDIF

30 IF(FN2.EQ.'DUMMY') THEN
  CLOSE(20,STATUS='DELETE')
ENDIF

STOP 'PROGRAM DCL'

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. ----------- COUNT, 1 -----------

* PROC,COUNT,LEFT=N/Y.
* COUNT COLUMN NUMBERS ON THE SCREEN.
CONNECT,ZICOUNT.
IF(SLEFTS=SNL), LEFTNY.
REWINZIC1.
COPY,ZIC2,ZICOUNT.
ELSE, LEFTNY.
REWINZIC2.
COPY,ZIC2,ZICOUNT.
ENDIF, LEFTNY.
RETURN,ZICOUNT,ZIC1,ZIC2.
REVERT.
* 
DATA,ZIC1.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. ----------- NOTE, 1 -----------

* PROC,NOTE,MES,MESSAGE,L=Z,NOTE,DISCON=N/Y.
* NOTE "MESSAGE" ON FILE L (DEFAULT CONNECTED).
* CALLING STATEMENT: NOTE,MES,MESSAGE.
* "MESSAGE" SHOULD NOT EXCEED 40 CHARACTERS.
IF($DISCON$=$N$) CONNECT,L.
IF($DISCON$=$N$), RETURN,L.
RETURN,ZIN.
REVERT.
* 
DATA,ZIN.
** MESSAGE **
-L.14-

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.12.09. --------- COUNT, 2 ---------

  1  2  3  4  5  6
12545678901234567890123456789012345678901234567890123456789012

* .DATA,ZZC2.

  1  2  3  4  5  6
1234567890123456789012345678901234567890123456789012345678901

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- CALC, 1 ---------

  .PROC,CALC,EXPR,A=1/,B=1/,C=1/,D=1/.
  * FORTRAN POCKET CALCULATOR.
  DMLOCK,ON.
  CONNECT,OUTPUT.
  REWIND,ZZCAL.
  FNS,=ZZCAL,#B=BIN,L=0,OPT=U,LO=S/-#A.
  BIN.
  RETURN,ZZCAL,BIN,OUTPUT.

  .DATA,ZZCAL.
  PROGRAM CAL
  DATA #A,#B,#C,#D/A,B,C,D/
  PI=3.1415926535898
  E=EXPR
  PRINT*,' =',E
  END

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- DIFFER, 1 ---------

  .PROC,DIFFER,FN1,FN2,L=OUTPUT/LIST.
  * COMPRE N1 AND N2 WITH PROGRAM DIFF OF LIBRARY PROGS OF DICK.
  * WINTER (MC.). DIFF IS WRITTEN IN A SPECIAL ASSEMBLER LANGUAGE
  * MAY NO LONGER BE SUPPORTED BY THE SYSTEM AT SOME FUTURE DATE.
  .IF,FILE(FN1,LO.OR.IN.OR.PF), OK.
  .ENDIF, OK.
  NOTE,&COMPARISON OF FILE FN1 AND FN2$.
  .ELSE, NOTEBEGIN.
  NOTE,&COMPARISON OF FILE FN1 AND FN2$.
  .ENDIF, NOTEEND.
  ATTACH,PROGS,ID=DW,LC=1,MR=1.
  LIBLOAD(PROGS,DIFF)
  EXECUTEDIFF,1=FN1,2=FN2,#L=L)
  RETURN,OUTPUT,PROGS.
  REVERT.
  .ENDIF, OK.
  NOTE,&NO FILE FN1 OR FN2$.
  REVERT,ABORT.

  .PROC,RUNS,FN,B=8IN/,OPT=2/0,L=OUTPUT/LIST,KEEP=N/Y,
  * FTNS COMPI LATION AND EXECUTION (UNLESS NOEX) OF FN.
  * IF,FILE(FN1,LO,OR.IN,OR.PF), OK.
  .IF,FILE(FN1,LO,OR.IN,OR.PF), OK.
  .IF,$KEEP=$N$, KEEPNY.
  .IF,$NOEX=$N$, RETURN,OUTPUT.
  .ELSE, KEEPNY.
  .ENDIF, KEEPNY.
  .ENDIF, KEEPNY.
  NOTE,&ERROR: FILE FN
  .RETURN,OUTPUT.
  .EXIT,S.
  NOTE,&ERROR:
  .RETURN,OUTPUT.
  NOTE,&ERROR: FILE FN DOES NOT EXIST.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- RUNS, 1 ---------

  .PROC,RUNS,FN,B=8IN/,OPT=2/0,L=OUTPUT/LIST,KEEP=N/Y,
  .IF,FILE(FN1,LO,OR.IN,OR.PF), OK.
  .RETURN,OUTPUT.
  .ELSE, KEEPNY.
  .ENDIF, KEEPNY.
  NOTE,&ERROR: FILE FN
  .RETURN,OUTPUT.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- RUNS, 2 ---------
REVERT, ABORT.
.*
EXIT,S.
NOTE, ERRORS.
REVERT, ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- WFO, 1 ---------
.PROC, WFO, FN, NOL=N/Y.
* WRITE FILE FN ON OUTPUT (DEFAULT: ADDING LINERNS).
.IF, $NOLS=SN, NOL, NOLN. * ADDING LINERNS (NOL=N).
ED, USE, LINERNS.
RETURN, EDLOG.
REWIND, OUT.
COPYSBF, OUT, OUTPUT.
.ELSE, NOLN. * NO LINERNS (NOL=Y).
REWIND, FN.
COPYSBF, FN, OUTPUT.
.ENDIF, NOLN.
RETURN, LINERNS, OUT.
COMMENT, FN WRITTEN ON OUTPUT.
REVERT.
.*
.DATA, LINERNS.
F, L=72
E, FN
W, OUT, K
SC, INIT
B, Q

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- PAG, 1 ---------
.PROC, PPRINT, FN, CY=N/.
* PRINT FILE FN AT LINEPRINTER, CALLING PROGRAM PAG.
.IF, FILE(FN, PF, OR, LO), OK.
RETURN, OUTPUT.
.IF, FILE(FN, PF). FILESTATUS. * PERMANENT FILE.
.IF, $CY=N, CHECKCY.
NOTE, ERRORS: YOU FORGOT TO SPECIFY #CY=N.
REVERT, ABORT.
.*
.ENDIF, CHECKCY.
RETURN, FN.
ATTACH, FN, CY=CY, ID=XXIDX, MR=1.
PAG, FN, $PERMANENT FILES, SFN #CY=CY.
.ELSE, FILESTATUS. * LOCAL FILE.
PAG, FN, $LOCAL FILES, $SFN.
.ENDIF, FILESTATUS.
COMMENT, FN WRITTEN ON OUTPUT.
ROUT.
REVERT.
.*
.ENDIF, OK.
NOTE, ERRORS: FILE FN DOES NOT EXISTS.
REVERT, ABORT.
.*
EXIT,S.
NOTE, ERRORS.
REVERT, ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- PPRINT, 1 ---------
.PROC, PPRINT, FN, CY=N/.
* PRINT FILE FN AT LINEPRINTER, CALLING PROGRAM PAG.
.IF, FILE(FN, PF, OR, LO), OK.
RETURN, OUTPUT.
.IF, FILE(FN, PF), FILESTATUS. * PERMANENT FILE.
.IF, $CY=N, CHECKCY.
NOTE, ERRORS: YOU FORGOT TO SPECIFY #CY=N.
REVERT, ABORT.
.*
.ENDIF, CHECKCY.
RETURN, FN.
ATTACH, FN, CY=CY, ID=XXIDX, MR=1.
PAG, FN, $PERMANENT FILES, SFN #CY=CY.
.ELSE, FILESTATUS. * LOCAL FILE.
PAG, FN, $LOCAL FILES, $SFN.
.ENDIF, FILESTATUS.
COMMENT, FN WRITTEN ON OUTPUT.
ROUT.
REVERT.
.*
.ENDIF, OK.
NOTE, ERRORS: FILE FN DOES NOT EXISTS.
REVERT, ABORT.
.*
EXIT, S.
NOTE, ERRORS.
REVERT, ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- PPRINT, 1 ---------
.PROC, PPRINT, FN, CY=N/.
* PRINT FILE FN AT LINEPRINTER, CALLING PROGRAM PAG.
.IF, FILE(FN, PF, OR, LO), OK.
RETURN, OUTPUT.
.IF, FILE(FN, PF), FILESTATUS. * PERMANENT FILE.
.IF, $CY=N, CHECKCY.
NOTE, ERRORS: YOU FORGOT TO SPECIFY #CY=N.
REVERT, ABORT.
.*
.ENDIF, CHECKCY.
RETURN, FN.
ATTACH, FN, CY=CY, ID=XXIDX, MR=1.
PAG, FN, $PERMANENT FILES, SFN #CY=CY.
.ELSE, FILESTATUS. * LOCAL FILE.
PAG, FN, $LOCAL FILES, $SFN.
.ENDIF, FILESTATUS.
COMMENT, FN WRITTEN ON OUTPUT.
ROUT.
REVERT.
.*
.ENDIF, OK.
NOTE, ERRORS: FILE FN DOES NOT EXISTS.
REVERT, ABORT.
.*
EXIT, S.
NOTE, ERRORS.
REVERT, ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- PPRINT, 1 ---------
.PROC, PPRINT, FN, CY=N/.
* PRINT FILE FN AT LINEPRINTER, CALLING PROGRAM PAG.
.IF, FILE(FN, PF, OR, LO), OK.
RETURN, OUTPUT.
.IF, FILE(FN, PF), FILESTATUS. * PERMANENT FILE.
.IF, $CY=N, CHECKCY.
NOTE, ERRORS: YOU FORGOT TO SPECIFY #CY=N.
REVERT, ABORT.
.*
.ENDIF, CHECKCY.
RETURN, FN.
ATTACH, FN, CY=CY, ID=XXIDX, MR=1.
PAG, FN, $PERMANENT FILES, SFN #CY=CY.
.ELSE, FILESTATUS. * LOCAL FILE.
PAG, FN, $LOCAL FILES, $SFN.
.ENDIF, FILESTATUS.
COMMENT, FN WRITTEN ON OUTPUT.
ROUT.
REVERT.
.*
.ENDIF, OK.
NOTE, ERRORS: FILE FN DOES NOT EXISTS.
REVERT, ABORT.
.*
EXIT, S.
NOTE, ERRORS.
REVERT, ABORT.
--- MASTErFILE MFCCl CY=30 --- 02/07/86 - 00.13.50. --------- PA6, 2 ---------

CALL GETPARM(FN,FN1,I)
CALL GETPARM(L,LEFT,I)

JL=INDEX(L,'/','/','/
)
IF(JL.GT.27) JL=27

LEFT=-----------------------------

LEFT(JL)=L(1:JL-1)

CALL GETPARM(R,RIGHT,I)

JR=INDEX(R,'/','/','/
)
IF(JR.GT.14) JR=14

RIGHT=-----------------------------

RIGHT(JR:14)='R(1:JR-1)

N1=1

CALL GETPARM(NN1,NNN1,I)
IF(I.EQ.1) N1=1

N2=IPMAX*ILMAX

CALL GETPARM(NN2,NNN2,I)

N2=IPMAX*ILMAX

OPEN(5,FILE=FN)
OPENC6,FILE='OUTPUT')

READ(5,'(A)',ERR=100,END=200)

LINE

WRITE(6,2) LINE,IL

CONTINUE

100 STOP 'ERROR'

200 STOP 'PROGRAM PAG'

--- MASTErFILE MFCCl CY=30 --- 02/07/86 - 00.13.50. --------- FOUT, 1 ---------

PROGRAM FOUT

*****************************************************************************
* PROGRAM WRITING FILES OF ATTACHED MASTERFILE ON OUTPUT. *
* CALL: 'FOUT,FN1/FN2/FN3/...' *
*****************************************************************************

CHARACTER*40 C1,C2
CHARACTER*27 MFH,MFNAME
CHARACTER*10 O,T
CHARACTER*8 XXFN,FN,FNAME
CHARACTER*72 LINE

IFMAX=10
IPMAX=150
ILMAX=60

C1='XXC=COMMENT,' CALL FGET(C1)
OPEN(4,FILE='XXC')
READ(4,'(/A)') MFH
I=INDEX(MFH,'/')

MFNAME='---------------------------

MFNAME(I:1)=MFH(1:I-1)

CLOSE(4,STATUS='DELETE')

CALL DATED)
CALL TIME(T)
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- FOUT, 2 -------

C
OPEN(6,FILE='OUTPUT')
OPEN(7,FILE='XXOUT')
CALL CONNEC(7)
DO 20 IF=1,IFMAX
CALL GETPARM(XXFN,FN,IF)
IF(IF.EQ.1) THEN
FN=XXFN
XXFN='XXXX'
ENDIF
IF(IF.EQ.-1) GOTO 30
C2=XXFN//1='1=1//'FN(1:J-1)
CALL FGETC2
OPEN(S,FILE=XXFN)
J=INDEXCFN,1)
FNAME= '------ 1
FNAME(9-J:8)=
1
1
//FN(1:J-1)
DO 10 IP=1,IPMAX
READ(5,'CA)1,ERR=100,END=15) LINE
WKITEC6,1)
MFNAME,D,T,FNAME,IP
IL1=(IP-1)*1LMAX+1
IL2=IL1+ILMAX-1
WRITE(6,2) LINE,IL1
DO 10 ll=IL1+1,IL2
READ CS, 'CA)1,ERR=100,END=15)
LINE
WRITEC6,2) LINE,Il
10 CONTINUE
15 WRITE(?,*) FN//'WRITTEN ON OUTPUT'
CLOSE(5,STATUS='DELETE')
20 CONTINUE
30 CLOSE(7,STATUS='DELETE')
STOP 'PROGRAM FOUT'

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- UFOUT, 1 -------

PROGRAM UFOUT
******************************************************************
*WRITE UPDATE SOURCE FILES OF ATTACHED MASTERFILE ON OUTPUT.
*CALL: "UFOUT,FN1/FN2/FN3/ ••• ".
******************************************************************
CHARACTER*72 LINE,SLINE
CHARACTER*40 C1,C2
CHARACTER*27 MFNAME,MNAME
CHARACTER*30 D,T
CHARACTER*8 XFN,FN,FNAME,DNAME,SDECK
CHARACTER*5 SDECK
DATA SDECK/*COMDECK*/
DATA SDECK /*DECK*/
IFMAX=10
IPMAX=15U
ILMAX=60
C
C1='XXC=COMMAND.'
CALL FGET(C1)
OPEN(4,FILE='XXC')
READ(4, '(A)') MFN
FNAME= '------ 1
MFNAME(1:1)=MFN(1:1)'/1'
CLOSE(4,STATUS='DELETE')
C
CALL DATE(O)
CALL TIME(O)
C
OPEN(6,FILE='OUTPUT')
OPEN(7,FILE='XXOUT')
CALL CONNEC(7)
DO 30 IF=1,IFMAX
CALL GETPARM(XXFN,FN,IF)
IF(IF.EQ.1) THEN
30
-L.18-

--- MASTERFILE MCCL CY=30 ---- 02/07/86 - 00.13.50. ---------- UFOUT, 2 ---------

FN=XXFN
XXFN='XXXX'
ENDIF
400
IF(IF.EQ.-1) GOTO 40
C2=XXFN//'='//FN// 1 • I
CALL FGET(C2)
440
OPEN(5,FILE=XXFN)
480
J=INDEX(FN, ' ')
520
FNAME='-------- 1
1 //FN(1:J-1)
550
LNR=O
580
L=O
610
KEEP=O
640
N=O
670
DO 20 IP=1,IPMAX
680
READ (5, 1 (A), 1, ERR=100, END=25) LINE
690
LNR=LNR+1
700
WRITE(6,1) MFNAME,D,T,FNAME,IP
710
lF(LINE(1:8).EQ.SCDECK)
720
THEN
730
N=1
740
L=1
750
K=INDEX(LINE(10:17), ' ')
760
dNAME=LINE(10:8+K)//'.I
770
WRITE(6,3) LINE,L
780
ELSEIF(LINE(1:5).EQ.SCDECK) THEN
790
L=1
800
K=INDEX(LINE(7:14), ' ')
810
dNAME=LINE(7:5+K)//'.I
820
WRITE(6,2) LNR,DNAME
830
ELSE
840
L=L+1
850
IF(NKEEP.EQ.1) THEN
860
WRITE(6,2) LNR-1,DNAME
870
WRITEC6,3) SLINE,L-1
880
ELSE
890
WRITEC6,2) LNR,DNAME
900
END IF
910
WRITEC6,3) SLINE,L
920
END IF
930
IL=IL+1
940
IF(NKEEP.EQ.1) THEN
950
IL=IL+1
960
NKEEP=0
970
ENDIF
980
* DO LOOP ON IL.
10 READ(S, '(A)', 1, ERR=100, END=25) LINE
11 LNR=LNR+1
12 IF(LINE(1:8).EQ.SCDECK) THEN
13 L=1
14 K=INDEX(LINE(10:17), ' ')
15 dNAME=LINE(10:8+K)//'.I
16 WRITE(6,2) LNR,DNAME
17 ENDIF
18 WRITE(6,3) LINE,L
19 ENDIF
20 GOTO 20
21 END
22
L.19

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. -------- UFOUT, 3 --------
IF(L.1.L.E.ILMAX) GOTO 10
C * END DO LOOP ON IL.
20 CONTINUE
C 25 WRITE(7,*) FN//'WRITTEN ON OUTPUT'
30 CONTINUE
C 40 CLOSE(7,STATUS='DELETE')
STOP 'PROGRAM UFOUT'
100 STOP 'ERROR'
C 1 FORMAT(' ',A27,A10,'---',A10,'---',A8,'---',I3,'-----')
2 FORMAT(' ',5X,'(LINE ',I4,'O) ','A8')
3 FORMAT(' ',A72,17)
END

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. -------- ROUT, 1 --------
.PROC,ROUT,OUTPUT,SHIFT=N/Y,TID=XXB/,FID=XXIDX/XXIX/N,
IC=DIS/ASCII.
.* ROUTE FILE OUTPUT TO THE LINEPRINTER.
* .IF,.NOT.FILE(OUTPUT,AS), LERROR.
NOTE,$ERROR: FILE OUTPUT DOES NOT EXISTS.
REVERT,ABORT.
.ENDIF, LERROR.
* .REWRITE,OUTPUT.
* .IF,$SHIFT=$N$, SHIFTNY.
COPY,OUTPUT,FILMPL.
.* ELSE, SHIFTNY.
COPY$BF,OUTPUT,FILMPL.
.ENDIF, SHIFTNY.
ROUTE,FILMPL,DC=PR,#TID=TID,#FID=FID,#IC=IC.
REVERT.
.* .EXIT,S.
NOTE,$ERRORS.
REVERT,ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. -------- RIN, 1 --------
.PROC,RIN,JOB,TID=N/XXB,FID=XXIDX.
.* ROUTE JOB TO INPUT QUEUE OF TID.
* .IF,.NOT.FILE(JOB,AS), NOJOB.
NOTE,$FILE JOB DOES NOT EXIST; TRY AGAIN.
REVERT,ABORT.
.ENDIF, NOJOB.
* .REWRITE,JOB.
COPY,JOB,FILMPL.
.* .IF($TIDS.EQ.N) ROUTE,FILMPL,DC=IN,#FID=FID.
.* .IF($TIDS.NE.N) ROUTE,FILMPL,DC=IN,#TID=TID,#FID=FID.
REVERT.
.* .EXIT,S.
NOTE,$ERRORS.
REVERT,ABORT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. -------- LOC, 1 --------
.PROC,LOC,ZZ.
* MAKE REMOTE OUTPUT FILE ZZ LOCAL UNDER THE NAME Z ZZ.
* PAGE, AND ROUTE TO THE LINEPRINTER.
* LOCAL,ZZ,Z ZZ.
IF,.NOT.FILE(Z ZZ,AS), LERROR.
NOTE,$ERROR: FILE DOES NOT EXISTS.
RETURN,ASK.
REVERT,ABORT.
.ENDIF, LERROR.
.* PAGE,Z ZZ.
ASK. -
RETURN,ASK.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- LOC, 2 -------
REVERT. 150
* 160
DATA,ASK. 170
PROC,ASK=1,
ANSWER? ROUTE FILE? (N/Y1='XX1D0'/Y2='XX12X') =
= (N=0,Y1=1,Y2=2,0,1,2).
190
IF,ANSWER=0, LROUTE. 200
RETURN,ASK.
REVERT,ABORT. 210
ELSE, LROUTE. 220
REWIND,Z_II. 230
COPY,Z_II,FILMPL. 240
IF(ANSWER=1) ROUTE,FILMPL,DC=PR,TID=XXB,FID=XX1D0. 250
ENDIF, LROUTE. 260
REVERT. 270
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. 0--------
COPYMF, 280
* COPY MASTERFILE MF (ALL FILES EXCEPT EX=.)
MFUSE. 290
MFUSE,MF,=OLD.ID=XX1D0,MR=1.
MFNEW,MF,=NEW.ID=XX1D0.
FGET,COMMENT,M=OLD.
UPCOM. 300
FAOD,COMMENT,M=NEW.
MFSET,MSG=F,ABT=F.
MFCOPY,M=OLD,N=NEW,REP.
.IF,SEX.IS.NE.SNS, EXCEPT.
FDEL,EX,M=NEW.
MFDELETE,M=NEW.
.ENDIF, EXCEPT.
MFSET,MSG=PREV.
RETURN,OLD,NEW,COMMENT.
MFUSE,MF,=XX1D0.
NOTE,NEW CY OF MASTERFILE MF ATTACHED.
REVERT. 310
* EXIT,S.
NOTE,ERRORS.
REVERT,ABORT.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- UPCOM, 1 -------
PROGRAM UPCOM 10
C ****************************************************************** 20
C * PROGRAM UPDATING COMMENT FILE OF A MASTERFILE WITH NEW CY AND *
C DATE. TO BE USED WHEN COPYING MASTERFILES WITH COPYMF. *
C****************************************************************** 30
C
CHARACTER*72 LINE
CHARACTER*10 D
C
OPEN(5,FILE='COMMENT') 40
OPEN(6,FILE='STORE') 50
RECONT 6 60
DO 10 IL=1,1000 70
READ(S,1,END=20) LINE 80
WRITE(6,1) LINE 90
10 CONTINUE 100
C 20
REWIND 6 110
DO 10 IL=1,1000 120
READ(S,1,END=20) LINE 130
WRITE(6,1) LINE 140
10 CONTINUE 150
C 20
REWIND 5 160
READ(5,1) LINE 170
WRITE(6,1) LINE 180
C 20
READ(5,1) LINE 190
WRITE(6,1) LINE 200
C 20
READ(5,1) LINE 210
WRITE(6,1) LINE 220
C
I=INDEX(LINE,'CY=') 230
READ(LINE(I+3:I+5),'(I3)') NCY 240
NCY=NCY+1 250
IF(NCY.LT.10) THEN 260
WRITE(LINE(I+3:I+3),'(11)') NCY 270
ELSE IF(NCY.GE.10).AND.(NCY.LT.100) THEN 280
WRITE(LINE(I+3:I+4),'(12)') NCY 290
ELSE IF(NCY.GE.100) THEN 300

```fortran
WRITE(1,1) LINE
ENDF
WRITE(5,1) D(2:9)
DO 30 1L=4,1000
   READ(6,1,END=40) LINE
   WRITE(5,1) LINE
30 CONTINUE
40 CLOSE(5)
CLOSE(6,STATUS='DELETE')
STOP 'PROGRAM UPCOM'
```

```fortran
PROC,DU,FN,NID=XXXXX/
IF,NCY=.NE.N, OK.
RETURN,FN.
ATTACH,FN,NCY=CY, ID=XXIDX.
MFKILL,DUM,M=XXIDX.
RETURN,COPY,DUM.
REVERT.
ENDIF, OK.
NOTE,ERROR: YOU FORGOT TO SPECIFY NCY=.
REVERT,ABORT.
.*
.* IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFKILL,MF_FN=CY, ID=XXIDX.
FTAKE,ZZN=M=FN M.
.* IF(SM,NE,SNS) FTAKE,ZZN=FN S.
.* IF(FILMPL,NUL=NOUL).
REWIND,FILMPL.
COPY,FILMPL, FN S.
ED,USE,EDSUB.
SUB, S=FN, M=FN M, NUL=NUL.
REWIND,EDJOB.
COPY,EDJOB,FILMPL.
REWIND,FILMPL.
IF,SM,EQ,SNS, LRENAM.
COPY,FILMPL, FN S.
ELSE, LRENAM.
COPY,FILMPL, FN M.
ENDIF, LRENAM.
IF,SNORS=SNS, LROUTE.
```

```fortran
PROC,NEW,FN,N/Y, M=N/, NOUL=N/Y, NORM=Y, TID=XXB.
IF, SNS, EQ, SNS, LERROR.
NOTE,ERROR: YOU FORGOT TO SPECIFY SNS=.
REVERT,ABORT.
.*
```

```fortran
MASTERFILE MFCCL CY=30 02/07/86 00.15.22. --- UPOM, 2 ---
WRITE(LINE(1+3:1+5),'(13)') NCY
```

```fortran
MERGE JOB MFM OF ID=XXIDX TO ID=NID.
IF,NCY.NE.N, OK.
RETURN,FN.
ATTACH,FN,NCY=CY, ID=XXIDX.
MFKILL, DUM, M=XXIDX.
RETURN, COPY, DUM.
REVERT.
ENDIF, OK.
NOTE,ERROR: YOU FORGOT TO SPECIFY NCY=.
REVERT, ABORT.
.*
```

```fortran
PROC,NEW,FN,S=N/,M=N/,NOUL=N/Y, NORM=N/Y, TID=XXB.
```

```fortran
IF(SNORS=SNS, LERROR.
NOTE,ERROR: YOU FORGOT TO SPECIFY SNS=.
REVERT, ABORT.
.*
```

```fortran
IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFKILL,MF_FN=CY, ID=XXIDX.
FTAKE,ZZN=M=FN M.
IF(SM,NE,SNS) FTAKE,ZZN=FN S.
COPY,EDJOB,FILMPL.
REWIND,EDJOB.
COPY,EDJOB,FILMPL.
REWIND,FILMPL.
IF,SM,NE,SNS, LRENAM.
COPY,FILMPL, FN S.
ELSE, LRENAM.
COPY,FILMPL, FN M.
ENDIF, LRENAM.
IF,SNORS=SNS, LROUTE.
```
-L.22-

