ASSOCIATIE EURATOM-FOM

OM-INSTITUUT VOOR PLASMAFYSICA

RIJNHUIZEN — NIEUWEGEIN — NEDERLAND

July 1986

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

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, reformating 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.

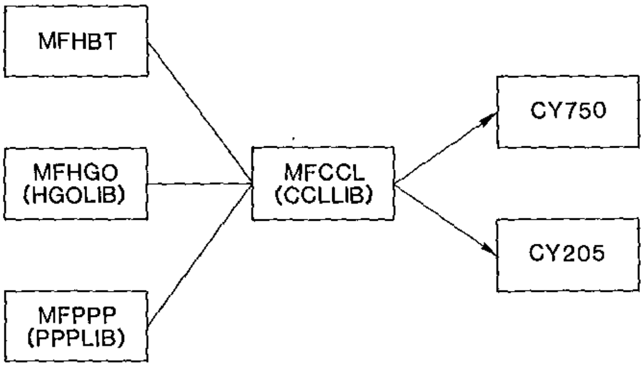


Fig. 1: Schematics of file handling

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 masterfiles (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 MFCCL, 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

[1] American National Standard Programming Language FORTRAN (American National Standard Institute, New York, 1978).
Extensions for the CDC computers are described in:
[1a] FORTRAN Version 5 Reference Manual, for use with the CDC NOS and NOS/BE systems (Control Data Corporation, Sunnyvale, California, 1983).
[1b] FORTRAN 200 Version 1 Reference Manual, for use with the CDC VSOS system (Control Data Corporation, Sunnyvale, California, 1983).

[2] J. Manickam, private communication.

[3] UPDATE Version 1 Reference Manual (Control Data Corporation, Sunnyvale, California, 1982).

[4] NOS/BE Version 1 Reference Manual (Control Data Corporation, Sunnyvale, California, 1984).

[5] CDC VSOS Version 2 Reference Manual (Control Data Corporation, Sunnyvale, California, 1984).

[6] U. Haanstra, Masterfiles, SARA-31 (Stichting Academisch Rekencentrum, Amsterdam, 1984).

[7] SARED, SARA-24 (Stichting Academisch Rekencentrum, Amsterdam 1980).

[8] J.P. Goedbloed, Computer Physics Communications **31** (1984) 123;
J.P. Goedbloed, G.M.D. Hogeweij, R. Kleibergen, J. Rem, R.M.O. Galvão, P.H. Sakanaka, Plasma Physics and Controlled Nuclear Fusion Research, 1984 (IAEA, Vienna, 1985), Vol. 2, p. 165.

[9] J.P. Goedbloed, G.M.D. Hogeweij, and D.W. Hewett, Plasma Physics Plotting Library PPPLIB (Rijnhuizen Report 86-166, 1986).

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 MFCCL, MFHBT, MFHGO, 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 MFCCL, the library CCLLIB is installed by means of the procedure NEWCCL (2) which adds one procedure (e.g., NEWCCL itself) to both MFCCL and CCLLIB. Next, files may be added, replaced and deleted in both MFCCL and CCLLIB by means of ADDCCL (3), REPCCL (4), and DELCCL (5). If this process of continuously adding and replacing files has created too much redundant space, the procedure COPYCCL (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.

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

Calling EDCY (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) Formating 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, equiped 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 ZZSYS1 (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,MF..." 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 PPPLIB. 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 PPPLIB 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 consitutes 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, wheras 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 NNHGO (92) of MFHGO, and the job NNPPP (95) and the modification deck MPPP10A (97) of MFPPP 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 looses 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

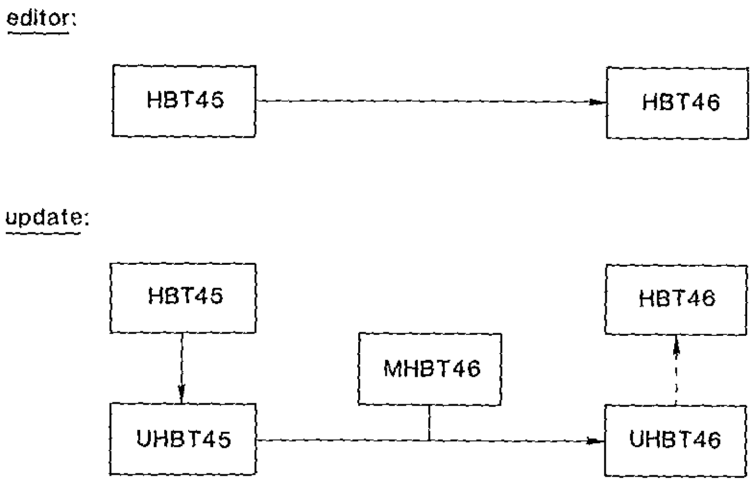


Fig. 2: Editing and updating programs

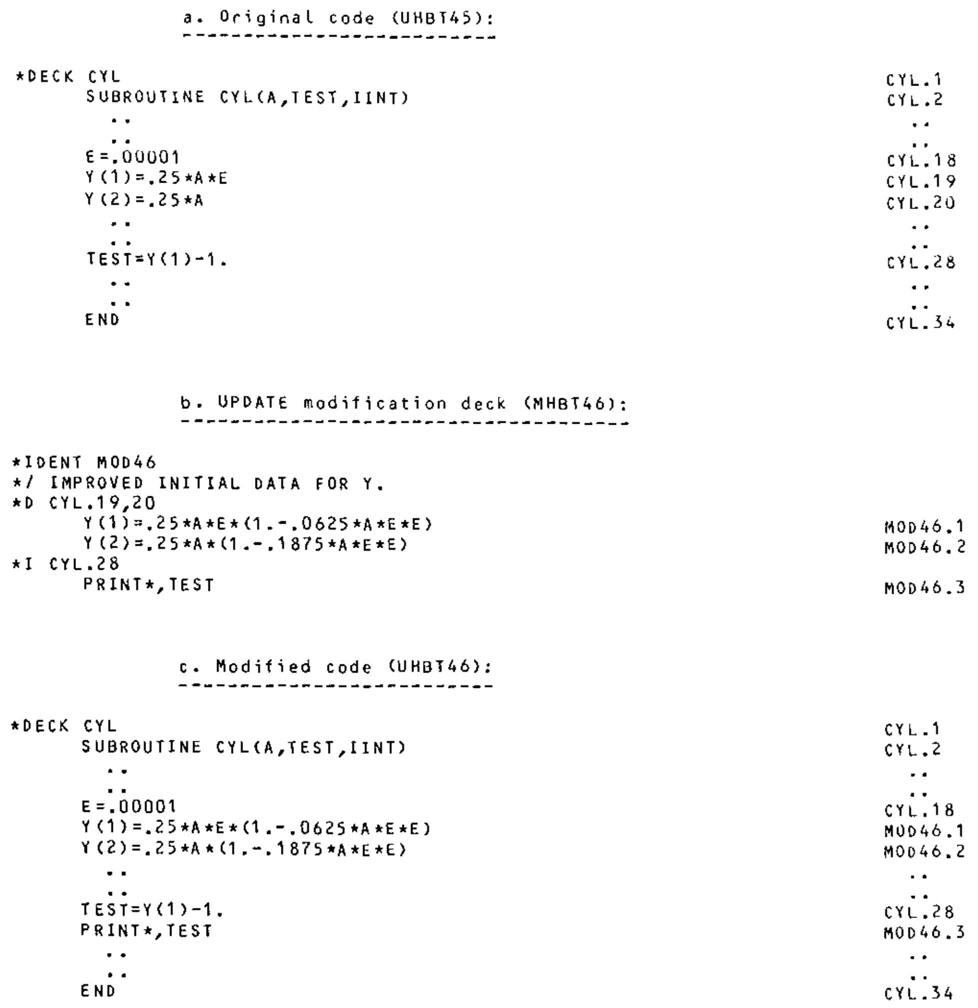
program) is impeded, 2) inadvertently, alongside the intended improvements of the code, trivial errors may have been introduced with the result that the new version of the code should be considered inferior to the old one. Consequently, the speed of this approach usually turns out to be illusory: much 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) preceeded 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.34. [We note in passing that a similar UPDATE directive exists for labeling COMMON blocks, viz. the *COMDECK directive. Together with the *CALL directive, which inserts the contents of the specified comdeck at a particular place, this cures one of the deficiences 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



```
          a. Original code (UHBT45):
          --------------------------

*DECK CYL                                                  CYL.1
      SUBROUTINE CYL(A,TEST,IINT)                          CYL.2
        ..                                                  ..
        ..                                                  ..
      E=.00001                                             CYL.18
      Y(1)=.25*A*E                                         CYL.19
      Y(2)=.25*A                                           CYL.20
        ..                                                  ..
        ..                                                  ..
      TEST=Y(1)-1.                                         CYL.28
        ..                                                  ..
        ..                                                  ..
      END                                                  CYL.34


          b. UPDATE modification deck (MHBT46):
          -------------------------------------

*IDENT MOD46
*/ IMPROVED INITIAL DATA FOR Y.
*D CYL.19,20
      Y(1)=.25*A*E*(1.-.0625*A*E*E)                        MOD46.1
      Y(2)=.25*A*(1.-.1875*A*E*E)                          MOD46.2
*I CYL.28
      PRINT*,TEST                                          MOD46.3


          c. Modified code (UHBT46):
          --------------------------

*DECK CYL                                                  CYL.1
      SUBROUTINE CYL(A,TEST,IINT)                          CYL.2
        ..                                                  ..
        ..                                                  ..
      E=.00001                                             CYL.18
      Y(1)=.25*A*E*(1.-.0625*A*E*E)                        MOD46.1
      Y(2)=.25*A*(1.-.1875*A*E*E)                          MOD46.2
        ..                                                  ..
        ..                                                  ..
      TEST=Y(1)-1.                                         CYL.28
      PRINT*,TEST                                          MOD46.3
        ..                                                  ..
        ..                                                  ..
      END                                                  CYL.34
```

Fig. 3. Example of the use of UPDATE

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) preceeded 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 preceeding 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 uniques 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

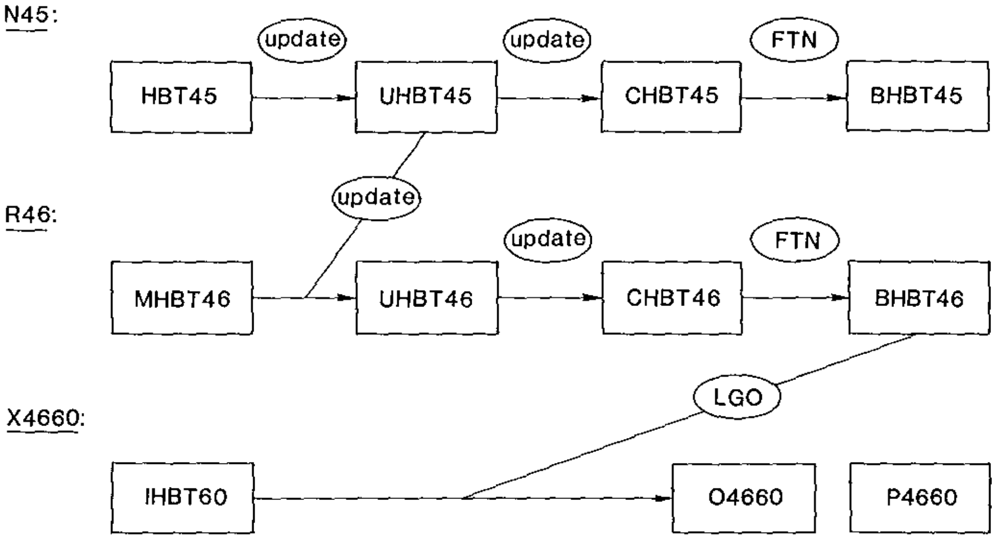


Fig. 4: Schematics of job control (procedures N, R, and X)

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 MFHBT 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 N45 is made. This run may be followed by a series of revisions R46, R46A, R46B, ..., where REVISE is the appropiate 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 (MFHBT), 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

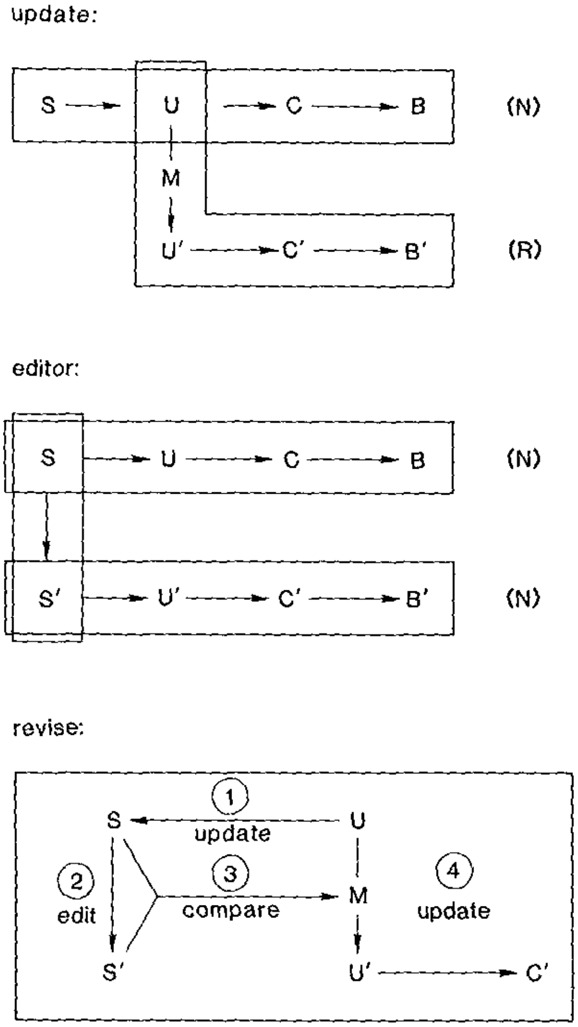


Fig. 5: Schematics of program control

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 abbrevations 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 → 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 COMPILE file C' (step ④). 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 loosing 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: page

Masterfile MFHBT:

Masterfile MFHGO:

Masterfile MFPPP:

Permanent files:

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.
            CALL: "NEWCCL,FN."
            FOLLOWING ALTERATIONS OF LIBRARY CCLLIB ARE EFFECTED WITH
            THE PROCEDURES ADDCCL, REPCCL, AND DELCCL.
  ADDCCL  - ADD FILE FN TO LIBRARY CCLLIB.
            CALL: "ADDCCL,FN,TYPE."
            TYPE MAY BE PROC, REL, ABS, OR PAS.
  REPCCL  - REPLACE OLD FILE FN BY NEW ONE IN LIBRARY CCLLIB.
            CALL: "REPCCL,FN,TYPE."
            TYPE MAY BE PROC, REL, ABS, OR PAS.
  DELCCL  - DELETE FILE FN FROM LIBRARY CCLLIB.
            CALL: "DELCCL,FN."
  COPYCCL - COPY MFCCL WITH MFCOPY AND LIBRARY CCLLIB WITH COPYLIB OF
            PIASLIB.
  LISTLIB - LIST CONTENTS OF LIBRARY LIB WITH LISTLIB OF PIASLIB USING
            PROGRAM RELIST.
            CALL: "LISTLIB,LIB(,PR)."
            LISTING APPEARS AT TERMINAL (UNLESS PR).
  RELIST  - PROGRAM REFORMATING OUTPUT OF LISTLIB OF PIASLIB INTO
            NUMBERED PAGES.
  LIBLIST - LIST ATTACHED LIBRARIES WITH LIBLIST OF PIASLIB.
  ZZFILES - LIST SYSTEM FILES STARTING WITH ZZZZZ BY CALLING ZZFILES OF
            PIASLIB.
  COST    - SHOW SYSTEM SECONDS USED SINCE LAST CALL OF COST BY CALLING
            COST OF PIASLIB.
            CALL: "COST.", OPERATIONS TO BE TIMED, "COST,R."
            THE TIMES ARE STORED IN SYSTEM FILE ZZZZZUC WHICH IS
            RETURNED BY THE PARAMETER R IN THE LAST CALL.
  EDCY    - EDIT WITH SARED ON THE CYBER INTERACTIVELY.
            CALL: "EDCY(,U)."
            THE PARAMETER U PREPARES A FILE B TO BE USED FOR REARRANGING
            LENGTHY COMMENT LINES WITH PROGRAM REAR.  WHEN EDITING, THE
            SET OF COMMANDS  "W,A,L1 L2", "DEL_L1 L2", "USE,B"  REPLACES
            THE COMMENT LINES L1-L2 WITH LINES‾OF 72 CHARACTERS.
  REAR    - PROGRAM REARRANGING LENGTHY COMMENT LINES TO 72 CHARACTERS.
            CALLED FROM FILE B, WHICH IS PRODUCED FOR USE IN AN
            INTERACTIVE SESSION OF SARED STARTED BY PROCEDURE EDCY.
```

```
EDDY    - EDIT WITH SARED ON THE CYBER INTERACTIVELY.
          THE EDITFILE 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,Q" DO NOT AFFECT THE
          CONTENTS OF EDFILE, UNLESS IT HAS BEEN CLEARED COMPLETELY
          WITH "DEL@".  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: LINENRS AFTER COLUMN 72) AND
          TRAILING BLANKS FROM A FILE.
          CALL: "STRIP,FN(,L=..)."
          THIS PROCEDURE IS USED, E.G., WHEN RECONSTRUCTING A SOURCE
          FILE FROM AN UPDATE PROGRAM LIBRARY (SEE PROCEDURE SU) IN
          ORDER TO CUT DOWN ON THE SIZE OF THE RESULTING FILE.
DCL     - DELETE COLUMNS I1 TO I2 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 I1 WILL BE DELETED.
NOTE    - NOTE "MESSAGE" ON FILE L (DEFAULT CONNECTED).
          CALL: "NOTE,$MESSAGE$(L=..,DISCON)."
          DEFAULT: "MESSAGE" APPEARS ON FILE OUTPUT (UNLESS L=..)
          WHICH IS CONNECTED (UNLESS DISCON).
COUNT   - COUNT COLUMN NUMBERS ON THE SCREEN.
CALC    - FORTRAN POCKET CALCULATOR.
          CALL: "CALC,$ANY FORTRAN EXPRESSION$."
DIFFER  - COMPARE FN1 AND FN2 WITH PROGRAM DIFF OF LIBRARY PROGS OF
          DICK WINTER (MC). DIFF IS WRITTEN IN A SPECIAL ASSEMBLER
          LANGUAGE THAT MAY NO LONGER BE SUPPORTED BY THE SYSTEM AT
          SOME FUTURE DATE.
          CALL: "DIFFER,FN1,FN2(,L=..)."
          DEFAULT: LISTING APPEARS ON CONNECTED FILE OUTPUT
          (UNLESS L=..).
RUN5    - FTN5 COMPILATION AND EXECUTION OF FILE FN.
          CALL: "RUN5,FN(,B=..,OPT=..,L=..,KEEP,PMD,NOEX)."
          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
          PMD), 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(,$LEFT TEXT$,$RIGHT TEXT$,N1=..,N2=..)."
          $LEFT TEXT$ AND $RIGHT TEXT$ 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/..."
UFOUT   - WRITE UPDATE SOURCES FROM ATTACHED MASTERFILE ON OUTPUT.
          CALL: "UFOUT,FN1/FN2/..."
ROUT    - ROUTE FILE OUTPUT TO LINEPRINTER.
          CALL: "ROUT(,OUTPUT)(,SHIFT,TID,FID,IC)."
          DEFAULT: FILE IS OUTPUT (UNLESS "OUTPUT" IS SPECIFIED
          DIFFERENTLY) WITH CONTROL CHARACTERS IN COLUMN1 (UNLESS
          SHIFT) AND IS WRIITEN IN DISPLAY CODE (UNLESS IC).
          BANNER IS LABELED WITH FID=XXIDX/XXI2X.
RIN     - ROUTE JOB TO INPUT QUEUE.
          CALL: "RIN,JOB(,TID=..,FID=..)."
          OUTPUT APPEARS AT TERMINAL (UNLESS TID).
LOC     - MAKE REMOTE OUTPUT FILE ZZ LOCAL UNDER THE NAME Z_ZZ,
          PAGE, AND ROUTE TO THE LINEPRINTER (WHEN GIVEN PERMISSSION).
          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=NID.
          CALL: "DU,FN,CY=..(,NID=..)."
          DEFAULT: NID=XXXXX.
NEW     - PRODUCE JOB NFN_S (OR NFN_M) FROM FILE NFN OF MFFN.
```

```
          THIS JOB PRODUCES NEW LIBRARY FNLIB FROM THE SOURCE FN_S
          (AND MODIFICATION MFN_M ,ALSO OF MFFN).
          CALL: "NEW,FN,S=..(,M=..,NOUL,NOR,TID=..)."
          DEFAULT: NO MODIFICATION MFN_M (UNLESS M),
          COMPLETE SOURCE LISTING (UNLESS NOUL),
          JOB NFN_S ROUTED TO INPUT QUEUE (UNLESS NOR),
          OUTPUT APPEARS AT TERMINAL (UNLESS TID=).
N       - PRODUCE JOB N_S FROM FILE NFN RESIDING IN MFFN.
          THIS JOB PRODUCES NEW UPDATE PL UFN_U FROM THE SOURCE FN_S
          AND COMPILES BFN_B.
          CALL: "N,FN,S=..(,U=..,B=..,NOUL,NOR,TID=..)."
          DEFAULT: S=U=B, COMPLETE SOURCE LISTING (UNLESS NOUL).
          PARAMETERS NOR,TID: SEE UNDER PROCEDURE NEW.
R       - PRODUCE JOB R_M FROM FILE RFN RESIDING IN MFFN.
          THIS JOB REVISES OLD UPDATE PL UFN_U WITH MODIFICATION
          MFN_M TO GET NEW UPDATE PL UFN_V, AND COMPILES BFN_B.
          CALL: "R,FN,U=..,M=..(,V=..,B=..,ULIST,NOR,TID=..)."
          DEFAULT: V=B=M, NO LISTING MODIFICATIONS (UNLESS ULIST).
          PARAMETERS NOR,TID: SEE UNDER PROCEDURE NEW.
X       - PROCEDURE CREATING JOB X_B_I FROM FILE XFN RESIDING IN
          MFFN. THIS JOB EXECUTES BFN_B WITH INPUT IFN_I.
          CALL:
          "X,FN,B=..,I=..(,P,O,D,T=..,IO=..,LP,NOR,TID=..).",
          P   - PLOTFILE P_B_I IS CATALOGED.
          O   - OUTPUT O_B_I IS CATALOGED.
          D   - DATA FILE D_B_I IS CATALOGED.
          PARAMETERS NOR,TID: SEE UNDER PROCEDURE NEW.
JOBCRD  - PROGRAM REWRITING JOBCARD OF JOB ON FILE FN. THIS PROGRAM
          IS CALLED FROM PROCEDURE X.
          CALL: "JOBCRD,FN,T=..,IO=..,LP=NP."
PLOUT   - CONVERT LOCAL GRAPHFILE "LFN" 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)."
          NOVERSA  - FILE NOT PLOTTED ON THE VERSATEC.
          NODELETE - COPY OF THE FILE KEPT ON THE 11/70.
SPLIT   - 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.
TOUT    - SEND LOCAL TEXT FILE "LFN" TO DESTINATION XXA (11/70).
          CALL: "TOUT,LFN(,NOVERSA,NODELETE)."
          NOVERSA  - FILE NOT PRINTED ON THE VERSATEC.
          NODELETE - COPY OF THE FILE KEPT ON THE 11/70.
DT      - PRINT DATE AND TIME.
SYS     - PROGRAM WRITING LOCAL FILE ZZSYS WITH PROCEDURE OF THE
          SAME NAME, SHOWING UPON CALLING WHICH FILES OF SYSBULL
          AND USERBUL CHANGED OVER THE PAST 7 DAYS.
          CALL: "SYS.", "ZZSYS." (SEE PROCEDURE INIT IN PROCFIL).
ZZSYS1  - PROCEDURE CALLING ZZSYS2 WITH THE DATE OF THE PREVIOUS
          SESSION. (THIS FILE HAS NO COUNTERPART IN CCLLIB).
ZZSYS2  - PROCEDURE CALLED FROM PROCEDURE ZZSYS1 WHICH IN TURN
          IS CALLED FROM PROCEDURE INIT ON PROCFIL TO DISPLAY THE
          FILES OF SYSBULL AND C205BUL THAT HAVE CHANGED SINCE THE
          LAST INTERACTIVE SESSION.  THIS PROCEDURE ALSO UPDATES
          ZZSYS1 WITH THE PRESENT DATE FOR USE NEXT TIME.  THE
          LATTER DATE RESIDES IN THE R1 REGISTER WHICH IS ASSIGNED
          TO THE D-PARAMETER OF SYSBULL, SO THAT A CALL TO SYSBULL
          CHANGES THE CONTENT OF R1 FROM THE PREVIOUS TO THE
          PRESENT DATE.
AUD     - COMPACT AUDIT OF ID=... (DEFAULT: AT TERMINAL) BY
          CALLING PROGRAM PASAUD, AND DIRECTORY OF MASTERFILES BY
          CALLING PROGRAM DIR IF LO IS SPECIFIED.
          IF "PF=.." IS SPECIFIED THE DATA OF ONE PERMFILE ARE SHOWN.
          CALL: "AUD(,PR,LO,PF=..,ID=..)."
DIR     - PROGRAM CONSTRUCTING DIRECTORY OF THE MASTERFILES SHOWN
          IN AUDIT OF ID=....
          FIRST CALL "AUDIT,ID=..,LF=DATA."., THEN "DIR,ID=...".
          THIS PROGRAM ONLY WORKS FOR MASTERFILES HAVING NAMES
          OF NO MORE THAN 10 CHARACTERS AND STARTING WITH "MF".
PASAUD  - PASCAL3 PROGRAM, WRITTEN BY HANS SCHRIJVER, REFORMATING
          OUTPUT OF SYSTEM AUDIT.
ATT750  - ATTACHES AND RETURNS ALL PERMFILES 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,AI=P,LF=ZZAUD,ID=.."
          AND WRITES THE PERMFILES FOUND ON A PROCEDURE FILE 'ZZATT',
          WHICH WILL ATTACH (AND SUBSEQUENTLY RETURN) ALL THESE FILES.
          CALL: "ATT750(,?,ID=..).".
          DEFAULT: ID=XXIDX.
```