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

.L.22-

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- NEW, 2 ----------
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. --------------- R, 2 -------

**DEFAULT:** V=B=M, NO LISTING OF UPDATE CHANGES (UNLESS ULIST),

**JOB** R=M ROUTED TO INPUT QUEUE (UNLESS NOR),

**OUTPUT** APPEARS AT TERMINAL (UNLESS TID=).

**IF,** SUB=EQ.NS. OR SMS=EQ.NS, LERROR.

**NOTE,** SERROR: YOU FORGOT TO SPECIFY #U= OR #M=.

**RETURN,** EDSUB, REVERT, ABORT.

**ENDIF,** LERROR.

**EXIT,** S.

**NOTE,** SERROR.

**RETURN,** EDSUB, REVERT, ABORT.

**ENDIF,** LERROR.

**IF,** ($V$.EQ.$N$.AND.$B$.EQ.$N$), SUB,#U=U,#M=M,#V=M,#B=M,#ULIST=ULIST.

**IF,** ($V$.NE.$N$.AND.$B$.EQ.$N$), SUB,#U=U,#M=M,#V=V,#B=M,#ULIST=ULIST.

**IF,** ($V$.EQ.$N$.AND.$B$.NE.$N$), SUB,#U=U,#M=M,#V=M,#B=B,#ULIST=ULIST.

**IF,** ($V$.NE.$N$.AND.$B$.NE.$N$), SUB,#U=U,#M=M,#V=V,#B=B,#ULIST=ULIST.

**IF,** ($NOR=N$), LROUT.

**REWIND,** R M.

**COPY,** R M; FILMPL.

**ENDIF,** LROUT.

**EXIT,** S.

**NOTE,** SERROR.

**RETURN,** EDSUB, REVERT, ABORT.

**ENDIF,** LROUT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. --------------- X, 1 -------

**PROC,** X FN, #H=I, #Y=I, P=N/Y, O=N/Y, D=N/Y, T=N/Y, I0=N/Y, LP=NP/LP,

**NOTE,** SERROR. YOU FORGOT TO SPECIFY #B= OR #I=.

**RETURN,** EDSUB, REVERT, ABORT.

**ENDIF,** LERROR.

**EXIT,** S.
**--- MASTERFILE MFCL CY=30 ---- 02/07/86 - 00.15.22. ---------- X, 2 ------**

**NOTE,ERRORS**, 350
RETURN,ZZ,ZZI,ZZJ,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.

**USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUES PARAMETERS IN X_FN.**

**DATA,EDSUB.**

10=PROC, #B= #, #I= #, #P= #, #D= #.
20=REVERT,
30=DATA,X_B_I.
#1,ZZ,30
W, #SUB, #
SC, INIT
#B, #

**--- MASTERFILE MFCL CY=30 ---- 02/07/86 - 00.15.22. ---------- JOBCRD, 1 ------**

**PROGRAM JOBCRD**

10  

**C***

**PROGRAM REWRITING JOBCARD OF JOB ON FILE FN FOR GIVEN T,IO,LP.**

**C***

**CHARACTER*7 FN,DUM,T,IO,LP,XX,XXX**

**DATA T,IO,LP/3*'N'/**

**CALL GETPARM(FN,DUM,DUM)**

**OPEN(5,FILE=FN)**

**OPEN(6,FILE='TEMP')**

**REWIND 5**

**REWIND 6**

**DO 10 IL=1,1000**

**READ(5,1,END=20) LINE**

**10 WRITE(6,1) LINE**

**REWIND 5**

**REWIND 6**

**C***

**DETERMINE DEFAULT VALUES OF T,IO,LP.**

**READ(6,1) LINE**

**I=INDEX(LINE,',T')**

**IF(I.NE.0) THEN**

**II=INDEX(LINE(I+3:72),',')**

**IF(II.NE.0) THEN**

**T=LINE(I+2:II+11) ENDIF**

**J=INDEX(LINE,',IO')**

**IF(J.NE.0) THEN**

**JJ=INDEX(LINE(J+4:72),',')**

**IF(JJ.NE.0) THEN**

**IO=LINE(J+3:JJ+1) ENDIF**

**K=INDEX(LINE,',NP')**

**IF(K.NE.0) THEN**

**LP='NP' ELSE**

**LP=LINE(K+1:2) ENDIF**

**L=INDEX(LINE,',')**

**C***

**DETERMINE OVERWRITE VALUES OF T,IO,LP.**

**DO 30 N=1,3**

**CALL GETPARM(XX,XXX,IX)**

**IF(IX.EQ.-1) GOTO 40**

**IF((XX.EQ.'T').AND.(XXX.NE.'N')) T=XXX**

**IF((XX.EQ.'IO').AND.(XXX.NE.'N')) IO=XXX**

**IF((XX.EQ.'LP').AND.(XXX.NE.'N')) LP=XXX**

**30 IF((T.EQ.'N').OR.(IO.EQ.'N').OR.(LP.EQ.'N')) GOTO 100**

**II=INDEX(T,'-')-1**

**JJ=INDEX(IO,'-')-1**

**C***

**CORRECT FIRST LINE OF FN.**

**M=L**

**IF(I.NE.0) M=MIN(I,M) IF(J.NE.0) M=MIN(J,M) IF(K.NE.0) M=MIN(K,M)**

**LINE(M:72)='T///T(1:11)//','IO//I(1:JJ)//','LP(1:2)//',.'**

**C***
**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- JOBCRD, 2 --------**

```fortran
WRITE(5,1) LINE
DO 50 IL=Z,1000
READ(6,1,END=60) LINE
50 WRITE(5,1)
60 CLOSE(6,STATUS=DELETE)
STOP 'PROGRAM JOBCRD'
```

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- PLOUT, 1 --------**

```fortran
10 .PROC:PLOUT,LFN,NOVERSA=VY/VN,NODELETE=DT/DN.
20 * CONVERT LOCAL GRAPHFILE "LFN" TO G-CODE FILE,
30 * INSERT FIRST LINE "\\LFN,P,NOVERSA,NODELETE",
40 * AND ROUTE FILE TO DESTINATION XXA.
50 .IF,FILE(LFN,AS), OK.
60 RETURN,ZIOUT,ZIINP,ZIJOB.
70 REQUEST,ZIJOB,QT.
80 ATTACH,TRANSF.
90 TRANSF,1=LFN,0=ZIINP,IC=GRAPHFILE,OC=GCODE.
100 SPLIT.
110 REWIND,ZIOUT.
120 COPYSF,ZIOUT,ZIJOB.
130 ROUTE,ZIJOB,OC=PR,PIO=PPPPP,TIO=XXA,IC=DIS.
140 RETURN,ZIINP,ZIOUT,TMP,TRANSF.
150 REVERT.
160 .**
170 .ENDIF, OK.
180 NOTE,$ERROR: FILE LFN DOES NOT EXISTS.
190 RETURN,TMP.
200 REVERT,ABORT.
210 .**
220 .EXIT,S.
230 NOTE,$SOMETHING WENT WRONGS.
240 RETURN,TMP,TRANSF,ZIINP,ZIOUT.
250 REVERT,ABORT.
260 .**
270 .DATA,TMP
280 "\\LFN,P,NOVERSA,NODELETE"
290```

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- SPLIT, 1 --------**