```
 ADDP    - ADD PROCEDURE TO LIBRARY LIB (DEFAULT: LIB="LIBRARY")         2090
           WHICH SHOULD BE ATTACHED WITH FULL PERMISSION.                2100
           IF NECESSARY DO: "RETURN,LIB.", "ATTACH,LIB,ID=XXIDX."        2110
           CALL: "ADDP(,?),FILENAME,LIBRARY NAME."                       2120
 DELP    - DELETE PROCEDURE FROM LIBRARY LIB (DEF.: LIB="LIBRARY")       2130
           WHICH SHOULD BE ATTACHED WITH FULL PERMISSION.                2140
           IF NECESSARY DO: "RETURN,LIB.", "ATTACH,LIB,ID=XXIDX."        2150
           CALL: "DELP(,?),FILENAME,LIBRARY NAME."                       2160
 GETP    - GET PROCEDURE FROM LIBRARY LIB (DEFAULT: LIB="LIBRARY").      2170
           CALL: "GETP(,?),FILENAME,LIBRARY NAME."                       2180
 REPP    - REPLACE PROCEDURE IN LIBRARY LIB (DEF.: LIB="LIBRARY")        2190
           WHICH SHOULD BE ATTACHED WITH FULL PERMISSION.                2200
           IF NECESSARY DO: "RETURN,LIB.", "ATTACH,LIB,ID=XXIDX."        2210
           CALL: "REPP(,?),FILENAME,LIBRARY NAME."                       2220
 ADD205  - ADD PERMFILE TO THE 205.                                      2230
           CALL: "ADD205(,?),FILENAME(,CODE)."                           2240
 DEL205  - DELETE PERMFILE(S) FROM THE 205.                              2250
           CALL: "DEL205(,?),FILENAME(,FN2,FN3,FN4,FN5)."                2260
 GET205  - GET PERMFILE FROM THE 205.                                    2270
           CALL: "GET205(,?),FILENAME(,CODE)."                           2280
 RNM205  - RENAME PERMFILE ON THE 205.                                   2290
           CALL: "RNM205(,?),OLD,NEW."                                   2300
 AUD205  - AUDIT OF PERMFILES ON THE 205.                                2310
           CALL: "AUD205(,?,LO,OUT)."                                    2320
 ATT205  - ATTACH ALL PERMFILES ON THE 205, AUDIT, AND RUN PROGRAM SAVE  2330
           TO RESET THE DATE OF LAST ACCESS TO TODAY.                    2340
           CALL: "ATT205(,?,U,AC,PA,TID,FID)."                           2350
           DEFAULTS: U=XXU1XX,AC=XXXACXXX,PA=XPAX,TID=XXB,FID=XXI2X.     2360
 BUD205  - GIVES THE 205-BUDGET LEFT FOR U=..(DEFAULT: U=XXU1XX).        2370
           CALL: "BUD205(,?,U=...)."                                     2380
 PER205  - GRANTS PERMISSION TO USER TO ACCESS U=XXU1XX PERMFILE ON      2390
           THE 205.  CALL: "PER205(,?),PFN(,USER=..,AC=..)."             2400
 Q205    - SHOWS THE QUEUES ON THE 205 FOR U=..(DEFAULT: U=XXU1XX).      2410
           CALL: "Q205(,?,U=...)."                                       2420
 RIN205  - ROUTE JOB TO INPUT QUEUE OF THE 205, WHERE AN INPUT RECORD    2430
           "IN1" AND ALSO A SECOND RECORD "IN2" MAY BE INCLUDED.         2440
           CALL: "RIN205(,?),JOB(,IN,INP,TID,FID)."                      2450
 PLT205  - CONVERTS THE BINARY PLOTFILE "NAME" FROM THE 205 TO A         2460
           GRAPHFILE TO BE VISUALIZED WITH GRIMAS.                       2470
           CALL: "PLT205(,?),NAME."                                      2480
 NNEW    - PRODUCES NEWPL UFN_S FROM THE SOURCE FN_S (OF MFFN),          2490
           OR UFN_M FROM UFN_S + MODIFICATION DECK MFN_M (OF MFFN), AND  2500
           CREATES A JOB NN_FN_S (OR NN_FN_M) FROM FILE NNFN (OF MFFN)   2510
           WHICH COMPILES LIBRARY FNLIB ON THE 205.                      2520
           CALL: "NNEW(,?),FN,S=..(,M=..,NOUL,NOCAT,FLIST,NOR).",        2530
           FN     - ROOT FILE NAME OF THE PROGRAM                        2540
           S      - IDENTIFICATION OF SOURCE FILE FN_S                   2550
           M      - NEWPL IS MODIFIED WITH MFN_M (FROM MFFN)             2560
           NOUL   - NO UPDATE LISTING OF THE SOURCE IS MADE              2570
           NOCAT - NEWPL IS NOT CATALOGED ON THE 750                     2580
           FLIST - A FTN200 LISTING IS MADE                              2590
           NOR    - JOB NN_FN_S IS CREATED BUT NOT SUBMITTED.            2600
 NN      - PRODUCES NEWPL UFN_U FROM THE SOURCE FN_S (OF MFFN),          2610
           AND CREATES A JOB NN_S FROM THE FILE NNFN (OF MFFN),          2620
           WHICH COMPILES BFN_B ON THE 205.                              2630
           CALL: "NN(,?),FN,S=..(,U=..,B=..,NOUL,NOCAT,FLIST,NOR).",     2640
           FN     - ROOT FILE NAME OF THE PROGRAM                        2650
           S      - IDENTIFICATION OF SOURCE FILE FN_S                   2660
           U      - IDENTIFICATION OF NEWPL UFN_U (ON 750)               2670
           B      - IDENTIFICATION OF BINARY BFN_B (ON 205)              2680
           NOUL   - NO UPDATE LISTING OF THE SOURCE IS MADE              2690
           NOCAT - NEWPL IS NOT CATALOGED ON THE 750                     2700
           FLIST - A FTN200 LISTING IS MADE                              2710
           NOR    - JOB NN_S IS CREATED BUT NOT SUBMITTED.               2720
 RR      - REVISES OLDPL UFN_U WITH MODIFICATION DECK MFN_M (OF MFFN)    2730
           TO GET NEWPL UFN_V AND CREATES A JOB RR_M FROM THE            2740
           FILE RR_M (OF MFFN) WHICH COMPILES BFN_B ON THE 205.          2750
           CALL:                                                         2760
           "RR(,?),FN,U=..,M=..(,V=..,B=..,ULIST,NOCAT,FLIST,NOR).",     2770
           FN     - ROOT FILE NAME OF THE PROGRAM                        2780
           U      - IDENTIFICATION OF OLDPL UFN_U (ON 750)               2790
           M      - IDENTIFICATION OF MODIFICATION MFN_M (FROM MFFN)     2800
           V      - IDENTIFICATION OF NEWPL UFN_V (ON 750)               2810
           B      - IDENTIFICATION OF BINARY BFN_B (ON 205)              2820
           ULIST - UPDATE LISTING OF THE CHANGES IS MADE                 2830
           NOCAT - NEWPL IS NOT CATALOGED                                2840
           FLIST - A FTN200 LISTING IS MADE                              2850
           NOR    - JOB RR_M IS CREATED BUT NOT SUBMITTED.               2860
 XX      - PROCEDURE CREATING JOB XX_B_I FROM FILE XXFN RESIDING IN      2870
           MFFN. THIS JOB EXECUTES BFN_B WITH INPUT IFN_I.               2880
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- COMMENT,  5 ------
               CALL: "XX(,?),FN,B=..,I=..(,P,O,D,TL=..,WS=..,LP=,NOR).",
               B    - IDENTIFICATION OF BINARY BFN_B (ON THE 205)
               I    - IDENTIFICATION OF INPUT FILE IFN_I (OF MFFN)
               P    - PLOT FILE P_B_I IS CATALOGED
               O    - OUTPUT O_B_I IS CATALOGED
               D    - DATA FILE D_B_I IS CATALOGED.
               NOR - JOB XX_B_I IS CREATED BUT NOT SUBMITTED.
      NU      - PRODUCES NEWPL U AND COMPILE FILE C FROM THE SOURCE S.
               CALL: "NU(,?),S(,U,C,NOUL,FID).",
               S       - SOURCE FILE
               U       - NEW UPDATE PROGRAM LIBRARY; DEFAULT: "NEWPL"
               C       - COMPILE FILE; DEFAULT:"COMPILE"; SUPPRESS:"C=0"
               NOUL   - NO UPDATE LISTING OF THE SOURCE IS MADE
               FID    - FID FOR UPDATE OUTPUT.
      RU      - REVISES OLDPL U WITH MODIFICATION DECK M TO PRODUCE NEWPL V
               AND COMPILE FILE C.
               CALL: "RU(,?),U,M=..(,V,C,ULIST,FID).",
               U       - OLD UPDATE PROGRAM LIBRARY
               M       - MODIFICATION DECK; "M=0" GIVES COMPILE OF OLDPL
               V       - NEW UPDATE PL; DEFAULT: "NEWPL"; SUPPRESS: "V=0"
               C       - COMPILE FILE; DEFAULT:"COMPILE"; SUPPRESS:"C=0"
               ULIST - UPDATE LISTING OF THE CHANGES IS MADE
               FID    - FID FOR UPDATE OUTPUT.
      SU      - RETRIEVES THE SOURCE S FROM AN OLD UPDATE PL U.
               CALL: "SU(,?),U,S=..(,NOUL,FID).",
               U     - OLD UPDATE PROGRAM LIBRARY
               S     - SOURCE FILE
               NOUL - NO UPDATE LISTING IS MADE
               FID  - FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".
      INSTAL  - INSTALS A LIBRARY FOR UPDATE PROCEDURES AND PROGRAMS WRITTEN
               BY JOS KOOT AND AMPLIFIED BY HANS GOEDBLOED.
               IT CONTAINS THE CCL PROCEDURES REVISE/ASKDOE AND THE FORTRAN
               PROGRAMS ASKDECK/MAKEFIL/MODGEN/USL/UML.
               CALL: "INSTAL(,LIB,PRCFILE).",
               LIB      - LFN OF THE LIBRARY; DEFAULT: "LIBRARY"
               PRCFILE - INSTRUCTS INSTAL TO READ FILES TO BE INSTALLED
                         FROM FILE INSTAL ITSELF.  NO OPTIONS FOR THE USER.
               TO USE INSTAL (WHICH IS SUPERFLUOUS HERE SINCE THE LIBRARY
               CCLLIB ALREADY CONTAINS THE FILES INSTALLED BY INSTAL),
               INSERT:
               AFTER 1ST EOR: PROCEDURE REVISE
               AFTER 2ND EOR: PROCEDURE ASKDOE
               AFTER 3RD EOR: FORTRAN PROGRAMS ASKDECK/MAKEFIL/MODGEN/USL/
                              UML (NOT SEPARATED BY COMMENTS OR EOR'S!).
      REVISE  - 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 COMPARED 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(,?),U(,D,M,CI,S,T,V,C,UL1,UL2).",
               U    - LFN OF THE OLDPL
               D    - DECKS TO BE UPDATED; DEFAULTS: $.ALL.$/$.ASK.$
               M    - MODIFICATION DECK; DEFAULT: "MODFILE"
               CI   - 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: 0/NEWPL
               C    - COMPILE FILE; DEFAULTS: 0/COMPILE
               UL1 - 1ST UPDATE LISTING (RETRIEVAL OF DECKS); DEFAULT: 0
               UL2 - 2ND UPDATE LISTING (CORRECTION RUN); DEFAULTS: 0/1.
      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.
               TYPICAL USE: ".IF(FILE(ZZZZZ1Z,AS)) ASKDOE."
      ASKDECK - PREPARES INPUT FILE FOR UPDATE RUN WHICH RETRIEVES SPECIFIED
               DECKS FROM OLDPL. THE DECKNAMES ARE OBTAINED INTERACTIVELY.
               CALLED FROM REVISE: "ASKDECK,INPFILE.".
      MAKEFIL - READS OUTPUT LSTFILE OF AN UPDATE RUN (WITH LIST OPTION L=7,
               WHICH HAS RETRIEVED SPECIFIED DECKS FROM OLDPL) AND PRODUCES
               TWO FILES: WITHSEQ AND WITHOUT. WITHSEQ CONTAINS LINE IMAGE
               AND SEQUENCE INFORMATION. WITHOUT CONTAINS LINE IMAGES ONLY.
               UNWANTED COMMON DECKS OBTAINED BY SELECTIVE UPDATE MODE ARE
               REMOVED BY COMPARING THE CONTENTS OF LSTFILE WITH THE DECK
               LIST REQUESTED ON THE UPDATE INPUT FILE INPFILE.
               CALLED FROM REVISE:
               "MAKEFIL,LSTFIL,WITHSEQ,WITHOUT(,INPFILE).".
      MODGEN  - COMPARES TWO FILES (NEWFILE AND OLDFILE) AND PREPARES AN
               UPDATE CORRECTION SET IN MODFILE.
               CALLED FROM REVISE: "MODGEN,OLDFILE,NEWFILE,MODFILE,CI.".
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.10.44. ------- COMMENT,  6 ------

  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=A124,O=FN.",
                OR "UPDATE,F,P=OLDPL,L=7,O=FN.",
             CALL: "USL,FN(,NOLIST)."
             THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF
             CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL.
  UML      - PROGRAM REFORMATING OUTPUT OF UPDATE MODIFICATIONS LISTING
             OF A CORRECTION RUN.
             AFTER "UPDATE,F,P=OLDPL,I=MOD,N,L=A1234,O=FN.",
             CALL: "UML,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:
             "RUN205(,?),NAME(,B,G,NOEX,I,P,OPT,UNS,L,LO,TL,WS,LP,NOR)",
             NAME - NAME OF THE FORTRAN PROGRAM
             B    - NAME OF THE BINARY TO BE CREATED OR ATTACHED
             G    - NAME OF THE GOFILE TO BE CREATED 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 (DPRSV)
             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.
  VAST205  - TRANSLATE PROGRAM NAME WITH VAST ON THE 205 AND CATALOG
             OUTPUT ON THE 750.
             CALL: "VAST(,?),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(,M=MF1),ID=XXIDX."
             "MFUSE,PFN2,M=SECOND,ID=......"
             "ALIAS(,?),FLIST(,MF1,MF2,ID,ACC,UN,I2,AC,U1,U2,PA,TA,TB).",
             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$")
             UN    - CY750 LOGIN NAME     (DEFAULT: "$XXUNXX$")
             I2    - CY205 FILE ID        (DEFAULT: "$XXI2X$")
             AC    - CY205 ACCOUNT NR     (DEFAULT: "$XXXACXXX$")
             U1    - CY205 USER NR 1      (DEFAULT: "$XXU1XX$")
             U2    - CY205 USER NR 2      (DEFAULT: "$XXU2XX$")
             PA    - CY205 PASSWORD       (DEFAULT: "$XPAX$")
             TA    - TID LOCAL COMPUTER   (DEFAULT: "$XXA$")
             TB    - TID LINE PRINTER     (DEFAULT: "$XXB$").
********
```

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

.PROC,NEWCCL,FN.
.* CREATING NEW LIBRARY CCLLIB, ADDING A FIRST FILE FN
.* WHICH SHOULD BE A CCL PROCEDURE.
.* FOLLOWING ALTERATIONS OF LIBRARY CCLLIB ARE EFFECTED
.* WITH THE PROCEDURES ADDCCL, REPCCL, AND DELCCL.
MFUSE,MFCCL,ID=XXIDX.
FADD,FN.
COMMENT.** FN ADDED TO MFCCL **
LIBRARY.
RETURN,CCLLIB.
REQUEST,LIB,*PF.
CONNECT,OUTPUT.
EDITLIB,I=ZZINP.
CATALOG,LIB,CCLLIB,ID=XXIDX.
RETURN,ZZINP,OUTPUT,LIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
```

```
--- 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. -------- ADDCCL,  1 ------

.PROC,ADDCCL,FN,TYPE.
.* ADD 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=XXIDX.
FADD,FN.
NOTE,$FN ADDED TO MFCCL$.
RETURN,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX.
CONNECT,OUTPUT.
.IF,$TYPE$=$PROC$, PROCEDURE.
EDITLIB,I=ZZPROC.
.ENDIF, PROCEDURE.
.IF,$TYPE$=$REL$, RELOCATABL.
FTN5,I=FN,B=REL,L=0,OPT=2,PL=10000.
EDITLIB,I=ZZREL.
.ENDIF, RELOCATABL.
.IF,$TYPE$=$ABS$.OR.$TYPE$=$PAS$, ABSOLUTE.
.IF($TYPE$=$ABS$) FTN5,I=FN,B=REL,L=0,OPT=2,PL=10000.
.IF($TYPE$=$PAS$) PAS3,FN,,REL,L-.
LOAD,REL.
NOGO,ABS.
EDITLIB,I=ZZABS.
.ENDIF, ABSOLUTE.
EXTEND,CCLLIB.
RETURN,REL,ABS,ZZPROC,ZZREL,ZZABS,OUTPUT,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: TYPE SHOULD BE PROC/REL/ABS/PAS$.
RETURN,ZZPROC,ZZREL,ZZABS.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZPROC,ZZREL,ZZABS,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
REVERT,ABORT.
.*
.DATA,ZZPROC.
LIBRARY(CCLLIB,OLD)
REWIND(FN)
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 ------

.PROC,REPCCL,FN,TYPE.
.* REPLACE FILE FN IN LIBRARY CCLLIB (TYPE: PROC/REL/ABS/PAS).
.IF, $TYPE$=$PROC$.OR.$TYPE$=$REL$.OR.$TYPE$=$ABS$.OR.$TYPE$=$PAS$, OK.
MFUSE,MFCCL,ID=XXIDX.
FREP,FN.
NOTE,$FN REPLACED IN MFCCL$.
RETURN,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX.
CONNECT,OUTPUT.
.IF,$TYPE$=$PROC$, PROCEDURE.
EDITLIB,I=ZZPROC.
.ENDIF, PROCEDURE.
.IF,$TYPE$=$REL$, RELOCATABL.
FTN5,I=FN,B=REL,L=0,OPT=2,PL=10000.
EDITLIB,I=ZZREL.
.ENDIF, RELOCATABL.
.IF,$TYPE$=$ABS$.OR.$TYPE$=$PAS$, ABSOLUTE.
.IF($TYPE$=$ABS$) FTN5,I=FN,B=REL,L=0,OPT=2,PL=10000.
.IF($TYPE$=$PAS$) PAS3,FN,,REL,L-.
LOAD,REL.
NOGO,ABS.
EDITLIB,I=ZZABS.
.ENDIF, ABSOLUTE.
EXTEND,CCLLIB.
RETURN,REL,ABS,ZZPROC,ZZREL,ZZABS,OUTPUT,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: TYPE SHOULD BE PROC/REL/ABS/PAS$.
RETURN,ZZPROC,ZZREL,ZZABS.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZPROC,ZZREL,ZZABS,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
REVERT,ABORT.
.*
.DATA,ZZPROC.
LIBRARY(CCLLIB,OLD)
REWIND(FN)
REPLACE(*,FN)
FINISH.
ENDRUN.
.*
.DATA,ZZREL.
LIBRARY(CCLLIB,OLD)
REWIND(REL)
REPLACE(*,REL)
SETAL(FN,1)
FINISH.
ENDRUN.
.*
.DATA,ZZABS.
LIBRARY(CCLLIB,OLD)
REWIND(ABS)
REPLACE(*,ABS,AL=1)
FINISH.
ENDRUN.


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

.PROC,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.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
```

```
--- 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
.* PROCEDURE COPYLIB OF PIASLIB.
MFUSE.
MFUSE,MFCCL,M=OLD,ID=XXIDX,MR=1.
MFNEW,MFCCL,M=NEW,ID=XXIDX.
FGET,COMMENT,M=OLD.
UPCOM.
FADD,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 ATTACHED **
.*
RETURN,CCLLIB.
ATTACH,CCLLIB,ID=XXIDX.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LIBLOAD(PIASLIB,COPYLIB)
EXECUTE(COPYLIB,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 ATTACHED **
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
REVERT,ABORT.
```

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

.PROC,LISTLIB,LIB,PR=N/Y.
.* LIST CONTENTS OF LIBRARY LIB WITH LISTLIB OF PIASLIB.
RETURN,OUTPUT.
ATTACH,PIASLIB,ID=PIASS,SN=S.
LIBLOAD(PIASLIB,LISTLIB)
.IF,$PR$=$N$, PRNY.
EXECUTE(LISTLIB,LIB)
.ELSE, PRNY.
EXECUTE(LISTLIB,LIB,OUTPUT)
RELIST.
ROUT.
.ENDIF, 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,TAPE6=OUTPUT)
C     ******************************************************************
C     * PROGRAM REFORMATING OUTPUT OF PROGRAM LISTLIB OF PIASLIB INTO  *
C     * NUMBERED PAGES.                                                *
C     ******************************************************************
C
      CHARACTER*73 HEADER,LINE
C
```

```
--- 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.'0') 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(3:30),HEADER(31:40),HEADER(41:50),IP
      IL1=(IP-1)*ILMAX+1
      IL2=IL1+ILMAX-1
      DO 30 IL=IL1,IL2
      READ(7,1,ERR=100,END=200) LINE
      WRITE(6,1) LINE
   30 CONTINUE
C
  100 STOP 'ERROR'
  200 CLOSE(7,STATUS='DELETE')
      STOP 'PROGRAM RELIST'
C
    1 FORMAT(A)
    2 FORMAT()
    3 FORMAT('1',A28,3X,A10,1X,A10,13X,'PAGE',I3/)
      END
```

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

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

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

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

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

.PROC,COST,R=N/Y.
.* 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($R$=$Y$) RETURN,ZZZZZUC.
RETURN,PIASLIB.
REVERT.
```

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

.PROC,EDCY,U=N/B.
.* EDIT WITH SARED ON THE CYBER INTERACTIVELY.
.* CALL: "EDCY(,U).".
.* PARAMETER U PREPARES FILE B TO BE USED FOR REARRANGING
.* COMMENT LINES WITH PROGRAM REAR. IN SARED, THE SET OF
.* COMMANDS  "W,A,L1 L2", "DELL1 L2", "USE,B"  REPLACES THE
.* COMMENT LINES L1-L2 WITH LINES OF 72 CHARACTERS.
```

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

.IF,$U$=$B$, USEB.
RETURN,B.
COPY,ZZB,B.
REWIND,B.
.ENDIF, USEB.
RETURN,ZZB.
ED.
REVERT.
.*
.DATA,ZZB.
!RETURN,C
!REAR
I,C
!RETURN,A
!REWIND,B
```

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

      PROGRAM REAR
C     ***************************************************************
C     * PROGRAM REARRANGING LENGTHY COMMENT LINES TO FIT LINES OF 72  *
C     * CHARACTERS.                                                   *
C     * THIS PROGRAM IS CALLED FROM FILE B, WHICH IS PRODUCED FOR USE *
C     * IN AN INTERACTIVE SESSION OF SARED STARTED BY PROCEDURE EDCY. *
C     ***************************************************************
C
      CHARACTER*85 LINE,BLANK
      CHARACTER*1000 STORE
      DATA BLANK/' '/
C
      OPEN(10,FILE='A')
      OPEN(20,FILE='C')
C
      K=1
      DO 30 I=1,100
      READ(10,1,END=40,ERR=100) LINE
      DO 10 J=85,1,-1
   10 IF(LINE(J:J).NE.BLANK) GOTO 20
   20 STORE(K:K+J-2)=LINE(2:J)
      K=K+J-1
   30 CONTINUE
C
   40 STORE(K:K)=BLANK
      N=1
      DO 70 L=1,100
      M1=N+69
      IF(K.LE.M1) M1=K
      DO 50 M=M1,N,-1
   50 IF(STORE(M:M).EQ.BLANK) GOTO 60
   60 WRITE(20,2,ERR=100) 'C '//STORE(N:M)//BLANK
      N=M+1
      IF(N.GE.K) GOTO 80
   70 CONTINUE
C
   80 STOP 'PROGRAM REAR'
  100 STOP 'ERROR'
C
    1 FORMAT(A)
    2 FORMAT(A71,'*')
      END
```

```
--- 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.
```

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

.PROC,FED,FILE.
.* PRODUCING FORMAT FILE FOR SARED.
.* FILE=FF: FORMAT FOR EDITING FORTRAN FILES.
.* FILE=FN: FORMAT FOR EDITING OTHER FILES.
.IF, $FILE$=$FF$.OR.$FILE$=$FN$, OK.
.IF, $FILE$.EQ.$FF$, FORTRAN.
COPY,ZZFF,FF.
REWIND,FF.
.ENDIF, FORTRAN.
.IF, $FILE$.EQ.$FN$, COMMENT.
COPY,ZZFN,FN.
REWIND,FN.
.ENDIF, COMMENT.
RETURN,ZZFF,ZZFN.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: SPECIFY "FED,FF" OR "FED,FN"$.
RETURN,ZZFF,ZZFN.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZFF,ZZFN.
REVERT,ABORT.
.*
.DATA,ZZFF.
FORMAT \\ FORTRAN
SET COUNT=1
SET LINE=19
SET EXP=1
SHOW
FORMAT SHOW
!REWIND FF
.*
.DATA,ZZFN.
FORMAT \\3 11 13 NO
SET COUNT=1
SET LINE=19
SET EXP=1
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: LINENRS AFTER COLUMN 72)
.* AND TRAILING BLANKS FROM A FILE.
ED,USE,INSTR.
COMMENT.** FN STRIPPED (L CHARS) **
RETURN,INSTR,EDLOG.
REVERT.
.*
EXIT,S.
COMMENT.** ERROR **
REVERT,ABORT.
.*
.DATA,INSTR.
E,FN
F,#L=L
W,FN,O
E,FN
W,FN,O
SC,INIT
B,Q
```

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

      PROGRAM DCL
C     ****************************************************************
C     * DELETE COLUMNS I1 TO I2 OF FILE FN1 AND WRITE THE RESULTING    *
C     * FILE ON FN2.                                                   *
C     * CALL: "DCL,FN1,FN2,I1(,I2)."                                   *
C     * IF FN1=FN2 THE ORIGINAL FILE FN1 IS OVERWRITTEN WITH FN2.      *
C     * IF I2 IS OMITTED ONLY COLUMN I1 WILL BE DELETED.               *
C     ****************************************************************
C
```

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

      IMPLICIT INTEGER(A-Z)
      PARAMETER(MXLINES=10000)
C
      CHARACTER*133 LINE
      CHARACTER*  7 FN1,FN2
      CHARACTER*  5 C1,C2
C
      CALL GETPARM(FN1,DUM,PARSTAT)
      CALL GETPARM(FN2,DUM,PARSTAT)
      IF(FN2.EQ.FN1) FN2='DUMMY'
      CALL GETPARM(C1,DUM,PARSTAT)
      CALL GETPARM(C2,DUM,PARSTAT)
      IF(PARSTAT.LT.0) C2=C1
      OPEN(10,FILE=FN1)
      OPEN(20,FILE=FN2)
      REWIND 10
      REWIND 20
C
      READ(C1,'(I5)') I1
      READ(C2,'(I5)') I2
      IF(I1.EQ.1) THEN
         DO 10 L=1,MXLINES
         READ(10,'(A)',ERR=100,END=30) LINE
         WRITE(20,'(A)') LINE(I2+1:133)
   10    CONTINUE
      ELSE
         DO 20 L=1,MXLINES
         READ(10,'(A)',ERR=100,END=30) LINE
         WRITE(20,'(A)') LINE(1:I1-1)//LINE(I2+1:133)
   20    CONTINUE
      ENDIF
C
   30 IF(FN2.EQ.'DUMMY') THEN
         REWIND(10)
         REWIND(20)
         DO 40 L=1,MXLINES
         READ(20,'(A)',ERR=100,END=50) LINE
         WRITE(10,'(A)') LINE
   40    CONTINUE
   50    CLOSE(20,STATUS='DELETE')
      ENDIF
      STOP 'PROGRAM DCL'
C
  100 STOP 'ERROR'
      END
```

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

.PROC,NOTE,MESSAGE,L=ZZNOTE,DISCON=N/Y.
.* NOTE "MESSAGE" ON FILE L (DEFAULT CONNECTED).
.* CALLING STATEMENT: NOTE,$MESSAGE$.
.* "MESSAGE" SHOULD NOT EXCEED 40 CHARACTERS.
.IF($DISCON$=$N$) CONNECT,L.
.IF($DISCON$=$Y$) DISCONT,L.
COPY,ZZN,L.
.IF($DISCON$=$N$) RETURN,L.
RETURN,ZZN.
REVERT.
.*
.DATA,ZZN.
 ** MESSAGE **
```

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

.PROC,COUNT,LEFT=N/Y.
.* COUNT COLUMN NUMBERS ON THE SCREEN.
CONNECT,ZZCOUNT.
.IF,$LEFT$=$N$, LEFTNY.
REWIND,ZZC1.
COPY,ZZC1,ZZCOUNT.
.ELSE, LEFTNY.
REWIND,ZZC2.
COPY,ZZC2,ZZCOUNT.
.ENDIF, LEFTNY.
RETURN,ZZCOUNT,ZZC1,ZZC2.
REVERT.
.*
.DATA,ZZC1.
```

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

                 *  1         2         3         4         5         6
           1234567890123456789012345678901234567890123456789012345678901234
.*
.DATA,ZZC2.
            *  1         2         3         4         5         6         7
 12345678901234567890123456789012345678901234567890123456789012345678901
```

```
--- 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.
DFMLOCK,ON.
CONNECT,OUTPUT.
REWIND,ZZCAL.
FTN5,I=ZZCAL,#B=BIN,L=0,OPT=0,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.
.* COMPARE FN1 AND FN2 WITH PROGRAM DIFF OF LIBRARY PROGS OF DICK
.* WINTER (MC). DIFF IS WRITTEN IN A SPECIAL ASSEMBLER LANGUAGE THAT
.* MAY NO LONGER BE SUPPORTED BY THE SYSTEM AT SOME FUTURE DATE.
.IF,FILE(FN1,AS).AND.FILE(FN2,AS), OK.
DFMLOCK,ON.
CONNECT,OUTPUT.
REWIND,FN1,FN2.
.IF,$L$.NE.$OUTPUT$, NOTEBEGIN.
RETURN,L.
NOTE,$COMPARISON OF FILE FN1 AND FN2$,#L=L,DISCON.
.ELSE, NOTEBEGIN.
NOTE,$COMPARISON OF FILE FN1 AND FN2$.
.ENDIF, NOTEBEGIN.
ATTACH,PROGS,ID=DW,LC=1,MR=1.
LIBLOAD(PROGS,DIFF)
EXECUTE(DIFF,1=FN1,2=FN2,#L=L)
RETURN,OUTPUT,PROGS.
REVERT.
.ENDIF, OK.
NOTE,$NO FILE FN1 OR FN2$.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,OUTPUT,PROGS.
REVERT,ABORT.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. ---------- RUN5,  1 ------

.PROC,RUN5,FN,B=BIN/,OPT=2/0,L=0/LIST,KEEP=N/Y,PMD=N/Y,NOEX=N/Y.
.* FTN5 COMPILATION AND EXECUTION (UNLESS NOEX) OF FN.
.IF,FILE(FN,LO.OR.IN.OR.PF), OK.
REWIND,B,FN.
.IF,$KEEP$=$N$, KEEPNY.
RETURN,OUTPUT.
CONNECT,OUTPUT.
.ELSE, KEEPNY.
REWIND,OUTPUT.
SKIPF,OUTPUT,99999.
.ENDIF, KEEPNY.
.IF($PMD$=$N$) FTN5,I=FN,#B=B,#L=L,#OPT=OPT,LO=S/-A,PL=10000.
.IF($PMD$=$Y$) FTN5,I=FN,#B=B,#L=L,#OPT=0,LO=S/-A,DB,PL=10000.
.IF($NOEX$=$N$) B.
.IF($KEEP$=$N$) RETURN,OUTPUT.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: FILE FN DOES NOT EXISTS$.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. ---------- RUN5,  2 ------

REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
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 LINENRS).
.IF,$NOL$=$N$, NOLNY.    * ADDING LINENRS (NOL=N).
ED,USE,LNRS.
RETURN,EDLOG.
REWIND,OUT.
COPYSBF,OUT,OUTPUT.
.ELSE, NOLNY.            * NO LINENRS (NOL=Y).
REWIND,FN.
COPYSBF,FN,OUTPUT.
.ENDIF, NOLNY.
RETURN,LNRS,OUT.
COMMENT.    FN WRITTEN ON OUTPUT
REVERT.
.*
.DATA,LNRS.
F,L=72
E,FN
W,OUT,K
SC,INIT
B,Q


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

.PROC,FPRINT,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,$ERROR: YOU FORGOT TO SPECIFY #CY=$.
REVERT,ABORT.
.ENDIF, CHECKCY.
RETURN,FN.
ATTACH,FN,#CY=CY,ID=XXIDX,MR=1.
PAG,FN,$PERMANENT FILE$,$FN #CY=CY$.
.ELSE, FILESTATUS.               * LOCAL FILE.
PAG,FN,$LOCAL FILE$,$FN$.
.ENDIF, FILESTATUS.
COMMENT.    FN WRITTEN ON OUTPUT
ROUT.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: FILE FN DOES NOT EXIST$.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$ERROR$.
REVERT,ABORT.