```fortran
PROGRAM SPLIT
10 100.
20 30.
30 40.
40 50.
50 60.
60 70.
70 80.
80 90.
90 100.
100 110.
110 120.
120 130.
130 140.
140 150.
150 160.
160 170.
170 180.
180 190.
190 200.
200 210.
210 220.
220 230.
230 240.
240 250.
250 260.
260 270.
270 280.
280 290.
```

--- Masterfile MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- JOBCRD, 2 --------

--- Masterfile MFCCL CY=30 ---- 02/07/86 - 00.15.22. ---------- PLOUT, 1 --------

--- Masterfile MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- SPLIT, 1 --------

--- Masterfile MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- SPLIT, 1 --------
**START OF THE MAIN LOOP: READ INPUT FILE LINE BY LINE.**

10 READ(5, (A), ERR=100, END=200) LINE
LEN=INDEX(LINE, ' ') - 1

20 IF(INI) THEN
  IEND=INDEX(LINE, '<')
  IF(IEND GT 0) THEN
    INIBUF(INIPTR+1:INIPTR+IEND)=LINE(1:IEND)
    WRITE(6, (A) ) INIBUF(INIPTR+1:INIPTR+IEND)
    LEN=LEN-IEND
    LINE(LEN+1:NCHR)=LINE(IEND+1:LEN)
    LEN=LEN-IEND
    INITST=.FALSE.
    OUTPTR=0
    GOTO 20
  ELSE
    NLEN=LEN
    IF(LINE(NLEN:NLEN), ' ') THEN
      NLEN=NLEN-1
      GOTO 30
    ENDIF
    IF(LINE(NLEN:NLEN), 'F') THEN
      NLEN=NLEN-1
      OUTBUF(OUTPTR+1:OUTPTR+NLEN)=LINE(1:NLEN)
      WRITE(6, (A) ) OUTBUF(1:OUTPTR+NLEN)
      OUTPTR=LEN-NLEN
      OUTBUF(1:OUTPTR)=LINE(NLEN+1:LEN)
    ENDIF
    ENDIF
  ELSE
    INITST=.TRUE.
    INIPTR=0
    GOTO 20
  ENDIF
ELSE
  ISTART=INDEX(LINE, '>')
  IF(ISTART GT 0) THEN
    OUTBUF(OUTPTR+1:OUTPTR+ISTART)=LINE(1:ISTART)
    WRITE(6, (A) ) OUTBUF(1:OUTPTR+ISTART)
    LEN=LEN-ISTART
    LINE(LEN+1:NCHR)=LINE(ISTART+1:LEN)
    LEN=LEN-ISTART
    INITST=.TRUE.
    INIPTR=ISTART
    GOTO 20
  ELSE
    NLEN=LEN
    IF(LINE(NLEN:NLEN), ' ') THEN
      NLEN=NLEN-1
      GOTO 30
    ENDIF
    IF(LINE(NLEN:NLEN), 'F') THEN
      NLEN=NLEN-1
      OUTBUF(OUTPTR+1:OUTPTR+NLEN)=LINE(1:NLEN)
      WRITE(6, (A) ) OUTBUF(1:OUTPTR+NLEN)
      OUTPTR=LEN-NLEN
      OUTBUF(1:OUTPTR)=LINE(1:NLEN)
    ENDIF
    ENDIF
  ENDIF
ENDIF

30 IF(LINE(NLEN:NLEN), '; ') THEN
  NLEN=NLEN-1
  GOTO 30
ENDIF

40 STOP 'READ ERROR'
200 STOP 'SPLIT FINISHED - END OF FILE'
END
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- TOUT, 2 -------

.DATA,EDJOB.
E,LKN
5\\\\\\\LFN,T,NOVERA,NODELETE
W,ZJOUT
B

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- DT, 1 -------

PROGRAM DT
C
C ******************************************************************************
C * PRINT DATE AND TIME.                                                    *
C ******************************************************************************
C
CHARACTER*10 D,T
CALL DATE(D)
CALL TIME(T)
T(10:1l)):=.:I
OPEN(10,FILE='XXOUT')
CALL CONNEC(10)
WRITE(10,*) D,T
CLOSE(10,STATUS='DELETE')
STOP 'DT'
END

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- SYS, 1 -------

PROGRAM SYS
C
C ******************************************************************************
C * PROGRAM WRITING LOCAL FILE ZZSYS, CONTAINING PROCEDURE OF THE            *
C * SAME NAME, WHICH UPON CALLING SHOWS WHICH FILES OF SYSBULL AND           *
C * USERBULL HAVE BEEN CHANGED OVER THE PAST 7 DAYS.                        *
C ******************************************************************************
C
CHARACTER*10 DATUM
INTEGER YEAR,MONTH,DAY
DIMENSION M(0:12)
OPEN(5,FILE= 'ZZSYS')
REWIND 5
CALL DATE(DATUM)
READ(DATUM,'(7X,I2)') YEAR
READ(DATUM,'(4X,I2)') MONTH
READ(DATUM,'(1X,I2)') DAY
M2=0
DO 10 K=U,MONTH-1
M2=M2+M(K)
10 CONTINUE
M2=M2+DAY
M2=M2-7
KK=O
DO 20 K=O,12
IF(M2-M(K).GT.0)
M2=M2-M(K)
KK=KK+1
20 CONTINUE
WRITE(5,21) M2,KK,YEAR
WRITE(5,22) M2,KK,YEAR
STOP 'PROGRAM SYS'
END

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ---------- ZZSYS1, 1 -------

.PROC,ZZSYS1.
ZZSYS1,44257.
REVERT.
**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ------- ZZSYS2, 1 -------**

.* PROC,ZZSYS2,D=0. 
.* THE CALL OF SYSBULL AND C20SBUL WITH THE DATE OF 
.* THE PREVIOUS SESSION AND PRODUCING NEW PROCEDURE ZZSYS1 
.* WITH THE PRESENT DATE (TO BE USED NEXT TIME). 
SET,R1=D. 
SET,R2=R1. 
SYSBULL,R0=R1. 
C20SBUL,R0=R2. 
ZIS,R1. 
RETURN,ZZS. 
REVERT. 
.* DATA,ZIS. 
.* PROC,ZZS,DD. 
FREP,ZZSYS1,M=MFCCL. 
RETURN,ZZSYS1. 
REVERT. 
.* SAVEMF,ZZSYS1. 
.* PROC,ZZSYS1. 
ZIS2,DD. 
RETURN. 
---

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ------- AUD, 1 -------**

.* PROC,AUD,PR=N/Y,LO=N/Y,PF=N/,ID=XXIDX. 
.* COMPACT AUDIT OF ID=... (DEFAULT: AT THE TERMINAL). 
.* DIRECTORY OF MASTERFILES IF LO IS SPECIFIED. 
.* IF,SPr=SNS, PRNT. * AUDIT SHOWN ON TERMINAL SCREEN. 
CONNECT,OUTPUT. 
ELSE, PRNY. * AUDIT PRINTED AT LINEPRINTER. 
RETURN,OUTPUT. 
REVERT. 
.* IF,SPR=$N$, PRNY. 
.* IF(SPR,NE.$N$) AUDIT,#ID=ID,LF=DATA. 
.* IF,SLO=SYS, DIRECTORY. 
.* ENDIF, DIRECTORY. 
.* IF(SPR=$Y$) ROUTE,OUTPUT,DC=#PR,TID=XXB,FID=XXIDX. 
RETURN,DATA,OUTPUT. 
REVERT. 
.* NOTE,$ATTACHED MASTERFILE RETURNED$.
ADD,#ID=ID. 
RETURN,MSTR. 
.* ENDIF, DIRECTORY. 
.* IF(SPR=$Y$) ROUTE,OUTPUT,DC=#PR,TID=XXB,FID=XXIDX. 
RETURN,DATA,OUTPUT. 
REVERT. 
.* NOTE,$ERROR$.
REVERT,ABORT. 
---

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ------- DIR, 1 -------**

PROGRAM DIR

C ******************************************************************************
C * PROGRAM CONSTRUCTING DIRECTORY OF CONTENTS OF THE MASTERFILES *
C * SHOWN IN AUDIT OF ID=... (DEFAULT: ID=XXIDX). *
C * HOE CALL "AUDIT, ID=... LF=DATA," THEN "DIR, ID=...". *
C ******************************************************************************
C
C CHARACTER=133 LINE,BLANK
C CHARACTER=10 MFNAME(50),D,T,SAVMF
C CHARACTER=5 ID,IDNAME
C CHARACTER=3 CY(50),SAVECY
C
C DATA BLANK/ "/
C NMAX=50
C LMAX=62
C
C CALL GETPARM(ID,IDNAME,1) 
IF(ID,EQ.'-') IDNAME='XXIDX' 
OPEN(10,FILE='DATA') 
OPEN(20,FILE='STORE') 
OPEN(30,FILE='OUTPUT') 
REWIND 10
C
C * MAKE LIST OF MASTERFILES.
PROGRAM PASAUD (DATA, OUTPUT);
(*$R-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- DIR, 2 -------

IFLAG=0
N=0
DO 10 I=1,1000
  READ(10,1,END=20,ERR=100) LINE
  IF(LINE(6:13).EQ.'AUDIT OF') IFLAG=1
  IF(LINE(6:24).EQ.'OWNER-ID STATISTICS') IFLAG=0
  IF(IFLAG.EQ.1).AND.(LINE(6:10).EQ.IDNAME).AND.
               (LINE(18:19).EQ.'MF') THEN
    N=N+1
  IF(N.GT.NMAX) GOTO 100
  MFNAME(N)=LINE(18:27)
  CY(N)=LINE(60:62)
END IF
10 CONTINUE

* ALPHABETICAL ORDER.
20 DO J=2,N
  DO K=J,N
    IF(MFNAME(J-1).LT.MFNAME(K)) GOTO 25
    IF((MFNAME(J-1).GT.MFNAME(K)).AND.(CY(J-1).LT.CY(K))) GOTO 25
    SAVEMF=MFNAME(K)
    SAVECY=CY(K)
    MFNAME(K)=MFNAME(J-1)
    CY(K)=CY(J-1)
    MFNAME(J-1)=SAVEMF
    CY(J-1)=SAVECY
  25 CONTINUE

CALL DATE(D)
CALL TIME(T)

* LIST CONTENTS OF THE MASTERFILES.
IP=1
DO 60 K=1,N
  REWIND 20
  CALL MFUSECMFNAMECK)//',M=MASTER,CY=1//CY(K)//1
  !D='//IDNAME//
  CALL MF LIST(1,M=MASTER,L=STORE,LO,C.
  60 CONTINUE
  WRITE(30,2) IONAME,D,T,IP,MFNAME(K),CY(K)
  L=L+1
  IF(L.LT.LMAX) GOTO 60
  IP=IP+1
  IF(IP.GT.IP MAX) GOTO 100
  WRITE(30,3)
  CLOSE(20,STATUS='DELETE')
END
100 STOP 'PROGRAM DIR'

FORMAT(*DIR AT(A)
2 FORMAT('DIRECTORY OF ID=1,A5,10X,A10,1X,A10,13X,'PAGE',13//
A 11X,'MASTERFILE',1,A10,'CY=',A3//
3 FORMAT('/ END DIRECTORY')

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD, 1 -------

PROGRAM PASAUD (DATA, OUTPUT);
(*$S-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD, 1 -------

PROGRAM PASAUD (DATA, OUTPUT);
(*$S-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD, 1 -------

PROGRAM PASAUD (DATA, OUTPUT);
(*$S-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD, 1 -------

PROGRAM PASAUD (DATA, OUTPUT);
(*$S-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD, 1 -------

PROGRAM PASAUD (DATA, OUTPUT);
(*$S-*

CONST LENGTH = 136; ENDLINE = 137;
WRD10 = 10;
C72 = 72;

TYPE LINE = ARRAY [1..ENDLINE] OF CHAR;
WORD = PACKED ARRAY[1..WRD10] OF CHAR;
ERRSTRING = PACKED ARRAY[1..20] OF CHAR;
PFARR = ARRAY [1..C72] OF CHAR;
PP = "PFINFO";
PFINFO = RECORD
NAME: PACKED ARRAY [1..10] OF CHAR;
CY: INTEGER;
INFO: PACKED ARRAY [1..72] OF CHAR;
P: PFP
END;

VAR DATA : TEXT;
INPOS,INLENGTH,TOTAL : INTEGER;
INLINE : LINE; TXT : WORD;
USERID : WORD;
NRFILES : INTEGER;
EOPF: BOOLEAN;
PFLIST: PFP;
PRUS,RBS,CYCLES,PRU2,RB2: INTEGER;
DAY,TIME: WORD;

PROCEDURE (*$E'PASAUD'*)
VAR I,J : INTEGER;
BEGIN J := 0;
INLINE[ENDLINE] := ' ';
FOR I:=1 TO LENGTH DO
IF EOLN(DATA) THEN
ELSE BEGIN J:=J+1;
READLN(DATA);
INPOS := 0;
INLENGTH := J
END; (* Of GETLINE *)

PROCEDURE SKIPSPACES;
DO ' ' INLINE[J];
END;
UNTIL (INPOS = ENDLINE) OR (INLINE[INPOS] <> 1);
INPOS := INPOS - 1;

PROCEDURE GETWOORD(VAR TXT: WORD; VAR TOTAL: INTEGER);
VAR I,J: INTEGER; A: LINE;
BEGIN SKIPSPACES;
I := 0;
INPOS := INPOS + 1;
WHILE (INPOS<ENDLINE) AND (INLINE[INPOS] <> ' ') DO
BEGIN I := I + 1;
A[I] := INLINE[INPOS]; INPOS := INPOS + 1
END;
INPOS := INPOS - 1;
FOR J := I+1 TO WRD10 DO A[J] := ' ';
TOTAL := I;
PACK(A,1,TXT)
END; (* Of GETWOORD *)

PROCEDURE FINDWOORD(INWOORD: WORD);
VAR FLAG: BOOLEAN; TXT: WORD; TOTAL: INTEGER;
BEGIN FLAG := TRUE;
WHILE (NOT EOF(DATA)) AND FLAG DO
BEGIN GETLINE;
GETWOORD(TXT,TOTAL);
FLAG := TXT <> INWOORD
END;
END; (* Of FINDWOORD *)

FUNCTION FINDAUDIT: BOOLEAN;
VAR TXT : WORD; TOTAL: INTEGER;
BEGIN REPEAT FINDWOORD('1 AUDIT');
GETWOORD(TXT,TOTAL);
UNTIL (TXT = 'AUDIT ') OR EOF(DATA);
FINDAUDIT := NOT EOF(DATA)
END; (* Of FINDAUDIT *)

PROCEDURE INSERT (ITEM: PFP);
VAR PF,PFPLUS: PFP;
ACY: INTEGER; ANAME : PACKED ARRAY[1..10] OF CHAR;
BEGIN WITH ITEM.
BEGIN P:= NIL; ANAME:= NAME; ACY:= CY END;
LASTPF'.P:= ITEM;
PF:= PFLIST; PFPLUS:= PF'.P;
WHILE ACY > PFPLUS'.NAME DO
BEGIN PF:= PF'.P; PFPLUS:= PF'.P END;
WHILE (ANAME=PFPLUS'.NAME) AND (ACY=PFPLUS'.CY) DO
BEGIN PF:= PF'.P; PFPLUS:= PF'.P END;
IF PFPLUS = ITEM THEN LASTPF:= ITEM ELSE
BEGIN PF'.P:= ITEM; ITEM'.P:= PFPLUS; LASTPF'.P:= NIL END;
END; (* Of INSERT *)
FUNCTION IGET (POS: INTEGER): INTEGER;
VAR I: INTEGER;
BEGIN I := 0; INPOS := POS; SKIPSPACES;
INPOS := INPOS + 1;
WHILE INLINE(INPOS) IN ['0'..'9'] DO
BEGIN I := 10 * I + ORD(INLINE(INPOS)) - ORD('0');
END;
INPOS := INPOS - 1;
IGET := I
END; (* OF IGET *)

PROCEDURE DAYANDTIME;
CONST D = 90; T = 76; VAR
TOTAL: INTEGER;
BEGIN INPOS := D; GETWOORD(DAY,TOTAL);
INPOS := T; GETWOORD(TIME,TOTAL);
END; (* OF DAYANDTIME *)

PROCEDURE GETLERR(A: ERRSTRING);
BEGIN IF EOF(DATA) THEN
BEGIN MESSAGE(A);
HALT END ELSE GETLINE
END; (* OF GETLERR *)

PROCEDURE GETPFINFO;
CONST PFNAAM = 17; CYNR = 59; VSNNR = 75; PRU = 85;
NOATTACHES = 24; TOTAL: INTEGER;
BEGIN NRFILES := NRFILES + 1;
NEW(PF);
FOR I := 1 TO C72 DO PFA[I] := 1;
PF.CY := IGET(CYNR);
PRUS := PRUS + IGET(PRU);
COPY(PFNAAM,1,10); COPY(CYNR,11,13);
COPY(VSNNR,67,72); COPY(PRU,22,25);
PACK(INLINE,CREATION+1,DATE1);
PACK(INLINE,ALTERATION+1,DATE2);
IF DATE1=DATE2 THEN BEGIN DATE1 := "TODAY ";
UNPACK(DATE2,INLINE,ALTERATION+1) END;
IF DATE1=DAY THEN BEGIN DATE1 := "TODAY ";
UNPACK(DATE1,INLINE,CREATION+1) END;
IF DATE2=DAY THEN BEGIN DATE2 := "TODAY ";
UNPACK(DATE2,INLINE,ALTERATION+1) END;
COPY(CREATION,32,39);
COPY(ALTERATION,62,69);
PACK(INLINE,ATTACH+1,DATE1);
IF DATE1=DAY THEN BEGIN DATE1 := "TODAY ";
UNPACK(DATE1,INLINE,ATTACH+1) END;
COPY(ATTACH,52,59);
PACK(INLINE,ATTACH+1,DATE1);
GETPFA('*' = EOF IN GETPFINFO');
RBS := RBS + IGET(RB);
COPY(NOATTACHES,61,64); COPY(_FLAGS,13,16); COPY(RB,27,29);
PACK(PFA,1,PF.NAME);
FOR I := 1 TO 10 DO IF PFA[I] = ' ' THEN PFA[I] := 'CHR(0);'
INSERT(PF);
END; (* OF GETPFINFO *)

PROCEDURE GETGENINFO;
VAR I: INTEGER;
8 E GIN CYCLES := IGET(INPOS);
PRU2 := IGET(INPOS);
PRU2 := IGET(INPOS);
END; (* OF GETGENINFO *)

PROCEDURE HEADER(VAR PAGENR: INTEGER);
BEGIN
FUNCTION IGET (POS: INTEGER): INTEGER;
VAR I: INTEGER;
BEGIN I := 0; INPOS := POS; SKIPSPACES;
INPOS := INPOS + 1;
WHILE INLINE(INPOS) IN ['0'..'9'] DO
BEGIN I := 10 * I + ORD(INLINE(INPOS)) - ORD('0');
END;
INPOS := INPOS - 1;
IGET := I
END; (* OF IGET *)

PROCEDURE DAYANDTIME;
CONST D = 90; T = 76; VAR
TOTAL: INTEGER;
BEGIN INPOS := D; GETWOORD(DAY,TOTAL);
INPOS := T; GETWOORD(TIME,TOTAL);
END; (* OF DAYANDTIME *)

PROCEDURE GETLERR(A: ERRSTRING);
BEGIN IF EOF(DATA) THEN
BEGIN MESSAGE(A);
HALT END ELSE GETLINE
END; (* OF GETLERR *)

PROCEDURE GETPFINFO;
CONST PFNAAM = 17; CYNR = 59; VSNNR = 75; PRU = 85;
NOATTACHES = 24; TOTAL: INTEGER;
BEGIN NRFILES := NRFILES + 1;
NEW(PF);
FOR I := 1 TO C72 DO PFA[I] := 1;
PF.CY := IGET(CYNR);
PRUS := PRUS + IGET(PRU);
COPY(PFNAAM,1,10); COPY(CYNR,11,13);
COPY(VSNNR,67,72); COPY(PRU,22,25);
PACK(INLINE,CREATION+1,DATE1);
PACK(INLINE,ALTERATION+1,DATE2);
IF DATE1=DATE2 THEN BEGIN DATE2 := "TODAY ";
UNPACK(DATE2,INLINE,ALTERATION+1) END;
IF DATE1=DAY THEN BEGIN DATE1 := "TODAY ";
UNPACK(DATE1,INLINE,CREATION+1) END;
IF DATE2=DAY THEN BEGIN DATE2 := "TODAY ";
UNPACK(DATE2,INLINE,ALTERATION+1) END;
COPY(CREATION,32,39);
COPY(ALTERATION,62,69);
PACK(INLINE,ATTACH+1,DATE1);
IF DATE1=DAY THEN BEGIN DATE1 := "TODAY ";
UNPACK(DATE1,INLINE,ATTACH+1) END;
COPY(ATTACH,52,59);
PACK(INLINE,ATTACH+1,DATE1);
GETPFA('*' = EOF IN GETPFINFO');
RBS := RBS + IGET(RB);
COPY(NOATTACHES,61,64); COPY(_FLAGS,13,16); COPY(RB,27,29);
PACK(PFA,1,PF.NAME);
FOR I := 1 TO 10 DO IF PFA[I] = ' ' THEN PFA[I] := 'CHR(0);'
INSERT(PF);
END; (* OF GETPFINFO *)

PROCEDURE GETGENINFO;
VAR I: INTEGER;
8 E GIN CYCLES := IGET(INPOS);
PRU2 := IGET(INPOS);
PRU2 := IGET(INPOS);
END; (* OF GETGENINFO *)

PROCEDURE HEADER(VAR PAGENR: INTEGER);
BEGIN
-- MASTERFILE MFCCL CY=50 ---- 02/07/86 - 00.16.38. -------- PASAUD, 4 --------

WRITE( 'AUDIT OF ID = ', USERID);  1730
WRITE( ',DAY ', DATE);  1740
WRITELN( 'PAGE ', PAGENR: 2);  1750
WRITE( '13,NRFILES=4,' FILES');  1760
WRITE(PRUS:6, ', PRUS');  1770
WRITE(RUS:5, ', RBS');  1780
WRITELN( '1 RDS=57 RUS=3648 WORDS');  1790
WRITE( 'AVERAGE: ', PRUS/NRFILES: 6: 1, ', PRUS/FILE');  1800
WRITE(RUS/NRFILES:4: 1, ' RBS/FILE');  1810
WRITELN( EFFICIENCY: ', PRUS/RBS: 6: 1, ' PROC.');  1820
WRITEC( 'FILENAME CY FG FT PRU RBS');  1830
WRITEC( 'CREATED ALTERED ATTACHED # AT VSN');  1840
WRITELN( ' (1 RB= 57 PRUS= 3648 WORDS)');  1850
WRITE( 'AV: ', PRUS/NRFILES: 6: 1, ', PRUS/FILE');  1860
WRITE(RBS/NRFILES:4: 1, ' RB/FILE');  1870
WRITELN( 'AUDIT FINISHED.');  1880
END; (* OF HEADER *)  1890

PROCEDURE PRINTLIST(VAR PF: PFP; NRLINES: INTEGER);  1900
CONST SIZE = 60; VAR I: INTEGER;  1910
BEGIN I := 0;
WHILE PF <> NIL AND I < NRLINES DO BEGIN I := I + 1;
WRITELN( '1PFA.INFO) ; PF := PFA.P;  1930
END;  1940
END; (* OF PRINTLIST *)  1950

PROCEDURE PRINTALL(NRLINES: INTEGER);  1960
VAR PF: PFP; PAGENR: INTEGER;  1970
BEGIN PF := PPLIST.P;
PAGENR := 1;
WHILE PF <> NIL DO BEGIN HEADER(PAGENR);
PRINTLIST(PF,NRLINES);
END;
WRITELN; WRITELN( 'AUDIT FINISHED.');  1990
END; (* OF PRINTALL *)  2000

BEGIN RESET (DATA);  2010
WHILE FINDAUDIT DO BEGIN DAYANDTIME;
FINDWORD( 'FLAGS=A=AR' );
NRFILES := 0;
PRUS := 0;
RUS := 0;
NEW(PPLIST); LASTPF := PPLIST;
PPLIST.P := NIL;
GETLERR( '*** EOF BEFORE PF-S');  2060
GETLINE;
GETWOORD(CUSERID,TOTAL);  2070
GETPFINFO;
EOPF := FALSE;
WHILE NOT(EOPF OR EOF(DATA)) DO BEGIN GETLINE;
GETWOORD(TXT,TOTAL);
IF TXT = USERID THEN GETPFINFO ELSE BEGIN
PRINTLIST(55);
END;  2080
END;  2090
END; (* OF WHILE FINDAUDIT *)  2100

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- ATT750, 1 --------

.PROC ATT750=1,
ID USER IDENTIFICATION = ] = ('N='XXIDX,'A').  10
* HELP /* NLIST.
ATT750 ATTACHES AND RETURNS ALL PERFILES OF ID.. ON THE 750 BY 20
RUNNING PROGRAM KEEP AND PROCEDURE ZZATT WHICH IT PRODUCES.
DEFAULT: ID=XXIDX.  30
.ENDHELP.
40
*** MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. --------- ATT750, 2 ---------

RETURN,ZZATT,ZZAUD,BIN.
CONNECT,OUTPUT.
AUDIT,#ID=10,LF=ZZAUD,AT=P.
REWIND,ZZPROG.
FTNS,1=ZZPROG,B=BIN,L=0.
BIN.
.IF,$IDS.EQ.$XXIDX$, LRET.
MUFSE.
NOTE,ATTACHED MASTERFILE RETURNED.
RETURN,CCLLIB.
.ENDIF, LRET.
ZZATT.
.IF,$10$,EQ.$XXIDX$, LATT.
ATTACH,CCLLIB,#ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
ENDIF, LATT.
AUDIT,#ID=ID,LF=DATA.
PAS AUD.
RETURN,ZZATT,ZZAUD,ZZPROG,BIN,DATA,OUTPUT.
REVERT.
* * EXIT,S.
RETURN,ZZATT,ZZAUD,ZZPROG,BIN,DATA.
REVERT,ABORT.**SOMETING WRONG**
DATA,ZZPROG.
PROGRAM KEEP((INPUT,OUTPUT,ZZAUD,ZZATT,TAPE5=ZZAUD,TAPE6=ZZATT)
C ******************************************************************
C THIS PROGRAM READS THE OUTPUT FILE 'ZZAUD' PRODUCED BY
C ''AUDIT,AI=P,LF=ZZAUD,ID=''
C AND WRITES THE PERMFILES FOUND ON A PROCEDURE FILE 'ZZATT', WHICH
C WILL ATTACH (AND SUBSEQUENTLY RETURN) ALL THESE FILES.
C THE PAGES OF ZZAUD SHOULD CONTAIN HEADERS OF AT MOST 54 FILES.
C ******************************************************************
C
CHARACTER PFN*40,#ID*9,DUMMY*1U
INTEGER CYCLE
DATA NF!LES/54/.
C
REWIND 5
REWIND 6
C
* WRITE CCL PROCEDURE ZZATT ON FILE ZZATT.
WRITE(6,1)
DO 10 IPAGE=1,10UO
DO 10 I=1,5
10 READ(5,1) DUMMY
DO 20 I=1,NFILES
READ(5,11,END=100,ERR=200) #ID,PFN,CYCLE
IF(PFN(1..1).EQ.1.IAND.#ID.EQ.1.IAND.CYCLE.EQ.U) GOTO 100
WRITE(6,12) PFN,#ID,CYCLE
20 CONTINUE
C
C * END OF AUDIT FILE REACHED.
100 WRITE(6,101)
STOP 'END AUDIT - NORMAL TERMINATION'
C
C * SIGNAL TROUBLES.
200 WRITE(6,201)
STOP 'ERROR WHILE READING AUDIT FILE'
C
C * END OCCURS IN HEADER.
300 IF(IPAGE.EQ.1) THEN
WRITE(6,301)
ELSE
GOTO 100
ENDIF
C
C * FORMATS.
1 FORMAT('PROC,ZZATT,'/'RETURN,A,')
11 FORMAT(5X,A9,3X,A40,2X,13)
12 FORMAT('ATTACH,A='/'RETURN,A,').
A ' #ID='A9,','CY='13,'MR=1. '/RETURN,A,'.
101 FORMAT('COMMENT.END OF AUDIT FILE REACHED')
201 FORMAT('COMMENT.SOMETHING WRONG WITH READING AUDIT FILE')
301 FORMAT('COMMENT.NO INFORMATION IN AUDIT FILE')

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. --------- ATT750, 2 ---------
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. ------- ATT750, 3 -------
END

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. ------- ADDP, 1 -------
.
PROC,ADDP*,
NAME [PROCEDURE NAME - ] = (*F),
**
HELP/,NULIST.
ADDP  ADDS A PROCEDURE TO A LIBRARY WHICH HAS TO BE ATTACHED
       WITH FULL PERMISSION.
       IF NECESSARY, FIRST DO: "RETURN,LIB", "ATTACH,LIB,ID=..".
       PARAMETERS:
       NAME - NAME OF THE PROCEDURE
       LIB - NAME OF THE LIBRARY, DEFAULT: "LIBRARY".

.ENDHELP.
**
NOTE,FILE(NAME,AS), NONAME.
IF,FILE(NAME), AS), NONAME.
NAME - NAME OF THE PROCEDURE.
LIB - NAME OF THE LIBRARY.
LIBRARY(LIB,OLD)
ADD(*,NAME,AL=1)
FINISH.
ENDRUN.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. ------- DELP, 1 -------
.
PROC,DELP*,
NAME [PROCEDURE NAME - ] = (*F),
**
HELP/,NULIST.
DELP  DELETES A PROCEDURE FROM A LIBRARY WHICH HAS TO BE ATTACHED
       WITH FULL PERMISSION.
       IF NECESSARY, FIRST DO: "RETURN,LIB", "ATTACH,LIB,ID=..".
       PARAMETERS:
       NAME - NAME OF THE PROCEDURE
       LIB - NAME OF THE LIBRARY, DEFAULT: "LIBRARY".

.ENDHELP.
**
NOTE,FILE(NAME,AS), NONAME.
IF,FILE(NAME), AS), NONAME.
NAME - NAME OF THE PROCEDURE.
LIB - NAME OF THE LIBRARY.
LIBRARY(LIB,OLD)
DELETE(NAME)
FINISH.
ENDRUN.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. ------- GETP, 1 -------
.
PROC,GETP*,
NAME [PROCEDURE NAME - ] = (*F),
**
HELP/,NULIST.
GETP  GETS A PROCEDURE FROM A LIBRARY.
       PARAMETERS:
       NAME - NAME OF THE PROCEDURE
       LIB - NAME OF THE LIBRARY, DEFAULT: "LIBRARY".

.ENDHELP.
**
NOTE,FILE(NAME,AS), NONAME.
RECEIVE,NAME.
NAME - NAME OF THE PROCEDURE.
LIB - NAME OF THE LIBRARY.
LIBRARY(LIB,OLD)
GET(*,NAME,AL=1)
FINISH.
ENDRUN.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. --------- GETP, 2 -------

IF, FILE(ZZGETP1, AS), OK.
RETURN, NAME.
REWIND, ZZGETP1.
COPY, ZZGETP1, NAME.
RETURN, ZZGETP1.
REWIND, NAME.
ENDIF, OK.

* DATA, ZZGETP2.
LIBRARY(DUM, NEW)
ADD(NAME, LIB, #LIB)
FINISH.
ENDRUN.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. --------- REPP, 1 -------

.PROC, REPP*I,
NAME [PROCEDURE NAME = ] = (*F),
LIB [LIBRARY NAME = ] = (*F, AN=LIBRARY).
* HELP, NOLIST.
REPP REPLACES A PROCEDURE IN A LIBRARY WHICH HAS TO BE ATTACHED
===
WITH FULL PERMISSION.
IF NECESSARY, FIRST DO: "RETURN, LIB", "ATTACH,LIB,ID=..".
===
PARAMETERS:
NAME - NAME OF THE PROCEDURE
LIB - NAME OF THE LIBRARY, DEFAULT: "LIBRARY".

.ENDHELP.

* *.IF, NOT. FILE(NAME, AS), NONAME.
NOTE, $FIL6 NAME DOES NOT EXIST; TRY AGAIN.
REVERT, ABORT.
* .ENDIF, NONAME.

* EDITLIB, I=ZZREP1, L=ZZREP2.
* IF(FILE(LIB, PF)) EXTEND, LIB.
RETURN, ZZREP1, ZZREP2.
* DATA, ZZREP1.
LIBRARY(LIB, OLD)
REPLACE(*, NAME, AL=1)
FINISH.
ENDRUN.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. --------- ADD205, 1 -------

.PROC, ADD205*I,
NAME [NAME OF THE FILE = ] = (*F),
CODE [CODE? CD/A] = J
* HELP, NOLIST.
ADD205 COPIES A FILE TO THE 205 AND MAKES IT PERMANENT THERE.
===
PARAMETERS:
NAME - NAME OF THE FILE TO BE COPIED
CODE - INDICATES WHETHER FILE IS DISPLAY OR ASCII CODED:
"D" OR "C6" INDICATES DISPLAY CODE (DEFAULT)
"A" OR "C8" INDICATES ASCII CODE.

.ENDHELP.

* *.IF, NOT. FILE(NAME, AS), NONAME.
REVERT.
* .ENDIF, NONAME.

* REWIND, ZZADD.
* CALL MFLINK.
NOTE, $MFLINK RESPONDS:
MFLINK, NAME, ST=205, DD=CODE, I=ZZADD.
* SKIP, OVEREXIT.
EXIT.
NOTE, $SOMETHING WENT WRONG.
ENDIF, OVEREXIT.
RETURN, ZZADD.
* *.INPUT FOR MFLINK.
* DATA, ZZADD.
USER(AC=XXXXACXX, U=XXU1XX, PA=XXPAX)
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- AD0205, 2 --------
MFTAKE,NAME.
DEFINE,NAME.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- DEL205, 1 --------

.PROC,DEL205*I,
NAME [NAME OF THE FILE -] = (*F),
NAM2 2ND FILE? (Y/N) - } = (*N=N,Y=N,F),
NAM3 3RD FILE? (Y/N) - } = (*N=N,Y=N,F),
NAM4 4TH FILE? (Y/N) - } = (*N=N,N=N,F).

HELP, NOLIST.
DEL205 DELETES PERMFILE(S) FROM THE 205.

PARAMETERS:
NAME - NAME OF THE FILE TO BE DELETED
NAM2 - 2ND FILE TO BE DELETED (OPTIONAL)
NAM3 - 3RD FILE TO BE DELETED (OPTIONAL)
NAM4 - 4TH FILE TO BE DELETED (OPTIONAL)
NAMS - 5TH FILE TO BE DELETED (OPTIONAL).

.ENDHELP.

REWIND, ZIDEL1.

* CALL MFLINK.
NOTE,$MFLINK RESPONDS:$.
MFLINK,ZIDEL2,ST=205,DD=CODE,I=ZIDEL1.

* SKIP, OVEREXIT.
EXIT.
NOTE,$SOMETHING WENT WRONG$.
ENDIF, OVEREXIT.
RETURN,ZIDEL1,ZIDEL2.

* INPUT FOR MFLINK.
.DAT,A,ZIDEL1.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
PURGE,NAME.

* IF($NAM2$.$NE.$N$)PURGE,NAM2.
* IF($NAM3$.$NE.$N$)PURGE,NAM3.
* IF($NAM4$.$NE.$N$)PURGE,NAM4.
* IF($NAM5$.$NE.$N$)PURGE,NAM5.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- GET205, 1 --------

.PROC, GET205*I,
NAME [NAME OF THE FILE -] = (*F),

HELP, NOLIST.
GET205 GETS A PERMFILE FROM THE 205.

PARAMETERS:
NAME - NAME OF THE FILE TO BE COPIED
CODE - INDICATES WHETHER FILE IS DISPLAY, ASCII, OR BINARY
CODE:
"D" OR "C6" INDICATES DISPLAY CODE (DEFAULT)
"A" OR "C8" INDICATES ASCII CODE
"B" OR "US" INDICATES BINARY CODE (USE FOR PLOTFILES).

.ENDHELP.

REWIND, ZIGET.

* CALL MFLINK.
NOTE,$MFLINK RESPONDS:$.
MFLINK,NAME,ST=205,DD=CODE,I=ZIGET.

* SKIP, OVEREXIT.
EXIT.
NOTE,$SOMETHING WENT WRONG$.
ENDIF, OVEREXIT.
RETURN,ZIGET.

* INPUT FOR MFLINK.
.DAT, ZIGET.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
ATTACH, NAME.
MFGIVE, NAME.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. ------- RNM205, 1 -------

.PROC,RNM205*I,
.OLD OLD NAME OF THE FILE -J = (*F),
*HELP, NOLIST.

RNM205 CHANGES THE NAME OF A FILE ON THE 205.

 PARAMETERS:
 OLD - OLD NAME OF THE FILE
 NEW - NEW NAME OF THE FILE.
.ENDHELP.
* IF,OLD$=$NEW$, LEQUAL.
NOTE,NEW NAME EQUALS OLD ONE.
REVERT.
.ENDIF, LEQUAL.

ROUTE,FILMPL,DC=IN,ST=205.

JOB~
DATA,FILMPL.

XXI2X,ST2U5.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCECTL=10,WS=128,LP=0)

* HELP,,NOLIST.

AUD205 GIVES A SURVEY OF THE PERMANENT FILES OF U=XXU1XX ON THE 205.

 PARAMETERS:
 LO - LENGTH OF THE AUDIT; P (PARTIAL, DEFAULT) / F (FULL)
 OUT - NAME OF THE OUTPUT FILE, DEFAULT IS THE TERMINAL.

REWIND,ZZAUD1.
NOTE,MFLINK RESPOND.$$.
MFLINK,ZZAUD2,ST=205,DD=C6,I=ZZAUD1.
* COPY 205 OUTPUT TO FILE OUT.
* IF,OUT$=$ZZTERM, TOTERM.
CONNECT,ZZTERM.
RETURN,EDLOG.

ED,USE,ZZAUD3.
REWIND,ZZAUD2.
COPYR,ZZAUD2,ZZTERM,1.
RETURN,ZZAUD3,ZZTERM,EDLOG.

ELSE, TOTERM.
REWIND,ZZAUD2.
COPYR,ZZAUD2,OUT.
ENDIF, TOTERM.

* INPUT FOR MFLINK.
* DATA,ZZAUD1.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
AUDIT,#LO=LO,L=OUT.
MFGIVE,OUT.
* USEFILE FOR THE EDITOR.
* DATA,ZZAUD3.
SC,F,ZZAUD4
F,L=90
E,ZZAUD2
/1="/1)cc*
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 09.17.58. --------- AUD205, 2 ----------

W, ZIAUD2, 0
B, Q
.ENDIF, USEFILE.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 09.17.58. --------- ATT205, 1 ----------

.PROC, ATT205, I,
   U, [USER IDENTIFICATION = ] = (\&N=XXU1XX, *A),
   AC, [ACCOUNT NUMBER = ] = (\&N=XXXACXXX, *A),
   PA, [PASSWORD = ] = (\&N=XPAX, *A),
   TID, [TERMINAL IDENTIFICATION = ] = (\&N=XXB, *A),

HELP, NOLIST.
ATT205 ATTACKS ALL PERMFILES ON THE 205, PERFORMS AN AUDIT, AND RUNS PROGRAM SAVE TO SET THE DATE OF LAST ACCESS ON TODAY.

.HELP, NOLIST.
ATT205 ATTACKS ALL PERMFILES ON THE 205, PERFORMS AN AUDIT, AND RUNS PROGRAM SAVE TO SET THE DATE OF LAST ACCESS ON TODAY.

.DEFAULTS: U=XXU1XX, AC=XXXACXXX, PA=XPAX, TID=XXB, FID=XXI2X.

.ENDHELP.

.*
RETURN, FILMPL.
REWIND, ZIJOB, ZIPROG.
COPYR, ZIJOB, FILMPL.
COPYR, ZIJOB, ZIPROG, FILMPL.
ROUTE, FILMPL, DC=1M, ST=205, #TID=TID, #FID=FID.

.*

JOB: DATA, ZIJOB.
FID, ST205, USER(#AC=AC, #U=U, #PA=PA)
RESOURCE(TL=100, WS=128, LP=1, PRI0=12)
COMMENT.*******************

.*
READ, ZIJOB, ZIPROG.

.PROC, PROGRAM, I,
   DATA, ZIJOB.
   FID, ST205.
   USER(#AC=AC, #U=U, #PA=PA)
   RESOURCE(TL=100, WS=128, LP=1, PRI0=12)
   COMMENT.*******************

.*
READ, ZIJOB, ZIPROG.

.PROC, PROGRAM, I,
   DATA, ZIJOB.
   FID, ST205.
   USER(#AC=AC, #U=U, #PA=PA)
   RESOURCE(TL=100, WS=128, LP=1, PRI0=12)
   COMMENT.*******************

.*
READ, ZIJOB, ZIPROG.

.PROC, PROGRAM, I,
   DATA, ZIJOB.
   FID, ST205.
   USER(#AC=AC, #U=U, #PA=PA)
   RESOURCE(TL=100, WS=128, LP=1, PRI0=12)
   COMMENT.*******************

.*
READ, ZIJOB, ZIPROG.

.PROC, PROGRAM, I,
   DATA, ZIJOB.
   FID, ST205.
   USER(#AC=AC, #U=U, #PA=PA)
   RESOURCE(TL=100, WS=128, LP=1, PRI0=12)
   COMMENT.*******************

.*
READ, ZIJOB, ZIPROG.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. ------ ATT205, 2 ------

C * READ LIST OF FILE NAMES. OPEN AND CLOSE EACH FILE IN ORDER TO
C * SET THE DATE OF LAST ACCESS TO TODAY.
00 40 I=1,100
READ(I, (A), ERR=100, END=200)
CLOSE(I, STATUS='DELETE')
WRITE(20, 2) LINE(1:19), LINE(101:106), LINE(111:112), LINE(115:116),
A LINE(101:106), LINE(111:112), LINE(115:116)
ELSEIF(LINE(1:19).EQ.'CYBER 200 AUDIT')
GOTO 20
ELSEIF(LINE(19:32).EQ.'AUDIT COMPLETE')
GOTO 200
ELSE WRITE(20, 'A') LINE
ENDIF
40 CONTINUE
STOP '*** ERROR IN EXECUTION OF PROGRAM SAVE ***'

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.19.23. ------- BU205, 1 -------

* PROC, BU205=
U [USER IDENTIFICATION -] = (*N=XXU1XX, *A).
**HELP, NOLIST.
BU205 GIVES THE 205-BUDGET LEFT FOR U=... (USER IDENTIFICATION).
===== DEFAULT: U=XXU1XX.
**ENDHELP.
*** ROUTE, FILMPL, DC=IN, ST=205.
*** DATA, FILMPL.
XXU1XX, ST=205.
#USER(#AC=XXXACXXX, U=XXU1XX, PA=XPAX)
RESOURCE(TL=1Q, WS=128, LP=0)
COMMENT.**********************************************************************
COMMENT. PERKU5: GRANTS PERMISSION TO ACCESS U=XXU1XX FILE ON THE 205.
PERKU5, U=USER, #AC=AC.
**********************************************************************
***
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. --------- RIN205, 1 ---------

.PROC,RIN205*I,

* USER IDENTIFICATION - J = (*N=XXU1XX,*A).

* HELP,NOLIST,

RIN205 SHOWS THE QUEUES ON THE 205 FOR U= ... (USER IDENTIFICATION).

****** DEFAULT: U=XXU1XX.

.*ENDHELP

." HELP,,NOLIST.

RIN205 ROUTE "JOB" TO THE INPUT QUEUE OF THE 205,

****** WHERE FILE "IN1" MAY BE AN INPUT RECORD,

AND FILE "IN2" MAY BE A SECOND INPUT RECORD.

TID IS THE TERMINAL IDENTIFICATION (DEFAULT: TERMINAL),

FID IS THE FILE IDENTIFICATION (DEFAULT: XXU1XX).

* ENDHELP.

.* IF,..NOT.FILE(JOB,AS), LERROR.

NOTE,FILE JOB DOES NOT EXIST; TRY AGAIN.

REVERT,ABORT.

.* ENDIF, LERROR.

.* RETURN,FILMPL.

REWIND,JOB.

COPYOR,JOB,FILMPL.

.* IF,$IN1$,NE.$N$, LCOPY1.

REWIND,IN1.

COPYIN,IN1,FILMPL.

.* ENDIF, LCOPY1.

.* IF,$IN2$,NE.$N$, LCOPY2.

REWIND,IN2.

COPYOR,IN2,FILMPL.

.* ENDIF, LCOPY2.

.* IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN,ST=205,#FID=FID.

.* IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,ST=205,#TID=TID,#FID=FID.

REVERT.

.* EXIT,S.

NOTE,$ERROR$.

REVERT,ABORT.

--- MASTERFILE MFCCL CY=3u ---- 02/07/86 - 00.26.29. --------- PLT205, 1 ---------

.PROC,PLT205*I,

NAME [NAME OF THE ZUS PLOT FILE - J] = ( *F ) ,

* HELP,,NOLIST.

PLT205 CONVERTS THE BINARY PLOTFILE "NAME" FROM THE 205 TO A

****** GRAPHFILE TO BE VISUALIZED WITH GRIMAS.

.*ENDHELP.

.* ATTACH,ABAQUS,ID=PUBLIC.

LIBRARY,ABAQUS,CCLLIB.

PLT205,NAME.

RETURN,ABAQUS.

LIBRARY,CCLLIB.

REVERT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. --------- NNEW, 1 ---------

.PROC,NNEW*I,

FN [ROOT FILE NAME OF THE PROGRAM - J] = (*A)\n
S [IDENTIFICATION OF THE SOURCE - J] = (*N=*A),

M [MODIFICATION OF NEWPL? (N/Y) - J] = (*N=*A),

NUUL [NO UPDATE SOURCE LISTING? (N/Y) - J] = (*N=0,*K=1,N=0,Y=1,0,1),

NOCL [NO CATALOG NEWPL ON 750? (N/Y) - J] = (*N=0,*K=1,N=0,Y=1,0,1),

FLIST [FTN2U00 LISTING? (N/Y) - J] = (*N=0,*K=1,N=0,Y=1,0,1),

20

30

40

50

60

70

80

90

100

10

20

30

40

50

60

70

80

90

100

110

120

130

140

150

160

170

180

190

200

210

220

230

240

250

260

270

280

290

300

310

320

330

340

350

360

370

380

390

400

410

420

430

440

450

460

470

480

490

500

510

520

530

540

550

560

570

580

590

600

610

620

630

640

650

660

670

680

690

700

710

720

730

740

750

760

770

780

790

800

810

820

830

840

850

860

870

880

890

900

910

920

930

940

950

960

970

980

990

1000
NOR - NO ROUTE TO INPUT QUEUE? \((N/Y) - [N=O,Y=1] \) = \((N=0,Y=1)\).

HELP, NOLIST.

NNEW PRODUCES NEWPL UFN S FROM THE SOURCE FN S (OF MASTERFILE MFFN), OR UFN M FROM UFN S + MODIFICATION MFN M (ALSO OF MFFN), AND CREATES A JOB NN FN S (OR NN FN M) FROM THE FILE NNFN (OF MFFN) WHICH COMPiles LIBRARY FNLIB ON THE 205.

PARAMETERS:
- FN - NAME OF THE PROGRAM SERVING AS A ROOT FOR THE NAMES OF THE DERIVED FILES
- S - IDENTIFICATION OF SOURCE FILE FN S (OF MASTERFILE MFFN)
- M - IF SPECIFIED, NEWPL IS MODIFIED WITH MFN M (FROM MFFN)
- NOL - IF SPECIFIED, NO UPDATE LISTING OF THE SOURCE IS MADE
- NOCAT - IF SPECIFIED, NEWPL IS NOT CATALOGED ON THE 750
- FLIST - IF SPECIFIED, A FTN200 LISTING IS MADE
- NOR - IF SPECIFIED, JOB NN FN S IS CREATED BUT NOT SUBMITTED.

*. HELP.

IF, S$=S$, LERROR.

NOTE, ERROR: YOU FORGOT TO SPECIFY #S=S.

RETURN, EDSUB, ASK, ZZINP.

REVERT, ABRQT.

* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN NN FN S/M SINCE IT MAY BECOME TOO LONG.

DATA, ASK.

PROC, ASK1=*

\(MFN = (HG\#H, PPP=PP, HG=\#G), \#M = (N=0, S=S), \#M\)

* IF(ANSWER) ROUTE, UPDATE, OUTPUT, UC=PB, TID=XXB, FID=\#FN_\#M.

* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN NN FN S/M SINCE IT MAY BECOME TOO LONG.

* A REPLACES FILE NAME NN FN S/M SINCE IT MAY BECOME TOO LONG.

* DATA, EDSUB.

10= PROC, SUB*, 11= NN FN = (NNHG=NNHG, NNP=NNPP, NNHG=NNNP), 12= \#M = (N=0, S=S), \#M, \#M= (\#A), \#MFLIST = (\#A).

20= IF, \#MOR=0, LROUTE.

30= REWIND, NN FN \#M.

40= COPYGR, NN FN \#M, FILMPL.

50= COPYGR, C FN \#M, FILMPL.

60= ROUTE, FILMPL, DC=IN, ST=205.

70= ENDIF, LROUTE.

80= REVERT.

90= DATA, NN FN \#M.

1= ZZINP, 100
--- MASTERFILE MFCL CY=30 ---- 02/07/86 - 00.26.29. -------------- NNEW, 3 ------

W, SUB, O
SC, INIT
B, G

--- MASTERFILE MFCL CY=30 ---- 02/07/86 - 00.26.29. -------------- NN, 1 ------

.PROC, NN=I.
FN [ROOT FILE NAME OF THE PROGRAM] = (*A1)
U [IDENTIFICATION OF NEWPL? (N/.)] = (*N=N, *A)
B [IDENTIFICATION OF BINARY? (N/.)] = (*N=N, *A)
NOL [NO UPDATE SOURCE LISTING? (N/Y)] = (*N=0, *K=1, N=0, Y=1, 0, 1)
NOCAT [NO CATALOG NEWPL ON 750? (N/Y)] = (*N=0, *K=1, N=0, Y=1, 0, 1)
FLIST [CFNF200 LISTING? (N/Y)] = (*N=0, *K=1, N=0, Y=1, 0, 1)
NOR [NO ROUTE TO INPUT QUEUE? (N/Y)] = (*N=0, *K=1, N=0, Y=1, 0, 1).

*HELP, NOLIST.
NN Produces newpl ufn u from the source fn s (of masterfile mfn),
and creates a job nn s from the file nnfn (of masterfile mfn),
which compiles bfn u on the 205.

PARAMETERS:
FN - NAME OF THE PROGRAM SERVING AS A ROOT FOR THE NAMES OF
THE DERIVED FILES
S - IDENTIFICATION OF SOURCE FILE FN S (IN MASTERFILE MFN)
U - IDENTIFICATION OF NEWPL UFN U (ON 750); DEFAULT: U=S
B - IDENTIFICATION OF BINARY BFN B (ON 205); DEFAULT: B=S
NOL - IF SPECIFIED, NO UPDATE LISTING OF THE SOURCE IS MADE
NOCAT - IF SPECIFIED, NEWPL IS NOT CATALOGED ON THE 750
FLIST - IF SPECIFIED, A CFNF200 LISTING IS MADE
NOR - IF SPECIFIED, JOB NN S IS CREATED BUT NOT SUBMITTED.

.ENDHELP.

*.
IF, $S=$S, LERROR.
NOTE,ERROR: YOU FORGOT TO SPECIFY $S=S$.
RETURN, EDSUB, ASK; ZZINP.
REVERT, ABORT.
ENDIF, LERROR.

*.
MFUSE, MF_FN, ID=XX1DX.
FTAKE, ZZINP, ASK, ZZINP.
RETURN, OUTPUT.
REWIND, ZZINP.

IF, SUS, EQ, SNS, LNAME.
UPDATE, F, $I=ZZINP, N=U, FN S, C=C, FN S, L=A124, 0=OUT.
IF, $I=0$ CATALOG, UFN S, ID=XX1DX.
ELSE, LNAME.
UPDATE, F, $I=ZZINP, N=U, FN S, C=C, FN S, L=A124, 0=OUT.
IF, $I=0$ CATALOG, UFN S, ID=XX1DX.
ENDIF, LNAME.
USL, OUT, NOLIST=NOL.
ASK.
RETURN, OUT, ASK, ZZINP, ZZ.

*.
FILE (ZZZZZZZZ), ASKDOE.
ED USE, EDSUB.

IF, $US, EQ, SNS, AND, $DS, EQ, SNS$ SUB, $S=S, RU=S, BU=S, #FLIST=FLIST.$
IF, $US, NE, SNS, AND, $DS, EQ, SNS$ SUB, $S=S, RU=S, BU=S, #FLIST=FLIST.$
IF, $US, EQ, SNS, AND, $DS, NE, SNS$ SUB, $S=S, RU=S, BU=S, #FLIST=FLIST.$
IF, $US, NE, SNS, AND, $DS, NE, SNS$ SUB, $S=S, RU=S, BU=S, #FLIST=FLIST.$
IF, NOR=0, LROUTE.
REWIND, NN S.
COPYBR, NN S, FILMPL.

IF, $US, EQ, SNS$ COPYBR, C, FN S, FILMPL.
IF, $US, NE, SNS$ COPYBR, C, FN S, FILMPL.
ROUTE, FILMPL, DC=IN, ST=205.
ENDIF, LROUTE.
RETURN, ZZIN, EDLOG, EDSUB, SUB.
REVERT.

*.
EXIT, $S$.
NOTE, ERRORS.
RETURN, OUT, ASK, ZZINP, ZZ, ZZIN, EDLOG, EDSUB, SUB, FILMPL.
REVERT, ABORT.

*.
USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN NN_FN.
DATA, EDSUB.
10=.PROC, SUB, $S=S, RU=S, BU=B, #FLIST=FLIST.
20=REVERT.
30=DATA, NN S.
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ NN, 2 -----

L.43

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ RR, 1 -----

**PROC,RR=1,**

U [IDENTIFICATION OF UPDATE OLDPL - ] J = (*M=N,A), 30
M [IDENTIFICATION OF MODIFICATIONS - ] J = (*M=N,A), 40
V [IDENTIFICATION OF NEWPL? (N/...) - ] J = (*M=N,A), 50
B [IDENTIFICATION OF BINARY? (N/...) - ] J = (*M=N,A), 60
ULIST [UPDATE LISTING OF CHANGES? (N/Y) - ] J = (*M=0, *K=1, N=0, Y=1,0), 70
NOCAT [NO CATALOG NEWPL ON 750? (N/Y) - ] J = (*M=0, *K=1, N=U, Y=1,0), 80
FLIST [FIN200 LISTING? (N/Y) - ] J = (*M=0, *K=1, N=U, Y=1,0), 90
NOR [NO ROUTE TO INPUT QUEUE? (N/Y) - ] J = (*M=0, *K=1, N=U, Y=1,0), 100

**HELP,** NOLIST.

RR REVISES OLDPL UFN U WITH MODIFICATION DECK MFU M (OF MASTERFILE 150
== MFN) TO GET NEWPL UFN V AND CREATES A JOB RR M FROM THE FILE 160
RR M (OF MFN) WHICH COMPILES BFN B ON THE 205. 150
PARAMETERS:
FN - NAME OF THE PROGRAM SERVING AS A ROOT FOR THE NAMES OF 170
THE DERIVED FILES
U - IDENTIFICATION OF OLDPL UFN UFN U (ON 750) 180
M - IDENTIFICATION OF MODIFICATION DECK MFU M (FROM MFN) 190
V - IDENTIFICATION OF NEWPL UFN V (ON 750); DEFAULT: V=M 200
B - IDENTIFICATION OF BINARY BFN B (ON 205); DEFAULT: B=M 210
ULIST - IF SPECIFIED, UPDATE LISTING OF THE CHANGES IS MADE 220
NOCAT - IF SPECIFIED, NEWPL IS NOT CATALOGED ON THE 750 230
FLIST - IF SPECIFIED, A FIN200 LISTING IS MADE 240
NOR - IF SPECIFIED, JOB RR M IS CREATED BUT NOT SUBMITTED. 250

**ENDHELP.**

**IF, SUB=SNS OR SMS=SNS, LERROR.**

NOTE, LERROR: YOU FORGOT TO SPECIFY NU= OR NM=. 270
RETURN, EDSUB, ASK.
REVERT, ABORT.
**ENDIF, LERROR.**

**MFUSE, MF FN, ID=XXIDX.**

**FTAKE, ZIM=M/ZZF=RR FN.**

**ATTACH, UFN U, ID=XXIDX, MR=1.**

**RETURN, OUTPUT.**

**IF(SVS=SNS, LNAME).**


**IF(NOCAT=Q) CATALOG, U, FN M, ID=XXIDX.**

**ELSE, LNAME.**


**ENDIF, LNAME.**

**RETURN, OUT.**

ASK.

**IF(FILE(ZZZZZ1Z, AS)) ASKDOE.**

**ED, USE, EDSUB.**

**IF(SVS=SN, N) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N, E) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N, E) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(NOCAT=Q) CATALOG, U, FN M, ID=XXIDX.**

**RETURN, OUT.**

**ASK.**

**IF(FILE(ZZZZZ1Z, AS)) ASKDOE.**

**ED, USE, EDSUB.**

**IF(SVS=SN, N, E) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N, E) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N, E) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**IF(SVS=SN, N) SUB, N=U, MN=U, MV=U, MB=M, FLIST=FLIST.**

**RETURN, OUT.**

**ASK.**
* ENDHELP
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN XX_FN.
.*
.* XX CREATE A JOB XX_B_I FROM THE FILE XXFN OF MASTERFILE MFFN.
.* THIS JOB EXECUTES THE BINARY BFN_B WITH INPUT IFN_I.
.
. ** MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- XX, 1 -----
.
. PROG,XX*I,
.
. FN [ROOT FILE NAME OF THE PROGRAM - ] = (*A)\n10 20
. I [IDENTIFICATION OF THE INPUT - ] = (*N=N,*A), 40
. P [CATALOG PLOTFILE? (N/Y) - ] = (*N=0,*K=1,N=0,Y=1,0,1), 50
. O [CATALOG OUTPUT? (N/Y) - ] = (*N=0,*K=1,N=0,Y=1,0,1), 60
. D [CATALOG DATA FILE? (N/Y) - ] = (*N=0,*K=1,N=0,Y=1,0,1), 70
. TL [CHANGE DEFAULT ON TL? (N/Y) - ] = (*N=0,N=0,N=0,Y=1), 80
. WS [CHANGE DEFAULT ON WS? (N/Y) - ] = (*N=0,N=0,N=0,Y=1), 90
. DP [CHANGE DEFAULT ON LP? (N/Y) - ] = (*N=0,N=0,N=0,Y=1), 100
. NOR [NO ROUTE TO INPUT QUEUE? (N/Y) - ] = (*N=0,N=0,N=0,Y=1), 110
.
. ENDHELP
.*
.*
.* IF,SBS=N$ OR SBS=S$, LERROR.
.
. ** ERROR: YOU FORGOT TO SPECIFY N$ OR $I$.
.
. ENDIF, LERROR.
.*
. ** IF(FILE(ZZZZZZ Dự, ASKDOE.
.
. ** MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- RR, 2 -----
.
. RETURN,ZZI,ZZR,ASK,*U_FN,EDLOG,EDSUB,SUB.
.
. ENDIF, LERROR.
.*
. ** USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN RR_FN.
.*
.* AND ASKS FOR REPLACEMENT OF FILE NAME XX_B_I IF IT IS TOO LONG.
.
. DATA,EDSUB.
. **
. ** MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- RR, 2 -----
.
. RETURN,ZZI,ZZR,ASK,*U_FN,EDLOG,EDSUB,SUB.
.
. ENDIF, LERROR.
.*
. ** USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN RR_FN.
.*
.* AND ASKS FOR REPLACEMENT OF FILE NAME XX_B_I IF IT IS TOO LONG.
.
. DATA,EDSUB.
.
. **
```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. --------------- XX, 1 -----