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

      PROGRAM PAG
C     ******************************************************************
C     * PROGRAM WRITING ATTACHED FILE IN NUMBERED PAGES ON OUTPUT.     *
C     * CALL: "PAG,FN,$LOCAL FILE$,$FN$."                              *
C     *   OR: "PAG,FN,$PERMANENT FILE$,$FN CY=..$."                    *
C     *   OR: "PAG,FN,$FROM FN CY=..$,$PART$,N1=..,N2=...".            *
C     ******************************************************************
C
      CHARACTER*27 L,LEFT
      CHARACTER*10 D,T
      CHARACTER*14 R,RIGHT
      CHARACTER* 7 FN,FN1,NN1,NNN1,NN2,NNN2
      CHARACTER*72 LINE
C
      IPMAX=150
      ILMAX=60
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. ----------- PAG,  2 ------
C
      CALL GETPARM(FN,FN1,I)
      CALL GETPARM(L,LEFT,I)
         JL=INDEX(L//'  ','  ')
         IF(JL.GT.27) JL=27
         LEFT='---------------------------'
         LEFT(1:JL)=L(1:JL-1)//' '
      CALL GETPARM(R,RIGHT,I)
         JR=INDEX(R//'  ','  ')
         IF(JR.GT.14) JR=14
         RIGHT='--------------'
         RIGHT(15-JR:14)=' '//R(1:JR-1)
      N1=1
      CALL GETPARM(NN1,NNN1,I)
      IF(I.NE.-1) THEN
         READ(NNN1,'(I7)') N1
         IF(N1.EQ.0) N1=1
      ENDIF
      N2=IPMAX*ILMAX
      CALL GETPARM(NN2,NNN2,I)
      IF(I.NE.-1) THEN
         READ(NNN2,'(I7)') N2
      ENDIF
      CALL DATE(D)
      CALL TIME(T)
C
      OPEN(5,FILE=FN)
      OPEN(6,FILE='OUTPUT')
      REWIND 5
C
      DO 10 N=1,N1-1
   10 READ(5,'(A)',ERR=100,END=200) LINE
      DO 20 IP=1,IPMAX
      IL1=N1+(IP-1)*ILMAX
      IF(IL1.GT.N2) GOTO 200
      WRITE(6,1) LEFT,D,T,RIGHT,IP
      IL2=IL1+ILMAX-1
      IF(IL2.GT.N2) IL2=N2
      DO 20 IL=IL1,IL2
      READ(5,'(A)',ERR=100,END=200) LINE
      WRITE(6,2) LINE,IL
   20 CONTINUE
C
  100 STOP 'ERROR'
  200 STOP 'PROGRAM PAG'
C
    1 FORMAT('1','--- ',A27,A10,'-',A10,' -',A14,',',I3,' ------'/)
    2 FORMAT(' ',A72,I6,'0')
      END


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

      PROGRAM FOUT
C     *****************************************************************
C     * PROGRAM WRITING FILES OF ATTACHED MASTERFILE ON OUTPUT.       *
C     * CALL: "FOUT,FN1/FN2/FN3/...".                                 *
C     *****************************************************************
C
      CHARACTER*40 C1,C2
      CHARACTER*27 MFN,MFNAME
      CHARACTER*10 D,T
      CHARACTER* 8 XXFN,FN,FNAME
      CHARACTER*72 LINE
C
      IFMAX=10
      IPMAX=150
      ILMAX=60
C
      C1='XXC=COMMENT.'
      CALL FGET(C1)
      OPEN(4,FILE='XXC')
      READ(4,'(/A)') MFN
      I=INDEX(MFN,'  ')
      MFNAME='---------------------------'
      MFNAME(1:I)=MFN(1:I-1)//' '
      CLOSE(4,STATUS='DELETE')
C
      CALL DATE(D)
      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//'='//FN//'.'
      CALL FGET(C2)
      OPEN(5,FILE=XXFN)
      J=INDEX(FN,' ')
      FNAME='--------'
      FNAME(9-J:8)=' '//FN(1:J-1)
C
      DO 10 IP=1,IPMAX
      READ(5,'(A)',ERR=100,END=15) LINE
      WRITE(6,1) MFNAME,D,T,FNAME,IP
      IL1=(IP-1)*ILMAX+1
      IL2=IL1+ILMAX-1
      WRITE(6,2) LINE,IL1
      DO 10 IL=IL1+1,IL2
      READ(5,'(A)',ERR=100,END=15) LINE
      WRITE(6,2) LINE,IL
   10 CONTINUE
   15 WRITE(7,*) FN//'WRITTEN ON OUTPUT'
      CLOSE(5,STATUS='DELETE')
C
   20 CONTINUE
   30 CLOSE(7,STATUS='DELETE')
      STOP 'PROGRAM FOUT'
  100 STOP 'ERROR'
C
    1 FORMAT('1','--- ',A27,A10,'-',A10,' -------',A8,',',I3,' ------'/)
    2 FORMAT(' ',A72,I6,'0')
      END
```

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

      PROGRAM UFOUT
C     ****************************************************************
C     * WRITE UPDATE SOURCE FILES OF ATTACHED MASTERFILE ON OUTPUT.  *
C     * CALL: "UFOUT,FN1/FN2/FN3/...".                               *
C     ****************************************************************
C
      CHARACTER*72 LINE,SLINE
      CHARACTER*40 C1,C2
      CHARACTER*27 MFN,MFNAME
      CHARACTER*10 D,T
      CHARACTER* 8 XXFN,FN,FNAME,DNAME,SCDECK
      CHARACTER* 5 SDECK
      DATA SCDECK/'*COMDECK'/
      DATA SDECK /'*DECK'/
C
      IFMAX=10
      IPMAX=150
      ILMAX=60
C
      C1='XXC=COMMENT.'
      CALL FGET(C1)
      OPEN(4,FILE='XXC')
      READ(4,'(/A)') MFN
      I=INDEX(MFN,' ')
      MFNAME='---------------------------'
      MFNAME(1:I)=MFN(1:I-1)//' '
      CLOSE(4,STATUS='DELETE')
C
      CALL DATE(D)
      CALL TIME(T)
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
```

```
         FN=XXFN
         XXFN='XXXX'
      ENDIF
      IF(IF.EQ.-1) GOTO 40
      C2=XXFN//'='//FN//'.'
      CALL FGET(C2)
      OPEN(5,FILE=XXFN)
      J=INDEX(FN,' ')
      FNAME='--------'
      FNAME(9-J:8)=' '//FN(1:J-1)
C
      LNR=0
      L=0
      IKEEP=0
      N=0
      DO 20 IP=1,IPMAX
      READ(5,'(A)',ERR=100,END=25) LINE
      LNR=LNR+1
      WRITE(6,1) MFNAME,D,T,FNAME,IP
      IF(LINE(1:8).EQ.SCDECK) THEN
         N=1
         L=1
         K=INDEX(LINE(10:17),' ')
         DNAME=LINE(10:8+K)//'.'
         WRITE(6,2) LNR,DNAME
         WRITE(6,3) LINE,L
      ELSEIF(LINE(1:5).EQ.SDECK) THEN
         L=1
         K=INDEX(LINE(7:14),' ')
         DNAME=LINE(7:5+K)//'.'
         WRITE(6,2) LNR,DNAME
         WRITE(6,3) LINE,L
      ELSE
         L=L+1
         IF(NKEEP.EQ.1) THEN
            WRITE(6,2) LNR-1,DNAME
            WRITE(6,3) SLINE,L-1
         ELSE
            WRITE(6,2) LNR,DNAME
         ENDIF
         WRITE(6,3) LINE,L
      ENDIF
      IL=2
      IF(NKEEP.EQ.1) THEN
         IL=IL+1
         NKEEP=0
      ENDIF
C     * DO LOOP ON IL.
   10 READ(5,'(A)',ERR=100,END=25) LINE
      LNR=LNR+1
      IF(LINE(1:8).EQ.SCDECK) THEN
         L=1
         K=INDEX(LINE(10:17),' ')
         DNAME=LINE(10:8+K)//'.'
         WRITE(6,2) LNR,DNAME
         IL=IL+1
         IF(IL.GT.ILMAX) THEN
            NKEEP=1
            SLINE=LINE
            GOTO 20
         ENDIF
         WRITE(6,3) LINE,L
      ELSEIF(LINE(1:5).EQ.SDECK) THEN
         L=1
         K=INDEX(LINE(7:14),' ')
         DNAME=LINE(7:5+K)//'.'
         IF(N.NE.1) WRITE(6,2) LNR,DNAME
         IL=IL+1
         IF((IL.GT.ILMAX).OR.(N.EQ.1)) THEN
            IF(N.EQ.1) N=0
            NKEEP=1
            SLINE=LINE
            GOTO 20
         ENDIF
         WRITE(6,3) LINE,L
      ELSE
         L=L+1
         WRITE(6,3) LINE,L
      ENDIF
      IL=IL+1
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.13.50. --------- UFOUT,  3 ------

      IF(IL.LE.ILMAX) GOTO 10
C     * END DO LOOP ON IL.
   20 CONTINUE
C
   25 WRITE(7,*) FN//'WRITTEN ON OUTPUT'
      CLOSE(5,STATUS='DELETE')
   30 CONTINUE
C
   40 CLOSE(7,STATUS='DELETE')
      STOP 'PROGRAM UFOUT'
  100 STOP 'ERROR'
C
    1 FORMAT('1','--- ',A27,A10,'-',A10,' -------',A8,',',I3,' ------')
    2 FORMAT(' ',57X,'(LINE ',I4,'0)  ',A8)
    3 FORMAT(' ',A72,I7)
      END
```

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

.PROC,ROUT,OUTPUT,SHIFT=N/Y,TID=XXB/,FID=XXIDX/XXI2X,
       IC=DIS/ASCII.
.* ROUTE FILE OUTPUT TO THE LINEPRINTER.
.*
.IF,.NOT.FILE(OUTPUT,AS), LERROR.
NOTE,$ERROR: FILE OUTPUT DOES NOT EXIST$.
REVERT,ABORT.
.ENDIF, LERROR.
.*
REWIND,OUTPUT.
.IF,$SHIFT$=$N$, SHIFTNY.
COPY,OUTPUT,FILMPL.
.ELSE, SHIFTNY.
COPYSBF,OUTPUT,FILMPL.
.ENDIF, SHIFTNY.
ROUTE,FILMPL,DC=PR,#TID=TID,#FID=FID,#IC=IC.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
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.
.*
REWIND,JOB.
COPY,JOB,FILMPL.
.IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN,#FID=FID.
.IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,#TID=TID,#FID=FID.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
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,$ERROR: FILE DOES NOT EXIST$.
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.
.*
.DATA,ASK.
.PROC,ASK*I,
   ANSWER [ROUTE FILE? (N/Y1='XXIDX'/Y2='XXI2X') -]
                             = (N=0,Y1=1,Y2=2,0,1,2).
IF,ANSWER=0, LROUTE.
RETURN,ASK.
REVERT,ABORT.
ELSE, LROUTE.
REWIND,Z_ZZ.
COPY,Z_ZZ,FILMPL.
IF(ANSWER=1) ROUTE,FILMPL,DC=PR,TID=XXB,FID=XXIDX.
IF(ANSWER=2) ROUTE,FILMPL,DC=PR,TID=XXB,FID=XXI2X.
ENDIF, LROUTE.
REVERT.
```

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

.PROC,COPYMF,MF,EX=N/.
.* COPY MASTERFILE MF (ALL FILES EXCEPT EX=..).
MFUSE.
MFUSE,MF,M=OLD,ID=XXIDX,MR=1.
MFNEW,MF,M=NEW,ID=XXIDX.
FGET,COMMENT,M=OLD.
UPCOM.
FADD,COMMENT,M=NEW.
MFSET,MSG=F,ABT=F.
MFCOPY,M=OLD,N=NEW,REP.
.IF,$EX$.NE.$N$, EXCEPT.
FDEL,EX,M=NEW.
MFCLEAN,M=NEW.
.ENDIF, EXCEPT.
MFSET,MSG=PREV.
RETURN,OLD,NEW,COMMENT.
MFUSE,MF,ID=XXIDX.
NOTE,$NEW CY OF MASTERFILE MF ATTACHED$.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
REVERT,ABORT.
```

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

      PROGRAM UPCOM
C     ****************************************************************
C     * PROGRAM UPDATING COMMENT FILE OF A MASTERFILE WITH NEW CY AND  *
C     * DATE. TO BE USED WHEN COPYING MASTERFILES WITH COPYMF.         *
C     ****************************************************************
C
      CHARACTER*72 LINE
      CHARACTER*10 D
C
      OPEN(5,FILE='COMMENT')
      OPEN(6,FILE='STORE')
      REWIND 5
      REWIND 6
      DO 10 IL=1,1000
      READ(5,1,END=20) LINE
      WRITE(6,1) LINE
   10 CONTINUE
C
   20 REWIND 5
      REWIND 6
C
      READ(6,1)LINE
      WRITE(5,1) LINE
C
      READ(6,1) LINE
      I=INDEX(LINE,'CY=')
      READ(LINE(I+3:I+5),'(I3)') NCY
      NCY=NCY+1
      IF(NCY.LT.10) THEN
         WRITE(LINE(I+3:I+3),'(I1)') NCY
      ELSE IF((NCY.GE.10).AND.(NCY.LT.100)) THEN
         WRITE(LINE(I+3:I+4),'(I2)') NCY
      ELSE IF(NCY.GE.100) THEN
```

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

         WRITE(LINE(I+3:I+5),'(I3)') NCY
      ENDIF
      WRITE(5,1) LINE
C
      READ(6,1) LINE
      CALL DATE(D)
      WRITE(5,1) D(2:9)
C
      DO 30 IL=4,1000
      READ(6,1,END=40) LINE
      WRITE(5,1) LINE
   30 CONTINUE
C
   40 CLOSE(5)
      CLOSE(6,STATUS='DELETE')
      STOP 'PROGRAM UPCOM'
C
    1 FORMAT(A)
      END
```

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

.PROC,DU,FN,CY=N/,NID=XXXXX/.
.* DUPLICATE PERMFILE FN OF ID=XXIDX TO ID=NID.
.IF,$CY$.NE.$N$, OK.
RETURN,FN.
ATTACH,FN,#CY=CY,ID=XXIDX.
MFNEW,DUM,M=DUM,ID=XXIDX.
FADD,FN=COPY,M=DUM.
RETURN,FN.
FTAKE,COPY,M=DUM.
CATALOG,COPY,FN,#CY=CY,ID=NID.
MFKILL,DUM,M=DUM,ID=XXIDX.
RETURN,COPY,DUM.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #CY=..$.
REVERT,ABORT.
.*
EXIT,#S.
NOTE,$ERROR$.
REVERT,ABORT.
```

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

.PROC,NEW,FN,S=N/,M=N/,NOUL=N/Y,NOR=N/Y,TID=N/XXB.
.* PRODUCE JOB NFN_S (OR NFN_M) FROM FILE NFN RESIDING IN MFFN.
.* THIS JOB PRODUCES NEW LIBRARY FNLIB FROM THE SOURCE
.* FN_S (AND MODIFICATION MFN_M ,ALSO RESIDING IN MFFN).
.* DEFAULT: NO MODIFICATION MFN_M (UNLESS M),
.* COMPLETE UPDATE SOURCE LISTING (UNLESS NOUL),
.* JOB NFN_S ROUTED TO INPUT QUEUE (UNLESS NOR),
.* OUTPUT APPEARS AT TERMINAL (UNLESS TID=).
.*
.IF,$S$.EQ.$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #S= $.
RETURN,EDSUB.
REVERT,ABORT.
.ENDIF, LERROR.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZN=N_FN/ZZS=FN_S.
.IF($M$.NE.$N$) FTAKE,ZZM=#M_FN_M.
COPYBR,ZZN,ZZN1.
ED,USE,EDSUB.
SUB,#S=S,#M=M,#NOUL=NOUL.
REWIND,ZZJOB.
COPYBR,ZZJOB,FILMPL.
COPYBR,ZZN,FILMPL.
REWIND,FILMPL.
.IF,$M$.EQ.$N$, LRENAME.
COPY,FILMPL,N_FN_S.
.ELSE, LRENAME.
COPY,FILMPL,N_FN_M.
.ENDIF, LRENAME.
.IF,$NOR$=$N$, LROUTE.
```

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

.IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN.
.IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,#TID=TID.
.ENDIF, LROUTE.
RETURN,ZZN,ZZS,ZZM,ZZN1,ZZJOB,EDLOG,EDSUB,SUB,FILMPL.
REVERT.
*
EXIT,#S.
NOTE,$ERROR$.
RETURN,ZZN,ZZS,ZZM,ZZN1,ZZJOB,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN N_FN.
.DATA,EDSUB.
10=.PROC,SUB,#S=S,#M=M,#NOUL=NOUL.
20=REVERT.
30=.DATA,ZZJOB.
I,ZZN1,30
W,SUB,0
SC,INIT
B,Q


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

.PROC,N,FN,S=N/,U=N/,B=N/,NOUL=N/Y,NOR=N/Y,TID=N/XXB.
.* PRODUCE JOB N_S FROM FILE NFN RESIDING IN MFFN.
.* THIS JOB PRODUCES NEW UPDATE PL UFN_U FROM THE SOURCE
.* FN_S AND COMPILES BFN_B.
.* DEFAULT: COMPLETE UPDATE SOURCE LISTING (UNLESS NOUL),
.* JOB N_S ROUTED TO INPUT QUEUE (UNLESS NOR),
.* OUTPUT APPEARS AT TERMINAL (UNLESS TID=).
.*
.IF,$S$.EQ.$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #S= $.
RETURN,EDSUB.
REVERT,ABORT.
.ENDIF, LERROR.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZN=N_FN/ZZS=FN_S.
COPYBR,ZZN,ZZN1.
ED,USE,EDSUB.
.IF($U$.EQ.$N$.AND.$B$.EQ.$N$) SUB,#S=S,#U=S,#B=S,#NOUL=NOUL.
.IF($U$.NE.$N$.AND.$B$.EQ.$N$) SUB,#S=S,#U=U,#B=S,#NOUL=NOUL.
.IF($U$.EQ.$N$.AND.$B$.NE.$N$) SUB,#S=S,#U=S,#B=B,#NOUL=NOUL.
.IF($U$.NE.$N$.AND.$B$.NE.$N$) SUB,#S=S,#U=U,#B=B,#NOUL=NOUL.
SKIPF,N_S.
COPYBR,ZZN,N_S.
.IF,$NOR$=$N$, LROUTE.
REWIND,N_S.
COPY,N_S,FILMPL.
.IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN.
.IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,#TID=TID.
.ENDIF, LROUTE.
RETURN,ZZN,ZZN1,ZZS,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,#S.
NOTE,$ERROR$.
RETURN,ZZN,ZZN1,ZZS,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUES PARAMETERS IN N_FN.
.DATA,EDSUB.
10=.PROC,SUB,#S=S,#U=U,#B=B,#NOUL=NOUL.
20=REVERT.
30=.DATA,N_S.
I,ZZN1,30
W,SUB,0
SC,INIT
#B,Q


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

.PROC,R,FN,U=N/,M=N/,V=N/,B=N/,ULIST=N/Y,NOR=N/Y,TID=N/XXB.
.* PRODUCE JOB R_M FROM FILE RFN RESIDING IN MFFN.
.* THIS JOB REVISES OLD UPDATE PL UFN_U WITH MODIFICATION
.* MFN_M TO GET NEW UPDATE PL UFN_V, AND COMPILES BFN_B.
```

```
--- 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,$U$.EQ.$N$.OR.$M$.EQ.$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #U= OR #M=$.
RETURN,EDSUB.
REVERT,ABORT.
.ENDIF, LERROR.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZR=R_FN/ZZM=#M_FN_M.
ATTACH,ZZU,#U_FN_U,ID=XXIDX,MR=1.
ED,USE,EDSUB.
.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,$NOR$=$N$, LROUTE.
REWIND,R_M.
COPY,R_M,FILMPL.
.IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN.
.IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,#TID=TID.
.ENDIF, LROUTE.
RETURN,ZZR,ZZM,ZZU,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZR,ZZM,ZZU,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN R_FN.
.DATA,EDSUB.
10=.PROC,SUB,#U=U,#M=M,#V=V,#B=B,#ULIST=ULIST.
20=REVERT.
30=.DATA,R_M.
I,ZZR,30
W,SUB,0
SC,INIT
#B,Q


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

.PROC,X,FN,B=N/,I=N/,P=N/Y,O=N/Y,D=N/Y,T=N/,IO=N/,LP=NP/LP,
          NOR=N/Y,TID=N/XXB.
.* PRODUCE JOB X_B_I FROM FILE XFN RESIDING IN MFFN.
.* THIS JOB EXECUTES BFN_B WITH INPUT IFN_I.
.* P: CATALOG PLOTFILE P_B_I.   O:CATALOG OUTPUT O_B_I.
.* D: CATALOG DATA FILE D_B_I.
.* JOB X_B_I ROUTED TO INPUT QUEUE (UNLESS NOR),
.* OUTPUT APPEARS AT TERMINAL (UNLESS TID=).
.*
.IF,$B$.EQ.$N$.OR.$I$.EQ.$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #B= OR #I=$.
RETURN,EDSUB.
REVERT,ABORT.
.ENDIF, LERROR.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZX=X_FN/ZZI=#I_FN_I.
ATTACH,ZZB,#B_FN_B,ID=XXIDX,MR=1.
.IF,$T$.NE.$N$.OR.$IO$.NE.$N$.OR.$LP$.NE.$NP$, LJOBCRD.
JOBCRD,ZZX,#T=T,#IO=IO,#LP=LP.
.ENDIF, LJOBCRD.
ED,USE,EDSUB.
SUB,#B=B,#I=I,#P=P,#O=O,#D=D.
.IF,$NOR$=$N$, LROUTE.
REWIND,X_B_I.
COPY,X_B_I,FILMPL.
.IF($TID$.EQ.$N$) ROUTE,FILMPL,DC=IN.
.IF($TID$.NE.$N$) ROUTE,FILMPL,DC=IN,#TID=TID.
.ENDIF, LROUTE.
RETURN,ZZX,ZZI,ZZB,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,S.
```

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

NOTE,$ERRORS.
RETURN,ZZX,ZZI,ZZB,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUES PARAMETERS IN X_FN.
.DATA,EDSUB.
10=.PROC,SUB,#B=B,#I=I,#P=P,#0=0,#D=D.
20=REVERT.
30=.DATA,X_B_I.
#I,ZZX,30
W,SUB,#0
SC,INIT
#B,Q


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

      PROGRAM JOBCRD
C     **************************************************************
C     * PROGRAM REWRITING JOBCARD OF JOB ON FILE FN FOR GIVEN T,IO,LP. *
C     * CALL: "JOBCRD,FN,T=..,IO=..,LP=NP."                         *
C     **************************************************************
C
      CHARACTER*7 FN,DUM,T,IO,LP,XX,XXX
      CHARACTER*72 LINE
      DATA T,IO,LP/3*'N'/
C
      CALL GETPARM(FN,DUM,IDUM)
      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
C
   20 REWIND 5
      REWIND 6
C
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.EQ.0) II=INDEX(LINE(I+3:72),'.')
         T=LINE(I+2:I+1+II)
      ENDIF
      J=INDEX(LINE,',IO')
      IF(J.NE.0) THEN
         JJ=INDEX(LINE(J+4:72),',')
         IF(JJ.EQ.0) JJ=INDEX(LINE(J+4:72),'.')
         IO=LINE(J+3:J+2+JJ)
      ENDIF
      K=INDEX(LINE,',NP')
      IF(K.NE.0) THEN
         LP='NP'
      ELSE
         K=INDEX(LINE,',LP')
         IF(K.NE.0) LP='LP'
      ENDIF
      L=INDEX(LINE,'.')
C
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
   30 IF((XX.EQ.'LP').AND.(XXX.NE.'N')) LP=XXX
   40 IF((T.EQ.'N').OR.(IO.EQ.'N').OR.(LP.EQ.'N')) GOTO 100
      II=INDEX(T,' ')-1
      JJ=INDEX(IO,' ')-1
C
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:II)//',IO'//IO(1:JJ)//','//LP(1:2)//'.'
C
```

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

      WRITE(5,1) LINE
      DO 50 IL=2,1000
      READ(6,1,END=60) LINE
   50 WRITE(5,1) LINE
   60 CLOSE(6,STATUS='DELETE')
C
      STOP 'PROGRAM JOBCRD'
  100 STOP 'ERROR IN CALL OF JOBCRD'
C
    1 FORMAT(A)
      END
```

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

.PROC,PLOUT,LFN,NOVERSA=VY/VN,NODELETE=DY/DN.
.* CONVERT LOCAL GRAPHFILE "LFN" TO G-CODE FILE,
.* INSERT FIRST LINE "\\\\LFN,P,NOVERSA,NODELETE",
.* AND ROUTE FILE TO DESTINATION XXA.
.*
.IF,FILE(LFN,AS), OK.
RETURN,ZZOUT,ZZINP,ZZJOB.
REQUEST,ZZJOB,*Q.
ATTACH,TRANSGF.
TRANSGF,I=LFN,O=ZZINP,IC=GRAPHFILE,OC=GCODE.
SPLIT.
REWIND,ZZOUT.
COPYSBF,ZZOUT,ZZJOB.
ROUTE,ZZJOB,DC=PR,FID=PPPPP,TID=XXA,IC=DIS.
RETURN,ZZINP,ZZOUT,TMP,TRANSGF.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: FILE LFN DOES NOT EXIST$.
RETURN,TMP.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$SOMETHING WENT WRONG$.
RETURN,TMP,TRANSGF,ZZINP,ZZOUT.
REVERT,ABORT.
.*
.DATA,TMP
\\\\LFN,P,NOVERSA,NODELETE
```

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

      PROGRAM SPLIT
C     ****************************************************************
C     THIS PROGRAM MAKES A LITTLE CONVERSION OF G-CODE FILES:
C     1. A FIRST LINE WITH THE NAME OF THE GRAPH FILE IS INSERTED,
C     2. THE INITIALIZING STRINGS OF THE DIFFERENT GRAPHS ARE WRITTEN
C        ON SEPARATE LINES,
C     3. THE FILE IS REARRANGED SUCH THAT PARTS THAT LOGICALLY BELONG
C        TOGETHER (LIKE "JNN/NN") ARE NOT DEVIDED OVER 2 LINES.
C     THESE CHANGES FACILITATE THE CONVERSION OF A G-CODE FILE BACK TO
C     A GRAPH FILE AGAIN.
C     THIS PROGRAM IS CALLED BY PROCEDURE PLOUT, WHICH CONVERTS A GRAPH
C     FILE AND SENDS THE RESULTING G-CODE FILE TO DESTINATION XXA.
C     ****************************************************************
C
      PARAMETER (NCHR=100)
      CHARACTER*(NCHR) LINE,INIBUF,OUTBUF
      LOGICAL INITST
      INTEGER ISTART,IEND,LEN,INIPTR,OUTPTR
C
      OPEN(4,FILE='TMP')
      OPEN(5,FILE='ZZINP')
      OPEN(6,FILE='ZZOUT')
      REWIND 4
      REWIND 5
C
C     * FIRST, COPY THE ONE-LINE FILE "TMP" (WHICH CONTAINS THE NAME OF
C     * THE ORIGINAL GRAPH FILE).
      READ(4,'(A)',ERR=100,END=200) LINE
      WRITE(6,'(A)') LINE(1:INDEX(LINE,' ')-1)
C
      INITST=.TRUE.
      INIPTR=0
```

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

      OUTPTR=0
C
C     * START OF THE MAIN LOOP: READ INPUT FILE LINE BY LINE.
C
   10 READ(5,'(A)',ERR=100,END=200) LINE
      LEN=INDEX(LINE,' ')-1
C
   20 IF(INITST) THEN
         IEND=INDEX(LINE,'<')
         IF(IEND.GT.0) THEN
            INIBUF(INIPTR+1:INIPTR+IEND)=LINE(1:IEND)
            WRITE(6,'(A)') INIBUF(1:INIPTR+IEND)
            LINE(1:LEN-IEND)=LINE(IEND+1:LEN)
            LEN=LEN-IEND
            LINE(LEN+1:NCHR)=' '
            INITST=.FALSE.
            OUTPTR=0
            GOTO 20
         ELSE
            INIBUF(INIPTR+1:INIPTR+LEN)=LINE(1:LEN)
            INIPTR=INIPTR+LEN
         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)
            LINE(1:LEN-ISTART)=LINE(ISTART+1:LEN)
            LEN=LEN-ISTART
            LINE(LEN+1:NCHR)=' '
            INITST=.TRUE.
            INIPTR=0
            GOTO 20
         ELSE
            NLEN=LEN
   30       IF(LLE(LINE(NLEN:NLEN),';')) THEN
               NLEN=NLEN-1
               GOTO 30
            ENDIF
            IF(LINE(NLEN:NLEN).EQ.'F') 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
C
      GOTO 10
C
  100 STOP 'READ ERROR'
  200 STOP 'SPLIT FINISHED - END OF FILE'
      END
```