.PROC,NU*I,
  S [SOURCE FILE - ] J = (*F)
  U [NEW UPDATE PL? (NEWPL/) - ] J = (*N=NEWPL,0,F),
  C [COMPILE FILE? (COMPILE/) - ] J = (*N=COMPILE,0,F),
  NOUL [NO UPDATE SOURCE LISTING? (Y/N) - ] J = (*N=0,*K=1,N=0,Y=1);,
  FID [FID FOR UPDATE OUTPUT? (XXIDX/) - ] J = (*N=XXIDX,0,A).

*HELP, NOLIST.
  NU PRODUCES NEWPL U AND COMPIL E FILE C FROM THE SOURCE S.
  = PARAMETERS:
    S - SOURCE FILE
    U - NEW UPDATE PROGRAM LIBRARY; DEFAULT: "NEWPL"
    C - COMPIL E FILE; DEFAULT: "COMPILE"; SUPPRESS: "C=0"
    NOUL - IF SPECIFIED, NO UPDATE LISTING OF THE SOURCE IS MADE
    FID - FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".

*ENDHELP.

*IF, NOT.FILE(S,AS), LERROR.
  NOTE,$FILE S DOES NOT EXIST; TRY AGAIN.
  RETURN, ASK, ZZINP.
  REVERT, ABDORT.
  *ENDIF, LERROR.