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

.PROC,TOUT,LFN,NOVERSA=VY/VN,NODELETE=DY/DN.
.* LOCAL TEXT FILE "LFN":
.* INSERT FIRST LINE "\\\\LFN,T,NOVERSA,NODELETE",
.* AND ROUTE FILE TO DESTINATION XXA.
.*
.IF,FILE(LFN,AS), OK.
RETURN,ZZOUT,ZZJOB.
REQUEST,ZZJOB,*Q.
ED,USE,EDJOB.
COPYSBF,ZZOUT,ZZJOB.
ROUTE,ZZJOB,DC=PR,FID=PPPPP,TID=XXA,IC=DIS.
RETURN,EDLOG,EDJOB,ZZOUT.
REVERT.
.*
.ENDIF, OK.
NOTE,$ERROR: FILE LFN DOES NOT EXIST$.
RETURN,EDJOB.
REVERT,ABORT.
.*
EXIT,S.
NOTE,$SOMETHING WENT WRONG$.
RETURN,EDLOG,EDJOB,ZZOUT.
REVERT,ABORT.
.*
```

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

.DATA,EDJOB.
E,LFN
5=\\\\LFN,T,NOVERSA,NODELETE
W,ZZOUT
B
```

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

      PROGRAM DT
C     *****************************************************************
C     * PRINT DATE AND TIME.                                          *
C     *****************************************************************
C
      CHARACTER*10 D,T
      CALL DATE(D)
      CALL TIME(T)
      T(10:10)=' '
      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     * PROGRAM WRITING LOCAL FILE ZZSYS, CONTAINING PROCEDURE OF THE  *
C     * SAME NAME, WHICH UPON CALLING SHOWS WHICH FILES OF SYSBULL AND *
C     * USERBUL HAVE BEEN CHANGED OVER THE PAST 7 DAYS.               *
C     *****************************************************************
C
      CHARACTER*10 DATUM
      INTEGER YEAR,MONTH,DAY
      DIMENSION M(0:12)
      DATA M /0,31,28,31,30,31,30,31,31,30,31,30,31/
C
      OPEN(5,FILE='ZZSYS')
      REWIND 5
C
      CALL DATE(DATUM)
      READ(DATUM,'(7X,I2)') YEAR
      READ(DATUM,'(4X,I2)') MONTH
      READ(DATUM,'(1X,I2)') DAY
C
      M2=0
      DO 10 K=0,MONTH-1
      M2=M2+M(K)
   10 CONTINUE
      M2=M2+DAY
      M2=M2-7
      KK=0
      DO 20 K=0,12
      IF(M2-M(K).GT.0) THEN
         M2=M2-M(K)
         KK=KK+1
      ENDIF
   20 CONTINUE
C
      WRITE(5,21) M2,KK,YEAR
      WRITE(5,22) M2,KK,YEAR
   21 FORMAT('.PROC,ZZSYS.'/
     A       'SYSBULL,D=',I2,'/',I2,'/',I2,'.')
   22 FORMAT('USERBUL,D=',I2,'/',I2,'/',I2,'.'/
     A       'RETURN,ZZSYS.')
      STOP 'PROGRAM SYS'
      END
```

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

.PROC,ZZSYS1.
ZZSYS2,44257.
REVERT.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- ZZSYS2,  1 ------

.PROC,ZZSYS2,D=0.
.* PROCEDURE CALLING SYSBULL AND C205BUL 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=D.
SYSBULL,#D=R1.
C205BUL,#D=R2.
ZZS,R1+.
RETURN,ZZS.
REVERT.
.*
.DATA,ZZS.
.PROC,ZZS,DD.
FREP,ZZSYS1,M=MFCCL.
RETURN,ZZSYS1.
REVERT.
.*
.##DATA,ZZSYS1.
.PROC,ZZSYS1.
ZZSYS2,DD.
REVERT.


--- 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,$PR$=$N$, PRNY.    * AUDIT SHOWN ON TERMINAL SCREEN.
CONNECT,OUTPUT.
.ELSE, PRNY.           * AUDIT PRINTED AT LINEPRINTER.
RETURN,OUTPUT.
REWIND,OUTPUT.
.ENDIF, PRNY.
.IF($PF$.EQ.$N$) AUDIT,#ID=ID,LF=DATA.
.IF($PF$.NE.$N$) AUDIT,#ID=ID,#PF=PF,LF=DATA.
PASAUD.
.IF,$LO$=$Y$, DIRECTORY.
MFUSE.
NOTE,$ATTACHED MASTERFILE RETURNED$.
DIR,#ID=ID.
RETURN,MASTER.
.ENDIF, DIRECTORY.
.IF($PR$=$Y$) ROUTE,OUTPUT,DC=#PR,TID=XXB,FID=XXIDX.
RETURN,DATA,OUTPUT.
REVERT.
.*
EXIT,S.
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     * FIRST CALL  "AUDIT,ID=...,LF=DATA.", THEN  "DIR,ID=...".       *
C     ******************************************************************
C
      CHARACTER*133 LINE,BLANK
      CHARACTER*10 MFNAME(50),D,T,SAVEMF
      CHARACTER*5 ID,IDNAME
      CHARACTER*3 CY(50),SAVECY
C
      DATA BLANK/' '/
      NMAX=50
      IPMAX=50
      LMAX=62
C
      CALL GETPARM(ID,IDNAME,I)
      IF(I.EQ.-1) IDNAME='XXIDX'
      OPEN(10,FILE='DATA')
      OPEN(20,FILE='STORE')
      OPEN(30,FILE='OUTPUT')
      REWIND 10
C
C     * MAKE LIST OF MASTERFILES.
```

```
--- 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.
     A   (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)
      ENDIF
   10 CONTINUE
C
C     * ALPHABETICAL ORDER.
   20 DO 25 J=2,N
      DO 25 K=J,N
      IF(MFNAME(J-1).LT.MFNAME(K)) GOTO 25
      IF((MFNAME(J-1).EQ.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
C
      CALL DATE(D)
      CALL TIME(T)
C
C     * LIST CONTENTS OF THE MASTERFILES.
      IP=1
      DO 60 K=1,N
      REWIND 20
      CALL MFUSE(MFNAME(K)//',M=MASTER,CY='//CY(K)//',ID='//IDNAME//
     A           ',MR=1.')
      CALL MFLIST('M=MASTER,L=STORE,LO,C.')
      REWIND 20
   30 WRITE(30,2) IDNAME,D,T,IP,MFNAME(K),CY(K)
      L=5
   40 READ(20,1,END=50,ERR=100) LINE
      IF((LINE(1:1).NE.'1').AND.(LINE.NE.BLANK)) THEN
         WRITE(30,1) LINE
         L=L+1
      ENDIF
      IF(L.LT.LMAX) GOTO 40
      IP=IP+1
      IF(IP.GT.IPMAX) GOTO 100
      GOTO 30
   50 IP=IP+1
      IF(IP.GT.IPMAX) GOTO 100
   60 CONTINUE
      WRITE(30,3)
      CLOSE(20,STATUS='DELETE')
C
      STOP 'PROGRAM DIR'
  100 STOP 'ERROR'
C
    1 FORMAT(A)
    2 FORMAT('1DIRECTORY OF ID=',A5,10X,A10,1X,A10,13X,'PAGE',I3///
     A       11X,'MASTERFILE  ',A10,'CY=',A3/)
    3 FORMAT(/' END DIRECTORY')
      END
```

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

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

CONST  LENGTH = 136; ENDLINE = 137;
    WRD10 = 10;
    C72 = 72;
TYPE   LINE = ARRAY [1..ENDLINE] OF CHAR;
    WOORD = PACKED ARRAY[1..WRD10] OF CHAR;
    ERRSTRING =PACKED ARRAY[1..20] OF CHAR;
    PFARR = ARRAY [1..C72] OF CHAR;
    PFP = ^PFINFO;
    PFINFO = RECORD
```

```
          NAME: PACKED ARRAY [1..10] OF CHAR;
          CY: INTEGER;
          INFO: PACKED ARRAY [1..C72] OF CHAR;
          P: PFP
       END;
VAR DATA : TEXT;
    INPOS,INLENGTH,TOTAL : INTEGER;
    INLINE : LINE; TXT : WOORD;
    USERID : WOORD;
    NRFILES: INTEGER;
    EOPF: BOOLEAN;
    PFLIST: PFP;
    LASTPF: PFP;
    PRUS,RBS,CYCLES,PRU2,RB2: INTEGER;
    DAY,TIME: WOORD;

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

PROCEDURE SKIPSPACES;
BEGIN  REPEAT INPOS := INPOS + 1
    UNTIL (INPOS = ENDLINE) OR (INLINE[INPOS] <> ' ');
    INPOS := INPOS - 1
END;  (*  OF SKIPSPACES  *)

PROCEDURE GETWOORD(VAR TXT: WOORD; 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: WOORD);
    VAR FLAG: BOOLEAN; TXT: WOORD; 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: WOORD; TOTAL: INTEGER;
BEGIN REPEAT FINDWOORD('1         ');
       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^ DO
    BEGIN P:= NIL; ANAME:= NAME; ACY:= CY END;
    LASTPF^.P:= ITEM;
    PF:= PFLIST; PFPLUS:= PF^.P;
    WHILE ANAME > 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');
       INPOS := INPOS + 1
    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;
       CREATION = 91; ATTACH = 113; ALTERATION = 124;
       NOATTACHES = 24; FLAGS = 79; RB = 87;
    VAR I,TOTAL: INTEGER;
       PF: PFP; TXT: WOORD; PFA: PFARR;
    DATE1,DATE2: WOORD;
    PROCEDURE COPY(POS,FROM,TOW: INTEGER);
       VAR I: INTEGER;
    BEGIN FOR I:= FROM TO TOW DO
       BEGIN POS := POS + 1;
          PFA[I] := INLINE[POS]
       END
    END;  (*  OF COPY  *)
BEGIN
    NRFILES := NRFILES + 1;
    NEW(PF);
    FOR I:= 1 TO C72 DO PFA[I] := ' ';
    PF^.CY := IGET(CYNR); PRUS := PRUS + IGET(PRU);
    COPY(PFNAAM,1,10); COPY(CYNR,11,13);
    COPY(VSNNR,67,72); COPY(PRU,22,25);
    IF (PFA[1]='M') AND (PFA[2]='F') THEN PFA[19]:='M';
    PACK(INLINE,CREATION+1,DATE1);
    PACK(INLINE,ALTERATION+1,DATE2);
    IF DATE1=DATE2 THEN BEGIN DATE2 := '   ID     ';
       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,42,49);
    PACK(INLINE,ATTACH+1,DATE1);
    IF DATE1=DAY THEN BEGIN DATE1 := ' TODAY    ';
       UNPACK(DATE1,INLINE,ATTACH+1) END;
    COPY(ATTACH,52,59);
    GETLERR('*** EOF IN GETPFINFO');
    RBS := RBS + IGET(RB);
    COPY(NOATTACHES,61,64); COPY(FLAGS,15,16); COPY(RB,27,29);
    PACK(PFA,1,PF^.INFO);
    FOR I:= 1 TO 10 DO IF PFA[I]=' ' THEN PFA[I]:=CHR(0);
    PACK(PFA,1,PF^.NAME);
    INSERT(PF);
END;  (*  OF GETPFINFO  *)

PROCEDURE GETGENINFO;
    VAR I: INTEGER;
BEGIN
    CYCLES := IGET(INPOS);
    PRU2 := IGET(INPOS);
    RB2 := IGET(INPOS);
END;  (*  OF GETGENINFO  *)

PROCEDURE HEADER(VAR PAGENR: INTEGER);
BEGIN
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.16.38. -------- PASAUD,  4 ------
    WRITE('1AUDIT OF ID = ',USERID);
    WRITE('           ',DAY,' ',TIME);
    WRITELN('              PAGE ',PAGENR:2);
    WRITELN;
    WRITE(' ':3,NRFILES:4,' FILES,');
    WRITE(PRUS:6,' PRUS,');
    WRITE(RBS:5,' RBS.');
    WRITELN('    (1 RB = 57 PRUS = 3648 WORDS)');
    WRITE('    AVERAGE: ',PRUS/NRFILES:6:1,' PRU/FILE,');
    WRITE(RBS/NRFILES:4:1,' RB/FILE.');
    WRITELN('   EFFICIENCY:',1.786*PRUS/RBS:6:1,' PROC.');
    WRITELN;
    WRITE(' FILENAME   CY FG FT   PRU  RB');
    WRITELN('   CREATED   ALTERED  ATTACHED  #AT   VSN');
    WRITELN;
    PAGENR := PAGENR + 1
END;  (*  OF HEADER  *)

PROCEDURE PRINTLIST(VAR PF:PFP; NRLINES: INTEGER);
    CONST SIZE = 60; VAR I: INTEGER;
BEGIN I := 0; IF NRLINES > SIZE THEN NRLINES := SIZE;
    WHILE (PF <> NIL) AND (I < NRLINES) DO
    BEGIN I := I + 1;
       WRITELN(' ',PF^.INFO);
       PF := PF^.P
    END
END;  (*  OF PRINTLIST  *)

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

BEGIN RESET (DATA);
    WHILE FINDAUDIT DO
    BEGIN DAYANDTIME;
       FINDWOORD('FLAGS-A=AR');
       NRFILES := 0;
       PRUS := 0; RBS := 0;
       NEW(PFLIST); LASTPF:= PFLIST; PFLIST^.P:= NIL;
       GETLERR('*** EOF BEFORE PF-S ');
       GETLINE; GETWOORD(USERID,TOTAL);
       GETPFINFO;
       EOPF := FALSE;
       WHILE NOT(EOPF OR EOF(DATA)) DO
       BEGIN GETLINE; GETWOORD(TXT,TOTAL);
             IF TXT=USERID THEN GETPFINFO ELSE
             BEGIN GETWOORD(TXT,TOTAL);
                EOPF := TXT = 'OWNER-ID  '
             END
       END;
       FINDWOORD('OWNER     ');
       CYCLES :=0; PRU2 := 0; RB2 := 0;
       IF EOF(DATA) THEN WRITELN(' *** PARTIAL AUDIT') ELSE
       BEGIN GETLINE;GETLINE;
          GETWOORD(TXT,TOTAL);
          IF TXT = USERID THEN GETGENINFO ELSE
          WRITELN(' *** ID-S DIFFERENT: ',USERID,' ',TXT);
       END;
       PRINTALL(55)
    END  (*  OF WHILE FINDAUDIT  *)
END .