*REWIND, S.
  RETURN, OUTFUT.
*IF(5CS,NE.105) RETURN, C.
  REMIND, ZZINP.
  UPDATE,F,=ZZINP,N=U, #C = C, L = A125, O = OUT.
  USL, OUT, NOLIST = NOUL.
  RETURN, OUT.
  ASK.
  RETURN, ASK, ZZINP.
  REVERT.
  *EXIT, WS.
  NOTE,$ERRORS.
  RETURN, ASK, ZZINP.
  REVERT, ABDORT.
  *DATA, ASK.
  *PROC, ASK1,
  ANSWER OK TO ROUTE UPDATE OUTPUT? (Y/N) - J = (Y=T,N=F).
  IF(ANSWER) ROUTE, OUTPUT, DC=PR,TID=XXB, #FID=FID.
  REVERT.
  *DATA, ZZINP.
  *LIMIT 10000
  *READ S

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. --------------- NU, 1 -----

.PROC, RU*1,
  U [OLD UPDATE PL - ] J = (*F)
  M [MODIFICATION DECK - ] J = (0,*F),
  V [NEW UPDATE PL? (NEWPL/) - ] J = (*N=NEWPL,0,F),
  C [COMPIL E FILE? (COMPILE/) - ] J = (*N=COMPILE,0,F),
  ULIST, SOURCE LISTING OF CHANGES? (Y/N) - J = (*N=0,*K=1,N=0,Y=1);,
  FID [FID FOR UPDATE OUTPUT? (XXIDX/) - ] J = (*N=XXIDX,0,A).
```

MASTERFILE MFCCL CY=30  02/07/86  00.26.29.  02/07/86  00.26.29.

HELP, NOLIST.
RU  REVISIONS OLDPL U WITH MODIFICATION DECK M TO PRODUCE NEWPL V AND
==  COMPARE FILE C.  PARAMETERS:
  U  -  OLD UPDATE PROGRAM LIBRARY
  M  -  MODIFICATION DECK; SUPPRESS: "M=O" (ONLY COMPARE OLDPL)
  V  -  NEW UPDATE PL; DEFAULT: "NEWPL"; SUPPRESS: "V=O"
  C  -  COMPARE FILE; DEFAULT: "COMPARE"; SUPPRESS: "C=O"

LIST  IF SPECIFIED, UPDATE LISTING OF THE CHANGES IS MADE
FID  -  FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".

ENDHELP.

*  IF, $M$, NE, $0$, LM, (, NOT, FILE(M, AS)), OR, (. NOT, (FILE(M, AS))), LERROR1.
NOTE, $FILE M$ DOES NOT EXIST; TRY AGAIN.
RETURN, ASK, REVERT, ABORT.
ENDIF, LERROR1.
ELSE, LM.
ELSE, $FILE U$ DOES NOT EXIST; TRY AGAIN.
RETURN, ASK, REVERT, ABORT.
ENDIF, LERROR2.
ELSE, LM.
REWIND, U.
RETURN, OUTPUT.
IF($C$, NE, $0$) RETURN, C.
REWIND, M.
REWIND, V.
UPDATE, F, P=U, I=M, N=V, #C=C, L=1234, O=OUT.
ENDIF, LABELV.
ELSE, LABELM.
UPDATE, F, P=U, I=M, #C=C, L=1234, O=OUT.
ENDIF, LABELM.
RETURN, INPUT.
UPDATE, F, P=U, #C=C, L=1234, O=OUT.
RETURN, INPUT.
ENDIF, LABELM.
UC, OUT, LIST=ULIST.
ASK.  RETURN, OUT, ASK.  REVERT.
*  EXIT, S.
NOTE, ERRORS.
RETURN, OUT, ASK.
REVERT, ABORT.
*  DATA, ASK.
PROC, ASK*1,
ANSWER [OK TO ROUTE UPDATE OUTPUT? (Y/N)] - = (Y=T, N=F).
IF(ANSWER) ROUTE, OUTPUT, DC=PR, TID=XXB#, #FID=FID.
REVERT.

--- MASTERFILE MFCCL CY=30  02/07/86  00.26.29.  02/07/86  00.26.29.

PROC, SU1.
U  [OLD UPDATE PL - ] = (*F)\ 
S  [SOURCE FILE - ] =(*F)\ 
NOUL [NO UPDATE LISTING? (N/Y) - ] = (*N=0, *K=1, H=U, Y=1, O, T),
*  HELP, NOLIST.
SU  RETRIEVE SOURCE S FROM AN OLD UPDATE PL U.
==  PARAMETERS:
  U  -  OLD UPDATE PROGRAM LIBRARY
  S  -  SOURCE FILE
NOUL  -  IF SPECIFIED, NO UPDATE LISTING IS MADE
FID  -  FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".
ENDHELP.
*  IF, NOT, FILE(U, AS), LERROR.
NOTE, $FILE U$ DOES NOT EXIST; TRY AGAIN.
RETURN, ASK1, ASK2.
MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- SU, 2 ------

**REVERT, ABORT.**

*ENDIF, LERROR.

**REWIND, U.

**IF(FILE(S, AS)) ASK1.
RETURN, OUTPUT.
UPDATE, F, P=U, S=S, C=Q, L=7, O=OUT.
**IF, NOUL=0, LIST.
USL, OUT, NO LIST=NOUL.
ASK2.
**ENDIF, LIST.
STRIP, S.
REVERT, INPUT, OUT, ASK1, ASK2.
REVERT.

**EXIT, WS.

**NOTE, ERRORS.
RETURN, INPUT, ASK1, ASK2.
REVERT, ABORT.

**DATA, ASK1.
**PROC, ASK1*1,
ANSWER (OK TO RETURN EXISTING FILE S? (Y/N) = (Y=T, N=F)).
40
**IF(.NOT., ANSWER) REVERT, ABORT.
RETURN, S.
REVERT.

**DATA, ASK2.
**PROC, ASK2*1,
ANSWER (OK TO ROUTE UPDATE OUTPUT? (Y/N) = (Y=T, N=F)).
470
**IF(ANSWER) ROUTE, OUTPUT, DC=PR, TID=XXB, #FID=FID.
REVERT.

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- INSTALL, 1 -------

**PROC, INSTALL*1;
LIB (LFN LIBRARY? (N=LIBRARY/..) = (N=LIBRARY, N=LIBRARY, #F))
PROFILE [NO OPTION, JUST TYPE "N" = (N=FILE, N=FILE).

**HELP, NOLIST.
INSTALL IS A PROCEDURE TO INSTALL A LIBRARY FOR UPDATE PROCEDURES AND
PROGRAMS WRITTEN BY JOS KOOT AND AMPLIFIED BY HANS GOEDBUDEL.
IT CONTAINS THE CCL PROCEDURES REVISE/ASKDOE AND THE FORTRAN
PROGRAMS ASKDECK/MAKEFIL/MODGEN/USL/UML.
PARAMETERS:
LIB - LFN OF THE LIBRARY; DEFAULT: "LIBRARY"
PROFILE - INSTRUCTS INSTALL TO READ FILES TO BE INSTALLED FROM
THE FILE INSTALL ITSELF. NO OPTIONS FOR THE USER.

**ENDHELP.

**RETURN, LIB, ZILST, ZILGO, ZZPRC1, ZZPRC2.
LIBRARY.
COPYBR, PROFILE, ZZPRC1.
COPYBR, PROFILE, ZZPRC2.
FNS, I=PROFILE, B=ZILGO, L=ZILST, E=ZILST, PL=10000.
EDITLIB, I=ZILNP, L=ZILST.
LIBRARY, LIB.
RETURN, ZILNP, ZILST, ZILGO, ZZPRC1, ZZPRC2.
REVERT.

**DATA, ZILNP.
LIBRARY(LIB, NEW)
ADD(*, ZZPRC1)
ADD(*, ZZPRC2)
ADD(*, ZILGO)
SETAL(ASKDECK, 1)
SETAL(MAKEFIL, 1)
SETAL(MODGEN, 1)
SETAL(USL, 1)
SETAL(USR, 1)
FINISH.
ENDRUN.

**BELOW,
**AFTER 1ST EOR: PROCEDURE REVISE.
**AFTER 2ND EOR: PROCEDURE ASKDOE.
**AFTER 3RD EOR: FORTRAN PROGRAMS ASKDECK/MAKEFIL/MODGEN/USL/UML

*(NOT SEPARATED BY COMMENT LINES OR EOR'S!).*
--- MASTERFILE FMCCL CY=30 ---- 02/07/86 - 00.27.57. --- REVISE, 1 ---

.L48.

**.PROC,REVISION=I,**

U IFN OF OLDPL - J = (*F)\nD [DECK NAME(S)? (N=S,ALL,$/Y=S,ASK,$,...) - ] = \n   (N=S,ALL,$,K=S,ASK,$,N=S,ALL,$,Y=S,ASK,$,A),
M [MODIFICATION DECK? (N=MODFILE/...)] = (N=S,MODFILE,N=MODFILE,F),
CI [CORRECTION IDENTIFIER? (N=MOD/...)] = (N=S,MOD,N=MOD,A),
S OLD SOURCE? (N=OLDSRC/... - ] = (N=S,OLDSRC,N=S,OLDSRC,F),
T NEW SOURCE? (N=NEWSRC/... - ] = (N=S,NEWSRC,N=S,NEWSRC,F),
V [COMPILE FILE? (N=COMPILE/... - ] = (N=S,NEWPL,N=S,NEWPL,F),
UL1 [1ST UPDATE LISTING? (N=Y,1/2) - ] = (N=S,MOD=1,N=Y,1,0,1,2),
*HELP,NOList.

**REVIEW**

REVIEW RETRIEVES ONE OR MORE DECKS FROM AN UPDATE LIBRARY U.

---

THE RETRIEVED DECKS D ARE PUT INTO AN EDITFILE S AND THE
EDITOR IS CALLED.  AFTER THE USER HAS FINISHED EDITING, THE
NEW VERSION T IS COMPARSED WITH THE OLD ONE AND A MODIFICATION
DECK M IS MADE.  IF WANTED, THIS DECK IS PRESENTED TO UPDATE
WHICH PRODUCES A NEW PROGRAM LIBRARY V AND A COMPILE FILE C.

**PARAMETERS:**

U = LFN OF THE OLDPL
D = DECKS TO BE UPDATED; DEFAULTS: $S,ALL,$/S,ASK,$
M = MODIFICATION DECK; DEFAULT: "MODFILE"
CI = CORRECTION IDENTIFIER; DEFAULT: "MOD"
S = OLD SOURCE DERIVED FROM OLDSRC; DEFAULTS: U/OLDSRC
T = NEW SOURCE AFTER EDITING; DEFAULTS: O/NEWSRC
V = NEWPL; DEFAULTS: O/NEWPL
C = COMPILE FILE; DEFAULTS: 0/COMPILE
UL1 = 1ST UPDATE LISTING IN FILE LFN; DEFAULT: 0
UL2 = 2ND UPDATE LISTING IN FILE LFN; DEFAULT: 0/1.

**HELP,U,NOLIST.**

U = THE LOGICAL FILE NAME OF THE OLD PROGRAM LIBRARY.

**HELP,D,NOLIST.**

D = THE NAME OF THE DECK(S) AND COMMON DECK(S) TO BE UPDATED.
OMITTED - ALL DECKS AND COMMON DECKS
D=S,ALL,$ - ALL DECKS AND COMMON DECKS
D=S,ASK,$ - DECKNAMES WILL BE ASKED FOR VIA TERMINAL
D=S,ASK,$/S,ASK,$ - DECKNAMES WILL BE ASKED VIA TERMINAL
D=DECKNAME - THE SPECIFIED DECK.

**HELP,M,NOLIST.**

M = THE LOGICAL FILE NAME OF THE MODIFICATION DECK.
OMITTED - THE MODIFICATION DECK IS WRITTEN ON "MODFILE"
M=LFN - THE MODIFICATION DECK IS WRITTEN ON FILE LFN

**HELP,CI,NOLIST.**

CI = THE CORRECTION SET IDENTIFIER. IT MUST NOT YET EXIST IN THE
OLD PROGRAM LIBRARY.
OMITTED - CORRECTION SET IDENTIFIER IS "MOD". UNLESS M HAS
BEEN SPECIFIED TO BE DIFFERENT FROM "MODFILE".
490
480
500
510

**HELP,S,NOLIST.**

S = THE LOGICAL FILE NAME OF THE OLD SOURCE DERIVED FROM OLDSRC.
OMITTED - OLD SOURCE IS RETURNED
S=OLDSRC - OLD SOURCE IS RETURNED
S=LFN - OLD SOURCE ON FILE OLDSRC

**HELP,T,NOLIST.**

T = THE LOGICAL FILE NAME OF THE NEW SOURCE OBTAINED AFTER
EDITING THE OLD SOURCE.
OMITTED - NEW SOURCE IS RETURNED
T=NEWSRC - NEW SOURCE IS RETURNED
T=LFN - NEW SOURCE ON FILE LFN

**HELP,V,NOLIST.**

V = THE LOGICAL FILE NAME OF THE NEW PROGRAM LIBRARY.
OMITTED - NO NEW LIBRARY PRODUCED
V=NEWLIBRARY - NEW LIBRARY PRODUCED
V=NEWLIBRARY - NEW LIBRARY ON FILE NEWPL
V=LFN - NEW LIBRARY ON FILE LFN

**HELP,C,NOLIST.**

C = THE LOGICAL FILENAME OF THE COMPILE FILE THAT IS TO RECEIVE
ALL MODIFIED CODE.
OMITTED - NO COMPILE FILE GENERATED
C=COMPILEFILE - COMPILE FILE GENERATED
C=LFN - COMPILE FILE GENERATED ON FILE LFN.

**HELP,UL1,NOLIST.**

UL1 INDICATES THAT THE LISTING OF THE 1ST UPDATE RUN (THE ONE THAT
RETRIEVES THE SPECIFIED DECKS FROM OLDSRC) IS TO BE KEPT AND
-L.49-

-- MASTERFILE MFCCL CY=30 -- 02/07/86 - 00.27.57. ------ REVISE, 2 ------

REFORMATTED BY MEANS OF PROGRAM USL.
OMITTED - UPDATE OUTPUT RETURNED
UL1 - UPDATE SOURCE LISTING OF OLDPL (CORRESPONDING TO FILE 3) WRITTEN ON "OUTPUT"
HELP,UL2,HOLIST.
UL2 INDICATES THAT THE LISTING OF THE 2ND UPDATE RUN (THE ONE THAT PRODUCES A NEWPL AND COMPIL) IS TO BE KEPT AND REFORMATTED BY MEANS OF PROGRAM UML.
OMITTED - UPDATE OUTPUT RETURNED
UL2=1 - LISTING OF THE MODIFICATION DECK WRITTEN ON "OUTPUT"
UL2=2 - LISTING OF THE MODIFICATION DECK + UPDATE LISTING OF ALL THE CHANGES PRODUCED WRITTEN ON "OUTPUT".

ENDHELP

** TEST ON PRESENCE OF OLDPL.
** IF NOT FILE(U,AS), LERROR.
COMMENT,** FILE U DOES NOT EXIST; TRY AGAIN **
RETURN,ZZUSE,ZZINP,ASKPRM1,ASKPRM2,ASKPRM3,ASKPRM4,ASKPRM5.
REVERT,ABORT.

** TEST ON CONFLICTING FILENAMES FOR OLDPL AND NEWPL.
** IF NEWPL IS TO BE WRITTEN AND FILE ALREADY EXISTS,
** ASK PERMISSION TO RETURN IT.
** IF,FILE(T,AS) ASKPRM2.

** IF OLD SOURCE IS TO BE WRITTEN AND FILE ALREADY EXISTS,
** ASK PERMISSION TO RETURN IT.
** IF,FILE(S,AS)) ASKPRM1.

** IF NEW SOURCE IS TO BE WRITTEN AND FILE ALREADY EXISTS, **ASK PERMISSION TO RETURN IT.
** IF,FILE(V,AS)) ASKPRM3.

** IF SOURCE IS TO BE WRITTEN AND FILE ALREADY EXISTS, ASK PERMISSION TO RETURN IT.
** IF,FILE(W,AS)) ASKPRM4.

** IF DECKNAMES ARE TO BE COMMUNICATED VIA THE TERMINAL, DO SO BY MEANS OF PROGRAM DECKNAME WHICH ASKS FOR DECKNAMES AND REPLACES FILE ZZINP.

** TRAP IN CASE A PERMISSION WAS NOT GRANTED.
** IF,FILE(ZZZZZ1Z,AS)) ASKDOE.

** CLEAN START.
** IF FILE(ZZNEW,M,OUTPUT,ASKPRM1,ASKPRM2,ASKPRM3,ASKPRM4).

** IF DECKNAMES ARE TO BE COMMUNICATED VIA THE TERMINAL, DO SO BY MEANS OF PROGRAM ASKDECK WHICH ASKS FOR DECKNAMES AND REPLACES FILE ZZINP.
** IF,FILE(ZZNEW,M,OUTPUT,ASKPRM1,ASKPRM2,ASKPRM3,ASKPRM4,ASKDECK).

** 1ST UPDATE RUN: RETRIEVE THE REQUESTED DECKS FROM OLDPL.
* CHECK LEGALITY OF CORRECTION SET IDENTIFIER.
REWIND, U.
.IF,$DS.EQ.$,ALL.$, FULLYN.
RETURN, INPUT.
.ELSE, FULLYN.
REWIND, ZINP.
.ENDIF, FULLYN.
RETURN, INPUT.
SKIP, EXIT.
EXIT,#S.
* PROGRAM MAKEFIL READS THE OUTPUT FILE ZZUL1 OF UPDATE AND PRODUCES
* TWO FILES, ZZOLD AND ZZNEW. ZZOLD CONTAINS LINE IMAGES AND SEQUENCE
* INFORMATION. ZZNEW CONTAINS LINE IMAGES ONLY.
REWIND, ZZUL1.
.IF,$DS.EQ.$,ALL.$, FULLYN.
MAKEFIL, ZZUL1, ZZOLD, ZZNEW.
.ELSE, FULLYN.
MAKEFIL, ZZUL1, ZZOLD, ZZNEW, ZINP.
.ENDIF, FULLYN.
.IF(UL1=1) USL, ZZUL1.
RETURN, ZZUL1, ZINP.
.IF,$SS$.NE.$0$, LS.
REWIND, ZZNEW.
COPYBR, ZZNEW, S.
REWIND, S.
.ENDIF, LS.
* PLACE ZZNEW IN THE EDITFILE AND HAVE THE USER EDIT HIS DECKS.
REWIND, ZZOLD, ZZNEW.
ED, USE, ZZUSE.
RETURN, ZZUSE.
ENDIF, EXIT.
* UPON RETURN FROM THE EDIT SESSION, ASK PERMISSION TO PROCEED
* AND REPLACE ZZNEW BY EDITED VERSION.
ASKPRMS.
SKIP, EXIT.
ENDIF, EXIT.
* MODGEN COMPARES ZZNEW AND ZZOLD. A CORRECTION SET IS WRITTEN
* ON MODIFICATION FILE M.
REWIND, ZZOLD, ZZNEW.
MODGEN, ZZOLD, ZZNEW, M, CI.
**RETURN,ZZOLD,ZZNEW,EDLOG.**  
*  
**2ND UPDATE RUN: PRESENT THE CORRECTIONS TO UPDATE, CREATING NEWPL**  
*  
**AND/OR COMPIL.**  
  
**REWIND,M.**  
  
**IF,(SVS,NE.,S$).OR.(S$S,NE.,S$S), VORC.**  
  
**IF,SVS,NE.,S$S, VYN.**  
  
**IF,$S$,EQ.,S,ALL.,$S$, FULLYN1.**  
  
**UPDATE,$F,P=U,1=M,N=V,#C=C,L=A1234,0=ZZUL2.**  
  
**ELSE, FULLYN1.**  
  
**UPDATE,$P=U,1=M,N=V,#C=C,L=A1234,0=ZZUL2.**  
  
**ENDIF, FULLYN1.**  
  
**ELSE, VYN.**  
  
**IF,$S$,EQ.,S,ALL.,$S$, FULLYN2.**  
  
**UPDATE,$P=U,1=M,N=V,#C=C,L=A1234,0=ZZUL2.**  
  
**ELSE, FULLYN2.**  
  
**UPDATE,$P=U,1=M,#C=C,L=A1234,0=ZZUL2.**  
  
**ENDIF, FULLYN2.**  
  
**ELSE, VYN.**  
  
**IF(UL2=1) UML,ZZUL2,LIST=0.**  
  
**IF(UL2=2) UML,ZZUL2,LIST=1.**  
  
**RETURN,ZZUL2.**  
  
**ENDIF, VORC.**  
  
**REVERT.**  
  
**EXIT,WS.**  
  
**COMMENT,** EXIT(7) IN PROCEDURE REVISE **  
  
**RETURN,ELOG,ZZOLD,ZZNEW,ZZUL2.**  
  
**REVERT,ABORT.**  
  
**EDITOR USEFILE (PREPARE EDITFILE).**  
  
**DATA,ZZUSE.**  
  
**SET,COUNT=1,LINES=19,EXP=1,PROMPT=##**  
  
**FORMAT,NO E,ZZNEW**  
  
**RETURN,** EXIT (?) IN PROCEDURE REVISE **  
  
**RETURN,EDLOG,ZZOLD,ZZNEW,ZZULZ.**  
  
**REVERT,** EXIT.**  
  
**INPUT FOR FIRST UPDATE RUN (RETRIEVAL OF DECKS).**  
  
**DATA,ZZINP.**  
  
**IDENT DUMMY,#U=DUMMY**  
  
**COMPILE O**  
  
**DATA,ASKPRM1.**  
  
**PROC,ASKPRM1*I,**  
  
**ANSWER EOK TO DISCARD EXISTING S? (Y/N) -] = (Y=HT,N=F).**  
  
**RETURN,ASKPRM1.**  
  
**IF(.NOT.ANSWER) REVERT,ABORT.**  
  
**RETURN,S.**  
  
**REVERT.**  
  
**DATA,ASKPRM2.**  
  
**PROC,ASKPRM2*I,**  
  
**ANSWER EOK TO DISCARD EXISTING T? (Y/N) -] = (Y=HT,N=F).**  
  
**RETURN,ASKPRM2.**  
  
**IF(.NOT.ANSWER) REVERT,ABORT.**  
  
**RETURN,T.**  
  
**REVERT.**  
  
**DATA,ASKPRM3.**  
  
**PROC,ASKPRM3*I,**  
  
**ANSWER EOK TO DISCARD EXISTING V? (Y/N) -] = (Y=HT,N=F).**  
  
**RETURN,ASKPRM3.**  
  
**IF(.NOT.ANSWER) REVERT,ABORT.**  
  
**RETURN,V.**  
  
**REVERT.**  
  
**DATA,ASKPRM4.**  
  
**PROC,ASKPRM4*I,**  
  
**ANSWER EOK TO DISCARD EXISTING C? (Y/N) -] = (Y=HT,N=F).**  
  
**RETURN,ASKPRM4.**  
  
**IF(.NOT.ANSWER) REVERT,ABORT.**  
  
**RETURN,C.**  
  
**REVERT.**  
  
**DATA,ASKPRM5.**  
  
**PROC,ASKPRM5*I,**  
  
**ANSWER EOK TO PROCEED UPDATING? (Y/N) -] = (Y=HT,N=F).**  
  
**RETURN,ASKPRM5.**  
  
**IF(.NOT.ANSWER) REVERT,ABORT.**  
  
**REVERT.**
**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. ------- ASKDOE, 1 ------**

```
.PROC,ASKDOE=I,
  ANSWER [OK TO DELETE OLD EDITFILE ? (Y/N) -3 = (Y=T,N=F)].
  .HELP,,NOLIST.
  ASKDOE ASKS FOR PERMISSION TO DELETE THE OLD EDITFILE. IT SHOULD 
  BE CALLED BEFORE ENTERING ED IN ORDER TO PREVENT MIXING WITH THE 
  CONTENTS OF AN EXISTING EDITFILE.
  .ENDHELP.
  IF(STATE,HELP,.PROC,ASKDOE*l, 10 
  ANSWER [OK TO DELETE OLD EDITFILE? (Y/N)] = (Y=T,N=F).
  20 
  HELP,,NOLIST.
  MAKEFIL ASKS FOR PERMISSION TO DELETE THE OLD EDITFILE. IT SHOULD 
  BE CALLED BEFORE ENTERING ED IN ORDER TO PREVENT MIXING WITH THE 
  CONTENTS OF AN EXISTING EDITFILE.
  .ENDHELP.
  IF(STATE,HELP,.PROC,MAKEFIL*1, 10 
  ANSWER [OK TO DELETE OLD EDITFILE? (Y/N)] = (Y=T,N=F).
  20 
  HELP,,NOLIST.
```

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. ------- ASKDOE, 1 ------**

```
**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. ------- ASKDOE, 1 ------**

C PROGRAM ASKDECK
C ******************************************************************
C * PREPARE INPUT INPFILE FOR UPDATE RUN WHICH RETRIEVES SPECIFIED * 
C * DECKS FROM OLDPL. THE DECKNAMES ARE OBTAINED INTERACTIVELY. * 
C ******************************************************************
C IMPLICIT INTEGER(A-Z)
C CHARACTER * 7 INPFILE
C CHARACTER *60 LINE
C CALL GETPARM(INPFILE,DUMSTR,PARSTAT)
C OPEN(UNIT=1,FILE=INPFILE)
C OPEN(UNIT=2,FILE= ' ZZTMI 1 )
C OPEN(UNIT=3,FILE= ' ZZTMO')
C CALL CONNEC(2)
C CALL CONNEC(3)
C REWIND(1)
C WRITE(1,'(A)') *IDENT DUMMY, U=DUMMY'
C WRITE(3,*) 'ENTER ONE OR MORE DECKNAMES IN THE FOLLOWING FORMAT:'
C WRITE(3,*) 'DECKNAME (ONE SINGLE DECK)'
C WRITE(3,*) 'DECK1,DECK2,... (SEVERAL DECKS)'
C WRITE(3,*) 'FIRSTDECK, LASTDECK (FOR A RANGE)' 
C READ(2,'(A)') LINE
C IF (LINE.NE. ' ') GOTO 10
C WRITE(1, '(A,A)') *COMPILE ',LINE'
C WRITE(3,*) 'ENTER MORE DECKNAMES OR A SPACE (FOR TERMINATION) -'
C READ(2,'(A)') LINE
C IF (LINE.NE. ' ') GOTO 10
C END
```

**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. ------- MAKEFIL, 1 ------**