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

.PROC,ATT750*I,
   ID [USER IDENTIFICATION -] = (*N=XXIDX,*A).
.*
.HELP,,NOLIST.
   ATT750   ATTACHES AND RETURNS ALL PERMFILES OF ID=.. ON THE 750 BY
   ======   RUNNING PROGRAM KEEP AND PROCEDURE ZZATT WHICH IT PRODUCES.
            DEFAULT: ID=XXIDX.
.ENDHELP.
```

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

.*
RETURN,ZZATT,ZZAUD,BIN.
CONNECT,OUTPUT.
AUDIT,#ID=ID,LF=ZZAUD,AI=P.
REWIND,ZZPROG.
FTN5,I=ZZPROG,B=BIN,L=0.
BIN.
.IF,$ID$.EQ.$XXIDX$, LRET.
MFUSE.
NOTE,$ATTACHED MASTERFILE RETURNED$.
RETURN,CCLLIB.
.ENDIF, LRET.
ZZATT.
.IF,$ID$.EQ.$XXIDX$, LATT.
ATTACH,CCLLIB,#ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
.ENDIF, LATT.
AUDIT,#ID=ID,LF=DATA.
PASAUD.
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 5 LINES (WHICH ARE
C     SKIPPED) + INFORMATION ON AT MOST 54 FILES.
C     ************************************************************
C
      CHARACTER PFN*40,#ID*9,DUMMY*10
      INTEGER CYCLE
      DATA NFILES/54/
C
      REWIND 5
      REWIND 6
C
C     * WRITE CCL PROCEDURE ZZATT ON FILE ZZATT.
      WRITE(6,1)
C
      DO 20 IPAGE=1,1000
      DO 10 I=1,5
   10 READ(5,'(A10)',END=300) DUMMY
      DO 20 I=1,NFILES
      READ(5,11,END=100,ERR=200) #ID,PFN,CYCLE
      IF(PFN(1:1).EQ.' '.AND.#ID.EQ.' '.AND.CYCLE.EQ.0) 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)
         STOP 'NO INFORMATION IN AUDIT FILE'
      ELSE
         GOTO 100
      ENDIF
C
C     * FORMATS.
    1 FORMAT('.PROC,ZZATT.'/'RETURN,A.')
   11 FORMAT(5X,A9,3X,A40,2X,I3)
   12 FORMAT('ATTACH,A=',A40,',',/,
     A       '#ID=',A9,',CY=',I3,',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,  3 ------

      END

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

.PROC,ADDP*I,
   NAME [PROCEDURE NAME -] = (*F),
   LIB  [LIBRARY NAME -  ] = (*F,*N=LIBRARY).
.*
.HELP,,NOLIST.
   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.
.*
.IF,.NOT.FILE(NAME,AS), NONAME.
NOTE,$FILE NAME DOES NOT EXIST; TRY AGAIN$.
REVERT,ABORT.
.ENDIF, NONAME.
.*
EDITLIB,I=ZZADDP1,L=ZZADDP2.
.IF(FILE(LIB,PF)) EXTEND,LIB.
RETURN,ZZADDP1,ZZADDP2.
.*
.DATA,ZZADDP1.
LIBRARY(LIB,OLD)
ADD(*,NAME,AL=1)
FINISH.
ENDRUN.

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

.PROC,DELP*I,
   NAME [PROCEDURE NAME -] = (*F),
   LIB  [LIBRARY NAME -  ] = (*F,*N=LIBRARY).
.*
.HELP,,NOLIST.
   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.
.*
EDITLIB,I=ZZDELP1,L=ZZDELP2.
RETURN,ZZDELP1,ZZDELP2.
.IF(FILE(LIB,PF)) EXTEND,LIB.
.*
.DATA,ZZDELP1.
LIBRARY(LIB,OLD)
DELETE(NAME)
FINISH.
ENDRUN.

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

.PROC,GETP*I,
   NAME [PROCEDURE NAME -] = (*F),
   LIB  [LIBRARY NAME -  ] = (*F,*N=LIBRARY).
.*
.HELP,,NOLIST.
   GETP     GETS A PROCEDURE FROM A LIBRARY.
   ====     PARAMETERS:
            NAME - NAME OF THE  PROCEDURE
            LIB  - NAME OF THE LIBRARY, DEFAULT: "LIBRARY".
.ENDHELP.
.*
RETURN,ZZGETP1.
EDITLIB,I=ZZGETP2.
REWIND,DUM.
COPYBR,DUM,ZZGETP1.
RETURN,DUM,ZZGETP2,OUTPUT.
.*
.* CHECK:
```

```
--- 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,*N=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,$FILE NAME DOES NOT EXIST; TRY AGAIN$.
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? (D/A) -     ] = (*N=C6,D=C6,A=C8,C6,C8).
.*
.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=XXXACXXX,U=XXU1XX,PA=XPAX)
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- ADD205,  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? (N/..) -] = (*N=N,N=N,*F),
   NAM3 [3RD FILE? (N/..) -] = (*N=N,N=N,*F),
   NAM4 [4TH FILE? (N/..) -] = (*N=N,N=N,*F),
   NAM5 [5TH FILE? (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)
           NAM5 - 5TH FILE TO BE DELETED (OPTIONAL).
.ENDHELP.
.*
REWIND,ZZDEL1.
.*
.* CALL MFLINK.
NOTE,$MFLINK RESPONDS:$.
MFLINK,ZZDEL2,ST=205,DD=C6,I=ZZDEL1.
.*
SKIP, OVEREXIT.
EXIT.
NOTE,$SOMETHING WENT WRONG$.
ENDIF, OVEREXIT.
RETURN,ZZDEL1,ZZDEL2.
.*
.* INPUT FOR MFLINK.
.DATA,ZZDEL1.
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),
   CODE [CODE? (D/A/B) -   ] = (*N=C6,D=C6,A=C8,B=US,C6,C8,US).
.*
.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
                  CODED:
                  "D" OR "C6" INDICATES DISPLAY CODE (DEFAULT)
                  "A" OR "C8" INDICATES ASCII CODE
                  "B" OR "US" INDICATES BINARY CODE (USE FOR PLOTFILES).
.ENDHELP.
.*
REWIND,ZZGET.
.*
.* CALL MFLINK.
NOTE,$MFLINK RESPONDS:$.
MFLINK,NAME,ST=205,DD=CODE,I=ZZGET.
.*
SKIP, OVEREXIT.
EXIT.
NOTE,$SOMETHING WENT WRONG$.
ENDIF, OVEREXIT.
RETURN,ZZGET.
.*
.* INPUT FOR MFLINK.
.DATA,ZZGET.
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 -] = (*F),
   NEW [NEW NAME OF THE FILE -] = (*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,ST205.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCE(TL=10,WS=128,LP=0)
COMMENT.****************************************
COMMENT.  RNM205:
COMMENT.  RENAME FILE "OLD" TO "NEW"
COMMENT.****************************************
ATTACH,OLD.
SWITCH,OLD,NEW.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- AUD205,  1 ------

.PROC,AUD205*I,
   LO  [LENGTH OPTION (P/F) -    ] = (*N=P,P,F),
   OUT [NAME OF THE OUTPUT FILE -] = (*F,*N=ZZTERM).
.*
.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.
.ENDHELP.
.*
REWIND,ZZAUD1.
NOTE,$MFLINK RESPONDS:$.
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.
COPYBR,ZZAUD2,ZZTERM,1.
RETURN,ZZAUD3,ZZTERM,EDLOG.
.ELSE,TOTERM.
REWIND,ZZAUD2.
COPYBR,ZZAUD2,OUT.
.ENDIF, TOTERM.
.*
SKIP,OVEREXIT.
EXIT.
NOTE,$SOMETHING WENT WRONG$.
ENDIF,OVEREXIT.
RETURN,ZZAUD1,ZZAUD2,ZZAUD3.
.*
.* INPUT FOR MFLINK.
.DATA,ZZAUD1.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
AUDIT,#LO=LO,L=OUT.
MFGIVE,OUT.
.*
.* USEFILE FOR THE EDITOR.
.IF,$OUT$=$ZZTERM$, USEFILE.
.DATA,ZZAUD3.
SC,F,ZZAUD4
F,L=90
E,ZZAUD2
/1/-/(1)@C*
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- AUD205,  2 ------

W,ZZAUD2,0
B,Q
.ENDIF, USEFILE.


--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.17.58. -------- 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),
   FID [FILE IDENTIFICATION -    ] = (*N=XXI2X,*A).
.*
.HELP,,NOLIST.
   ATT205  ATTACHES 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,ZZJOB,ZZPROG.
COPYBR,ZZJOB,FILMPL.
COPYBR,ZZPROG,FILMPL.
ROUTE,FILMPL,DC=IN,ST=205,#TID=TID,#FID=FID.
RETURN,ZZJOB,ZZPROG.
.*
.* JOB:
.DATA,ZZJOB.
FID,ST205.
USER(#AC=AC,#U=U,#PA=PA)
RESOURCE(TL=100,WS=128,LP=1,PRIO=12)
COMMENT.****************************************
COMMENT.  ATT205:
COMMENT.  ATTACH ALL PERMFILES ON THE 205,
COMMENT.  AUDIT, AND RUN PROGRAM SAVE TO RESET
COMMENT.  THE DATE OF LAST ACCESS.
COMMENT.****************************************
ATTACH,*.
PURGE,OUT,GOF.
AUDIT,LO=F,L=OUT.
FTN200,I=INPUT,B=BIN,L=0.
LOAD,BIN,CN=GOF,L=0.
GOF.
COMMENT.**INPUT RECORD SAVE AFTER EOR**
.*
.* PROGRAM:
.DATA,ZZPROG.
      PROGRAM SAVE
C     ****************************************************************
C     THIS PROGRAM READS THE OUTPUT FILE 'OUT' FROM "AUDIT,LO=F,L=OUT",
C     REDUCES IT TO A COMPACT FORMAT, AND WRITES IT ONTO FILE 'OUTPUT'.
C     EACH FILE OF WHICH THE NAME IS FOUND ON THE LIST IS OPENED AND
C     CLOSED IN ORDER TO SET THE DATE OF LAST ACCESS TO TODAY.
C     ****************************************************************
C
      CHARACTER*6 UNAME
      CHARACTER*8 FNAME
      CHARACTER*136 LINE
C
      OPEN(10,FILE='OUT')
      OPEN(20,FILE='OUTPUT')
      REWIND(10)
C
C     * READ 1ST LINE OF A PAGE.
   10 READ(10,'(A)',ERR=100,END=100) LINE
      IF(LINE(2:16).NE.'CYBER 200 AUDIT') GOTO 10
      UNAME=LINE(36:41)
   20 WRITE(20,1) LINE(1:43),LINE(47:49),LINE(44:46),LINE(50:74),
     A            LINE(116:121)
C
C     * READ HEADERS.
   30 READ(10,'(A)',ERR=100,END=100) LINE
      IF(LINE(9:11).EQ.'FSN') THEN
         WRITE(20,2)
      ELSE
         WRITE(20,'(A)') LINE
         GOTO 30
      ENDIF
C
```

```
--- 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.
      DO 40 I=1,100
      READ(10,'(A)',ERR=100,END=200) LINE
      IF(LINE(23:28).EQ.UNAME) THEN
         FNAME=LINE(12:19)
         OPEN(30,FILE=FNAME)
         CLOSE(30,STATUS='DELETE')
         WRITE(20,3) LINE(9:19),LINE(30:46),LINE(55:60),LINE(64:69),
     A               LINE(73:76),LINE(89:90),LINE(87:88),LINE(91:92),
     B               LINE(113:114),LINE(111:112),LINE(115:116),
     C               LINE(101:102),LINE(99:100),LINE(103:104)//'*'
      ELSEIF(LINE(2:16).EQ.'CYBER 200 AUDIT') THEN
         GOTO 20
      ELSEIF(LINE(19:32).EQ.'AUDIT COMPLETE') THEN
         GOTO 200
      ELSE
         WRITE(20,'(A)') LINE
      ENDIF
   40 CONTINUE
C
  100 STOP '*** ERROR IN EXECUTION OF PROGRAM SAVE ***'
  200 WRITE(20,4)
      STOP 'END OF PROGRAM SAVE'
C
C     * FORMATS.
    1 FORMAT(A,A,A,A,A)
    2 FORMAT(1X,'FSN NAME   TYP FC RT BT  ACS  DEVICE  DSET',
     A       '  FLEN   CREATED  MODIFIED  ACCESSED')
    3 FORMAT(1X,A11,1X,A17,1X,A6,1X,A6,1X,A4,2X,
     A       A2,'/',A2,'/',A2,2X,A2,'/',A2,'/',A2,2X,A2,'/',A2,'/',A3)
    4 FORMAT(71X,'(*:TODAY)'/,6X,'AUDIT COMPLETE')
      END
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.19.23. -------- BUD205,  1 ------

.PROC,BUD205*I,
   U [USER IDENTIFICATION -] = (*N=XXU1XX,*A).
.*
.HELP,,NOLIST.
   BUD205  GIVES THE 205-BUDGET LEFT FOR U=...(USER IDENTIFICATION).
   ======  DEFAULT: U=XXU1XX.
.ENDHELP
.*
RETURN,BUDGET.
ATTACH,BUDGET,ID=PUBLIC.
BUDGET,U205=U.
RETURN,BUDGET.
REVERT.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.19.23. -------- PER205,  1 ------

.PROC,PER205*I,
   PFN  [PERMANENT FILE ON THE 205 -] = (*F),
   USER [USER GRANTED PERMISSION -  ] = (*N=XXU2XX,*A),
   AC   [ACCESS (R/W/X/A/M) -       ] = (*N=R,*S5(RWXAM)).
.*
.HELP,,NOLIST.
   PER205  GRANTS PERMISSION TO ACCESS U=XXU1XX FILE ON THE 205.
   ======  PARAMETERS:
           PFN  - NAME OF 205-PERMFILE BELONGING TO U=XXU1XX
           USER - USER GRANTED ACCESS TO THIS FILE, DEFAULT: U=XXU2XX
           AC   - ACCESS PERMISSION (R/W/X/A/M), DEFAULT: AC=R.
.ENDHELP.
.*
ROUTE,FILMPL,DC=IN,ST=205.
.*
.DATA,FILMPL.
XXI2X,ST205.
#USER(#AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCE(TL=10,WS=128,LP=0)
COMMENT.****************************************
COMMENT.  PER205:
COMMENT.  PERMISSION FOR U=USER TO ACCESS
COMMENT.  U=XXU1XX PERMFILES ON THE 205.
COMMENT.****************************************
ATTACH,PFN.
PERMIT,PFN,U=USER,#AC=AC.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.19.23. ---------- Q205,  1 ------

.PROC,Q205*I,
   U [USER IDENTIFICATION -] = (*N=XXU1XX,*A).
.*
.HELP,,NOLIST.
   Q205  SHOWS THE QUEUES ON THE 205 FOR U=...(USER IDENTIFICATION).
   ====  DEFAULT: U=XXU1XX.
.ENDHELP
.*
QSTAT,UN=U.
REVERT.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. -------- RIN205,  1 ------

.PROC,RIN205*I,
   JOB [JOB -                                ] = (*F),
   IN1 [INPUT FILE TO BE INCLUDED? (N/..) - ] = (*N=N,*F),
   IN2 [SECOND FILE TO BE INCLUDED? (N/..) -] = (*N=N,*F),
   TID [TERMINAL IDENTIFICATION? (N/..) -   ] = (*N=N,*A),
   FID [FILE IDENTIFICATION? (N/..) -       ] = (*N=XXI2X,*A).
.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 IDENTICATION (DEFAULT: XXI2X).
.ENDHELP.
.*
.IF,.NOT.FILE(JOB,AS), LERROR.
NOTE,$FILE JOB DOES NOT EXIST; TRY AGAIN$.
REVERT,ABORT.
.ENDIF, LERROR.
.*
RETURN,FILMPL.
REWIND,JOB.
COPYBR,JOB,FILMPL.
.IF,$IN1$.NE.$N$, LCOPY1.
REWIND,IN1.
COPYBR,IN1,FILMPL.
.ENDIF, LCOPY1.
.IF,$IN2$.NE.$N$, LCOPY2.
REWIND,IN2.
COPYBR,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=30 ---- 02/07/86 - 00.26.29. -------- PLT205,  1 ------

.PROC,PLT205*I,
   NAME [NAME OF THE 205 PLOT FILE -] = (*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.
PLOT205,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 -  ] = (*A)\
   S     [IDENTIFICATION OF THE SOURCE -   ] = (*N=N,*A),
   M     [MODIFICATION OF NEWPL? (N/..) -  ] = (*N=N,*A),
   NOUL  [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 [FTN200 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.
   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)
         NOUL  - 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.
.ENDHELP.
.*
.IF,$S$=$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #S=$.
RETURN,EDSUB,ASK,ZZINP.
REVERT,ABORT.
.ENDIF, LERROR.
.*
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZS=FN_S.
RETURN,OUTPUT.
REWIND,ZZINP.
UPDATE,F,I=ZZINP,N=U_FN_S,C=C_FN_S,L=A124,O=OUT1.
USL,OUT1,NOLIST=NOUL.
.IF,$M$=$N$, MNY.
.IF(NOCAT=0) CATALOG,U_FN_S,ID=XXIDX.
.ELSE, MNY.
FTAKE,ZZM=#M_FN_M.
RETURN,C_FN_S.
UPDATE,F,P=U_FN_S,I=ZZM,N=V_FN_M,C=C_FN_M,L=A1234,O=OUT2.
RETURN,U_FN_S.
.IF(NOCAT=0) CATALOG,V_FN_M,U_FN_M,ID=XXIDX.
UML,OUT2.
.ENDIF, MNY.
ASK,#FN=FN,#M=M.
RETURN,ZZS,ZZM,OUT1,OUT2,ASK,ZZINP.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
FTAKE,ZZN=NN_FN.
ED,USE,EDSUB.
SUB,NN_FN,#M=M,#NOR=NOR,#FLIST=FLIST.
RETURN,ZZN,EDLOG,EDSUB,SUB,FILMPL.
REVERT.
.*
EXIT,#S.
NOTE,$ERROR$.
RETURN,ZZS,ZZM,OUT1,OUT2,ASK,ZZINP,ZZN,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.DATA,ZZINP.
*READ ZZS
.*
.DATA,ASK.
.PROC,ASK*I,
   #FN = (HGO=HG,PPP=PP,HG,PP), #M = (N=S,S,M),
   ANSWER [OK TO ROUTE UPDATE OUTPUT? (Y/N) -] = (Y=T,N=F).
IF(ANSWER) ROUTE,OUTPUT,DC=PR,TID=XXB,FID=#FN_#_#M.
REVERT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN NN_FN
.* AND REPLACES FILE NAME NN_FN_S/M SINCE IT MAY BECOME TOO LONG.
.DATA,EDSUB.
10=.PROC,SUB*I,
11=   NN_FN = (NNHGO=NNHG,NNPPP=NNPP,NNHG,NNPP),
12=   #M = (N=S,S,M), #NOR = (*A), #FLIST = (*A).
20=.IF,#NOR=0, LROUTE.
30=REWIND,NN_FN_#_#M.
40=COPYBR,NN_FN_#_#M,FILMPL.
50=COPYBR,C_FN_#_#M,FILMPL.
60=ROUTE,FILMPL,DC=IN,ST=205.
70=.ENDIF, LROUTE.
80=REVERT.
90=.DATA,NN_FN_#_#M.
I,ZZN,100
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ---------- NNEW,  3 ------

W,SUB,0
SC,INIT
B,Q


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

.PROC,NN*I,
   FN    [ROOT FILE NAME OF THE PROGRAM -    ] = (*A)\
   S     [IDENTIFICATION OF THE SOURCE -     ] = (*N=N,*A),
   U     [IDENTIFICATION OF NEWPL? (N/..) - ] = (*N=N,*A),
   B     [IDENTIFICATION OF BINARY? (N/..) -] = (*N=N,*A),
   NOUL  [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 [FTN200 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 MFFN),
   ==     AND CREATES A JOB NN_S FROM THE FILE NNFN (OF MASTERFILE MFFN),
          WHICH COMPILES BFN_B 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 MFFN)
          U      - IDENTIFICATION OF NEWPL UFN_U (ON 750); DEFAULT:U=S
          B      - IDENTIFICATION OF BINARY BFN_B (ON 205); DEFAULT: B=S
          NOUL   - 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_S IS CREATED BUT NOT SUBMITTED.
.ENDHELP.
.*
.IF,$S$=$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #S=$.
RETURN,EDSUB,ASK,ZZINP.
REVERT,ABORT.
.ENDIF, LERROR.
.*
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZS=FN_S/ZZN=NN_FN.
RETURN,OUTPUT.
REWIND,ZZINP.
.IF,$U$.EQ.$N$, LNAMES.
UPDATE,F,I=ZZINP,N=#U_FN_S,C=C_FN_S,L=A124,O=OUT.
.IF(NOCAT=0) CATALOG,#U_FN_S,ID=XXIDX.
.ELSE, LNAMES.
UPDATE,F,I=ZZINP,N=#U_FN_U,C=C_FN_U,L=A124,O=OUT.
.IF(NOCAT=0) CATALOG,#U_FN_U,ID=XXIDX.
.ENDIF, LNAMES.
USL,OUT,NOLIST=NOUL.
ASK.
RETURN,OUT,ASK,ZZINP,ZZS.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
ED,USE,EDSUB.
.IF($U$.EQ.$N$.AND.$B$.EQ.$N$) SUB,#S=S,#U=S,#B=S,#FLIST=FLIST.
.IF($U$.NE.$N$.AND.$B$.EQ.$N$) SUB,#S=S,#U=U,#B=S,#FLIST=FLIST.
.IF($U$.EQ.$N$.AND.$B$.NE.$N$) SUB,#S=S,#U=S,#B=B,#FLIST=FLIST.
.IF($U$.NE.$N$.AND.$B$.NE.$N$) SUB,#S=S,#U=U,#B=B,#FLIST=FLIST.
.IF,NOR=0, LROUTE.
REWIND,NN_S.
COPYBR,NN_S,FILMPL.
.IF($U$.EQ.$N$) COPYBR,C_FN_S,FILMPL.
.IF($U$.NE.$N$) COPYBR,C_FN_U,FILMPL.
ROUTE,FILMPL,DC=IN,ST=205.
.ENDIF, LROUTE.
RETURN,ZZN,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,#S.
NOTE,$ERROR$.
RETURN,OUT,ASK,ZZINP,ZZS,ZZN,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN NN_FN.
.DATA,EDSUB.
10=.PROC,SUB,#S=S,#U=U,#B=B,#FLIST=FLIST.
20=REVERT.
30=.DATA,NN_S.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ NN,  2 ------

I,ZZN,30
W,SUB,0
SC,INIT
#B,Q
.*
.DATA,ASK.
.PROC,ASK*I,
   ANSWER [OK TO ROUTE UPDATE OUTPUT? (Y/N) -] = (Y=T,N=F).
IF(ANSWER) ROUTE,OUTPUT,DC=PR,TID=XXB,FID=NN_S.
REVERT.
.*
.DATA,ZZINP.
*LIMIT 10000
*READ ZZS


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

.PROC,RR*I,
   FN     [ROOT FILE NAME OF THE PROGRAM -    ] = (*A)\
   U      [IDENTIFICATION OF UPDATE OLDPL -   ] = (*N=N,*A),
   M      [IDENTIFICATION OF MODIFICATIONS - ] = (*N=N,*A),
   V      [IDENTIFICATION OF NEWPL? (N/..) - ] = (*N=N,*A),
   B      [IDENTIFICATION OF BINARY? (N/..) -] = (*N=N,*A),
   ULIST  [UPDATE LISTING OF CHANGES? (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  [FTN200 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.
   RR     REVISES OLDPL UFN_U WITH MODIFICATION DECK MFN_M (OF MASTERFILE
   ==     MFFN) TO GET NEWPL UFN_V AND CREATES A JOB RR_M FROM THE FILE
          RR_M (OF MFFN) WHICH COMPILES BFN_B ON THE 205.
          PARAMETERS:
          FN     - NAME OF THE PROGRAM SERVING AS A ROOT FOR THE NAMES OF
                   THE DERIVED FILES
          U      - IDENTIFICATION OF OLDPL UFN_U (ON 750)
          M      - IDENTIFICATION OF MODIFICATION DECK MFN_M (FROM MFFN)
          V      - IDENTIFICATION OF NEWPL UFN_V (ON 750); DEFAULT: V=M
          B      - IDENTIFICATION OF BINARY BFN_B (ON 205); DEFAULT: B=M
          ULIST  - IF SPECIFIED, UPDATE LISTING OF THE CHANGES IS MADE
          NOCAT  - IF SPECIFIED, NEWPL IS NOT CATALOGED ON THE 750
          FLIST  - IF SPECIFIED, A FTN200 LISTING IS MADE
          NOR    - IF SPECIFIED, JOB RR_M IS CREATED BUT NOT SUBMITTED.
.ENDHELP.
.*
.IF,$U$=$N$.OR.$M$=$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #U= OR #M=$.
RETURN,EDSUB,ASK.
REVERT,ABORT.
.ENDIF, LERROR.
.*
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZM=#M_FN_M/ZZR=RR_FN.
ATTACH,#U_FN,#U_FN_U,ID=XXIDX,MR=1.
RETURN,OUTPUT.
.IF,$V$=$N$, LNAMES.
UPDATE,F,P=#U_FN,I=ZZM,N=#U_FN_M,C=C_FN_M,L=A1234,O=OUT.
.IF(NOCAT=0) CATALOG,#U_FN_M,ID=XXIDX.
.ELSE, LNAMES.
UPDATE,F,P=#U_FN,I=ZZM,N=#U_FN_V,C=C_FN_V,L=A1234,O=OUT.
.IF(NOCAT=0) CATALOG,#U_FN_V,ID=XXIDX.
.ENDIF, LNAMES.
UML,OUT,LIST=ULIST.
RETURN,OUT.
ASK.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
ED,USE,EDSUB.
.IF($V$.EQ.$N$.AND.$B$.EQ.$N$) SUB,#U=U,#M=M,#V=M,#B=M,#FLIST=FLIST.
.IF($V$.NE.$N$.AND.$B$.EQ.$N$) SUB,#U=U,#M=M,#V=V,#B=M,#FLIST=FLIST.
.IF($V$.EQ.$N$.AND.$B$.NE.$N$) SUB,#U=U,#M=M,#V=M,#B=B,#FLIST=FLIST.
.IF($V$.NE.$N$.AND.$B$.NE.$N$) SUB,#U=U,#M=M,#V=V,#B=B,#FLIST=FLIST.
.IF,NOR=0, LROUTE.
REWIND,RR_M.
COPYBR,RR_M,FILMPL.
.IF($V$.EQ.$N$) COPYBR,C_FN_M,FILMPL.
.IF($V$.NE.$N$) COPYBR,C_FN_V,FILMPL.
ROUTE,FILMPL,DC=IN,ST=205.
.ENDIF, LROUTE.
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ RR,  2 ------

RETURN,ZZM,ZZR,ASK,#U_FN,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZM,ZZR,ASK,#U_FN,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN RR_FN.
.DATA,EDSUB.
10=.PROC,SUB,#U=U,#M=M,#V=V,#B=B,#FLIST=FLIST.
20=REVERT.
30=.DATA,RR_M.
I,ZZR,30
W,SUB,0
SC,INIT
#B,Q
.*
.DATA,ASK.
.PROC,ASK*I,
   ANSWER [OK TO ROUTE UPDATE OUTPUT? (Y/N) -] = (Y=T,N=F).
IF(ANSWER) ROUTE,OUTPUT,DC=PR,TID=XXB,FID=RR_M.
REVERT.


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

.PROC,XX*I,
   FN  [ROOT FILE NAME OF THE PROGRAM - ] = (*A)\
   B   [IDENTIFICATION OF THE BINARY -  ] = (*N=N,*A),
   I   [IDENTIFICATION OF THE INPUT -   ] = (*N=N,*A),
   P   [CATALOG PLOTFILE? (N/Y) -       ] = (*N=0,*K=1,N=0,Y=1,0,1),
   O   [CATALOG OUTPUT? (N/Y) -         ] = (*N=0,*K=1,N=0,Y=1,0,1),
   D   [CATALOG DATA FILE? (N/Y) -      ] = (*N=0,*K=1,N=0,Y=1,0,1),
   TL  [CHANGE DEFAULT ON TL? (N/..) -  ] = (*N=0,N=0,*S4(0123456789)),
   WS  [CHANGE DEFAULT ON WS? (N/..) -  ] = (*N=0,N=0,*S4(0123456789)),
   LP  [CHANGE DEFAULT ON LP? (N/..) -  ] = (*N=0,N=0,*S2(0123456789)),
   NOR [NO ROUTE TO INPUT QUEUE? (N/Y) -] = (*N=0,*K=1,N=0,Y=1,0,1).
.*
.HELP,,NOLIST.
   XX      CREATES A JOB XX_B_I FROM THE FILE XXFN OF MASTERFILE MFFN.
   ==      THIS JOB EXECUTES THE BINARY BFN_B WITH INPUT IFN_I.
           PARAMETERS:
           FN  - NAME OF THE PROGRAM SERVING AS A ROOT FOR THE NAMES OF
                 THE DERIVED FILES
           B   - IDENTIFICATION OF BINARY FILE BFN_B (ON THE 205)
           I   - IDENTIFICATION OF INPUT FILE IFN_I (IN MASTERFILE MFFN)
           P   - IF SPECIFIED, PLOTFILE IS CATALOGED UNDER THE NAME P_B_I
           O   - IF SPECIFIED, OUTPUT IS CATALOGED UNDER THE NAME O_B_I
           D   - IF SPECIFIED, DATAFILE IS CATALOGED UNDER THE NAME D_B_I
           TL  - IF SPECIFIED, DEFAULT VALUE OF TL IN XXFN IS OVERWRITTEN
           WS  - IF SPECIFIED, DEFAULT VALUE OF WS IN XXFN IS OVERWRITTEN
           LP  - IF SPECIFIED, DEFAULT VALUE OF LP IN XXFN IS OVERWRITTEN
           NOR - IF SPECIFIED, JOB XX_B_I IS CREATED BUT NOT SUBMITTED.
.ENDHELP
.*
.IF,$B$=$N$.OR.$I$=$N$, LERROR.
NOTE,$ERROR: YOU FORGOT TO SPECIFY #B= OR #I=$.
RETURN,EDSUB.
REVERT,ABORT.
.ENDIF, LERROR.
.*
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
MFUSE,MF_FN,ID=XXIDX.
FTAKE,ZZX=XX_FN/ZZI=#I_FN_I.
ED,USE,EDSUB.
SUB,XX_B_I,#B=B,#I=I,#P=P,#O=O,#D=D,#NOR=NOR.
RETURN,ZZX,ZZI,EDLOG,EDSUB,SUB.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,ZZX,ZZI,EDLOG,EDSUB,SUB,FILMPL.
REVERT,ABORT.
.*
.* USEFILE TO PRODUCE PROCEDURE WHICH SUBSTITUTES PARAMETERS IN XX_FN
.* AND ASKS FOR REPLACEMENT OF FILE NAME XX_B_I IF IT IS TOO LONG.
.DATA,EDSUB.
10=.PROC,SUB*#I,
11=   XX_B_I [REPLACE XX_B_I BY Y_B_I? (Y/..) -] = (Y=Y_B_I,*F),
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ XX,  2 ------

12=   #B=(*A), #I=(*A), #P=(*A), #O=(*A), #D=(*A), #NOR=(*A).
20=.IF,#NOR=0, LROUTE.
30=REWIND,XX_B_I.
40=COPYBR,XX_B_I,FILMPL.
50=COPYBR,ZZI,FILMPL.
60=ROUTE,FILMPL,DC=IN,ST=205.
70=.ENDIF, LROUTE.
80=REVERT.
90=.DATA,XX_B_I.
#I,ZZX,100
.IF,TL.NE.0.OR.WS.NE.0.OR.LP.NE.0, LRESOURCE.
DEL/RESOURCE/@
.=RESOURCE(#TL=TL,#WS=WS,#LP=LP,PRIO=12)
.ENDIF, LRESOURCE.
W,SUB,#0
SC,INIT
#B,Q


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

.PROC,NU*I,
   S     [SOURCE FILE -                       ] = (*F)\
   U     [NEW UPDATE PL? (NEWPL/..) -         ] = (*N=NEWPL,*F),
   C     [COMPILE FILE? (COMPILE/..) -        ] = (*N=COMPILE,0,*F),
   NOUL [NO UPDATE SOURCE LISTING? (N/Y) -  ] = (*N=0,*K=1,N=0,Y=1,0,1),
   FID  [FID FOR UPDATE OUTPUT? (XXIDX/..) -] = (*N=XXIDX,*A).
.*
.HELP,,NOLIST.
   NU   PRODUCES NEWPL U AND COMPILE FILE C FROM THE SOURCE S.
   ==   PARAMETERS:
        S    - SOURCE FILE
        U    - NEW UPDATE PROGRAM LIBRARY; DEFAULT: "NEWPL"
        C    - COMPILE 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,ABORT.
.ENDIF, LERROR.
.*
REWIND,S.
RETURN,U,OUTPUT.
.IF($C$.NE.$0$) RETURN,C.
REWIND,ZZINP.
UPDATE,F,I=ZZINP,N=U,#C=C,L=A124,O=OUT.
USL,OUT,NOLIST=NOUL.
RETURN,OUT.
ASK.
RETURN,ASK,ZZINP.
REVERT.
.*
EXIT,#S.
NOTE,$ERROR$.
RETURN,ASK,ZZINP.
REVERT,ABORT.
.*
.DATA,ASK.
.PROC,ASK*I,
   ANSWER [OK TO ROUTE UPDATE OUTPUT? (Y/N) -] = (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. ------------ RU,  1 ------

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

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.26.29. ------------ RU,  2 ------

.*
.HELP,,NOLIST.
   RU     REVISES OLDPL U WITH MODIFICATION DECK M TO PRODUCE NEWPL V AND
   ==     COMPILE FILE C.  PARAMETERS:
          U      - OLD UPDATE PROGRAM LIBRARY
          M      - MODIFICATION DECK; SUPPRESS: "M=0" (ONLY COMPILE OLDPL)
          V      - NEW UPDATE PL; DEFAULT: "NEWPL"; SUPPRESS: "V=0"
          C      - COMPILE FILE; DEFAULT: "COMPILE"; SUPPRESS: "C=0"
          ULIST  - IF SPECIFIED, UPDATE LISTING OF THE CHANGES IS MADE
          FID    - FID FOR UPDATE OUTPUT; DEFAULT: "XXIDX".
.ENDHELP.
.*
.IF,$M$.NE.$0$, LM.
.IF,(.NOT.(FILE(U,AS))).OR.(.NOT.(FILE(M,AS))), LERROR1.
NOTE,$FILE U OR M DOES NOT EXIST$.
NOTE,$TRY AGAIN$.
RETURN,ASK.
REVERT,ABORT.
.ENDIF, LERROR1.
.ELSE, LM.
.IF,.NOT.FILE(U,AS), LERROR2.
NOTE,$FILE U DOES NOT EXIST; TRY AGAIN$.
RETURN,ASK.
REVERT,ABORT.
.ENDIF, LERROR2.
.ENDIF, LM.
.*
REWIND,U.
RETURN,OUTPUT.
.IF($C$.NE.$0$) RETURN,C.
.IF,$M$.NE.$0$, LABELM.
REWIND,M.
.IF,$V$.NE.$0$, LABELV.
RETURN,V.
UPDATE,F,P=U,I=M,N=V,#C=C,L=A1234,O=OUT.
.ELSE, LABELV.
UPDATE,F,P=U,I=M,#C=C,L=A1234,O=OUT.
.ENDIF, LABELV.
.ELSE, LABELM.
RETURN,INPUT.
UPDATE,F,P=U,#C=C,L=A1234,O=OUT.
RETURN,INPUT.
.ENDIF, LABELM.
UML,OUT,LIST=ULIST.
ASK.
RETURN,OUT,ASK.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,OUT,ASK.
REVERT,ABORT.
.*
.DATA,ASK.
.PROC,ASK*I,
   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. ------------ SU,  1 ------

.PROC,SU*I,
   U     [OLD UPDATE PL -                        ] = (*F)\
   S     [SOURCE FILE -                          ] = (*F),
   NOUL  [NO UPDATE LISTING? (N/Y) -             ] = (*N=0,*K=1,N=0,Y=1,0,1),
   FID   [FID FOR UPDATE OUTPUT? (XXIDX/..) -] = (*N=XXIDX,*A).
.*
.HELP,,NOLIST.
   SU    RETRIEVES THE 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=0,L=7,O=OUT.
.IF,NOUL=0, LIST.
USL,OUT,NOLIST=NOUL.
ASK2.
.ENDIF, LIST.
STRIP,S.
RETURN,INPUT,OUT,ASK1,ASK2.
REVERT.
.*
EXIT,#S.
NOTE,$ERRORS.
RETURN,INPUT,ASK1,ASK2.
REVERT,ABORT.
.*
.DATA,ASK1.
.PROC,ASK1*I,
   ANSWER [OK TO RETURN EXISTING FILE S? (Y/N) -] = (Y=T,N=F).
IF(.NOT.ANSWER) REVERT,ABORT.
RETURN,S.
REVERT.
.*
.DATA,ASK2.
.PROC,ASK2*I,
   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. -------- INSTAL,  1 ------

.PROC,INSTAL*I\
  LIB     [LFN LIBRARY? (N=LIBRARY/..) -] = (*N=LIBRARY,N=LIBRARY,*F),
  PRCFILE [NO OPTION. JUST TYPE "N" -    ] = (*N=#FILE,N=#FILE).
.*
.HELP,,NOLIST.
  INSTAL  IS A PROCEDURE TO INSTAL A LIBRARY FOR UPDATE PROCEDURES AND
  ======  PROGRAMS WRITTEN BY JOS KOOT AND AMPLIFIED BY HANS GOEDBLOED.
          IT CONTAINS THE CCL PROCEDURES REVISE/ASKDOE AND THE FORTRAN
          PROGRAMS ASKDECK/MAKEFIL/MODGEN/USL/UML.
          PARAMETERS:
          LIB     - LFN OF THE LIBRARY; DEFAULT: "LIBRARY"
          PRCFILE - INSTRUCTS INSTAL TO READ FILES TO BE INSTALLED FROM
                    THE FILE INSTAL ITSELF.  NO OPTIONS FOR THE USER.
.ENDHELP.
.*
RETURN,LIB,ZZLST,ZZLGO,ZZPRC1,ZZPRC2.
LIBRARY.
COPYBR,PRCFILE,ZZPRC1.
COPYBR,PRCFILE,ZZPRC2.
FTN5,I=PRCFILE,B=ZZLGO,L=ZZLST,E=ZZLST,PL=10000.
EDITLIB,I=ZZINP,L=ZZLST.
LIBRARY,LIB.
RETURN,ZZINP,ZZLST,ZZLGO,ZZPRC1,ZZPRC2.
REVERT.
.*
.DATA,ZZINP.
LIBRARY(LIB,NEW)
ADD(*,ZZPRC1)
ADD(*,ZZPRC2)
ADD(*,ZZLGO)
SETAL(ASKDECK,1)
SETAL(MAKEFIL,1)
SETAL(MODGEN,1)
SETAL(USL,1)
SETAL(UML,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 MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- REVISE,  1 ------

.PROC,REVISE*I,
  U    [LFN OF OLDPL -                       ] = (*F)\
  D    [DECK NAME(S)? (N=$.ALL.$/Y=$.ASK.$/..) -] =
                    (*N=$.ALL.$,*K=$.ASK.$,N=$.ALL.$,Y=$.ASK.$,*A),
  M    [MODIFICATION DECK? (N=MODFILE/..)-] = (*N=MODFILE,N=MODFILE,*F),
  CI   [CORRECTION IDENTIFIER? (N=MOD/..)-] = (*N=MOD,N=MOD,*A),
  S    [OLD SOURCE? (N=0/OLDSRC/..) -     ] = (*N=0,*K=OLDSRC,N=0,0,*F),
  T    [NEW SOURCE? (N=0/NEWSRC/..) -     ] = (*N=0,*K=NEWSRC,N=0,0,*F),
  V    [NEWPL? (N=0/NEWPL/..) -           ] = (*N=0,*K=NEWPL,N=0,0,*F),
  C    [COMPILE FILE? (N=0/COMPILE/..) -  ] = (*N=0,*K=COMPILE,N=0,0,*F),
  UL1  [1ST UPDATE LISTING? (N/Y) -       ] = (*N=0,*K=1,N=0,Y=1,0,1),
  UL2  [2ND UPDATE LISTING? (N=0/Y=1/2) - ] = (*N=0,*K=1,N=0,Y=1,0,1,2).
.*
.HELP,,NOLIST.
  REVISE  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 COMPARED 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: $.ALL.$/$.ASK.$
          M    - MODIFICATION DECK; DEFAULT: "MODFILE"
          CI   - 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: 0/NEWPL
          C    - COMPILE FILE; DEFAULTS: 0/COMPILE
          UL1  - 1ST UPDATE LISTING (RETRIEVAL OF DECKS); DEFAULT: 0
          UL2  - 2ND UPDATE LISTING (CORRECTION RUN); DEFAULTS: 0/1.
.HELP,U,NOLIST.
  U       IS THE LOGICAL FILE NAME OF THE OLD PROGRAM LIBRARY.
.HELP,D,NOLIST.
  D       IS THE NAME OF THE DECK(S) AND COMMON DECK(S) TO BE UPDATED.
          OMITTED     - ALL DECKS AND COMMON DECKS
          D=$.ALL.$   - ALL DECKS AND COMMON DECKS
          D             - DECKNAMES WILL BE ASKED FOR VIA TERMINAL
          D=$.ASK.$   - DECKNAMES WILL BE ASKED FOR VIA TERMINAL
          D=DECKNAME ⊢ THE SPECIFIED DECK.
.HELP,M,NOLIST.
  M       IS 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      IS 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".
                    IN THAT CASE, CI IS CHANGED TO "MOD_MODFILENR".
          M=IDENT - THE SPECIFIED IDENTIFIER IS USED.
.HELP,S,NOLIST.
  S       IS THE LOGICAL FILE NAME OF THE OLD SOURCE DERIVED FROM OLDPL.
          OMITTED - OLD SOURCE IS RETURNED
          S=0      - OLD SOURCE IS RETURNED
          S        - OLD SOURCE ON FILE OLDSRC
          S=LFN    - OLD SOURCE ON FILE LFN.
.HELP,T,NOLIST.
  T       IS THE LOGICAL FILE NAME OF THE NEW SOURCE OBTAINED AFTER
          EDITING THE OLD SOURCE.
          OMITTED - NEW SOURCE IS RETURNED
          T=0      - NEW SOURCE IS RETURNED
          T        - NEW SOURCE ON FILE NEWSRC
          T=LFN    - NEW SOURCE ON FILE LFN.
.HELP,V,NOLIST.
  V       IS THE LOGICAL FILE NAME OF THE NEW PROGRAM LIBRARY.
          OMITTED - NO NEW LIBRARY PRODUCED
          V=0      - NO NEW LIBRARY PRODUCED
          V        - NEW LIBRARY ON FILE NEWPL
          V=LFN    - NEW LIBRARY ON FILE LFN.
.HELP,C,NOLIST.
  C       IS THE LOGICAL FILENAME OF THE COMPILE FILE THAT IS TO RECEIVE
          ALL MODIFIED CODE.
          OMITTED - NO COMPILE FILE GENERATED
          C=0      - NO COMPILE FILE GENERATED
          C        - COMPILE FILE WRITTEN ON FILE COMPILE
          C=LFN    - COMPILE FILE WRITTEN ON FILE LFN.
.HELP,UL1,NOLIST.
  UL1     INDICATES THAT THE LISTING OF THE 1ST UPDATE RUN (THE ONE THAT
          RETRIEVES THE SPECIFIED DECKS FROM OLDPL) IS TO BE KEPT AND
          REFORMATTED BY MEANS OF PROGRAM USL.
          OMITTED - UPDATE OUTPUT RETURNED
          UL1     - UPDATE SOURCE LISTING OF OLDPL (CORRESPONDING TO
                    FILE S) WRITTEN ON "OUTPUT"
.HELP,UL2,NOLIST.
  UL2     INDICATES THAT THE LISTING OF THE 2ND UPDATE RUN (THE ONE THAT
          PRODUCES A NEWPL AND COMPILE) 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.
.ENDIF, LERROR.
.*
.* IF OLD SOURCE IS TO BE WRITTEN AND FILE ALREADY EXISTS,
.* ASK PERMISSION TO RETURN IT.
.IF,$S$.NE.$0$, LASK1.
.IF(FILE(S,AS)) ASKPRM1.
.ENDIF, LASK1.
.*
.* IF NEW SOURCE IS TO BE WRITTEN AND FILE ALREADY EXISTS,
.* ASK PERMISSION TO RETURN IT.
.IF,$T$.NE.$0$, LASK2.
.IF(FILE(T,AS)) ASKPRM2.
.ENDIF, LASK2.
.*
.* TEST ON CONFLICTING FILENAMES FOR OLDPL AND NEWPL.
.* IF NEWPL IS TO BE WRITTEN AND FILE ALREADY EXISTS,
.* ASK PERMISSION YO RETURN IT.
.IF,$V$.NE.$0$, LASK3.
.IF,$U$=$V$, LERROR.
COMMENT.** OLDPL EQUALS NEWPL; TRY AGAIN **
RETURN,ZZUSE,ZZINP,ASKPRM1,ASKPRM2,ASKPRM3,ASKPRM4,ASKPRM5.
REVERT,ABORT.
.ENDIF, LERROR.
.IF(FILE(V,AS)) ASKPRM3.
.ENDIF, LASK3.
.*
.* IF COMPILE IS TO BE WRITTEN AND FILE ALREADY EXISTS,
.* ASK PERMISSION TO RETURN IT.
.IF,$C$.NE.$0$, LASK4.
.IF(FILE(C,AS)) ASKPRM4.
.ENDIF, LASK4.
.*
.* IF EDITFILE EXISTS, ASK PERMISSION TO RETURN IT.
.IF(FILE(ZZZZZ1Z,AS)) ASKDOE.
.*
.* TRAP IN CASE A PERMISSION WAS NOT GRANTED.
SKIP, LEXIT.
EXIT,#S.
COMMENT.** EXIT(1) IN PROCEDURE REVISE:    **
COMMENT.** REQUIRED PERMISSION NOT GRANTED **
RETURN,ZZUSE,ZZINP,ASKPRM1,ASKPRM2,ASKPRM3,ASKPRM4,ASKPRM5.
REVERT.
ENDIF, LEXIT.
.*
.* CLEAN START.
RETURN,ZZUL1,ZZOLD,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,$D$.EQ.$.ASK.$, LASKDECK.
ASKDECK,ZZINP.
RETURN,ZZTMI,ZZTMO.
SKIP, LEXIT.
EXIT,#S.
COMMENT.** EXIT(2) IN PROCEDURE REVISE **
RETURN,ZZINP,ZZUSE,ASKPRM5,ZZTMI,ZZTMO.
REVERT,ABORT.
ENDIF, LEXIT.
.ENDIF, LASKDECK.
.*
.* 1ST UPDATE RUN: RETRIEVE THE REQUESTED DECKS FROM OLDPL.
.* CHECK LEGALITY OF CORRECTION SET IDENTIFIER.
REWIND,U.
.IF,$D$.EQ.$.ALL.$, FULLYN.
RETURN,INPUT.
UPDATE,F,P=U,#C=0,L=7,O=ZZUL1.
.ELSE, FULLYN.
REWIND,ZZINP.
UPDATE,P=U,I=ZZINP,#C=0,L=7,O=ZZUL1.
.ENDIF, FULLYN.
RETURN,INPUT.
SKIP, LEXIT.
EXIT,#S.
COMMENT.** EXIT(3) IN PROCEDURE REVISE: **
COMMENT.** RETRIEVAL OF DECK(#S) FAILED. **
COMMENT.** WRONG OLDPL SPECIFIED OR     **
COMMENT.** WRONG DECKNAME OR            **
COMMENT.** WRONG CORRECTION IDENTIFIER? **
RETURN,ZZUL1,INPUT,ZZINP,ZZUSE,ASKPRM5.
REVERT,ABORT.
ENDIF, LEXIT.
.*
.* 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,$D$.EQ.$.ALL.$, FULLYN.
MAKEFIL,ZZUL1,ZZOLD,ZZNEW.
.ELSE, FULLYN.
MAKEFIL,ZZUL1,ZZOLD,ZZNEW,ZZINP.
.ENDIF, FULLYN.
.IF(UL1=1) USL,ZZUL1.
RETURN,ZZUL1,ZZINP.
.IF,$S$.NE.$0$, LS.
REWIND,ZZNEW.
COPYBR,ZZNEW,S.
REWIND,S.
.ENDIF, LS.
IF,FILE(ZZOLD,.NOT.AS).OR.FILE(ZZNEW,.NOT.AS), LEXIT.
EXIT,#S.
COMMENT.** EXIT(4) IN PROCEDURE REVISE: **
COMMENT.** RETRIEVAL OF DECK(#S) FAILED. **
COMMENT.** WRONG DECKNAME OR            **
COMMENT.** WRONG CORRECTION IDENTIFIER? **
RETURN,ZZUL1,ZZOLD,ZZNEW,ZZINP,ZZUSE,ASKPRM5.
REVERT,ABORT.
ENDIF, LEXIT.
.*
.* PLACE ZZNEW IN THE EDITFILE AND HAVE THE USER EDIT HIS DECKS.
REWIND,ZZOLD,ZZNEW.
ED,USE,ZZUSE.
RETURN,ZZUSE.
SKIP, LEXIT.
EXIT,#S.
COMMENT.** EXIT(5) IN PROCEDURE REVISE **
RETURN,ZZZZZ1Z,ZZZZZ3Z,EDLOG,ZZOLD,ZZNEW,ZZUSE,ASKPRM5.
REVERT,ABORT.
ENDIF, LEXIT.
ED.
.*
.* UPON RETURN FROM THE EDIT SESSION, ASK PERMISSION TO PROCEED
.* AND REPLACE ZZNEW BY EDITED VERSION.
ASKPRM5.
SKIP, LEXIT.
EXIT,#S.
COMMENT.** EXIT(6) IN PROCEDURE REVISE:    **
COMMENT.** REQUIRED PERMISSION NOT GRANTED.**
RETURN,EDLOG,ZZOLD,ZZNEW,ASKPRM5.
REVERT.
ENDIF, LEXIT.
ED,W,ZZNEW,0.
.IF,$T$.NE.$0$, LT.
REWIND,ZZNEW.
COPYBR,ZZNEW,T.
REWIND,T.
.ENDIF, LT.
.*
.* 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 COMPILE.
REWIND,M.
.IF,($V$.NE.$0$).OR.($C$.NE.$0$), VORC.
.IF,$V$.NE.$0$, VYN.
.IF,$D$.EQ.$.ALL.$, FULLYN1.
UPDATE,F,P=U,I=M,N=V,#C=C,L=A1234,O=ZZUL2.
.ELSE, FULLYN1.
UPDATE,P=U,I=M,N=V,#C=C,L=A1234,O=ZZUL2.
.ENDIF, FULLYN1.
.ELSE, VYN.
.IF,$D$.EQ.$.ALL.$, FULLYN2.
UPDATE,F,P=U,I=M,#C=C,L=A1234,O=ZZUL2.
.ELSE, FULLYN2.
UPDATE,P=U,I=M,#C=C,L=A1234,O=ZZUL2.
.ENDIF, FULLYN2.
.ENDIF, VYN.
.IF(UL2=1) UML,ZZUL2,LIST=0.
.IF(UL2=2) UML,ZZUL2,LIST=1.
RETURN,ZZUL2.
.ENDIF, VORC.
REVERT.
.*
EXIT,#S.
COMMENT.** EXIT(7) IN PROCEDURE REVISE **
RETURN,EDLOG,ZZOLD,ZZNEW,ZZUL2.
REVERT,ABORT.
.*
.* EDITOR USEFILE (PREPARE EDITFILE).
.DATA,ZZUSE.
SET,COUNT=1,LINES=19,EXP=1,PROMPT=##
FORMAT,NO
E,ZZNEW
.*
.* INPUT FOR FIRST UPDATE RUN (RETRIEVAL OF DECKS).
.DATA,ZZINP.
*IDENT DUMMY,#U=DUMMY
*COMPILE D
.*
.DATA,ASKPRM1.
.PROC,ASKPRM1*I,
  ANSWER [OK TO DISCARD EXISTING S? (Y/N) -] = (Y=#T,N=F).
RETURN,ASKPRM1.
IF(.NOT.ANSWER) REVERT,ABORT.
RETURN,S.
REVERT.
.*
.DATA,ASKPRM2.
.PROC,ASKPRM2*I,
  ANSWER [OK TO DISCARD EXISTING T? (Y/N) -] = (Y=#T,N=F).
RETURN,ASKPRM2.
IF(.NOT.ANSWER) REVERT,ABORT.
RETURN,T.
REVERT.
.*
.DATA,ASKPRM3.
.PROC,ASKPRM3*I,
  ANSWER [OK TO DISCARD EXISTING V? (Y/N) -] = (Y=#T,N=F).
RETURN,ASKPRM3.
IF(.NOT.ANSWER) REVERT,ABORT.
RETURN,V.
REVERT.
.*
.DATA,ASKPRM4.
.PROC,ASKPRM4*I,
  ANSWER [OK TO DISCARD EXISTING C? (Y/N) -] = (Y=#T,N=F).
RETURN,ASKPRM4.
IF(.NOT.ANSWER) REVERT,ABORT.
RETURN,C.
REVERT.
.*
.DATA,ASKPRM5.
.PROC,ASKPRM5*I,
  ANSWER [OK TO PROCEED UPDATING? (Y/N) -] = (Y=#T,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) -] = (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(.NOT.ANSWER) REVERT,ABORT.
RETURN,ZZZZZ1Z,ZZZZZ3Z.
REVERT.
```

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

      PROGRAM ASKDECK
C
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
      CHARACTER *60 LINE
C
      CALL GETPARM(INPFILE,DUMSTR,PARSTAT)
      OPEN(UNIT=1,FILE=INPFILE)
      OPEN(UNIT=2,FILE='ZZTMI')
      OPEN(UNIT=3,FILE='ZZTMO')
      CALL CONNEC(2)
      CALL CONNEC(3)
      REWIND(1)
C
      WRITE(1,'(A)') '*IDENT DUMMY,U=DUMMY'
      WRITE(3,*) 'ENTER ONE OR MORE DECKNAMES IN THE FOLLOWING FORMAT:'
      WRITE(3,*) 'DECKNAME                (ONE SINGLE DECK)  '
      WRITE(3,*) 'DECK1,DECK2,...            (SEVERAL DECKS)  '
      WRITE(3,*) 'FIRSTDECK.LASTDECK           (FOR A RANGE) -'
      READ(2,'(A)') LINE
   10 WRITE(1,'(A,A)') '*COMPILE ',LINE
      WRITE(3,*) 'ENTER MORE DECKNAMES OR A SPACE (FOR TERMINATION) -'
      READ(2,'(A)') LINE
      IF(LINE.NE.' ') GOTO 10
C
      REWIND(1)
      CLOSE(1,STATUS='KEEP')
      CLOSE(2,STATUS='DELETE')
      CLOSE(3,STATUS='DELETE')
      END
```

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

      PROGRAM MAKEFIL
C
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
      CHARACTER * 10 CHCKONE
      CHARACTER * 72 IMAGE,INPLINE
      CHARACTER *  9 CHCKTWO
      CHARACTER *  9 SEQID
      CHARACTER *  7 SEQNR
      CHARACTER * 25 CHCKTHR
      CHARACTER *  1 CHARRAY(132)
      EQUIVALENCE(CHARRAY(  1),LSTLINE)
      EQUIVALENCE(CHARRAY(  1),CHCKONE)
      EQUIVALENCE(CHARRAY( 11),IMAGE  )
      EQUIVALENCE(CHARRAY( 83),CHCKTWO)
      EQUIVALENCE(CHARRAY( 92),SEQID  )
      EQUIVALENCE(CHARRAY(101),SEQNR  )
      EQUIVALENCE(CHARRAY(108),CHCKTHR)
      CHARACTER * 7 LSTFILE,WITHSEQ,WITHOUT,INPFILE
      CHARACTER * 9 DEKNAME
      CHARACTER *10 MOLDONE(MXNRDEK)
      CHARACTER * 7 MOLDTWO
      CHARACTER *24 MOLDTHR
      CHARACTER *40 DUMSTR
      PARAMETER(LSTUNIT=1,WITHUNT=2,WOUTUNT=3,INPUNIT=4,DUMUNIT=5)
      PARAMETER(MOLDTWO='       ')
      PARAMETER(MOLDTHR='     A                  ')
      LOGICAL FULLUP
C
      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.GT.0) THEN
               CALL GETPARM(INPFILE,DUMSTR,PARSTAT)
               IF(PARSTAT.LT.0) THEN
                  FULLUP=.TRUE.
                  GOTO 10
               ELSE
                  FULLUP=.FALSE.
                  GOTO 10
               ENDIF
            ENDIF
         ENDIF
      ENDIF
      CALL ERRMSG(1)
C
   10 OPEN(UNIT=LSTUNIT,FILE=LSTFILE,STATUS='OLD',ERR=12)
      OPEN(UNIT=WITHUNT,FILE=WITHSEQ,ERR=13)
      OPEN(UNIT=WOUTUNT,FILE=WITHOUT,ERR=14)
      IF(.NOT.FULLUP)
     A   OPEN(UNIT=INPUNIT,FILE=INPFILE,STATUS='OLD',ERR=15)
      GOTO 20
   12 CALL ERRMSG(2)
   13 CALL ERRMSG(3)
   14 CALL ERRMSG(4)
   15 CALL ERRMSG(5)
C
   20 REWIND(LSTUNIT)
      REWIND(WITHUNT)
      REWIND(WOUTUNT)
C
      IF(.NOT.FULLUP) THEN
         RANGE=0
         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
               RANGE=1
               NROFDEK=NROFDEK+1
               MOLDONE(NROFDEK)=' '//INPLINE(L:L+K-1)
               IF(NROFDEK.GE.MXNRDEK) GOTO 50
            ELSE
   40          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 30
                  ELSE
                     CALL ERRMSG(7)
                  ENDIF
               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.'1UNLABELED OLDPL') THEN
         IF(.NOT.FULLUP) THEN
            OPEN(UNIT=DUMUNIT,FILE='DUMFILE',ERR=53)
            GOTO 54
   53       CALL ERRMSG(10)
   54       REWIND(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,I7)') NUM
            WRITE(WITHUNT,'(A,1X,A,''.'',I7.7)') IMAGE,SEQID(1:I-1),NUM
            WRITE(WOUTUNT,'(A)') IMAGE
            IF(.NOT.FULLUP) WRITE(DUMUNIT,'(A)') LSTLINE
            NLINE=NLINE+1
         ENDIF
      ENDIF
      GOTO 60
C
  100 IF(.NOT.FULLUP) THEN
         REWIND(DUMUNIT)
         REWIND(LSTUNIT)
         DO 110 I=1,10000
            READ(DUMUNIT,'(A)',END=115) LSTLINE
            WRITE(LSTUNIT,'(A)') LSTLINE
  110    CONTINUE
  115    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
      END
C =====================================================================
      SUBROUTINE ERRMSG(I)
      CHARACTER *40 MSG
      WRITE(MSG,'(''** PROGRAM MAKEFIL - ERROR NR'',I3,'' **'',5X)') I
      CALL REMARK(MSG)
      CALL ABORT
      END
          IDENT ABORT
          ENTRY ABORT
 ABORT
          ABORT ,ND
 LABEL    XJ
          EQ    LABEL
```

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

         END

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

      PROGRAM MODGEN
C
C     ****************************************************************
C     * COMPARE TWO FILES (NEWFILE AND OLDFILE) AND PREPARE AN UPDATE  *
C     * CORRECTION SET IN MODFILE.                                     *
C     ****************************************************************
C
      IMPLICIT INTEGER (A-Z)
C
      PARAMETER(OLDBFL =1000)
      PARAMETER(NEWBFL =1000)
      PARAMETER(LINELEN=  72)
      PARAMETER(SEQLENG=  18)
      PARAMETER(SINGLEN=  40)
      PARAMETER(RANGLEN=  40)
      PARAMETER(BIGNUM =1000)
      PARAMETER(MINMATS=   5)
C
      CHARACTER *(LINELEN) OLDLINE(0:OLDBFL-1)
      CHARACTER *(LINELEN) NEWLINE(0:NEWBFL-1)
      CHARACTER *(LINELEN) LINE
      CHARACTER *(SEQLENG) SEQNR  (0:OLDBFL-1)
      CHARACTER *(   *   ) IDENT,INSERT,DELETE,BEFORE,FORMAT,LINEFMT
      CHARACTER *(SEQLENG) LIN1SEQ,SEQONE,SEQTWO,SEQTHRE
      CHARACTER *(RANGLEN) RANGE
      CHARACTER *(SINGLEN) SINGLE
      CHARACTER *        7  OLDFILE,NEWFILE,MODFILE
      CHARACTER *        9  CSIDENT
      CHARACTER *       10  DUMSTR1,DUMSTR2
      LOGICAL               OLDEOF,NEWEOF
C
      COMMON/OLD/OLDFILE,OLDLINE,SEQNR
      COMMON/NEW/NEWFILE,NEWLINE
C
      PARAMETER(OLDUNIT=1)
      PARAMETER(NEWUNIT=2)
      PARAMETER(MODUNIT=3)
      PARAMETER(FORMAT ='(A,A,A)')
      PARAMETER(LINEFMT='(A)     ')
      PARAMETER(IDENT  ='*IDENT ')
      PARAMETER(INSERT ='*I')
      PARAMETER(DELETE ='*D')
      PARAMETER(BEFORE ='*B')
C
C     ****************************************************************
C     * READ CONTROL STATEMENT PARAMETERS. THE PARAMETERS ARE :       *
C     *    OLDFILE : LFN OF FILE CONTAINING OLD DECK.                 *
C     *    NEWFILE : LFN OF FILE CONTAINING NEW DECK.                 *
C     *    MODFILE : LFN OF FILE TO RECEIVE CORRECTIONS SET.          *
C     *    CSIDENT : CORRECTION SET IDENTIFIER.                       *
C     * ALL PARAMETERS ARE REQUIRED. NO DEFAULTS ARE DEFINED.         *
C     ****************************************************************
C
      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
      CALL ERROR(1)
   10 CONTINUE
C
C     ****************************************************************
C     * IF THE CORRECTION SET IDENTIFIER CSIDENT STILL HAS THE DEFAULT *
C     * VALUE "MOD", IT IS CHANGED TO A NAME DERIVED FROM THE NAME OF  *
C     * THE MODIFICATION DECK MODFILE, PROVIDED THE LATTER HAS BEEN    *
C     * SPECIFIED TO BE DIFFERENT FROM "MODFILE".                      *
```

```
C     ******************************************************************
C
      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
   11    CONTINUE
   12    CSIDENT='MOD'//MODFILE(J:K)
      ENDIF
C
C     ******************************************************************
C     * OPEN THE FILES. OLDFILE AND NEWFILE MUST NOT BE EMPTY.         *
C     * READ SEQUENCE LABEL OF FIRST LINE OF OLDFILE AND MEMORIZE      *
C     * IT IN LOCATION LIN1SEQ. WRITE IDENTIFICATION LINE ON MODFILE.  *
C     ******************************************************************
C
      CALL OPENOLD(DUMMY1)
      CALL OPENNEW(DUMMY1)
      OPEN(UNIT=MODUNIT,FILE=MODFILE,ERR=20)
      GOTO 30
   20 CALL ERROR(4)
   30 CONTINUE
      REWIND(MODUNIT)
      CALL READOLD(1,KOLD,DUMMY1,OLDEOF)
      IF(OLDEOF) CALL ERROR(5)
      LIN1SEQ=SEQNR(KOLD)
      CALL READNEW(1,DUMMY1,DUMMY2,NEWEOF)
      IF(NEWEOF) CALL ERROR(6)
      WRITE(MODUNIT,FORMAT) IDENT,CSIDENT
C
C     ******************************************************************
C     * READ OLDFILE AND NEWFILE IN PARALLEL UNTIL TWO DIFFERENT LINES *
C     * ARE ENCOUNTERED OR EOF IS HIT.                                 *
C     ******************************************************************
C
      IOLD=0
      INEW=0
C
   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
C
C     ******************************************************************
C     * FILL THE BUFFERS OF OLDFILE AND NEWFILE.                       *
C     * THEN RUN ALONG SUBSEQUENT DIAGONALS OF CONSTANT SUM OF OLD AND *
C     * NEW LINE NUMBER.                                               *
C     * CONTINUE UNTIL BUFFERS ARE EXHAUSTED OR A MATCH IS FOUND.      *
C     ******************************************************************
C
   50 CONTINUE
         CALL READOLD(IOLD+OLDBFL-1,DUMMY1,NOLD,OLDEOF)
         CALL READNEW(INEW+NEWBFL-1,DUMMY1,NNEW,NEWEOF)
         DO 60 SUM=IOLD+INEW+3,NOLD+NNEW
         DO 60 JOLD=MAX0(IOLD+1,SUM-NNEW),MIN0(NOLD,SUM-INEW-1)
            JNEW=SUM-JOLD
            CALL READOLD(JOLD,KOLD,DUMMY1,DUMMY2)
            CALL READNEW(JNEW,KNEW,DUMMY1,DUMMY2)
            IF(OLDLINE(KOLD).EQ.NEWLINE(KNEW)) THEN
               IF(OLDLINE(KOLD)(1:5).EQ.'*DECK'.OR.
     A            OLDLINE(KOLD)(1:8).EQ.'*COMDECK') THEN
                  ACTMATS=1
               ELSEIF(JOLD+MINMATS-1.LE.NOLD.AND.
     A                JNEW+MINMATS-1.LE.NNEW) THEN
                  DO 51 I=1,MINMATS-1
                     CALL READOLD(JOLD+I,KOLD,DUMMY1,DUMMY2)
                     CALL READNEW(JNEW+I,KNEW,DUMMY1,DUMMY2)
                     IF(OLDLINE(KOLD).NE.NEWLINE(KNEW)) GOTO 60
   51             CONTINUE
                  ACTMATS=MINMATS
               ELSE
                  IF(.NOT.(OLDEOF.OR.NEWEOF)) GOTO 130
                  ACTMATS=MIN0(NOLD-JOLD+1,NNEW-JNEW+1)
                  DO 52 I=1,ACTMATS-1
```

```
                     CALL READOLD(JOLD+I,KOLD,DUMMY1,DUMMY2)
                     CALL READNEW(JNEW+I,KNEW,DUMMY1,DUMMY2)
                     IF(OLDLINE(KOLD).NE.NEWLINE(KNEW)) GOTO 60
   52             CONTINUE
               ENDIF
               GOTO 70
            ENDIF
   60    CONTINUE
      IF(OLDEOF.OR.NEWEOF) GOTO 90
      GOTO 130
C
C     ******************************************************************
C     * RECORD APPROPRIATE MODIFICATIONS IN MODFILE.                   *
C     ******************************************************************
C
   70 CONTINUE
         LOLD=JOLD-IOLD-1
         IF(LOLD.GT.1) THEN
            CALL READOLD(IOLD+1,KOLD1,DUMMY1,DUMMY2)
            CALL READOLD(JOLD-1,KOLD2,DUMMY1,DUMMY2)
            WRITE(MODUNIT,FORMAT)DELETE,RANGE(SEQNR(KOLD1),SEQNR(KOLD2))
         ELSEIF(LOLD.EQ.1) THEN
            CALL READOLD(IOLD+1,KOLD,DUMMY1,DUMMY2)
            WRITE(MODUNIT,FORMAT) DELETE,SINGLE(SEQNR(KOLD))
         ELSEIF(IOLD.GT.0) THEN
            CALL READOLD(IOLD,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(KNEW)
   80    CONTINUE
         IOLD=JOLD+ACTMATS-1
         INEW=JNEW+ACTMATS-1
      GOTO 40
C
C     ******************************************************************
C     * POST EOF PROCESSING.                                           *
C     ******************************************************************
C
   90 CONTINUE
         IF(IOLD.EQ.0.OR..NOT.(OLDEOF.OR.NEWEOF)) GOTO 130
         CALL READNEW(INEW+NEWBFL,DUMMY1,NNEW,NEWEOF)
         CALL READOLD(IOLD,KOLD,DUMMY1,DUMMY2)
         SEQONE=SEQNR(KOLD)
         CALL READOLD(IOLD+1,KOLD,NOLD,OLDEOF)
         IF(NOLD.GT.IOLD) SEQTWO=SEQNR(KOLD)
  100    IF(.NOT.OLDEOF) THEN
            CALL READOLD(NOLD+BIGNUM,DUMMY1,NOLD,OLDEOF)
            GOTO 100
         ENDIF
         CALL READOLD(NOLD,KOLD,DUMMY1,DUMMY2)
         SEQTHRE=SEQNR(KOLD)
         LOLD=NOLD-IOLD
         IF(LOLD.GT.1) THEN
            WRITE(MODUNIT,FORMAT) DELETE,RANGE(SEQTWO,SEQTHRE)
         ELSEIF(LOLD.EQ.1) THEN
            WRITE(MODUNIT,FORMAT) DELETE,SINGLE(SEQTWO)
         ELSEIF(NNEW.GT.INEW.OR..NOT.NEWEOF) THEN
            WRITE(MODUNIT,FORMAT) INSERT,SINGLE(SEQONE)
         ENDIF
  110    CONTINUE
            INEW=INEW+1
            CALL READNEW(INEW,KNEW,DUMMY1,NEWEOF)
            IF(NEWEOF) GOTO 120
            WRITE(MODUNIT,LINEFMT) NEWLINE(KNEW)
         GOTO 110
  120    CONTINUE
         CLOSE(MODUNIT,STATUS='KEEP')
         CLOSE(OLDUNIT)
         CLOSE(NEWUNIT)
         STOP 'PROGRAM MODGEN'
C
C     ******************************************************************
C     * IF NO MATCH HAS BEEN FOUND, PREPARE CORRECTION SET THAT        *
C     * REPLACES A FULL DECK.                                          *
C     ******************************************************************
C
```

```
 130 CONTINUE
        REWIND(NEWUNIT)
        REWIND(MODUNIT)
        WRITE(MODUNIT,FORMAT) IDENT,CSIDENT
 140    CONTINUE
           IOLD=NOLD+1
           CALL READOLD(IOLD+BIGNUM,DUMMY1,NOLD,OLDEOF)
        IF(.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)
        ENDIF
 150    CONTINUE
           READ(NEWUNIT,LINEFMT,END=160) LINE
           WRITE(MODUNIT,LINEFMT) LINE
        GOTO 150
 160    CONTINUE
        CLOSE(MODUNIT,STATUS='KEEP')
        CLOSE(OLDUNIT)
        CLOSE(NEWUNIT)
        STOP 'PROGRAM MODGEN'
     END
C ==================================================================
     SUBROUTINE READOLD(RECNR,BUFLOC,LASTNR,EOFLAG)
C
     IMPLICIT INTEGER(A-Z)
C
     PARAMETER(BFL     =1000)
     PARAMETER(LINELEN=  72)
     PARAMETER(SEQLENG=  18)
     PARAMETER(OLDUNIT=   1)
C
     LOGICAL EOF,EOFLAG
     CHARACTER *       7  OLDFILE
     CHARACTER *(LINELEN) LINE (0:BFL-1)
     CHARACTER *(SEQLENG) SEQNR(0:BFL-1)
     COMMON/OLD/OLDFILE,LINE,SEQNR
C
     SAVE BEGIN,EIND,BUFPNT,EOF
C
     IF(RECNR.LT.BEGIN) CALL ERROR(7)
     IF(.NOT.EOF) THEN
  10    IF(RECNR.GT.EIND ) THEN
           I=MOD(BUFPNT+EIND-BEGIN+1,BFL)
           READ(OLDUNIT,'(A,A)',END=20) LINE(I),SEQNR(I)
           EIND=EIND+1
           IF(EIND-BEGIN+1.GT.BFL) THEN
              BEGIN=BEGIN+1
              BUFPNT=MOD(BUFPNT+1,BFL)
           ENDIF
           GOTO 10
        ENDIF
     ENDIF
     IF(RECNR.LE.EIND) THEN
        BUFLOC=MOD(BUFPNT+RECNR-BEGIN,BFL)
        LASTNR=EIND
        EOFLAG=.FALSE.
        RETURN
     ENDIF
C
  20 LASTNR=EIND
     EOF=.TRUE.
     EOFLAG=.TRUE.
     RETURN
C
     ENTRY OPENOLD(DUMMY)
     OPEN(UNIT=OLDUNIT,FILE=OLDFILE,ERR=30,STATUS='OLD')
     REWIND(OLDUNIT)
     EOF=.FALSE.
     BEGIN=0
     EIND=0
     BUFPNT=0
     RETURN
  30 CALL ERROR(2)
     END
C ==================================================================
     SUBROUTINE READNEW(RECNR,BUFLOC,LASTNR,EOFLAG)
C
```