```
**MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. ------- MAKEFIL, 1 ------**

C PROGRAM MAKEFIL
C ******************************************************************
C * READ OUTPUT LSTFILE OF AN UPDATE RUN (WITH LIST OPTION L=7, * 
C * AND HAVING RETRIEVED SPECIFIED DECKS FROM OLDPL) AND PRODUCE * 
C * TWO FILES: WITHSEQ AND WITHOUT. WITHSEQ CONTAINS LINE IMAGES * 
C * AND SEQUENCE INFORMATION. WITHOUT CONTAINS LINE IMAGES ONLY. * 
C * UNWANTED COMMON DECKS OBTAINED BY SELECTIVE UPDATE MODE ARE * 
C * REMOVED BY COMPARING THE CONTENTS OF LSTFILE WITH THE DECK * 
C * LIST REQUESTED ON THE UPDATE INPUT FILE INPFILE. * 
C ******************************************************************
C IMPLICIT INTEGER(A-Z)
C PARAMETER(MXNRDEK=100)
C CHARACTER *132 LSTLINE
C CHARACTER *10 CHCKONE
C CHARACTER *72 IMAGL, INPLINE
C CHARACTER *9 CHCKTWO
C CHARACTER *9 SEQID
C CHARACTER *7 SEQNR
C CHARACTER *25 CHCKTH
C CHARACTER *1 CHARRAY(132)
C EQUIVALENCE(CHARRAY(1),LSTLINE)
C EQUIVALENCE(CHARRAY(1),CHCKONE)
```
EQUIVALENCE(CHARARRAY( 11), IMAGE )
EQUIVALENCE(CHARARRAY( 83), CHCKTWO)
EQUIVALENCE(CHARARRAY( 92), SEWID )
EQUIVALENCE(CHARARRAY(101), SEUNR )
CHARACTER * 9 LSTFILE, WITHSEQ, WITHOUT, INPFILE
CHARACTER * 9 DEKNAME
CHARACTER *10 MOLDONE(MXNRDEK)
CHARACTER * 7 MOLDTWO
CHARACTER *24 MOLDTHR
CHARACTER *40 DUMSTR
PARAMETER(CLSTUNIT=1, WITHUNT=2, WOUTUNT=3, INPUNIT=4, DUMUN1T=5)
PARAMETER(MOLDTWO=' 1 ')
PARAMETER(MOLDTHR= 1 ' A ')
LOGICAL FULLUP
CALL GETPARM(LSTFILE, DUMSTR, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(WITHSEQ, DUMSTR, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(WITHOUT, DUMSTR, PARSTAT)
IF(PARSTAT.LT.0) THEN
FULLUP=.TRUE.
GOTO 10
ELSE
FULLUP=.FALSE.
GOTO 10
ENDIF
END IF
ENDIF
CALL ERRMSG(1)
10 OPEN(UNIT=LSTUNIT, FILE=LSTFILE, STATUS='OLD', ERR=12)
OPEN(UNIT=WOUTUNIT, FILE=WOUTFILE, ERR=13)
OPEN(UNIT=INPUNIT, FILE=INPFIL1E, STATUS='OLD', ERR=15)
GOTO 20
12 CALL ERRMSG(2)
13 CALL ERRMSG(3)
14 CALL ERRMSG(4)
15 CALL ERRMSG(5)
20 REWIND(LSTUNIT)
REWIND(WOUTUNIT)
REWIND(INPUNIT)
C
IF(.NOT.FULLUP) THEN
REWIND(INPUNIT)
READ(INPUNIT,'(A)') INPLINE
IF(INPLINE(1:7).NE. 'IDENT') CALL ERRMSG(6)
NROFDEK=0
30 READ(INPUNIT,'(A), END=50') INPLINE
IF(INPLINE(1:9).EQ. 'COMPILE ') THEN
L=10
K=INDEX(INPLINE(L:L+9),',')-1
IF(K.NE.-1) THEN
L=K+1
NROFDEK=NROFDEK+1
MOLDONE(NROFDEK)= ' //INPLINE(L:L+K-1)
IF(NROFDEK.GE.MXNRDEK) GOTO 50
L=L+1
GOTO 40
ELSE
J=INDEX(INPLINE(L:L+9),',')-1
IF(J.NE.-1) THEN
NROFDEK=NROFDEK+1
MOLDONE(NROFDEK)= ' //INPLINE(L:L+J-1)
IF(NROFDEK.GE.MXNRDEK) GOTO 50
L=L+J+1
GOTO 40
ELSE
I=INDEX(INPLINE(L:L+9),',')-1
IF(I.NE.-1) THEN
NROFDEK=NROFDEK+1
MOLDONE(NROFDEK)= ' //INPLINE(L:L+I-1)
IF(NROFDEK.GE.MXNRDEK) GOTO 50
GOTO 50
ENDIF
ENDIF
ENDIF
CALL ERRMSG(7)
700
ELSE
CALL ERRMSG(7)
ENDIF
ENDIF
ELSE
CALL ERRMSG(8)
ENDIF
50
CLOSE(INPUNIT)
ENDIF

C
READ(LSTUNIT,'(A)',END=51) LSTLINE
GOTO 52
51
CALL ERRMSG(9)
52
IF(LSTLINE(1:16),EQ,'UNLABELED OLDPL') THEN
IF(.NOT.FULLUP) THEN
OPEN(UNIT=DUMUNIT,FILE='DUMFILE',ERR=53)
GOTO 54
ENDIF
ELSE
CALL ERRMSG(10)
ENDIF
54
WRITE(DUMUNIT)
WRITE(DUMUNIT,'(A)') LSTLINE
ENDIF
ELSE
CALL ERRMSG(11)
ENDIF
NLINE=0
60
READ(LSTUNIT,'(A)',END=100) LSTLINE
IF(CHCKTWO.EQ.MOLDTWO.AND.CHCKTHR.EQ,MOLDTHR) THEN
IF(.NOT.FULLUP) AND (.RANGE.NE.2) THEN
DO 70 IDEK=1,NROFDEK
IF(CHCKONE.EQ.MOLDONE(IDEK)) THEN
IF(RANGE.EQ,1)
RANGE=2
GOTO 80
ENDIF
70
CONTINUE
GOTO 60
ENDIF
80
I=INDEX(SEQID,' ')
IF(I.GT.1.AND.I.LE.10) THEN
READ(SEQNR,'(BN,17)') NUM
WRITE(WITHUNT,'(A,1X,A,11•11,I7.7)') IMAGE,SEQID(1:I-1),NUM
ENDIF
ELSE
WRITE(WITHUNT,'(A)') IMAGE
ENDIF
IF(.NOT.FULLUP) WRITE(DUMUNIT,'(A)') LSTLINE
NLINE=NLINE+1
ENDIF
100
IF(.NOT.FULLUP) THEN
REWIND(DUMUNIT)
REWIND(LSTUNIT)
DO 110 I=1,10000
READ(DUMUNIT,'(A)',END=115) LSTLINE
WRITE(LSTUNIT,'(A)') LSTLINE
CONTINUE
110
CLOSE(DUMUNIT,STATUS='DELETE')
ENDIF
IF(NLINE.GT.0) THEN
CLOSE(LSTUNIT)
CLOSE(WITHUNT,STATUS='KEEP')
CLOSE(WOUTUNT,STATUS='KEEP')
ELSE
CLOSE(LSTUNIT,STATUS='DELETE')
CLOSE(WITHUNT,STATUS='DELETE')
CLOSE(WOUTUNT,STATUS='DELETE')
ENDIF
ENDIF
END

C
SUBROUTINE ERRMSG(1)
CHARACTER *40 MSG
WRITE(MSG,'(** PROGRAM MAKEFIL - ERROR NR'',13,'','**',5X)') I
CALL ABORT
END
IDENT ABORT
ENTRY ABORT
ABORT
ABORT,C
LABEL
XJ
EQ
LABEL
PROGRAM MODGEN

******************************************************************
* COMPARE TWO FILES (NEWFILE AND OLDFILE) AND PREPARE AN UPDATE *
* CORRECTION SET IN MODFILE.                                      *
******************************************************************

IMPLICIT INTEGER (A-Z)
PARAMETER(OLDBFL = 100)
PARAMETER(NEWBFL = 100)
PARAMETER(LINELEN = 72)
PARAMETER(SEQLENG = 18)
PARAMETER(SINGLEN = 40)
PARAMETER(RANGLEN = 40)
PARAMETER(BIGNUM = 1000)
PARAMETER(MINMATS = 5)

CHARACTER * (LINELEN)
CHARACTER * (LINELEN)
CHARACTER * (LINELEN)
CHARACTER * (SEQLENG)
CHARACTER * (SEQLENG)
CHARACTER * (RANGLEN)
CHARACTER * (SINGLEN)
CHARACTER * (SINGLEN)
CHARACTER * (RANGLEN)
CHARACTER * (SINGLEN)

LOGICAL OLDLINE(O:OLDBFL-1)
LOGICAL NEWLINE(O:NEWBFL-1)
INTEGER SEQNR(O:OLDBFL-1)
CHARACTER IDENT(18)
CHARACTER INSERT(1)
CHARACTER DELETE(1)
CHARACTER BEFORE(1)
CHARACTER FORMAT(1)
CHARACTER LINEFMT(1)
CHARACTER DUMSTR1(1)
CHARACTER DUMSTR2(1)
CHARACTER OLDFILE(1)
CHARACTER NEWFILE(1)
CHARACTER MODFILE(1)
CHARACTER CSIDENT(1)
CHARACTER DUMSTR(1)

COMMON/OLD/OLDFILE, OLDLINE, SEQNR
COMMON/NEW/NEWFILE, NEWLINE

PARAMETER(OLDUNIT = 1)
PARAMETER(NEWUNIT = 2)
PARAMETER(MODUNIT = 3)
PARAMETER(FORMAT = '(A,A,A)'
PARAMETER(LINEFMT = '(')
PARAMETER(IDENT = '*IDENT'
PARAMETER(INSERT = '*I'
PARAMETER(DELETE = '*D'
PARAMETER(BEFORE = '*B'

CALL GETPARM(OLDFILE, DUMSTR1, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(NEWFILE, DUMSTR1, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(MODFILE, DUMSTR1, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(CSIDENT, DUMSTR1, PARSTAT)
IF(PARSTAT.GT.0) THEN
CALL GETPARM(DUMSTR2, DUMSTR1, PARSTAT)
IF(PARSTAT.LT.0) GOTO 10
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
CALL ERROR(1)
CONTINUE

******************************************************************
* IF THE CORRECTION SET IDENTIFIER CSIDENT STILL HAS THE DEFAULT *
* IT IS CHANGED TO A NAME DERIVED FROM THE NAME OF MODFILE. *
* SPECIFIED TO BE DIFFERENT FROM "MODFILE".                    *
******************************************************************
**IF((CSIDENT.EQ.'MOD').AND.(MODFILE.NE.'MODFILE')) THEN**

**K=INDEX(MODFILE,',')-1**

**IF(K.EQ.-1) K=7**

**DO 11 J=1,K**

**I=ICHAR(MODFILE(J:J))**

**IF(16.LE.I.AND.I.LE.25) GOTO 12**

**CONTINUE**

**12 CSIDENT= 1 MOD 1 //MODF1LECJ:K)**

**ENDIF**

**OPEN THE FILES. OLDFILE AND NEWFILE MUST NOT BE EMPTY.**

**READ SEQUENCE LABEL OF FIRST LINE OF OLDFILE AND MEMORIZE IT IN LOCATION LIN1SEQ. WRITE IDENTIFICATION LINE ON MODFILE.**

**CALL OPENOLO(DUMMY1)**

**CALL OPENNEW(DUMMY1)**

**OPENCUNIT=MODUNIT,FILE=MODFILE,ERR=20)**

**GOTO 30**

**20 CALL ERROR(4)**

**RETRIEVE(MODUNIT)**

**CALL READOLD(IOLD+1,KOLD,NOLD,OLDEOF)**

**IF(OLDEOF) CALL ERROR(5)**

**LIN1SEQ=SEQNR(KOLD)**

**CALL READNEW(INEW+1,DUMMY1,DUMMY2,NEWEOF)**

**IF(NEWEOF) CALL ERROR(6)**

**WRITE(MODUNIT,FORMAT) IDENT,CSIDENT**

**REAO OLDFILE AND NEWFILE IN PARALLEL UNTIL TWO DIFFERENT LINES ARE ENCOUNTERED OR EOF IS HIT.**

**IOLD=0**

**INEW=0**

**40 CONTINUE**

**CALL READOLD(IOLD+1,KOLD,NOLD,OLDEOF)**

**CALL READNEW(INEW+1,KNEW,NNEW,NEWEOF)**

**IF(OLDEOF.OR.NEWEOF) GOTO 90**

**IF OLDLINE(KOLD).NE.NEWLINE(KNEW) GOTO 50**

**IOLD=IOLD+1**

**INEW=INEW+1**

**GOTO 40**

**FILL THE BUFFERS OF OLDFILE AND NEWFILE. THEN RUN ALONG SUCCEEDING DIAGONALS OF CONSTANT NEW LINE NUMBER. CONTINUE UNTIL BUFFERS ARE EXHAUSTED OR A MATCH IS FOUND.**

**50 CONTINUE**

**CALL READOLD(IOLD+OLDBFL-1,DUMMY1,NOLD,OLDEOF)**

**CALL READNEW(INEW+NEWBFL-1,DUMMY1,NNEW,NEWEOF)**

**DO 60 JOLD=IOLD+1,NOLD+MINMATS-1,NOLD+MINMATS**

**JNEW=SUM-JOLD**

**CALL READOLD(JOLD,KOLD,DUMMY1,DUMMY2)**

**CALL READNEW(JNEW,KNEW,DUMMY1,DUMMY2)**

**IF(OLDLINE(KOLD).NE.NEWLINE(KNEW)) THEN**

**IF(OLDLINE(KOLD)(1:5).EQ. '*DECK', OR, )**

**OLDLINE(KOLD)(1:8).EQ. '*COMDECK' THEN**

**ACTMATS=1**

**ELSEIF(JOLD+MINMATS-1.LE.NOLD.AND. )**

**JNEW+MINMATS-1.LE.NEW**

**DO 51 I=1,MINMATS-1**

**CALL READOLD(JOLD+1,KOLD,DUMMY1,DUMMY2)**

**CALL READNEW(JNEW+1,KNEW,DUMMY1,DUMMY2)**

**IF(OLDLINE(KOLD).NE.NEWLINE(KNEW)) GOTO 60**

**51 CONTINUE**

**ACTMATS=MINMATS**

**ELSE**

**IF(NEW.EOF. OR.NEWEOF)) GOTO 130**

**ACTMATS=MINU(NOLD+JOLD+1,NNEW-JNEW+1)**

**DO 52 I=1,ACTMATS-1**
CALL READOLD(JOLD+1,KOLD,DUMMY1,DUMMY2)
CALL READNEW(JNEW+1,KNEW,DUMMY1,DUMMY2)
IF(OLDLINE(KOLD).NE.NEWLINE(KNEW)) GOTO 60
CONTINUE
ENDIF
GOTO 70
ENDIF
60 CONTINUE
IF(OLDEOF.OR.NEWEOF) GOTO 90
GOTO 130
CONTINUE

******************************************************************
* RECORD APPROPRIATE MODIFICATIONS IN MODFILE.                  *
******************************************************************
70 CONTINUE
LOLD=JOLD-IOLD-1
IF(LOLD.GT.1) THEN
CALL READOLD(JOLD+I,KOLD1,DUMMY1,DUMMY2)
CALL READOLD(JOLD+2,KOLD2,DUMMY1,DUMMY2)
WRITE(MODUNIT,FORMAT) DELETE,RANGE(SEQNR(KOLD1),SEQNR(KOLD2))
ELSEIF(LOLD.EQ.1) THEN
CALL READOLD(JOLD+1,KOLD,DUMMY1,DUMMY2)
WRITE(MODUNIT,FORMAT) DELETE,SINGLE(SEQNR(KOLD))
ELSEIF(LOLD.GT.0) THEN
CALL READOLD(JOLD,KOLD,DUMMY1,DUMMY2)
WRITE(MODUNIT,FORMAT) INSERT,SINGLE(SEQNR(KOLD))
ELSE
WRITE(MODUNIT,FORMAT) BEFORE,SINGLE(LIN1SEQ)
ENDIF
DO 80 MNEW=INEW+1,JNEW-1
CALL READNEW(MNEW,KNEW,DUMMY1,DUMMY2)
WRITE(MODUNIT,LINEFMT) NEWLINE(SEQNRKNEW)
80 CONTINUE

******************************************************************
* POST EOF PROCESSING.                                          *
******************************************************************
90 CONTINUE
IF(JOLD.EQ.0.OR.,NOT.(OLDEOF.OR.NEWEOF)) GOTO 130
CALL READOLD(IOLD,KOLD,DUMMY1,DUMMY2)
CALL READNEW(INew,KNEW,DUMMY1,DUMMY2)
WRITE(MODUNIT,FORMAT) NEWLINE(KNEW)
CONTINUE

******************************************************************
* IF NO MATCH HAS BEEN FOUND, PREPARE CORRECTION SET THAT       *
* REPLACES A FULL DECK.                                         *
******************************************************************

130 CONTINUE
REWD(NEWUNIT)
REWD(MODUNIT)
WRITE(MODUNIT,FORMAT) IDENT,CSIDENT
140 CONTINUE
IOLO=NOLD+1
CALL READOLD(IOLD+BIGNUM,DUMMY1,NOLD,OLDEOF)
IFC.NOT.OLDEOF) GOTO 140
CALL READOLD(NOLD,KOLD,DUMMY1,DUMMY2)
IF(NOLD.GT.1) THEN
WRITE(MODUNIT,FORMAT) DELETE,RANGE(LIN1SEQ,SEQNR(KOLD))
ELSE
WRITE(MODUNIT,FORMAT) DELETE,SINGLE(LIN1SEQ)
END IF
150 CONTINUE
READ(NEWUNIT,LINEFMT ,END=160) LINE
WRITE(MODUNIT,LINEFMT)
GOTO 150
160 CONTINUE
CLOSECMODUNIT,STATUS='KEEP')
CLOSE(OLDUNIT)
CLOSE(NEWUNIT)
STOP 'PROGRAM MODGEN'
END

SUBROUTINE READOLD(RECNR,BUFLOC,LASTNR,EOFLAG)
IMPLICIT INTEGER(A-Z)
PARAMETER CB
=1000)
PARAMET£R(LINELEN= 72)
PARAMETER(SEQLENG= 18)
PARAMETER(OLDUNIT= 1)
LOGICAL EOF,EOFLAG
CHARACTER * (LINELEN) LINE CO:BFL-1)
CHARACTER •(SEQLENG) SEQNRCO:OFL-1)
IMPLICIT INTEGER(A-Z)
PARAMETER(BFL=1000)
PARAMETER(LINELEN=72)
PARAMETER(NEWUNIT=2)
LOGICAL EOF, EFLAG
CHARACTER* (LINELEN) LINE (0:BFL-1)
COMMON/NEW/NEWFILE, LINE

SAVE BEGIN, EIND, BUFPNT, EOF

IF(RECNR.LT.BEGIN) CALL ERROR(8)
IF(EOF) THEN
  IF(RECNR.GT.EIND) THEN
    I=MOD(BUFPNT+EIND-BEGIN+1,BFL)
    READ(NEWUNIT,1(A),END=20) LINE(I)
    EIND=EIND+1
    IF(EIND-BEGIN+1.GT.BFL) THEN
      BEGIN=BEGIN+1
      BUFPNT=MOD(BUFPNT+1,BFL)
    END IF
    GOTO 10
  END IF
END IF
IF(RECNR.LE.EIND) THEN
  BUFLOC=MOD(BUFPNT+RECNR-BEGIN,BFL)
  LASTNR=EIND
  EOF=.FALSE.
  RETURN
END IF

ENTRY OPENNEW(DUMMY)
OPEN(UNIT=NEWUNIT, FILE=NEWFILE, ERR=30, STATUS='OLD')
REWIND(NEWUNIT)
EOF=.FALSE.
BEGIN=0
EIND=0
BUFPNT=0
RETURN

CALL ERROR(3)
END

FUNCTION SINGLE(A)
CHARACTER*(*) SINGLE,A
L=LEN(A)
K=INDEX(A(2:L),',')
IF(K.EQ.0) K=L
I=INDEX(A(1:K),',')
DO 10 J=I+1,K
  IF(A(J:J).NE.'0') GOTO 20
10 CONTINUE
SINGLE=A(1:I)//A(J:K)
END

FUNCTION RANGE(A,B)
CHARACTER*(*) RANGE,A,B
LA=LEN(A)
KA=INDEX(A(2:LA),',')
IF(KA.EQ.0) KA=LA
IA=INDEX(A(1:KA),',')
DO 10 JA=IA+1,KA
  IF(A(JA).NE.'0') GOTO 20
10 CONTINUE
RANGE=A(1:IA)//A(JA:KA)//',',//B(JB:KB)
ELSE
  RANGE=A(1:IA)//A(JA:KA)//',',//B(JB:KB)
END
**PROGRAM MODGEN**

- **ERROR NR**, I3, **,**6X)

**REMARK(MSG)**

**ABORT**

**END**

**PROGRAM USL**

* **PROGRAM REFORMATING OUTPUT OF UPDATE SOURCE LISTING OBTAINED**

* FROM A CREATION RUN OR FROM AN AUDIT RUN OF OLDPL.

* AFTER "UPDATE,F,1=SOURCE,N,L=A124,0=FN.",

* OR "UPDATE,F,P=OLDPL,L=7,0=FN.",

* CALL: "USL,FN(,NOLIST).

* THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF

* CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL (*COMDECKS

* AND *DECKS).}

**CHARACTER*133 LINE,BLANK,ZERO**

**CHARACTER*116 HEADER**

**CHARACTER* 30 MASTER**

**CHARACTER* 26 CARDS**

**CHARACTER* 116 CORR,HDR2,LISTOFC**

**CHARACTER* 14 HDR1**

**CHARACTER* 12 COMMON,COMPIL**

**CHARACTER* 4 DECK**

**CHARACTER* 10 YANKDCK,ACTIVE,STARS,SLASHES**

**CHARACTER* 10 COMMON,COMPIL**

**CHARACTER* 8 SCDECK**

**CHARACTER* 7 SERROR,NAME**

**CHARACTER* 6 SDECK**

**CHARACTER* 5 SREAD**

**CHARACTER* 4 DECK**

**CHARACTER* 1 NOLIST,NY**

**DATA CARDS** /CARDS ENCOUNTERED IN INPUT/ 280

**DATA MASTER** /MASTER AUDIT, IDENT CARD TOTAL/ 290

**DATA LISTOFC** /LIST OF CONTROL/ 300

**DATA YANKDCK** /YANK$$/ 310

**DATA ACTIVE** /A/ 320

**DATA STARS** /****/ 330

**DATA SLIMIT** /LIMIT/ 340

**DATA SREAD** /READ/ 350

**DATA SCDECK** /COMDECK/ 360

**DATA SDECK** /DECK/ 370

**DATA SERROR** /ERROR*/ 380

**DATA HRR1** /CREATION RUN/ 390

**DATA HRR2** /UNLAUNCHED OLDPL/ 400

**DATA CORR** /CORRECTION IDENT/ 410

**DATA BLANK** /'/ 420

**DATA ZERO** /0/ 430

**DATA DECK** /DECK*/ 440

**DATA COMMON** /COMMON DECKS*/ 450

**DATA COMPIL** /COMPILE FILE/ 460

**IPMAX=150**

**ILMAX=60**

**LRAX=IPMAX*ILMAX**

**CALL GETPARM(FN,DUM,IDUM)**

**OPEN(10,FILE=FN)**

**OPEN(20,FILE='OUTPUT')**

**REWIND 10**

**CALL GETPARM(NOLIST,NY,NOLIST)**

**IF(NOLIST.EQ.-1) NOLIST='N'**

**IF(NOLIST.EQ. 1) NOLIST='Y'**

**IF(NOLIST.EQ.0) THEN**

**IF(NY.EQ.'1') NY='Y'**

**NOLIST=NY**

**ENDIF**

**DO 10 I=1,1000**

**DO 10 I=1,1000**
READ(10,1,END=100) LINE
IF(LINE(1:14).EQ.HDR1) THEN
 LOPT=4
  GOTO 20
ELSEIF(LINE(32:57).EQ.CARDS) THEN
  LOPT=4
  LINE(1:14)=HDR1
  GOTO 20
ELSEIF(LINE(1:16).EQ.HDR2) THEN
  LOPT=7
  IF(NOLIST.EQ.'Y') THEN
    DO 5 J=1,1000
    READ(10,1,END=100) LINE
    IF(LINE(11:26).EQ.LISTOFC) THEN
      LINE(1:116)=HEADER(1:30) //MASTER//HEADER(62:116)
      GOTO 20
    ENDIF
    5 CONTINUE
  ENDIF
  GOTO 20
ELSE
  WRITE(20,1)
  ENDIF
  10 CONTINUE
  GOTO 20
ENDIF

C * REFORMATING "CARDS ENCOUNTERED IN INPUT" (LOPT=4, CREATION RUN)
C * OR "MASTER AUDIT, IDENT CARD TOTAL" (LOPT=7, AUDIT RUN).
20 HEADER=LINE(1:116)
IF(NOLIST.EQ.'Y') THEN
  IP=1
  DO 25 J=1,1000
  READ(10,1,ERR=100,END=200) LINE
  IF(LINE(2:8).EQ.SERROR) THEN
    WRITE(20,4) LINE(2:133)
  ELSEIF(INDEX(LINE,CORR).NE.0) THEN
    GOTO 50
  END IF
  WRITE(20,14) LINE(11:92),LINE(92:98),LINE(104:107)
  ELSEIF(NSS.EQ.1).AND.(LINE(11:18).EQ.SREAO)) THEN
    WRITE(20,13) LINE(11:82)
  ELSEIF(NSS.EQ.1).AND.(LINE(11:15).EQ.SREAD)) THEN
    WRITE(20,15) LINE(11:82)
  ELSEIF(NSS.EQ.1).AND.(LINE(11:16).EQ.SCOECK)) THEN
    IF(N.EQ.1) THEN
      WRITE(20,6) IL=IL+4
      NAME=LINE(20:26)
      WRITE(20,*)
    ENDIF
  ELSE
    WRITE(20,16) NAME=LINE(20:26)
    WRITE(20,*)
  ENDIF
ELSE
  WRITE(20,1)
  ENDIF
  25 CONTINUE
  GOTO 200
ENDIF

30 WRITE(20,2) HEADER,IP
C * START LOOP ON IP.
31 IF=1
C * START LOOP ON IL.
32 IL=1
IF(NKEEP.EQ.1) THEN
  WRITE(20,3) L,L=1
  NKEEP=0
  L=L+1
  IL=IL+1
ENDIF
40 READ(10,1,ERR=100,END=200) LINE
IF((LINE(1:10).EQ.STARS).OR.(LINE(1:10).EQ.SLASHES)) NSS=1
IF((LOPT.EQ.7) NSS=1
IF((MASTER.EQ.1).AND.(LINE(11:16).EQ.SLIMIT)) THEN
  WRITE(20,15) LINE(11:82)
ELSEIF(LINE(1:10).EQ.YANKDCK) THEN
  LYA=LYA+1
  IF(LYA.EQ.1) THEN
    WRITE(20,1)' IN YANK$$ DECK:'
    L=IL+2
  ENDIF
  WRITE(20,14) LINE(11:82),LINE(92:98),LINE(104:107)
  ELSEIF(LINE(11:10).EQ.SREAD)) THEN
    WRITE(20,15) LINE(11:82)
  ELSEIF(LINE(11:16).EQ.SCOECK)) THEN
    IF(N.EQ.1) THEN
      WRITE(20,6) IL=IL+4
    ENDIF
  ELSE
    WRITE(20,16) NAME=LINE(20:26)
    WRITE(20,*)
  ENDIF
ELSE
  WRITE(20,1)
  ENDIF
IL=IL+1
IEQ=INDEX(LINE, '==')
IF(IEQ.GT.0) THEN
  WRITE(20,15) BLANK(1:(IEQ-10)/2)//LINE(IEQ:82)
  IL=IL+3
END IF
IF(IL.GT.ILMAX) THEN
  NKEEP=1
  GOTO 45
END IF
WRITE(20,3) L,LINE(11:82),LINE(92:98),LINE(104:107)
L=L+1
ELSEIF((NSS.EQ.1).AND.(LINE(11:15).EQ.'SDECK')) THEN
  IF(LOPT.NE.1).OR.(N.EQ.2)) THEN
    IF((N.EQ.2).AND.(IL.NE.1)) THEN
      IP=IP+1
      WRITE(20,2) HEADER,IP
      IL=1
    END IF
    WRITE(20,7)
    IL=IL+4
    N=3
  END IF
  NAME=LINE(17:23)
  WRITE(20,*)
  IL=IL+1
ELSEIF(INDEX(LINE,'CORR').NE.0) THEN
  GOTO 50
ELSEIF((LINE.NE.BLANK).AND.(LOPT.NE.7)) THEN
  WRITE(20,4) LINE(2:133)
ELSE
  GOTO 40
ENDIF
IL=IL+1
IF(IL.LE.ILMAX) GOTO 40

CZ * END LOOP ON IL.
  IP=IP+1
  IF(IP.LE.IPMAX) GOTO 30
C1 * END LOOP ON IP.
  GOTO 200
C * LAST PAGES OF UPDATE OUTPUT FILE (LISTING OF DECKNAMES).
  50 K=1
  L=1
  K=1
60 READ(10,1,ERR=100,END=200) LINE
  IF(INDEX(LINE,DECK).NE.0).AND.(K.EQ.1)) THEN
    IF(IP.EQ.1) HEADER(31:60)='DECK LIST AS WRITEN, IF NEWPL'
    WRITE(20,2) HEADER,IP
    WRITE(20,8)
    WRITE(20,9)
    K=2
  ELSEIF(INDEX(LINE,COMMON).NE.0).AND.(L.EQ.1)) THEN
    WRITE(20,11)
    L=2
  ELSEIF(INDEX(LINE,COMPIL).NE.0).AND.(M.EQ.1)) THEN
    WRITE(20,12)
    M=2
**Program UML**