```
      IMPLICIT INTEGER(A-Z)
C
      PARAMETER(BFL     =1000)
      PARAMETER(LINELEN=  72)
      PARAMETER(NEWUNIT=   2)
C
      LOGICAL EOF,EOFLAG
      CHARACTER *       7  NEWFILE
      CHARACTER *(LINELEN) LINE (0:BFL-1)
      COMMON/NEW/NEWFILE,LINE
C
      SAVE BEGIN,EIND,BUFPNT,EOF
C
      IF(RECNR.LT.BEGIN) CALL ERROR(8)
      IF(.NOT.EOF) THEN
   10    IF(RECNR.GT.EIND ) THEN
            I=MOD(BUFPNT+EIND-BEGIN+1,BFL)
            READ(NEWUNIT,'(A)',END=20) LINE(I)
            EIND=EIND+1
            IF(EIND-BEGIN+1.GT.BFL) THEN
               BEGIN=BEGIN+1
               BUFPNT=MOD(BUFPNT+1,BFL)
            ENDIF
            GOTO 10
         ENDIF
      ENDIF
      IF(RECNR.LE.EIND) THEN
         BUFLOC=MOD(BUFPNT+RECNR-BEGIN,BFL)
         LASTNR=EIND
         EOFLAG=.FALSE.
         RETURN
      ENDIF
C
   20 LASTNR=EIND
      EOF=.TRUE.
      EOFLAG=.TRUE.
      RETURN
C
      ENTRY OPENNEW(DUMMY)
      OPEN(UNIT=NEWUNIT,FILE=NEWFILE,ERR=30,STATUS='OLD')
      REWIND(NEWUNIT)
      EOF=.FALSE.
      BEGIN=0
      EIND=0
      BUFPNT=0
      RETURN
   30 CALL ERROR(3)
      END
C ====================================================================
      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
   20 SINGLE=A(1:I)//A(J:K)
      END
C ====================================================================
      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:JA).NE.'0') GOTO 20
   10 CONTINUE
   20 LB=LEN(B)
      KB=INDEX(B(2:LB),' ')
      IF(KB.EQ.0) KB=LB
      IB=INDEX(B(1:KB),'.')
      DO 30 JB=IB+1,KB
         IF(B(JB:JB).NE.'0') GOTO 40
   30 CONTINUE
   40 IF(A(2:IA).EQ.B(2:IB)) THEN
         RANGE=A(1:IA)//A(JA:KA)//','//B(JB:KB)
      ELSE
```

```
--- MASTERFILE MFCCL CY=30 ---- 02/07/86 - 00.27.57. -------- MODGEN,  6 ------

         RANGE=A(1:IA)//A(JA:KA)//','//B(2:IB)//B(JB:KB)
      ENDIF
      END
C ===================================================================
      SUBROUTINE ERROR(I)
      CHARACTER *40 MSG
      WRITE(MSG,'(''** PROGRAM MODGEN - ERROR NR'',I3,'' **'',6X)') I
      CALL REMARK(MSG)
      CALL ABORT
      END


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

      PROGRAM USL
C
C     ***************************************************************
C     * PROGRAM REFORMATING OUTPUT OF UPDATE SOURCE LISTING OBTAINED  *
C     * FROM A CREATION RUN OR FROM AN AUDIT RUN OF OLDPL.            *
C     * AFTER "UPDATE,F,I=SOURCE,N,L=A124,O=FN.",                     *
C     *    OR "UPDATE,F,P=OLDPL,L=7,O=FN.",                           *
C     * CALL: "USL,FN(,NOLIST)."                                      *
C     * THE PARAMETER "NOLIST" SWITCHES OFF THE COMPLETE LISTING OF   *
C     * CARDS ENCOUNTERED IN INPUT OR ACTIVE CARDS ON OLDPL (*COMDECKS *
C     * AND *DECKS).                                                  *
C     ***************************************************************
C
      CHARACTER*133 LINE,BLANK,ZERO
      CHARACTER*116 HEADER
      CHARACTER* 30 MASTER
      CHARACTER* 26 CARDS
      CHARACTER* 16 CORR,HDR2,LISTOFC
      CHARACTER* 14 HDR1
      CHARACTER* 12 COMMON,COMPIL
      CHARACTER* 10 YANKDCK,ACTIVE,STARS,SLASHES
      CHARACTER*  8 SCDECK
      CHARACTER*  7 FN,DUM,SERROR,NAME
      CHARACTER*  6 SLIMIT
      CHARACTER*  5 SDECK,SREAD
      CHARACTER*  4 DECK
      CHARACTER*  1 NOLIST,NY
      DATA CARDS  /'CARDS ENCOUNTERED IN INPUT'/
      DATA MASTER /'MASTER AUDIT, IDENT CARD TOTAL'/
      DATA LISTOFC/'LIST OF CONTROL,'/
      DATA YANKDCK/' YANK$$$  '/
      DATA ACTIVE /'     A    '/
      DATA STARS  /' *****    '/
      DATA SLASHES/' /////    '/
      DATA SLIMIT /'*LIMIT'/
      DATA SREAD  /'*READ'/
      DATA SCDECK /'*COMDECK'/
      DATA SDECK  /'*DECK'/
      DATA SERROR /'*ERROR*'/
      DATA HDR1   /'1CREATION RUN '/
      DATA HDR2   /'1UNLABELED OLDPL'/
      DATA CORR   /'CORRECTION IDENT'/
      DATA BLANK  /' '/
      DATA ZERO   /'0'/
      DATA DECK   /'DECK'/
      DATA COMMON /'COMMON DECKS'/
      DATA COMPIL /'COMPILE FILE'/
C
      IPMAX=150
      ILMAX=60
      LMAX=IPMAX*ILMAX
C
      CALL GETPARM(FN,DUM,IDUM)
      OPEN(10,FILE=FN)
      OPEN(20,FILE='OUTPUT')
      REWIND 10
      CALL GETPARM(NOLIST,NY,INOLIST)
      IF(INOLIST.EQ.-1) NOLIST='N'
      IF(INOLIST.EQ. 1) NOLIST='Y'
      IF(INOLIST.EQ. 0) THEN
         IF(NY.EQ.'1') NY='Y'
         NOLIST=NY
      ENDIF
C
C     * START READING UPDATE OUTPUT FILE.
      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') GOTO 200
         IF(LINE(31:61).NE.MASTER) THEN
            HEADER=LINE(1:116)
            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) LINE
      ENDIF
   10 CONTINUE
      GOTO 100
C
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=0
         DO 25 I=1,10000
         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
         ENDIF
   25    CONTINUE
         GOTO 200
      ENDIF
      L=1
      N=1
      LYA=0
      NKEEP=0
C1    * START LOOP ON IP.
      IP=1
   30 WRITE(20,2) HEADER,IP
C2     * START LOOP ON IL.
      IL=1
      IF(NKEEP.EQ.1) THEN
         WRITE(20,3) L,LINE(11:82),LINE(92:98),LINE(104:107)
         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((NSS.EQ.1).AND.(LINE(11:16).EQ.SLIMIT)) THEN
         WRITE(20,13) LINE(11:82)
      ELSEIF(LINE(1:10).EQ.YANKDCK) THEN
         LYA=LYA+1
         IF(LYA.EQ.1) THEN
            WRITE(20,1) '0      IN YANK$$$ DECK:'
            IL=IL+2
         ENDIF
         WRITE(20,14) LINE(11:82),LINE(92:98),LINE(104:107)
      ELSEIF((NSS.EQ.1).AND.(LINE(11:15).EQ.SREAD)) THEN
         WRITE(20,13) LINE(11:82)
      ELSEIF((NSS.EQ.1).AND.(LINE(11:18).EQ.SCDECK)) THEN
         IF(N.EQ.1) THEN
            WRITE(20,6)
            IL=IL+4
            N=2
         ENDIF
         NAME=LINE(20:26)
         WRITE(20,*)
```

```
        IL=IL+1
        IEQ=INDEX(LINE,'==')
        IF(IEQ.GT.0) THEN
          WRITE(20,15) BLANK(1:(IEQ-10)/2)//LINE(IEQ:82)
          LINE(IEQ:82)=' '
          IL=IL+3
        ENDIF
        IF(IL.GT.ILMAX) THEN
          NKEEP=1
          GOTO 45
        ENDIF
        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((N.EQ.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
          ENDIF
          WRITE(20,7)
          IL=IL+4
          N=3
        ENDIF
        NAME=LINE(17:23)
        WRITE(20,*)
        IL=IL+1
        IEQ=INDEX(LINE,'==')
        IF(IEQ.GT.0) THEN
          WRITE(20,15) BLANK(1:(IEQ-10)/2)//LINE(IEQ:82)
          LINE(IEQ:82)=' '
          IL=IL+3
        ENDIF
        IF(IL.GT.ILMAX) THEN
          NKEEP=1
          GOTO 45
        ENDIF
        WRITE(20,3) L,LINE(11:82),LINE(92:98),LINE(104:107)
        L=L+1
      ELSEIF(LINE(2:8).EQ.SERROR) THEN
        WRITE(20,4) LINE(2:133)
      ELSEIF(((LOPT.EQ.4).AND.(LINE(92:98).EQ.NAME)).OR.
     A           ((LOPT.EQ.7).AND.(LINE(108:117).EQ.ACTIVE))) THEN
        WRITE(20,3) L,LINE(11:82),LINE(92:98),LINE(104:107)
        L=L+1
      ELSEIF((LINE(1:14).EQ.HDR1).OR.(LINE(1:16).EQ.HDR2)) THEN
        GOTO 40
      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
C2    * END LOOP ON IL.
   45 IP=IP+1
      IF(IP.LE.IPMAX) GOTO 30
C1    * END LOOP ON IP.
      GOTO 200
C
C     * LAST PAGES OF UPDATE OUTPUT FILE (LISTING OF DECKNAMES).
   50 K=1
      L=1
      M=1
   60 READ(10,1,ERR=100,END=200) LINE
      IF((INDEX(LINE,DECK).NE.0).AND.(K.EQ.1)) THEN
        IP=IP+1
        HEADER(31:60)='DECK LIST AS WRITTEN, 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
```

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

      ELSEIF((LINE.EQ.BLANK).OR.(LINE.EQ.ZERO)
     A       .OR.(LINE(1:1).EQ.'1')) THEN
      ELSE
         IF(LINE(3:13).EQ.'THIS UPDATE') WRITE(20,1) '0'
         IF(K.NE.1) WRITE(20,1) LINE
      ENDIF
      GOTO 60
C
  100 STOP '**ERROR IN USL**'
  200 STOP 'PROGRAM USL'
C
C     * FORMATS.
    1 FORMAT(A)
    2 FORMAT(A116,'PAGE ',I3/)
    3 FORMAT(' ',I4,'0 ',A72,' ',A7,'.',A4)
    4 FORMAT(' ',A132)
    6 FORMAT(/37X,'==============',
     A       /37X,'= *COMDECKS  =',
     B       /37X,'==============')
    7 FORMAT(/37X,'===========',
     A       /37X,'= *DECKS  =',
     B       /37X,'===========')
    8 FORMAT(26X,'(LISTING OF CORRECTION IDENTS SKIPPED)')
    9 FORMAT(/'0DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A',
     A       ' NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE.'/)
   11 FORMAT(//10X,'COMMON DECKS ENCOUNTERED'/)
   12 FORMAT(//10X,'DECKS WRITTEN TO COMPILE FILE'/)
   13 FORMAT(' ',6X,A72)
   14 FORMAT(' ',6X,A72,' ',A7,'.',A4)
   15 FORMAT(/,' ',6X,A,/)
      END
```

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

      PROGRAM UML
C
C     ****************************************************************
C     * PROGRAM REFORMATING OUTPUT OF UPDATE MODIFICATIONS LISTING OF  *
C     * A CORRECTION RUN.                                              *
C     * AFTER "UPDATE,F,P=OLDPL,I=MOD,N,L=A1234,O=FN.",                *
C     * CALL: "UML,FN(,LIST).".                                        *
C     * THE PARAMETER "LIST" SWITCHES THE LISTING OF MODIFICATIONS ON. *
C     ****************************************************************
C
      CHARACTER*133 LINE,BLANK,ZERO
      CHARACTER*116 HEADER
      CHARACTER* 72 THREE
      CHARACTER* 30 LEFT,HDR1,HDR2,HDR3
      CHARACTER* 16 MODIF,CORR
      CHARACTER* 14 VALUES
      CHARACTER* 12 COMMON,COMPIL
      CHARACTER* 10 ONE,UNLAB,STARS,SLASHES,SERROR
      CHARACTER*  7 FN,DUM,NAME,NAME2
      CHARACTER*  4 DECK
      CHARACTER*  3 TWO,SIDENT,SCOMNT,SYANK,SCALL,SDELETE,SINSERT
      CHARACTER*  1 LIST,NY
C
      DATA BLANK  /' '/
      DATA ZERO   /'0'/
      DATA UNLAB  /'1UNLABELED'/
      DATA STARS  /' *****    '/
      DATA SLASHES/' /////    '/
      DATA SERROR /' *ERROR*  '/
      DATA SIDENT /'*ID'/
      DATA SCOMNT /'*/ '/
      DATA SYANK  /'*YA'/
      DATA SCALL  /'*CA'/
      DATA SDELETE/'*DE'/
      DATA SINSERT/'*IN'/
      DATA LEFT   /'1CORRECTION RUN               '/
      DATA HDR1   /'CARDS ENCOUNTERED IN INPUT    '/
      DATA HDR2   /'MODIFICATIONS PROGRAM LIBRARY '/
      DATA HDR3   /'DECK LIST AS WRITTEN, IF NEWPL'/
      DATA MODIF  /'MODIFICATIONS / '/
      DATA CORR   /'CORRECTION IDENT'/
      DATA DECK   /'DECK'/
      DATA COMMON /'COMMON DECKS'/
      DATA VALUES /'VALUES DEFINED'/
      DATA COMPIL /'COMPILE FILE'/
```

```
C
      IPMAX=150
      ILMAX=60
      LMAX=IPMAX*ILMAX
C
      CALL GETPARM(FN,DUM,IDUM)
      OPEN(10,FILE=FN)
      OPEN(20,FILE='OUTPUT')
      REWIND 10
      CALL GETPARM(LIST,NY,ILIST)
      IF(ILIST.EQ.-1) LIST='N'
      IF(ILIST.EQ. 1) LIST='Y'
      IF(ILIST.EQ. 0) THEN
         IF(NY.EQ.'1') NY='Y'
         LIST=NY
      ENDIF
C
C1    * START READING UPDATE OUTPUT FILE.
      DO 10 I=1,1000
      READ(10,1,END=100) LINE
      IF(LINE(1:10).EQ.UNLAB) THEN
         HEADER=LEFT//HDR1//LINE(61:116)
         IF(INDEX(LINE,MODIF).NE.0) THEN
            IP=0
            GOTO 50
         ENDIF
         GOTO 20
      ELSE
         WRITE(20,1) LINE
      ENDIF
   10 CONTINUE
      GOTO 100
C
C2    * REFORMATING "CARDS ENCOUNTERED IN INPUT".
   20 L=1
      N=1
      NERR=0
      DO 40 IP=1,IPMAX
      WRITE(20,2) HEADER,IP
C     * START LOOP ON IL.
      IL=1
   30 READ(10,1,END=50,ERR=100) LINE
      ONE=LINE(1:10)
      TWO=LINE(11:13)
      THREE=LINE(11:82)
      IF((LINE.EQ.BLANK).OR.(LINE.EQ.ZERO)) THEN
         GOTO 30
      ELSEIF(INDEX(LINE,MODIF).NE.0) THEN
         GOTO 50
      ELSEIF(INDEX(LINE,CORR).NE.0) THEN
         GOTO 80
      ELSEIF(ONE.EQ.UNLAB) THEN
         GOTO 30
      ELSEIF((ONE.EQ.STARS).OR.(ONE.EQ.SLASHES)) THEN
         IF(TWO.EQ.SIDENT) THEN
            IF(N.EQ.1) THEN
               WRITE(20,7)
               IL=IL+4
               N=2
            ENDIF
            WRITE(20,5) BLANK
            IL=IL+1
            NAME=LINE(18:24)
            NR=1
            WRITE(20,3) L,THREE
         ELSEIF(TWO.EQ.SCOMNT) THEN
            WRITE(20,3) L,THREE
         ELSEIF((TWO.EQ.SYANK).OR.(TWO.EQ.SCALL).OR.
     A           (LINE(11:17).EQ.'*DEFINE').OR.
     B           (LINE(11:15).EQ.'*DECK').OR.
     C           (LINE(11:18).EQ.'*COMDECK')) THEN
            WRITE(20,4) L,THREE,NAME,NR
            NR=NR+1
         ELSEIF((TWO.EQ.SDELETE).OR.(TWO.EQ.SINSERT)) THEN
            DO 35 M=19,28
            IF(LINE(M:M).NE.BLANK) GOTO 36
   35       CONTINUE
   36       THREE=LINE(11:18)//LINE(M:82)
            WRITE(20,3) L,THREE
         ENDIF
      ELSEIF(ONE.NE.BLANK) THEN
         IF(ONE.EQ.SERROR) THEN
            NERR=NERR+1
            IF(NERR.GT.20) GOTO 100
         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
      IL=IL+1
      IF(IL.LE.ILMAX) GOTO 30
C     * END LOOP ON IL.
   40 CONTINUE
      GOTO 100
C
C3   * REFORMATING "MODIFICATIONS".
   50 IF((LIST.NE.'Y').OR.(NERR.NE.0)) THEN
         IPP=IP
         GOTO 81
      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
   60 READ(10,1,END=100) LINE
      IF(INDEX(LINE,CORR).NE.0) THEN
         GOTO 80
      ELSEIF(LINE(2:133).NE.BLANK) THEN
         IF((LINE(114:114).EQ.'I').OR.(LINE(116:116).EQ.'D')) THEN
            WRITE(20,6) LINE(2:83),LINE(92:98),
     A                  LINE(104:107),LINE(112:133)
            IB=1
            IF(LINE(114:114).EQ.'I') THEN
               NAME2=LINE(92:98)
               READ(LINE(104:107),'(I4)') NR2
               NR2=NR2+1
            ENDIF
         ELSEIF(LINE(2:4).EQ.'***') THEN
            WRITE(20,1) LINE
         ELSE
            GOTO 60
         ENDIF
      ELSEIF(LINE.EQ.ZERO) THEN
         IF((IL.NE.1).AND.(IB.LE.2)) THEN
            WRITE(20,5) BLANK
            IB=IB+2
         ELSE
            GOTO 60
         ENDIF
      ELSEIF(LINE.EQ.BLANK) THEN
         GOTO 60
      ENDIF
      IL=IL+1
      IF(IL.LE.ILMAX) GOTO 60
C     * END LOOP ON IL.
   70 CONTINUE
      GOTO 100
C
C     * LAST PAGES OF UPDATE OUTPUT FILE (LISTING OF DECKNAMES).
   80 IF((NAME2.NE.NAME).OR.(NR2.NE.NR)) GOTO 101
   81 K=1
      L=1
      M=1
      N=1
   90 READ(10,1,ERR=100,END=200) LINE
```

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

      IF(((INDEX(LINE,'DECK LIST AS WRITTEN').NE.0).OR.
     A    (INDEX(LINE,'0DECKS ARE LISTED').NE.0)).AND.(K.EQ.1)) THEN
         IP=IPP+1
         HEADER(1:60)=LEFT//HDR3
         WRITE(20,2) HEADER,IP
         WRITE(20,9)
         WRITE(20,11)
         K=2
      ELSEIF((INDEX(LINE,COMMON).NE.0).AND.(L.EQ.1)) THEN
         WRITE(20,12)
         L=2
      ELSEIF((INDEX(LINE,VALUES).NE.0).AND.(M.EQ.1)) THEN
         WRITE(20,13)
         M=2
      ELSEIF((INDEX(LINE,COMPIL).NE.0).AND.(N.EQ.1)) THEN
         WRITE(20,14)
         N=2
      ELSEIF(LINE(2:4).EQ.'***') THEN
         WRITE(20,1) LINE
      ELSEIF((LINE.EQ.BLANK).OR.(LINE.EQ.ZERO)
     A       .OR.(LINE(1:10).EQ.UNLAB)
     B       .OR.(LINE(1:10).EQ.'0DECKS ARE')) THEN
      ELSE
         IF(LINE(3:13).EQ.'THIS UPDATE') WRITE(20,1) '0'
         IF(K.NE.1) WRITE(20,1) LINE
      ENDIF
      GOTO 90
C
  100 STOP '**ERROR IN UML**'
  101 STOP 'ERROR: MODIFICATIONS DO NOT MATCH INSTRUCTIONS ON *IDENT'
  200 STOP 'PROGRAM UML'
C
C     * FORMATS.
    1 FORMAT(A)
    2 FORMAT(A116,'PAGE ',I3/)
    3 FORMAT(' ',I4,'0 ',A72)
    4 FORMAT(' ',I4,'0 ',A72,' ',A7,'.',I4)
    5 FORMAT(' ',A132)
    6 FORMAT(' ',A82,A7,'.',A4,A22)
    7 FORMAT(/37X,'============',
     A       /37X,'= *IDENTS  =',
     B       /37X,'============')
    8 FORMAT(/37X,'=================',
     A       /37X,'= MODIFICATIONS =',
     B       /37X,'=================')
    9 FORMAT(26X,'(LISTING OF CORRECTION IDENTS SKIPPED)')
   11 FORMAT(/'0DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A',
     A        ' NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE'/)
   12 FORMAT(//10X,'COMMON DECKS ENCOUNTERED'/)
   13 FORMAT(//10X,'VALUES DEFINED FOR THIS UPDATE'/)
   14 FORMAT(//10X,'DECKS WRITTEN TO COMPILE FILE'/)
      END


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

.PROC,RUN205*I,
  NAME[NAME OF THE PROGRAM? (N/..) - ] = (*N=N,N=N,*F)\
  B   [BINARY? (N/..) -              ] = (*N=BIN,N=BIN,*F),
  G   [GOFILE? (N/..) -              ] = (*N=GOF,N=GOF,*F),
  NOEX[NO EXECUTION? (N/Y) -         ] = (*N=0,*K=1,N=0,Y=1,0,1),
  I   [INPUT FILE? (N/..) -          ] = (*N=N,N=N,*F),
  P   [PLOT FILE? (N/..) -           ] = (*N=N,N=N,*F),
  OPT [OPTIMIZATION? (N=1/1/DPRSV) - ] = (*N=1,*K=1,N=1,0,1,*S5(DPRSV)),
  UNS ['UNSAFE' OPTIMIZATION? (N/Y) -] = (*N=0,*K=1,N=0,Y=1,0,1),
  L   [FTN200 LISTING? (N/Y) -       ] = (*N=0,*K=1,N=0,Y=1,0,1),
  LO  [LIST OPTIONS? (N=S/SX/AMSX) - ] = (*N=S,*K=SX,N=S,*S4(AMSX)),
  TL  [TIME LIMIT? (N=100/..) -      ] = (*N=100,N=100,*S4(0123456789)),
  WS  [WORKING SET? (N=256/..) -     ] = (*N=256,N=256,*S4(0123456789)),
  LP  [NUMBER LARGE PAGES? (N=5/..) -] = (*N=5,N=5,*S2(0123456789)),
  NOR [NO ROUTE OF THE JOB? (N/Y) -  ] = (*N=0,*K=1,N=0,Y=1,0,1).
.*
.HELP,,NOLIST.
  RUN205  CREATES A JOB FOR THE 205 WHICH PERFORMS A COMPLETE COMPILE,
  ======  LOAD, AND EXECUTE SEQUENCE OF FORTRAN PROGRAM "NAME", OR ONLY
          PART OF IT RESULTING IN A PERMANENT BINARY "B" OR GOFILE "G".
          ALTERNATIVELY, IF "NAME" IS NOT SPECIFIED, A RUN IS PERFORMED
          STARTING FROM EITHER "B" OR "G".
          PARAMETERS:
          NAME - NAME OF THE FORTRAN PROGRAM
          B    - NAME OF THE BINARY TO BE COMPILED OR ATTACHED
          G    - NAME OF THE GOFILE 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 (DPRSV)
          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: "N"), EXECUTION REQUIRES
          SPECIFICATION OF A PREVIOUSLY DEFINED "B" OR "G".
.HELP,B,NOLIST.
  B       IS THE NAME OF THE BINARY FILE; DEFAULT: "BIN".
          IF "NAME" IS SPECIFIED, THE BINARY IS MADE PERMANENT UNDER
          THE NAME "B". IF "NAME" IS NOT SPECIFIED, EXECUTION STARTS BY
          ATTACHING AN EXISTING PERMFILE "B".
.HELP,G,NOLIST.
  G       IS THE NAME OF THE GOFILE; DEFAULT: "GOF".
          IF "NAME" IS SPECIFIED, THE GOFILE IS MADE PERMANENT UNDER
          THE NAME "G". IF "NAME" IS NOT SPECIFIED, EXECUTION STARTS BY
          ATTACHING AN EXISTING PERMFILE "G".
.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 COMPILE-TIME COMPUTABLE RESULTS
          R - REMOVE REDUNDANT CODE
          S - SCHEDULE INSTRUCTIONS
          V - VECTORIZE DO LOOPS
          DEFAULT (OPT=1), ALL THESE OPTIONS ARE SPECIFIED.
.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 ASSIGMENTS
          S - SOURCE LISTING
          X - CROSS REFERENCE MAP
          DEFAULT: LO=S.
.HELP,TL,NOLIST.
  TL      SPECIFIES THE TIME LIMIT. DEFAULT: 100.
.HELP,WS,NOLIST.
  WS      SPECIFIES THE WORKING SET SIZE LIMIT IN NUMBER OF BLOCKS
          (DECIMAL). DEFAULT: WS=256.
.HELP,LP,NOLIST.
  LP      SPECIFIES THE LARGE PAGE LIMIT (DECIMAL). DEFAULT: LP=5.
.HELP,NOR,NOLIST.
  NOR     INDICATES THAT THE JOB IS TO BE CREATED, BUT NOT ROUTED TO THE
          INPUT QUEUE OF THE CY205.
.ENDHELP.
.*
.IF,($NAME$.NE.$N$).AND.(.NOT.FILE(NAME,AS)), LERROR1.
NOTE,$FILE NAME DOES NOT EXIST; TRY AGAIN$.
RETURN,JOB,EDJOB,ZZUSE.
REVERT,ABORT.
.ENDIF, LERROR1.
.IF,($I$.NE.$N$).AND.(.NOT.FILE(I,AS)), LERROR2.
NOTE,$FILE I DOES NOT EXIST; TRY AGAIN$.
RETURN,JOB,EDJOB,ZZUSE.
REVERT,ABORT.
.ENDIF, LERROR2.
```

```
.*
.IF,NOR=0, LNOR.
EDJOB.
RETURN,FILMPL.
REWIND,JOB.
COPYBR,JOB,FILMPL.
.IF,$NAME$.NE.$N$, LNAME.
REWIND,NAME.
COPYBR,NAME,FILMPL.
.ENDIF, LNAME.
.IF,$I$.NE.$N$, LI.
REWIND,I.
COPYBR,I,FILMPL.
.ENDIF, LI.
ROUTE,FILMPL,DC=#IN,ST=205.
.ENDIF, LNOR.
RETURN,EDJOB,ZZUSE.
REVERT.
.*
EXIT,S.
NOTE,$ERROR$.
RETURN,EDJOB,ZZUSE.
REVERT,ABORT.
.*
.* JOB FOR THE 205.
.DATA,JOB.
XXI2X,ST205.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCE(#TL=TL,#WS=WS,#LP=LP,PRIO=12)
COMMENT.****************************************
COMMENT.  RUN205:
.IF,$NAME$.NE.$N$, LNAME.
COMMENT.  FORTRAN PROGRAM NAME COMPILED,
.IF(NOEX=0)COMMENT.  LOADED AND EXECUTED
.IF($I$.NE.$N$)COMMENT.  WITH INPUT I.
.IF($B$.NE.$BIN$)COMMENT.  BINARY B MADE PERMANENT.
.IF($G$.NE.$GOF$)COMMENT.  GOFILE G MADE PERMANENT.
.ELSE, LNAME.
.IF($B$.NE.$BIN$)COMMENT.  COMPILE, LOAD, AND EXECUTE B
.IF($G$.NE.$GOF$)COMMENT.  LOAD AND EXECUTE G
.IF($I$.NE.$N$)COMMENT.  WITH INPUT I.
.ENDIF, LNAME.
.IF($P$.NE.$N$)COMMENT.  PLOTFILE P MADE PERMANENT.
COMMENT.****************************************
PATTACH,SARALIB.
.IF,$NAME$.NE.$N$, LCOMPILE.
.IF($B$.NE.$BIN$)PURGE,B.
.IF,L=0, LNY.
FTN200,#I=INPUT,#B=B/500,#OPT=OPT,#UNS=UNS,SC=1,#L=0,#LO=LO.
.ELSE, LNY.
FTN200,#I=INPUT,#B=B/500,#OPT=OPT,#UNS=UNS,SC=1,#L=OUTPUT/800,#LO=LO.
.ENDIF, LNY.
.IF($B$.NE.$BIN$)DEFINE,B.
.ENDIF, LCOMPILE.
.IF(($NAME$.EQ.$N$).AND.($B$.NE.$BIN$))ATTACH,B.
.IF(($NAME$.NE.$N$).AND.($G$.NE.$GOF$))PURGE,G.
.IF,(($NAME$.NE.$N$).AND.((NOEX=0).OR.((NOEX.NE.0).AND.($G$.NE.$GOF$))))
     .OR.(($NAME$.EQ.$N$).AND.($B$.NE.$BIN$)), LLOAD.
ATTACH,HGOLIB.
ATTACH,NAG,U=555555.
ATTACH,PPPLIB.
.IF,LP=0, LLP.
LOAD,B,CN=G/256,CDF=512,#L=0,
       LIB=F200LIB,HGOLIB,PLOTFTN,PPPLIB,NAG.
.ELSE, LLP.
LOAD,B,CN=G/1000,CDF=6400,#L=0,
       LIB=F200LIB,HGOLIB,PLOTFTN,PPPLIB,NAG,GRLPALL= .
.ENDIF, LLP.
.ENDIF, LLOAD.
.IF(($NAME$.NE.$N$).AND.($G$.NE.$GOF$))DEFINE,G.
.IF,NOEX=0, LEXECUTE.
.IF(($NAME$.EQ.$N$).AND.($G$.NE.$GOF$))ATTACH,G.
.IF($P$.NE.$N$)PURGE,TAPE99.
.IF($P$.NE.$N$)DEFINE,TAPE99/1500.
G.
.IF($P$.NE.$N$)SWITCH,TAPE99,P.
.ENDIF, LEXECUTE.
.IF,$NAME$=$N$, NAMENY.
.IF($I$.NE.$N$)COMMENT.**INPUT RECORD I AFTER EOR**
.ELSE, NAMENY.
```