```
**-----------------------------------------------------**
** PROGRAM UML **
**-----------------------------------------------------**
```

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- USL, 1 ---------

```
1000 STOP '**ERROR IN USL**'
2000 STOP 'PROGRAM USL'

**FORMATS.**

1 FORMAT(A)
2 FORMAT(A116,'PAGE',13/)
3 FORMAT(A11,14,'0',A72,'0',A7,'0',A4)
4 FORMAT(' ',A132)
6 FORMAT(/37X,'EQUALS',I==============,A37X,1 =COMDECKS=',
137X,'EQUALS ',B37X,'============== 1 )
7 FORMAT(/37X,'EQUALS ',A37X,'=',B37X,'=','0DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A',
223X,'NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE.'/)
8 FORMAT(' ',6X,A72)
9 FORMAT(' ',6X,A72,' ',A7,'.',A4)
10 FORMAT(/,6X,A,/) END
```

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- UML, 1 ---------

```
**-----------------------------------------------------**
** PROGRAM UML **
**-----------------------------------------------------**
```

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- USL, 1 ---------

```
**-----------------------------------------------------**
** PROGRAM UML **
**-----------------------------------------------------**
```

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- UML, 1 ---------
**--- MASTERFILE MFCC CY=30 --- 02/07/86 - 00.27.57. --------- UML, 2 ------**

```
c
IPMAX=150  460
ILMAX=60  470
LMAX=IPMAX*ILMAX  480

CALL GETPARM(FN, DUM, IDUM)  490
OPEN(10, FILE=FN)  500
OPEN(20, FILE='OUTPUT')  510
REWIND 10  520
CALL GETPARM(LIST, NY, ILLST)  530
IF(LIST.EQ.'-1') LIST='N'  540
IF(LIST.EQ.' 1 ') LIST='Y'  550
LIST=NY  560
ENDIF  570

C1  580
* START READING UPDATE OUTPUT FILE.  590
DO 10 I=1,1000  600
READ(10,1,END=1000) LINE  610
IF(LINE(1:10).EQ.UNLAB) THEN  620
HEADER=LEFT//HDR1//LINE(61:116)  630
IF(INDEX(LINE,MODIF).NE.0) THEN  640
IP=0  650
GOTO 50  660
ENDIF  670
GOTO 20  680
ELSE  690
WRITE(20,1) LINE  700
ENDIF  710
GOTO 100  720

C2  730
* REFORMATING "CARDS ENCOUNTERED IN INPUT".  740
L=1  750
N=1  760
NERR=0  770
DO 40 IP=1,IPMAX  780
WRITE(20,2) HEADER, IP  790
ENDIF  800

30 READ(10,1,END=50,ERR=100) LINE  810
ONE=LINE(1:10)  820
TWO=LINE(11:13)  830
THREE=LINE(11:82)  840
IF(LINE.EQ.BLANK).OR.(LINE.EQ.ZERO)) THEN  850
GOTO 30  860
ELSEIF(INDEX(LINE, MODIF).NE.0) THEN  870
GOTO 50  880
ELSEIF(INDEX(LINE, CORR).NE.0) THEN  890
GOTO 80  900
ELSEIF(ONE.EQ.UNLAB) THEN  910
GOTO 30  920
ELSEIF(ONE.EQ. STARS).OR.(ONE.EQ.SLASHES) THEN  930
IF(TWO.EQ.IDENT) THEN  940
IF(N.EQ.1) THEN  950
WRITE(20,7)  960
IL=IL+4  970
N=2  980
ENDIF  990
WRITE(20,5) BLANK  1000
IL=IL+1  1010
NAME=LINE(18:24)  1020
NR=1  1030
WRITE(20,3) L,THREE  1040
ENDIF  1050

ELSEIF(TWO.EQ.COMMT) THEN  1060
WRITE(20,3) L,THREE  1070
ELSEIF(TWO.EQ.STACK).OR.(TWO.EQ.SCALL).OR.  1080
A (LINE(11:17).EQ."*DEF(NF)."))  1090
B (LINE(11:15).EQ."*DECK(S)."))  1100
C (LINE(11:16).EQ."*COMDECK")) THEN  1110
WRITE(20,4) L,THREE, NAME, NR  1120
NR=NR+1  1130
ELSEIF(TWO.EQ.SDELETE).OR.(TWO.EQ.SINSERT) THEN  1140
DO 35 M=19,28  1150
IF(LINE(M,4).NE.BLANK) GOTO 36  1160
35 CONTINUE  1170
36 THRE=LINE(11:18)/LINE(M:82)  1180
WRITE(20,3) L,THREE  1190
ENDIF  1200
```
ELSEIF (ONE .NE. 0 .AND. NB .NE. 0) THEN
    IF (ONE .EQ. ERROR) THEN
        NERR=NERR+1
    ENDIF
    WRITE (20, 5) LINE (2:133)
    L=L-1
ELSE
    WRITE (20, 4) L, THREE, NAME, NR
    NR=NR+1
ENDIF
L=L+1
IF (L.LT. LMAX) GOTO 30
C * END LOOP ON L.
C 40 CONTINUE
GOTO 100
C C3 * REFORMATING "MODIFICATIONS".
50 IF (LIST.NE.'Y').OR.(NERR.NE.0)) THEN
    IPP=IP
    GOTO 61
ENDIF
IF (NR.EQ.1) THEN
    NAME2=NAME
    NR2=NR
ENDIF
N=1
DO 70 IPP=IP+1,IPMAX
    HEADER (1:60)=LEFT//HDR2
    WRITE (20, 2) HEADER, IPP
C * START LOOP ON IL.
    IL=1
    IB=1
    IF (N.EQ.1) THEN
        WRITE (20, 8) IL=IL+4
        N=2
    ENDIF 
70 CONTINUE
GOTO 100
C C3 * LAST PAGES OF UPDATE OUTPUT FILE (LISTING OF DECKNAMES).
80 IF (NAME2.NE.NAME).OR.(NR2.NE.NR) GOTO 101
K=1
L=1
M=1
N=1
90 READ (10,1,ERR=100,END=200) LINE
--- MASTERFILE MFCCY=30 ---- 02/07/86 - 00.27.57. ----------- UML, 4 --------

IF((INDEX(LINE,'DECK LIST AS WRITTEN').NE.0).OR.
A (INDEX(LINE,'DECKS ARE LISTED').NE.0)).AND.(K.EQ.1)) THEN
\[2060\]
  JP=IP+1
\[2070\]
  HEADER(1:60)=LEFT'/HDR3
\[2080\]
  WRITE(20,2) HEADER,IP
\[2090\]
  WRITE(20,9)
\[2100\]
  WRITE(20,11)
\[2110\]
  K=2
\[2120\]
ELSE IF((INDEX(LINE,COMMON).NE.0).AND.(L.EQ.1)) THEN
\[2150\]
  WRITE(20,12)
\[2150\]
  L=2
\[2160\]
ELSE IF((INDEX(LINE,VALUES).NE.0).AND.(M.EQ.1)) THEN
\[2170\]
  WRITE(20,13)
\[2180\]
  M=2
\[2190\]
ELSE IF((INDEX(LINE,COMPIL).NE.0).AND.(N.EQ.1)) THEN
\[2200\]
  WRITE(20,14)
\[2210\]
  N=2
\[2220\]
ELSE IF(LINE(2:4),EQ,'***') THEN
\[2230\]
  WRITE(20,1) LINE
\[2240\]
ELSE IF(LINE(,EQ.BLANK).OR.(LINE,EQ.ZERO)
A .OR.(LINE(1:10),EQ,UNLAB)
B .0X.(LINE(1:10).EQ,'DECKS ARE') THEN
\[2250\]
ELSE
\[2260\]
  IF(LINEXC3:13).EQ.'THIS UPDATE') WRITE(20,1) 'U'
\[2270\]
  IF(NE.1) WRITE(20,1) LINE
\[2280\]
ENDIF
\[2290\]
GOTO 90
\[2300\]
C

100 STOP **ERROR IN UML**
\[2340\]
101 STOP **ERROR: MODIFICATIONS DO NOT MATCH INSTRUCTIONS ON *IDENT**
\[2350\]
200 STOP **PROGRAM UML**
\[2360\]
C
\[2370\]
C + FORMATS,
\[2380\]
1 FORMAT(A)
\[2390\]
2 FORMAT(A16,'PAGE ',I3/)
\[2400\]
3 FORMAT(' ',I4,' ',A72)
\[2410\]
4 FORMAT(' ',14,' ',A72, ' ',A7, ' ','14)
\[2420\]
5 FORMAT(' ',A32)
\[2430\]
6 FORMAT(' ',A82, A7, ' ',A42)
\[2440\]
7 FORMAT(37X, '************', A
\[2450\]
8 /37X, '************')
\[2460\]
A /37X, '************', 'A
\[2470\]
B /37X, '************')
\[2480\]
8 FORMAT(37X, '************')
\[2490\]
9 FORMAT(26X,'(LISTING OF CORRECTION IDENTS SKIPPED)')
\[2500\]
11 FORMAT('/DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON *A',
\[2510\]
A ' NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE')
\[2520\]
12 FORMAT('/10X,COMON DECKS ENCOUNTED')
\[2530\]
13 FORMAT('/10X,VALUES DEFINED FOR THIS UPDATE')
\[2540\]
14 FORMAT('/10X,DECKS WRITTEN TO COMPILE FILE')
\[2550\]
END

--- MASTERFILE MFCCY=30 ---- 02/07/86 - 00.27.57. ----------- RUN205, 1 --------

*PROC**,RUN205*,!
NAME(=NAME OF THE PROGRAM? (N/..) - ) = (N=N,S,N=*F)
\[10\]
B (BINARY? (N/..) - ) = (N=N,S,N=BIN,*F)
\[20\]
G (GOFILE? (N/..) - ) = (N=N,S,N=GOF,*F)
\[30\]
NOEXECUTION? (N/Y) - ) = (N=N,Y=1,N=0,Y=1,0,1)
\[40\]
I INPUT? (N/..) - ) = (N=N,N=I,*F)
\[50\]
P PLOT? (N/..) - ) = (N=N,N=I,*F)
\[60\]
OPT [OPTIMIZATION? (N=1/1/BPRS) - ] = (N=1,K=1,N=1,Y=0,1,1,
\[70\]
UNS [UNSAFE] OPTIMIZATION? (N/Y) - ] = (N=N,Y=1,N=0,Y=1,0,1)
\[80\]
L LFTN20U LISTING? (N/..) - ) = (N=N,Y=1,N=0,Y=1,0,1)
\[90\]
LO LIST OPTIONS? (N=5/SX/AAMSX) - ] = (N=5,.K=5,.S=1,.A=1,.N=1)
\[100\]
TL TIME LIMIT? (N=100/..) - ] = (N=100,N=100,N=100,.S=S(0123456789))
\[110\]
WS WORKING SET? (N=256/..) - ] = (N=256,N=256,.S=S(0123456789))
\[120\]
LP (NUMBER LARGE PAGES? (N=5/..) - ] = (N=5,.K=5,.S=1,.A=1,.N=1)
\[130\]
NOR NO ROUTE OF THE JOU? (N/Y) - ] = (N=N,Y=1,N=0,Y=1,0,1)
\[140\]
*HELP, NOLIST.

RUN205, CREATES A JOB FOR THE 205 WHICH PERFORMS A COMPLETE COMPIL.
\[150\]
LOAD, AND EXECUTE SEQUENCE OF FORTAN PROGRAM NAME, OR ONLY
\[160\]
PART OF IT RESULTING IN A PERMANENT BINARY "B" OR GOFILE "G".
\[170\]
ALTERNATIVELY, IF "NAME" IS NOT SPECIFIED, A RUN IS PERFORMED
\[180\]
STARTING FROM EITHER "B" OR "G".
\[190\]
PARAMETERS:
\[200\]
NAME - NAME OF THE FORTAN PROGRAM
\[210\]
260
G - NAME OF THE BINARY TO BE COMPILED OR ATTACHED
NOEX - PROGRAM IS NOT EXECUTED ("B" OR "G" IS SPECIFIED)
I - NAME OF THE INPUT FILE
P - NAME OF THE PLOT FILE TO BE PRODUCED
OPT - OPTIMIZATIONS (OPTSV)
UNS - POTENTIALLY UNSAFE OPTIMIZATIONS ARE PERMITTED
L - FORTRAN LISTING OF THE PROGRAM IS PRODUCED
LO - FTN200 LISTING OPTIONS
TL - TIME LIMIT
WS - WORKING SET
LP - NUMBER OF LARGE PAGES
NOR - JOB IS CREATED BUT NOT ROUTED TO THE INPUT QUEUE.

HELP,NAME,NOLIST.
NAME MUST BE THE NAME OF A LOCAL FILE WITH THE FTN200 PROGRAM. IF "NAME" IS NOT SPECIFIED (TYPE: "L"), EXECUTION REQUIRES SPECIFICATION OF A PREVIOUSLY DEFINED "B" OR "G".

HELP,B,NOLIST.
B IS THE NAME OF THE BINARY FILE; DEFAULT: "BIN".

HELP,G,NOLIST.
G IS THE NAME OF THE GOFILE; DEFAULT: "GOF".

HELP,NOEX,NOLIST.
NOEX INDICATES THAT THE PROGRAM IS TO BE COMPILED, AND POSSIBLY LOADED, BUT NOT EXECUTED. THIS ONLY MAKES SENSE IF EITHER "B" OR "G" IS SPECIFIED. DEFAULT: NOEX=0 (EXECUTION).

HELP,I,NOLIST.
I MUST BE THE NAME OF A LOCAL FILE WITH THE INPUT; TO BE SPECIFIED IF EXECUTION OF THE PROGRAM REQUIRES INPUT.

HELP,P,NOLIST.
P INDICATES THAT PLOTS ARE TO BE MADE; THE PLOT FILE IS MADE PERMANENT UNDER THE NAME "P".

HELP,OPT,NOLIST.
OPT INDICATES THE POSSIBLE OPTIMIZATIONS OF THE FTN200 COMPILER:
D - OPTIMIZE DO-LOOPS
P - PROPAGATE COMPIL-TIME COMPUTABLE RESULTS
R - REMOVE REDUNDANT CODE
S - SCHEDULE INSTRUCTIONS
V - VECTORIZE DO LOOPS

HELP,UNS,NOLIST.
UNS PERMITS THE COMPILER TO PERFORM UNSAFE OPTIMIZATIONS. DEFAULT: UNS=0.

HELP,L,NOLIST.
L INDICATES THAT A COMPLETE FTN200 LISTING OF THE PROGRAM IS DESIRED. DEFAULT: L=0 (NO LISTING).

HELP,LO,NOLIST.
LO SPECIFIES THE FTN200 LISTING OPTIONS:
A - ASSEMBLY LISTING OF OBJECT CODE
M - MAP OF REGISTER FILE AND STORAGE ASSIGNMENTS
S - SOURCE LISTING
X - CROSS REFERENCE MAP

HELP,TL,NOLIST.
TL SPECIFIES THE TIME LIMIT. DEFAULT: 100.

HELP,WS,NOLIST.

HELP,LP,NOLIST.

HELP,NOR,NOLIST.
NOR INDICATES THAT THE JOB IS TO BE CREATED, BUT NOT ROUTED TO THE INPUT QUEUE OF THE CYC205.

ENDHELP.

IF,($NAME,NE,$SN$),AND,(.NOT.FILE(NAME,AS)), LERROR1.
NOTE: FILE NAME DOES NOT EXIST; TRY AGAIN.
RETURN,JOB,EDJOB,ZZUSE.
RETURN,EDJOB,ZZUSE.
RETURN,ABORT.
ENDIF, LERROR1.

IF,($1$E,NE,$SN$),AND,(.NOT.FILE(1,AS)), LERROR2.
NOTE: FILE I DOES NOT EXIST; TRY AGAIN.
RETURN,EDJOB,ZZUSE.
RETURN,ABORT.
ENDIF, LERROR2.
**MASTERFILE**

```
--- MASTFILM MFCC L CY=30 02/07/86 - 00.27.57. RUN205, 3 ---

* IF, NOR=0, L3. NOR.
ANI JOB.
RETURN, FILMPL.
REWIND, JOB.
COPYB, J0B, FILMPL.
* IF, SNAME, NE, SNS, LNAME.
REWIND, NAME.
* IF, $S$, NE, SNS, LNAME.
REWIND, I.
ROUTE, FILMPL, DC=#IN, ST=205.
* TYPE, EDJOB, ZZUSE.
REVERT.
* EXIT, S.
NOTE, SERRORS.
RETURN, EDJOB, ZZUSE.
REVERT, ABORT.
* JOB FOR THE 205.
* DATA, JOB.
XX12X, STUS.
-6.68- 02/07/86 - 00.27.57.  ---
EXIT, S.
NOTE, SERRORS.
RETURN, EDJOB, ZZUSE.
REVERT, ABORT.
* JOB FOR THE 205.
* DATA, JOB.
XX12X, STUS.
```

**RUN205:**