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

.IF($I$.NE.$N$)COMMENT.**INPUT RECORDS NAME AND I AFTER EOR**
.IF($I$.EQ.$N$)COMMENT.**INPUT RECORD NAME AFTER EOR**
.ENDIF, NAMENY.
.*
.* PROCEDURE FOR FURTHER EDITING JOB.
.DATA,EDJOB.
.PROC,EDJOB*#I,
   ANSWER [EDIT JOB? (DELETE OLD EDITFILE!) (N/Y) -] = (N=F,Y=T).
IF(.NOT.ANSWER) REVERT.
RETURN,ZZZZZ1Z,ZZZZZ3Z.
ED,USE,ZZUSE.
ED.
ED,W,JOB,0.
RETURN,EDLOG,ZZZZZ1Z,ZZZZZ3Z.
REVERT.
.*
.* USEFILE FOR EDITING JOB.
.DATA,ZZUSE.
SET,COUNT=1,LINES=19,EXP=1,PROMPT=##
FORMAT,NO
E,JOB


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

.PROC,VAST205*I,
   NAME [NAME OF THE FTN200 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
   ======= 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.
COPYBR,ZZVAST,FILMPL.
COPYBR,NAME,FILMPL.
.*
DFMLOCK,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,$REPLY OF ROUTE:$.
DFMLOCK,OFF.
.*
.* JOB FOR THE 205.
.DATA,ZZVAST.
XXI2X,ST205.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCE(TL=10,WS=512,JCAT=NORMAL)
COMMENT.****************************************
COMMENT.  VAST205:
COMMENT.  NAME TRANSLATED BY VAST ON THE 205,
COMMENT.  OUTPUT OUT CATALOGED ON THE 750.
COMMENT.****************************************
VAST(TAPE1=INPUT,TAPE2=S,TAPE3=T)
MFLINK(T,ST=NBE,JCS="ACCOUNT,XXXACCXXX,XXUNXX.",
                    "CATALOG,OUT,ID=XXIDX.")


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

.PROC,ALIAS*I,
   FLIST [FILES? ($FN1/FN2/..$) - ] = (*A)\
   MF1   [1ST MF? (N=MASTER/..) - ] = (*N=MASTER,N=MASTER,*F),
   MF2   [2ND MF? (N=SECOND/..) - ] = (*N=SECOND,N=SECOND,*F),
   ID    [750 FILE ID? (N/..) -   ] = (*N=$XXIDX$,N=$XXIDX$,*A),
   ACC   [750 ACCOUNT NR? (N/..) -] = (*N=$XXXACCXXX$,N=$XXXACCXXX$,*A),
   UN    [750 LOGIN NAME? (N/..) -] = (*N=$XXUNXX$,N=$XXUNXX$,*A),
   I2     [205 FILE ID? (N/..) -   ] = (*N=$XXI2X$,N=$XXI2X$,*A),
   AC     [205 ACCOUNT NR? (N/..) -] = (*N=$XXXACXXX$,N=$XXXACXXX$,*A),
   U1     [205 USER NR 1? (N/..) - ] = (*N=$XXU1XX$,N=$XXU1XX$,*A),
   U2     [205 USER NR 2? (N/..) - ] = (*N=$XXU2XX$,N=$XXU2XX$,*A),
   PA     [205 PASS WORD? (N/..) - ] = (*N=$XPAX$,N=$XPAX$,*A),
   TA     [LOCAL COMPUTER? (N/..) -] = (*N=$XXA$,N=$XXA$,*A),
   TB     [LINE PRINTER? (N/..) -  ] = (*N=$XXB$,N=$XXB$,*A).
.*
.HELP,,NOLIST.
   ALIAS  TRANSFERS FILES FROM ATTACHED MASTERFILE PFN1,M=MF1,ID=XXIDX
   =====  TO A SECOND ATTACHED MASTERFILE PFN2,M=MF2,ID=..., 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$")
          UN    - CY750 LOGIN NAME      (DEFAULT: "$XXUNXX$")
          I2    - CY205 FILE ID         (DEFAULT: "$XXI2X$")
          AC    - CY205 ACCOUNT NR      (DEFAULT: "$XXXACXXX$")
          U1    - CY205 USER NR 1       (DEFAULT: "$XXU1XX$")
          U2    - CY205 USER NR 2       (DEFAULT: "$XXU2XX$")
          PA    - CY205 PASSWORD        (DEFAULT: "$XPAX$")
          TA    - TID LOCAL COMPUTER    (DEFAULT: "$XXA$")
          TB    - TID LINE PRINTER      (DEFAULT: "$XXB$").
.ENDHELP.
.*
RETURN,GIVE.
MFLIST,FLIST,M=MF1,CCL=GIVE/TRANS.
ED,DEL@.
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,INIT.
ED,B,Q.
FREP,FN,M=MF2.
RETURN,FN.
REVERT.
.*
EXIT,S.
COMMENT.** ERROR IN TRANS **
RETURN,FN.
REVERT,ABORT.
.*
.DATA,USEFILE.
/XXIDX/ID/U@C*
/XXXACCXXX/ACC/U@C*
/XXUNXX/UN/U@C*
/XXI2X/I2/U@C*
/XXXACXXX/AC/U@C*
/XXU1XX/U1/U@C*
/XXU2XX/U2/U@C*
/XPAX/PA/U@C*
/XXA/TA/U@C*
/XXB/TB/U@C*
```

```
--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 15.27.52. ------- COMMENT,  1 ------

COMMENT.
MASTERFILE MFHBT CY=45
14/06/86
********
  MASTERFILE MFHBT CONTAINS THE FILES PERTAINING TO PROGRAM HBT.
********
  FILES:
  COMMENT  - THIS FILE.
  NHBT     - 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.
  RHBT     - FRAMEWORK FOR A JOB REVISING OLDPL UHBT_U WITH MODIFI-
             CATION 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.
  XHBT     - FRAMEWORK FOR A JOB EXECUTING BHBT_B WITH INPUT IHBT_I
             (PLOTFILE 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,I(,P,O,D,T,IO,LP).
  NNHBT    - FRAMEWORK FOR A JOB COMPILING BHBT_B ON THE 205 FROM
             CHBT_U (DUE TO NEWPL UHBT_U FROM NEW SOURCE HBT_S).
             PROCEDURE NN OF CCLLIB PRODUCES BOTH THE COMPILE FILE
             AND THE ACTUAL JOB NN_S, INSERTING VALUES FOR S,U,B.
  RRHBT    - FRAMEWORK FOR A JOB COMPILING BHBT_B ON THE 205 FROM
             CHBT_V (DUE TO NEWPL UHBT_V, FROM MODIFICATION MHBT_M ON
             OLDPL UHBT_U).
             PROCEDURE RR OF CCLLIB PRODUCES BOTH THE COMPILE FILE
             AND THE ACTUAL JOB RR_M, INSERTING VALUES FOR U,M,V,B.
  XXHBT    - FRAMEWORK FOR A JOB EXECUTING BHBT_B WITH INPUT IHBT_I.
             PROCEDURE XX OF CCLLIB PRODUCES THE JOB XX_B_I FROM
             THIS FILE, INSERTING VALUES FOR B,I(,P,O,D,TL,WS,LP).
 *HBT45    - SOURCE FILE OF PROGRAM HBT, D.D. 14/06/86.
  MHBT46   -       .........
             (UPDATE MODIFICATION DECKS)
  IHBT60   -       .........
             (NAMELIST INPUT FILES)
********
  NOTICE:
  BECAUSE OF SIZE CONSIDERATIONS, AND IN ORDER TO AVOID ALTERATIONS
  OF THE MASTERFILE DURING BATCHJOBS, THE FILES UHBT_U, BHBT_B,
  P_B_I, AND O_B_I ARE STORED AS SEPARATE PERMFILES.
********


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

XXIDX,CM110000,T50,I0250,NP.
ACCOUNT,XXXACCXXX,XXUNXX.
COMMENT.****************************************
COMMENT.  N_S:
COMMENT.  NEW UPDATE PL UHBT_U FROM HBT_S,
COMMENT.  COMPILATION OF BHBT_B.
COMMENT.****************************************
REDUCE.
MFUSE,MFHBT,ID=XXIDX,MR=1.
FTAKE,HBT=HBT_S.
UPDATE,F,I=INPUT,N=UHBT,C=CHBT,L=A124,O=OUT.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
USL,OUT,NOLIST=NOUL.
CATALOG,UHBT,UHBT_U,ID=XXIDX.
FTN5,I=CHBT,#B=BHBT,OPT=2,LO=#S/-A,PL=10000,L=0.
CATALOG,BHBT,BHBT_B,ID=XXIDX.
EXIT,#S.
CATALOG,OUT,ON_S,ID=XXIDX.
COMMENT.**UPDATE DIRECTIVES AFTER EOR**
*LIMIT 10000
*READ HBT


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

XXIDX,CM110000,T40,I0250,NP.
ACCOUNT,XXXACCXXX,XXUNXX.
COMMENT.****************************************
COMMENT.  R_M:
COMMENT.  REVISION OLD UPDATE PL UHBT_U WITH
COMMENT.  MODIFICATION MHBT_M, NEW PL UHBT_V,
COMMENT.  COMPILATION OF BHBT_B.
```

```
--- MASTERFILE MFHBT CY=45 ---- 02/07/86 - 00.29.27. ---------- RHBT,  2 ------

COMMENT.****************************************                         80
REDUCE.                                                                  90
ATTACH,UHBT,UHBT_U,ID=XXIDX,MR=1.                                       100
MFUSE,MFHBT,ID=XXIDX,MR=1.                                              110
FTAKE,MHBT=MHBT_M.                                                      120
UPDATE,F,P=UHBT,I=MHBT,N=VHBT,C=CHBT,L=A1234,O=OUT.                     130
ATTACH,CCLLIB,ID=XXIDX,MR=1.                                            140
LIBRARY,CCLLIB.                                                         150
UML,OUT,LIST=ULIST.                                                     160
CATALOG,VHBT,UHBT_V,ID=XXIDX.                                           170
FTN5,I=CHBT,#B=BHBT,OPT=2,LO=S/-A,PL=10000,L=0.                         180
CATALOG,BHBT,BHBT_B,ID=XXIDX.                                           190
EXIT,S.                                                                 200
CATALOG,OUT,OR_M,ID=XXIDX.                                              210


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

XXIDX,CM275000,T100,IO20,NP.                                             10
ACCOUNT,XXXACCXXX,XXUNXX.                                                20
COMMENT.****************************************                         30
COMMENT.  X_B_I:                                                         40
COMMENT.  EXECUTION BHBT_B WITH INPUT IHBT_I.                            50
.IF($P$.NE.$N$)COMMENT.  PLOT FILE #P_B_I CATALOGED.                     60
.IF($O$.NE.$N$)COMMENT.  OUTPUT #O_B_I CATALOGED.                        70
.IF($D$.NE.$N$)COMMENT.  DATA FILE #D_B_I CATALOGED.                     80
COMMENT.****************************************                         90
REDUCE.                                                                 100
ATTACH,BHBT,BHBT_B,ID=XXIDX,MR=1.                                       110
MFUSE,MFHBT,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($P$.NE.$N$)CATALOG,PHBT,#P_B_I,ID=XXIDX.                            190
.IF,$O$.NE.$N$, LCATO.                                                  200
REWIND,OUTPUT.                                                          210
COPY,OUTPUT,OUT.                                                        220
CATALOG,OUT,#O_B_I,ID=XXIDX.                                            230
.ENDIF, LCATO.                                                          240
.IF($D$.NE.$N$)CATALOG,DHBT,#D_B_I,ID=XXIDX.                            250


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

XXI2X,ST205.                                                             10
USER(AC=XXXACXXX,#U=XXU1XX,PA=XPAX)                                      20
RESOURCE(TL=100,WS=256,LP=0,PRIO=12)                                     30
COMMENT.****************************************                         40
COMMENT.  NN_S:                                                          50
COMMENT.  COMPILATION OF BHBT_B FROM CHBT_U                              60
COMMENT.  (ORIGINATING FROM NEW SOURCE HBT_S).                           70
COMMENT.****************************************                         80
PURGE,BHBT_B.                                                            90
.IF(FLIST=0)FTN200,I=INPUT,#B=BHBT_B/500,L=0,OPT=1.                     100
.IF(FLIST=1)FTN200,I=INPUT,#B=BHBT_B/500,OPT=1.                         110
DEFINE,BHBT_B.                                                          120
COMMENT.**INPUT RECORD CHBT_U AFTER EOR**                               130


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

XXI2X,ST205.                                                             10
USER(AC=XXXACXXX,#U=XXU1XX,PA=XPAX)                                      20
RESOURCE(TL=100,WS=256,LP=0,PRIO=12)                                     30
COMMENT.****************************************                         40
COMMENT.  RR_M:                                                          50
COMMENT.  COMPILATION OF BHBT_B FROM CHBT_V                              60
COMMENT.  (DUE TO REVISION MHBT_M ON UHBT_U).                            70
COMMENT.****************************************                         80
PURGE,BHBT_B.                                                            90
.IF(FLIST=0)FTN200,I=INPUT,#B=BHBT_B/500,L=0,OPT=1.                     100
.IF(FLIST=1)FTN200,I=INPUT,#B=BHBT_B/500,OPT=1.                         110
DEFINE,BHBT_B.                                                          120
COMMENT.**INPUT RECORD CHBT_V AFTER EOR**                               130
```

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

XXI2X,ST205.
USER(AC=XXXACXXX,U=XXU1XX,PA=XPAX)
RESOURCE(TL=200,WS=700,LP=5,PRIO=12)
COMMENT.****************************************
COMMENT.  XX_B_I:
COMMENT.  EXECUTION BHBT_B WITH INPUT IHBT_I.
.IF(P=1)COMMENT.  PLOT FILE #P_B_I CATALOGED.
.IF(O=1)COMMENT.  OUTPUT #O_B_I CATALOGED.
.IF(D=1)COMMENT.  DATA FILE #D_B_I CATALOGED.
COMMENT.****************************************
PATTACH,SARALIB.
ATTACH,BHBT_B.
ATTACH,HGOLIB.
ATTACH,NAG,U=555555.
ATTACH,PPPLIB.
LOAD,BHBT_B,CN=GHBT_B/2000,CDF=6400,
      LIB=F200LIB,HGOLIB,NAG,PLOTFTN,PPPLIB,
      GRLP=*FIVE,*SEVEN,L=0.
.IF(P=1)PURGE,TAPE99.
.IF(P=1)DEFINE,TAPE99/1500.
.IF(D=1)PURGE,DHBT.
.IF(D=1)DEFINE,DHBT.
.IF(O=0)GHBT_B.
.IF(O=1)GHBT_B,**INPUT,#O_B_I,TAPE10=INPUT,TAPE20=#O_B_I.
.IF(P=1)SWITCH,TAPE99,#P_B_I.
.IF(O=1)DEFINE,#O_B_I.
.IF(D=1)SWITCH,DHBT,#D_B_I.
COMMENT.**INPUT RECORD IHBT_I AFTER EOR**
```

```
--- MASTERFILE MFHGO CY=27 ---- 02/07/86 - 15.52.04. ------- COMMENT,  1 ------

COMMENT.
MASTERFILE MFHGO CY=27
21/08/85
********
  MASTERFILE MFHGO CONTAINS THE SOURCE HGO_S OF LIBRARY HGOLIB AND
  SOME AUXILIARY FILES.
********
  FILES:
  COMMENT - THIS FILE.
  NHGO    - JOB CREATING NEW LIBRARY HGOLIB FROM HGO_S ( + MHGO_M) ON
            THE 750. THIS JOB IS EDITED AND LAUNCHED BY PROCEDURE NEW
            OF CCLLIB.
            CALL: "NEW,HGO,S=..(,M=..,NOUL,NOR,TID).".
  NNHGO   - JOB COMPILING NEW LIBRARY HGOLIB FROM CHGO_S (OR CHGO_M)
            ON THE 205. THIS JOB IS EDITED AND LAUNCHED BY PROCEDURE
            NNEW OF CCLLIB.
            CALL: "NNEW,HGO,S=..,(M=..,NOUL,NOCAT,FLIST,NOR)."
 *HG027   - SOURCE FILE OF LIBRARY HGOLIB.
********
```

```
--- MASTERFILE MFHGO CY=27 ---- 02/07/86 - 00.30.06. ---------- NHGO,  1 ------

XXIDX,CM100000,T100,IO200,NP.
ACCOUNT,XXXACCXXX,XXUNXX.
COMMENT.****************************************
.IF($M$.EQ.$N$)COMMENT.  NHGO_S:
.IF($M$.NE.$N$)COMMENT.  NHGO_M:
COMMENT.  NEW LIBRARY HGOLIB FROM SOURCE HGO_S
.IF($M$.NE.$N$)COMMENT.  MODIFIED WITH MHGO_M.
COMMENT.****************************************
REDUCE.
MFUSE,MFHGO,ID=XXIDX,MR=1.
FTAKE,HGO=HGO_S.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=HGO,N=UHGO,C=CHGO,L=A124,O=OUT1.
USL,OUT1,NOLIST=NOUL.
.IF,$M$.EQ.$N$, MNY.
CATALOG,UHGO,UHGO_S,ID=XXIDX,MR=1.
FTN5,I=CHGO,B=BHGO,OPT=2,LO=#S/-A,L=0.
.ELSE, MNY.
FTAKE,MHGO=MHGO_M.
UPDATE,F,P=UHGO,I=MHGO,N=NEWPL,C=COMPILE,L=A1234,O=OUT2.
CATALOG,NEWPL,UHGO_M,ID=XXIDX,MR=1.
UML,OUT2.
FTN5,I=COMPILE,B=BHGO,OPT=2,LO=#S/-A,L=0.
.ENDIF, MNY.
REQUEST,LIB,*PF.
EDITLIB(USER,L=SCR)
CATALOG,LIB,HGOLIB,ID=XXIDX,MR=1.
ITEMIZE,LIB.
COMMENT.**EDITLIB DIRECTIVES AFTER EOR**
LIBRARY(LIB,NEW)
ADD(*,BHGO)
FINISH.
ENDRUN.
```

```
--- MASTERFILE MFHGO CY=27 ---- 02/07/86 - 00.30.06. --------- NNHGO,  1 -----

XXI2X,ST205.
USER,AC=XXXACXXX,U=XXU1XX,PA=XPAX.
RESOURCE,TL=100,WS=256,LP=0,PRIO=12.
COMMENT.****************************************
COMMENT.  NNHGO_M:
COMMENT.  CREATION OF HGOLIB FROM CHGO_M.
COMMENT.****************************************
PURGE,HGOLIB.
.IF(FLIST=0)FTN200,I=INPUT,B=BHGO_M/500,OPT=1,L=0.
.IF(FLIST=1)FTN200,I=INPUT,B=BHGO_M/500,OPT=1.
OLE,I=BHGO_M,N=HGOLIB,OUTPUT=LIST.
DEFINE,HGOLIB.
COMMENT.**INPUT RECORD CHGO_M AFTER EOR**
```

```
--- 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.
********
  FILES:
  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).".
  NNPPP   - JOB COMPILING NEW LIBRARY PPPLIB FROM CPPP_S (OR CPPP_M)
            ON THE 205. THIS JOB IS EDITED AND LAUNCHED BY PROCEDURE
            NNEW OF CCLLIB.
            CALL: "NNEW,PPP,S=..,(M=..,NOUL,NOCAT,FLIST,NOR)."
 *PPP10   - SOURCE FILE FOR LIBRARY PPPLIB.
  MPPP10  - 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 ------

XXIDX,CM100000,T100,I0200,NP.
ACCOUNT,XXXACCXXX,XXUNXX.
COMMENT.****************************************
.IF($M$.EQ.$N$)COMMENT.  NPPP_S:
.IF($M$.NE.$N$)COMMENT.  NPPP_M:
COMMENT.  NEW LIBRARY PPPLIB FROM SOURCE PPP_S
.IF($M$.NE.$N$)COMMENT.  MODIFIED WITH MPPP_M.
COMMENT.****************************************
REDUCE.
MFUSE,MFPPP,ID=XXIDX,MR=1.
FTAKE,PPP=PPP_S.
ATTACH,CCLLIB,ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
UPDATE,I=PPP,N=UPPP,C=CPPP,L=A124,O=OUT1.
USL,OUT1,NOLIST=NOUL.
.IF,$M$.EQ.$N$, MNY.
CATALOG,UPPP,UPPP_S,ID=XXIDX,MR=1.
FTN5,I=CPPP,B=BPPP,OPT=2,LO=#S/-A,L=0.
.ELSE, MNY.
FTAKE,MPPP=MPPP_M.
UPDATE,F,P=UPPP,I=MPPP,N=NEWPL,C=COMPILE,L=A1234,O=OUT2.
CATALOG,NEWPL,UPPP_M,ID=XXIDX,MR=1.
UML,OUT2.
FTN5,I=COMPILE,B=BPPP,OPT=2,LO=#S/-A,L=0.
.ENDIF, MNY.
REQUEST,LIB,*PF.
EDITLIB(USER,L=SCR)
CATALOG,LIB,PPPLIB,ID=XXIDX,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. --------- NNPPP,  1 ------

XXI2X,ST205.
USER,AC=XXXACXXX,U=XXU1XX,PA=XPAX.
RESOURCE,TL=100,WS=256,LP=0,PRIO=12.
COMMENT.****************************************
COMMENT.  NNPPP_M:
COMMENT.  CREATION OF PPPLIB FROM CPPP_M.
COMMENT.****************************************
PURGE,PPPLIB.
.IF(FLIST=0)FTN200,I=INPUT,B=BPPP_M/500,OPT=1,L=0.
.IF(FLIST=1)FTN200,I=INPUT,B=BPPP_M/500,OPT=1.
OLE,I=BPPP_M,N=PPPLIB,OUTPUT=LIST.
DEFINE,PPPLIB.
COMMENT.**INPUT RECORD CPPP_M AFTER EOR**
```

```
--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.31.10. -------- MPPP10,  1 ------

*IDENT MOD10                                                                  10
*DEFINE SARA,CY750                                                            20

--- MASTERFILE MFPPP CY=10 ---- 02/07/86 - 00.31.10. ------- MPPP10A,  1 ------

*IDENT MOD10A                                                                 10
*DEFINE SARA,CY205                                                            20
```

```
--- PERMANENT FILE ID=PUBLIC -- 03/07/86 - 13.58.56. -- PROCFIL CY=1,  1 ------
     (ADAPTED TEXT)

.PROC,INIT*I,
   ID  [YOUR 'ID' -                  ] = (    *A),
   PAR [1ST PARAMETER OF YOUR 'INIT' -] = (*N=,*A),
   PW  [PASSWORD OF YOUR 'PROCFIL'? - ] = (*N=,*A),
   SN  [SETNAME OF YOUR 'PROCFIL'? -  ] = (*N=,*A).
.HELP,,NOLIST
   INIT  ALLOWS YOU TO EXECUTE A PRIVATE PROCEDURE WITH A SIMPLE CALL.
   ====  YOUR PRIVATE PROCEDURE SHOULD ALSO HAVE THE NAME 'INIT' AND
         SHOULD BE LOCATED ON A PERMANENT FILE WITH THE NAME 'PROCFIL'.
         PARAMETERS:
         ID  - IDENTIFICATION OF YOUR OWN PERMFILE 'PROCFIL'
         PAR - OPTIONAL PARAMETER TO BE PASSED TO YOUR PRIVATE 'INIT'
         PW  - PASSWORD, POSSIBLY NEEDED TO ATTACH YOUR 'PROCFIL'
         SN  - SETNAME OF YOUR 'PROCFIL', IF DIFFERENT FROM PRESENT ONE.
         THE PRESENT PROCEDURE 'INIT' MAY BE CALLED WITH
              "BEGIN,INIT,,ID,PAR,PW,SN.",
         WHERE 'ID' IS THE ONLY REQUIRED PARAMETER.
         YOUR PRIVATE PROCEDURE 'INIT' IS THEN CALLED WITH
              "BEGIN,INIT,,PAR."  OR  "BEGIN,INIT,,PAR,PW=PW.",
         SO THAT THE VALUE OF 'PAR' HAS BEEN TRANSFERRED.
.HELP,ID,NOLIST
   ID    IS THE 'ID' OF YOUR PRIVATE PERMANENT FILE 'PROCFIL'.
.HELP,PAR,NOLIST
   PAR   IS AN OPTIONAL PARAMETER TO BE PASSED TO YOUR PRIVATE PROCEDURE
         'INIT'.
.HELP,PW,NOLIST
   PW    IS A PASSWORD, POSSIBLY NEEDED TO ATTACH YOUR OWN 'PROCFIL'.
.HELP,SN,NOLIST
   SN    IS THE SETNAME FOR YOUR PRIVATE 'PROCFIL' WHICH IS ONLY NEEDED
         WHEN IT DIFFERS FROM THE PRESENT SETNAME.
.ENDHELP
.*
RETURN,PROCFIL.
.IF(.NOT.$SN$) ATTACH,PROCFIL,PROCFIL,#ID=ID,#PW=PW.
.IF($SN$) ATTACH,PROCFIL,PROCFIL,#ID=ID,#PW=PW,#SN=SN.
.*
SKIP, LZZZZZC4.
EXIT.
COMMENT.** ATTACH FAILED, ZZZZZC4 TRIED **
.IF(.NOT.$SN$) ATTACH,PROCFIL,ZZZZZC4,#ID=ID,#PW=PW.
.IF($SN$) ATTACH,PROCFIL,ZZZZZC4,#ID=ID,#PW=PW,#SN=SN.
ENDIF, LZZZZZC4.
.*
.IF(.NOT.$PW$) BEGIN,INIT,PROCFIL,PAR.
.IF($PW$) BEGIN,INIT,PROCFIL,PAR,#PW=PW.
```

```
--- PERMANENT FILE ID=XXIDX --- 03/07/86 - 13.55.32. - PROCFIL CY=10,  1 ------

.PROC,INIT,PAR,PW=.
RETURN,CCLLIB.
ATTACH,CCLLIB,#ID=XXIDX,MR=1.
LIBRARY,CCLLIB.
RETURN,PROCFIL.
FINDX.
FINDX,/XXB.
FED,FF.
ETL,200.
RETURN,MFCCL,MASTER.
MFUSE,MFCCL,M=MFCCL,#ID=XXIDX.
FGET,ZZSYS1,M=MFCCL.
ZZSYS1.
RETURN,MFCCL.
MFUSE,PAR,#ID=XXIDX.
REVERT.
```