```
02/07/86 - 00.27.57.
```

**USER: AC=XXXACXXX, U=XXU1XX, PA=XPAX**

**RESOUCRC: TL=TL, NSW=WS, NIL=NL, PRIO=12**

```
**COMMENT:**
```

**PATTACH, S_Service**

```
```

**PATTACH, S_Service**

```
```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- RUN205, 4 --------

 IF ($IS,NE,NS) COMMENT.**INPUT RECORDS NAME AND I AFTER EOR**
 IF ($IS,EQ,NS) COMMENT.**INPUT RECORD NAME AFTER EOR**
 ENDIF, NAMENY.

* * 
PROCEDURE FOR FURTHER EDITING JOB.
DATA,EDJOB.
PROC,EDJOB=1,

 IF (.NOT. ANSWER) REVERT.
 IF (.NOT. ANSWER) REVER.
 RETURN,ZZZZZ1Z,ZZZZZ3Z.
 REVERT.

* * USEFILE FOR EDITING JOB.
DATA,ZZUSE.
SET,COUNT=1,LINES=19,EXP=1,PROMPT=##
FORMAT,NO

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- VAST205, 1 --------

 PROC, VAST205=1,
 NAME (NAME OF THE F7200 PROGRAM ) = ($F),
 OUT (NAME OF THE VAST OUTPUT ) = ($F,$N=TRANSL).

 HELP, NOLIST.
 VAST205 CREATES A JOB FOR THE 205 WHICH CALLS VAST AND CATALOGES THE
 EQUALS OUTPUT OF VAST (THE TRANSLATED PROGRAM) ON THE 750.
 PARAMETERS:
 NAME = IS THE NAME OF THE FORTRAN PROGRAM
 OUT = IS THE NAME OF THE PERMANENT OUTPUT FILE ON THE 750,
 DEFAULT IS "TRANSL".

 ENDHELP.

 IF, NOT, FILE(NAME, AS), NONAME.
 NOTE, FILE NAME DOES NOT EXIST; TRY AGAIN.
 RETURN, ZZVAST.
 REVERT, ABORT.

 ENDIF, NONAME.

 REWIND, ZZVAST, NAME.
 COPYR, ZZVAST, FILMPL.
 COPYR, NAME, FILMPL.

 FMLOCK, ON.
 ROUTE, FILMPL, DC=IN, ST=205.
 RETURN, ZZVAST.

 NOTE, WHEN THE JOB IS DONE, YOU CAN GET THE.
 NOTE, OUTPUT WITH "ATTACH,OUT,ID=XXIDX.".
 NOTE, REPLY OF ROUTE:

 FMLOCK, OFF.

* * JOB FOR THE 205.
 DATA, ZZVAST.

 USER(AC=$XXXACXXX, U=$XXU1XX, PA=$XPAX)

 RESOURCE(TL=10, WS=512, JCAT=NORMAL)

 COMMENT.***********************************************************

 VAST205: NAME TRANSLATED BY VAST ON THE 205,
 COMMENT. OUTPUT OUT CATALOGED ON THE 750.
 COMMENT.***********************************************************

 VAST(TAPE1=INPUT, TAPEZ=$S, TAPE3=$T)
 MFLINK(T, ST=MBE, JCS="ACCOUNT,$XXXACXXX, XXUNXX.",
 "CATALOG,OUT,ID=XXIDX.")

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- ALIAS, 1 --------

 PROC, ALIAS=1,
 FLIST [FILES? ($FN1/FN2/...$) = ($A)]
 MF1 [TST MF? (N=MASTER/..) = ($AN=MASTER,N=MASTER,$F),
 MF2 [2ND MF? (N=SECOND/..) = ($AN=SECOND,N=SECOND,$F),
 ID [Tape1 FILE ID? (N/..) = ($AN=$XXXACXXX,N=$XXIDX,X,$A),
 UN [Tape1 LOGIN NAME? (N/..) = ($AN=$XXUNXX,X=$XXUNXX,$A),

--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. --------- ALIAS, 2  ---------

12 C205 FILE ID? (N/..) - 3 = (*N=*XXIDX,X,N=*XXIDX,X,*A), 80
AC C205 ACCOUNT NR? (N/..) - 9 = (*N=*XXXACCXXX,X,N=*XXXACCXXX,X,*A),
U1 C205 USER NR 1? (N/..) - 3 = (*N=*XXXU1XX,X,N=*XXXU1XX,X,*A), 100
U2 C205 USER NR 2? (N/..) - 3 = (*N=*XXXU2XX,X,N=*XXXU2XX,X,*A), 110
PA C205 PASS WORD? (N/..) - 3 = (*N=*XPAXS,X,N=*XPAXS,X,*A), 120
TA LOCAL COMPUTER? (N/..) - 3 = (*N=*XXAXS,X,N=*XXAXS,X,*A),
TB LINE PRINTER? (N/..) - 3 = (*N=*XXBS,X,N=*XXBS,X,*A), 140

*HELP, NOLIST.

ALIAS TRANSFERS FILES FROM ATTACHED MASTERFILE PFN1, M=MF1, ID=XXIDX
TO A SECOND ATTACHED MASTERFILE PFN2, M=MF2, ID=XXIDX, WHILE
CHANGING ALL PERSONAL ID'S, ACCOUNTS, AND PASSWORDS INTO
THE SPECIFIED ONES.

PARAMETERS:

- FLIST - LIST OF FILES TO BE TRANSFERRED:
  - "FN" - ONE FILE
  - "$FN1/FN2/..." - A FEW FILES (STRING <= 40 CHARs!)
  - "$ $" - ALL FILES

- MF1 - MAIN MASTERFILE (DEFAULT: "MASTER")
- MF2 - RECEIVING MASTERFILE (DEFAULT: "SECOND")
- ID - CY750 FILE ID (DEFAULT: "$XXIDX")
- ACC - CY750 ACCOUNT NR (DEFAULT: "$XXXACCXXX")
- U1 - CY750 USER NR 1 (DEFAULT: "$XXXU1XX")
- U2 - CY750 USER NR 2 (DEFAULT: "$XXXU2XX")
- PA - CY750 PASS WORD (DEFAULT: "$XPAXS")
- TA - TID LOCAL COMPUTER (DEFAULT: "$XXAXS")
- TB - TID LINE PRINTER (DEFAULT: "$XXBS").

*ENDHELP.

** RETURN, GIVE.
MFLIST, FLIST, M=MF1, CCL=GIVE/TRANS.
ED, DEDB.
GIVE.
RETURN, GIVE, TRANS, USEFILE, EDLOG.
REVERT.

* EXIT, S.
COMMENT, ** ERROR IN ALIAS **
RETURN, GIVE, TRANS, USEFILE, EDLOG.
REVERT, ABORT.

* DATA, TRANS.
PROC, TRANS, FN, MF1, DATE, TIME, RAND, COM, LOCK, SIZE.
FGET, FN, M=MF1.
REWIND, USEFILE.
ED, E, FN.
ED, USE, USEFILE.
ED, W, FN, 0.
ED, SC, X, INIT.
ED, Q, Q.
FREP, FN, M=MF2.
RETURN, FN.
REVERT.

* EXIT, S.
COMMENT, ** ERROR IN TRANS **
RETURN, FN.
REVERT, ABORT.

* DATA, USEFILE.
// XXIDX/ID/UBC*
// XXIDX/ACC/UBC*
// XXIDX/U1/UBC*
// XXIDX/U2/UBC*
// XXXACCXXX/ACC/UBC*
// XXXU1XX/U1/UBC*
// XXXU2XX/U2/UBC*
// XXXACCXXX/ACC/UBC*
// XXXU1XX/U1/UBC*
// XXXU2XX/U2/UBC*
// XPAX/PA/UBC*
// XXA/TA/UBC*
// XXB/TB/UBC*
FILE:

COMMENT - THIS FILE.

**HBT** - FRAMEWORK FOR A JOB CREATING NEW UPDATE PL UHBT U FROM SOURCE HBT_S AND COMPILING BHBT B.
PROCEDURE N OF CCLLIB PRODUCES THE ACTUAL JOB N_S FROM THIS FILE, INSERTING VALUES FOR S,U,B.

**RHB** - FRAMEWORK FOR A JOB REVISI NG OLDPL UHBT U WITH MODIFICATION MHBT M (NEWPL IS UHBT V) AND COMPILING BHBT B.
PROCEDURE R OF CCLLIB PRODUCES THE ACTUAL JOB R_M FROM THIS FILE, INSERTING VALUES FOR U,M,V,B.

**XHB** - FRAMEWORK FOR A JOB EXECUTING BHBT B WITH INPUT UB (PL邮轮 P.B.I AND OUTPUT O.B.I MAY BE CATALOGED).
PROCEDURE X OF CCLLIB PRODUCES THE ACTUAL JOB X_B.I FROM THIS FILE, INSERTING VALUES FOR B,L.P,T,L,M,T,LP.

**RHBT** - FRAMEWORK FOR A JOB COMPILING BH BT B ON THE 205 FROM CHBT V (DUE TO NEWPL UHBT V FROM MODIFICATION MHBT M, OLDPL UHBT U).
PROCEDURE RR OF CCLLIB PRODUCES BOTH THE COMPILE FILE AND THE ACTUAL JOB R_M FROM THIS FILE, INSERTING VALUES FOR U,M,V,B.

**XHBT** - FRAMEWORK FOR A JOB EXECUTING BHBT B WITH INPUT UB.
PROCEDURE X OF CCLLIB PRODUCES THE ACTUAL JOB X_B.I FROM THIS FILE, INSERTING VALUES FOR B,IC,P,O,D,T,LP.

**XXHBT** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.

**NNHBT** - FRAMEWORK FOR A JOB REVISING OLDPL UHBT U WITH MODIFICATION MHBT M (NEWPL IS UHBT V) AND COMPILING BHBT B.
PROCEDURE NN OF CCLLIB PRODUCES THE ACTUAL JOB NN S FROM THIS FILE, INSERTING VALUES FOR S,U,A.

**RHBT** - FRAMEWORK FOR A JOB EXECUTING BHBT B WITH INPUT UB.

**XXHBT** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.

**RFUSE,MFHT, ID=XXIDX, MR=1.**

**TTAKE,HBT=HBT S.**

**UPDATE,F_I=INPUT,N=UHBT,C=CHBT,L=A124,0=OUT.**

**ATTACH,CCLLIB, ID=XXIDX, MR=1.**

**LIBRARY,CCLLIB.**

**F755,OUT, NOLIST=NOL.**

**CATALOG, UHBT, UHBT_1, ID=XXIDX.**

**F455, L=CHBT, RB=MHBT, OPT=3,0=L,PL=10000,L=0.**

**CATALOG, BHBT, BHBT_1, ID=XXIDX.**

**EXIT, FS, 16U.**

**CATALOG, OUT, ON, S, ID=XXIDX.**

**LIMIT 1000U.**

**READ HAT**

-- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. -------- RHBT, 1 --------

XXIDX, CM110000, 150, 10250, WP.
ACCOUNT, XXXACXX, XXUNXX.
COMMENT.**********                       **********

**RHBT** - FRAMEWORK FOR A JOB REVISI NG OLDPL UHBT U WITH MODIFICATION MHBT M, NEW PL-UHBT V, COMPILING BHBT B.
PROCEDURE R OF CCLLIB PRODUCES THE ACTUAL JOB R_M FROM THIS FILE, INSERTING VALUES FOR U,M,V,B.

**XRB** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.

**RHBT** - FRAMEWORK FOR A JOB REVISING OLDPL UHBT U WITH MODIFICATION MHBT M (NEWPL IS UHBT V) AND COMPILING BHBT B.
PROCEDURE RR OF CCLLIB PRODUCES BOTH THE COMPILE FILE AND THE ACTUAL JOB R_M FROM THIS FILE, INSERTING VALUES FOR U,M,V,B.

**XHBT** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.

**RHBT** - FRAMEWORK FOR A JOB EXECUTING BHBT B WITH INPUT UB.

**XXHBT** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.

**RHBT** - FRAMEWORK FOR A JOB REVISING OLDPL UHBT U WITH MODIFICATION MHBT M (NEWPL IS UHBT V) AND COMPILING BHBT B.
PROCEDURE RR OF CCLLIB PRODUCES BOTH THE COMPILE FILE AND THE ACTUAL JOB R_M FROM THIS FILE, INSERTING VALUES FOR U,M,V,B.

**XHBT** - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.
--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. -------- RHBT, 2 --------

COMMENT.****************************************                  80
REDUCE.                                                     90
ATTACH, UHBT, ID=XXIDX, MR=1.                               100
MFUSE, MHBT, ID=XXIDX, MR=1.                                110
FTAKE, MHBT=MHBT_M.                                          120
UPDATE, P=UHBT, I=UHBT, M=UHBT, C=CHBT, L=81234, O=OUT.    130
ATTACH, CCLLIB, ID=XXIDX, MR=1.                             140
LIBRARY, CCLLIB.                                             150
UML, OUT, LIST=ULIST.                                       160
CATALOG, UHBT, ID=XXIDX, MR=1.                              170
FNT5, I=CH8T, #B=BH8T, OPT=2, LO=S/-A, PL=10000, L=O.        180
CATALOG, UHBT, ID=XXIDX.                                   190
EXIT.                                                        200
CATALOG, OUT, OR_M, ID=XXIDX.                               210

--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. -------- RMHBT, 1 --------

XXIDX, CM275U00, T1000, I020, NP.                           10
ACCOUNT, XXXACCXXX, XXUNXX.                                 20
COMMENT.****************************************              30
COMMENT. X_B_I.                                             40
COMMENT. EXECUTION UHBT_B WITH INPUT IHBT.                  50
 IF($PS,N.E.SN$)COMMENT. _PLT FILE #P_B_1 CATALOGED.        60
 IF($SO,N.E.SN$)COMMENT. OUTPUT NO_B_I CATALOGED.           70
 IF($SO,N.E.SN$)COMMENT. DATA FILE #D_B_I CATALOGED.        80
MEMORY,**********                                           90
REDUCE.                                                     100
ATTACH, UHBT, ID=XXIDX, MR=1.                               110
MFUSE, MHBT, ID=XXIDX, MR=1.                                120
FTAKE, IHBT=IHBT_I.                                         130
ATTACH, HGOLIB, ID=XXIDX, MR=1.                             140
ATTACH, NAG.                                                150
ATTACH, PPPLIB, ID=XXIDX, MR=1.                             160
LDSET, LIB=HGOLIB/NAG/PPPLIB/CALCOMP.                       170
BHBT, IHBT.                                                 180
 IF($PS,N.E.SN$)CATALOG, PHBT, #P_B_I, ID=XXIDX.            190
 IF($SO,N.E.SN$)LCATO.                                      200
REWIND, OUTPUT.                                             210
COPY, OUTPUT, OUT.                                          220
CATALOG, OUT, #O_B_1, ID=XXIDX.                            230
ENDIF, LCATO.                                               240
IF($SO,N.E.SN$)CATALOG, QHBT, #D_B_1, ID=XXIDX.             250

--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. -------- RHBT, 1 --------

XX12X, ST20S.                                                10
USER(AC=XXXXXXX, #U=XXU1XX, PA=XPAX)                        20
RESOURCE(TL=100, WS=256, LP=0, PRI0=12)                    30
COMMENT.****************************************            40
COMMENT. N_S:                                               50
COMMENT. COMPILED UHBT_B FROM CHBT_U.                       60
COMMENT. (ORIGINATING FROM NEW SOURCE MBT_S).              70
MEMORY,**********                                           80
PURGE, UHBT_B.                                              90
 IF(FLIST=0)FTN200, I=INPUT, #B=UHBT_B/500, LO=O, OPT=1.   100
 IF(FLIST=1)FTN200, I=INPUT, #B=UHBT_B/500, OPT=1.        110
DEFINE, UHBT_B.                                            120
COMMENT.**INPUT RECORD CHBT_U AFTER EOR**                  130

--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. -------- RMHBT, 1 --------

XX12X, ST205.                                                10
USER(AC=XXXXXXX, #U=XXU1XX, PA=XPAX)                        20
RESOURCE(TL=100, WS=256, LP=0, PRI0=12)                    30
COMMENT.****************************************            40
COMMENT. R_S:                                               50
COMMENT. COMPILED UHBT_B FROM CHBT_V.                       60
COMMENT. (ORIGINATING FROM NEW SOURCE MBT_M ON UHBT_U).    70
MEMORY,**********                                           80
PURGE, UHBT_B.                                              90
 IF(FLIST=0)FTN200, I=INPUT, #B=UHBT_B/500, LO=O, OPT=1.   100
 IF(FLIST=1)FTN200, I=INPUT, #B=UHBT_B/500, OPT=1.        110
DEFINE, UHBT_B.                                            120
COMMENT.**INPUT RECORD CHBT_V AFTER EOR**                  130
L.73

--- MASTERFILE MFHBT CY=45 ----- 02/07/86 - 00.29.27. --------- XXHBT, 1 -------

XX12X,ST205.
USER(AC=XXA(XXX, U=XXU1XX, PA=XXPA)
RESOURCE(TL=200, WS=700, LP=3, PRI0=12)
COMMENT.**************************************************
COMMENT. XX_B_I:
COMMENT. EXECUTION DHST_B WITH INPUT IMBT_I.
.IF(P=1)COMMENT. PLOT FILE #P_B_I CATALOGED.
.IF(D=1)COMMENT. OUTPUT #D_B_I CATALOGED.
.IF(D=1)COMMENT. DATA FILE #D_B_I CATALOGED.
COMMENT.**************************************************
PATTACH,SAKALIB.
110
ATTACH,BHBT_B,
120
ATTACH,SAKLIB.
130
ATTACH,STBT_B.
140
ATTACH,STBT_B.
150
LOAD,BHBT_B, CN=STBT_B/2000, COF=6400,
   LIB=F0ULIB,F0ULIB,NAG,PLOTTLN,PPPLIB,
   GRLP=FIVE,SEVEN,L=0.
160
.IF(P=1)PURGE,TAPE99.
170
.IF(D=1)PURGE,TAPE99.
180
.IF(D=1)DEFINE,OHBT.
190
.IF(D=1)DEFINE,OHBT.
200
.IF(D=1)DEFINE,OHBT.
210
.IF(D=1)DEFINE,OHBT.
220
.IF(D=1)DEFINE,OHBT.
230
.IF(D=1)DEFINE,OHBT.
240
.IF(D=1)DEFINE,OHBT.
250
.IF(D=1)DEFINE,OHBT.
260
.IF(D=1)DEFINE,OHBT.
270
.IF(D=1)DEFINE,OHBT.
280
COMMENT.**INPUT RECORD IMBT_I AFTER EOR**
-- MASTERFILE MFHGO CY=27 ---- 02/07/86 - 00.30.06. --------- NHGO, 1 --------

COMMENT:
--- MASTERFILE MFHGO CY=27 ---- 02/07/86 - 00.30.06. --------- HHGO, 1 --------

XXIOM,CM1000000,T100,10200,NP.
ACCOUNT,XXARCCXXXX,XXUNXX.
COMMENT,**********************************************************
.IF($M$.EQ.$N$)COMMENT.
.NHGO:
.IF($M$.NE.$N$)COMMENT.
.NHGO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($N$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIMX,MR=1.
ATTACH,CCLLIB,1D=XXIMX,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOS,N=NHGO,S=CHGO,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
MUSE,FMHGO,1D=XXIOM,MR=1.
ATTACH,CCLLIB,1D=XXIOM,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGOM,N=NHGOM,S=CHGOM,L=A1244,0=OUT1.
USL,OUT1,HGOLIB=NOUL.
.IF($M$.EQ.$N$)COMMENT.
.XXXO:
.IF($M$.NE.$N$)COMMENT.
.XXXO-M:
COMMENT,NEW LIBRARY HGOLIB FROM SOURCE HGO S
.IF($M$.NE.$N$)COMMENT. MODIFIED WITH MHGO M.
COMMENT,**********************************************************
REDUCE.
--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 15.53.07. ------- COMMENT, 1 -------

COMMENT.
MASTERFILE MFPPP CY=10
03/05/86
*******
MASTERFILE MFPPP CONTAINS THE SOURCE PPP S OF PLOTTING LIBRARY
PPPLIB AND SOME AUXILIARY FILES.
*******

FILE:
COMMENT - THIS FILE.
NPPP - JOB CREATING NEW LIBRARY PPPLIB FROM PPP S (+ MPPP M) ON
THE 750. THIS JOB IS EDITED AND LAUNCHED BY PROCEDURE NEW
OF CCLLIB.
CALL: "NEW,PPP, S=(M=,NOUL,NOR,TID)."

NPMLP - JOB COMPILED NEW LIBRARY PPPLIB FROM CPPP S (OR CPPP M)
ON THE 205. THIS JOB IS EDITED AND LAUNCHED BY PROCEDURE
NEW OF CCLLIB.
CALL: "NEW,PPP, S=(M=,NOUL,NOCAT,FLIST,NOR)."

*PPP10 - SOURCE FILE FOR LIBRARY PPPLIB.

MPPP10A - UPDATE CORRECTION *IDENT (CONTAINING "*DEFINE SARA, CY750")
FOR COMPILATION OF PPPLIB, CY=10 FROM PPP10 ON THE CY750.

MPPP10A - UPDATE CORRECTION *IDENT (CONTAINING "*DEFINE SARA, CY205")
FOR COMPILATION OF PPPLIB, CY=10 FROM PPP10 ON THE CY205.

*******
--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.30.37. ------- NPPP, 1 -------

NPPP, XXI0X, CM700000, T100, 1020, NP.
ACCOUNT, XXAXACXX, XXUXNXX.
COMMENT,******************************
.*IFS(MS.EQ.SNS),COMMENT, NPPP_S:
.*IFS(MS.NE.SNS),COMMENT, NPPP_M:
COMMENT, NEW LIBRARY PPPLIB FROM SOURCE PPP S
.*IFS(MS.EQ.SNS),COMMENT, MODIFIED WITH MPPP_M:
COMMENT,******************************
REDUCE.
MFUSE,MFPPP, ID=XXI0X, MR=1.
FTAKE,PPP=PPP_S.
ATTACH, CCLLIB, ID=XXI0X, MR=1.
LIBRARY, CCLLIB.
UPDATE, I=PPP, N=UPPP, c=CPPP, L=A124, 0=OUT1.
USL, OUT1, NOLIST=NOUL.
IF, SMS.EQ.SNS, NY.
CATALOG, UPPP, UPPP_S, 10=XXI0X, MR=1.
FIN5, I=CPPP, B=PPP, OPT=2, 0=45/-A, L=0.
ELSE, NY.
FTAKE, MPPP=MPPP_M.
UPDATE, F=P=UPPP, I=PPP, N=NEWPL, C=Compile, L=A1234, 0=OUT2.
CATALOG, NEWPL, UPPP_M, ID=XXI0X, MR=1.
UNIT, OUT2.
FT5S, I=Compile, B=PPP, OPT=2, 0=45/-A, L=0.
ENDIF, NY.
REQUEST, LIB, *PF.
EDITLIB(USER, L=$CRI)
CATALOG, LIB, PPPLIB, ID=XXI0X, MR=1.
ITEMIZE, LIB.
COMMENT, **EDITLIB DIRECTIVES AFTER EOR**
LIBRARY(LIB,NEW)
ADD(*, BPPP)
FINISH.
ENDRUN.

--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.30.37. ------- NPPP, 1 -------

NPPP, XXI0X, CM700000, T100, 1020, NP.
ACCOUNT, XXAXACXX, XXUXNXX.
COMMENT,******************************
.*IFS(MS.EQ.SNS),COMMENT, NPPP_S:
.*IFS(MS.NE.SNS),COMMENT, NPPP_M:
COMMENT, NEW LIBRARY PPPLIB FROM SOURCE PPP S
.*IFS(MS.EQ.SNS),COMMENT, MODIFIED WITH MPPP_M:
COMMENT,******************************
REDUCE.
MFUSE,MFPPP, ID=XXI0X, MR=1.
FTAKE,PPP=PPP_S.
ATTACH, CCLLIB, ID=XXI0X, MR=1.
LIBRARY, CCLLIB.
UPDATE, I=PPP, N=UPPP, c=CPPP, L=A124, 0=OUT1.
USL, OUT1, NOLIST=NOUL.
IF, SMS.EQ.SNS, NY.
CATALOG, UPPP, UPPP_S, 10=XXI0X, MR=1.
FIN5, I=CPPP, B=PPP, OPT=2, 0=45/-A, L=0.
ELSE, NY.
FTAKE, MPPP=MPPP_M.
UPDATE, F=P=UPPP, I=PPP, N=NEWPL, C=Compile, L=A1234, 0=OUT2.
CATALOG, NEWPL, UPPP_M, ID=XXI0X, MR=1.
UNIT, OUT2.
FT5S, I=Compile, B=PPP, OPT=2, 0=45/-A, L=0.
ENDIF, NY.
REQUEST, LIB, *PF.
EDITLIB(USER, L=$CRI)
CATALOG, LIB, PPPLIB, ID=XXI0X, MR=1.
ITEMIZE, LIB.
COMMENT, **EDITLIB DIRECTIVES AFTER EOR**
LIBRARY(LIB,NEW)
ADD(*, BPPP)
FINISH.
ENDRUN.
--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.31.10. ------- MPPP10, 1 ------

*IDENT MOD10
*DEFINE SARA, CY?50

--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.31.10. ------- MPPP10A, 1 ------

*IDENT MOD10A
*DEFINE SARA, CY?05
PERMANENT FILE ID=PUBLIC -- 03/07/86 - 13.58.56. -- PROCFIL CY=1, 1 -------

(ADAPTED TEXT)

--- PERMANENT FILE ID=XXIDX --- 03/07/86 - 13.55.32. -- PROCFIL CY=10, 1 -------