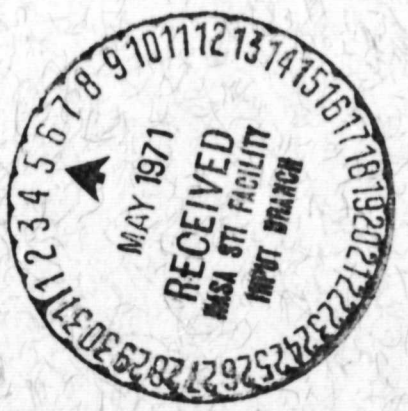
X-603-71-144
PREPRINT
NASA TM X-65501

SUFFER

A PROGRAM FOR CONSOLE EXECUTION OF UTILITY PROGRAMS

JOHN S. CAVALLINI
DENNIS M. GIBLIN



FEBRUARY 1971

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-603-71-144

SUFFER

A PROGRAM FOR CONSOLE EXECUTION OF UTILITY PROGRAMS

John S. Cavallini
Dennis M. Giblin
SESD Computer Center

February 1971

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

SUFFER

A PROGRAM FOR CONSOLE EXECUTION OF UTILITY PROGRAMS

John S. Cavallini
Dennis M. Giblin
SESD Computer Center

ABSTRACT

SUFFER, System Utility Facility For Easy Recovery, is a systems program written for use on IBM S/360 computers. It allows the composition and execution of utility programs at any operator's console. Associated with SUFFER are procedure and control statement libraries. SUFFER's value, use, logic flow and installation are discussed. Examples and source listings are included.

CONTENTS

	Page
INTRODUCTION	1
THE VALUE OF SUFFER	1
HOW TO USE SUFFER	6
SUFFER LOGIC FLOW	7
INSTALLATION OF SUFFER	8
ACKNOWLEDGEMENT	9
APPENDIX A	A-1
APPENDIX B	B-1
APPENDIX C	C-1
APPENDIX D	D-1

SUFFER

A PROGRAM FOR CONSOLE EXECUTION OF UTILITY PROGRAMS

INTRODUCTION

This paper describes and offers motivation for the use of a program written for IBM S/360 computers. The program is called SUFFER - System Utility Facility For Easy Recovery. It is an on-line tool to be used by systems programs and operators from an operator's console. Using SUFFER, the system programmer can construct and execute utility programs designed to aid in both the investigation of system problems and normal system software maintenance. The use of two data sets, SYS2.SUFFER.PROCLIB and SYS2.SUFFER.CONTROL, containing JCL and utility control statements respectively, allows the construction of these programs from previously defined building blocks. In addition, a job may be created in toto at the operator's console and executed using SUFFER.

The potential values of SUFFER, how to use it, its logic flow and installation are discussed in this paper. Program listings and examples are offered in the Appendices.

THE VALUE OF SUFFER

System Programming covers a wide range of functions associated with operating systems and extends from the conception and design of an operating system through to the responsibility of maintaining and debugging an operating system which is already in existence. In this latter aspect of system programming, the most important concern is to keep the operating system performing with a minimal amount of "down time." Fast and accurate isolation of problems causing down time and a solution to the problem, such as a temporary programming fix, constitute a major function of the systems programmer and, hence, it is necessary in attempting to solve the problems of a large operating system to have as much information as possible available to the systems programmer for analysis of the system at that point in time when the problem occurs. Keeping down time to a minimum also involves a program of system maintenance allowing the operating system to continue its operation essentially error free and providing a reliable backup of the software and data stored in the computing system on secondary storage. Obviously, problems are being isolated and solved and information is being stored about present operating systems or there would not be an operating system working. However, any reduction in the amount of time consumed in this area further reduces the amount of time the system is inoperable and,

therefore, increases the time the computing system can be used for production work. The intent of this paper is to discuss how "down time" can be reduced by enhancing the system programmer's ability to communicate with the operating system (thus providing himself with more information about the system) through SUFFER.

Multiple Console Support, providing the features of routing codes, descriptor codes, multiple and dedicated consoles, hard copy and the user exit, could in the extreme case provide the user's system programmer a console unto himself with its own routing and descriptor codes (which could be changed dynamically), however, this is an unnecessary extravagance. Yet, systems work will be proposed to be done with the normal configuration for any installation.

IBM offers the user four different ways of organizing data on secondary storage in the computing system and also uses two of these organizations (partitioned data set and sequential data set) to store the operating system programs and tables. IBM places the responsibility for maintaining data sets of any organization on the user. However, IBM does provide a set of standard utility programs for creating and maintaining operating system data. There are essentially two categories for these utility programs. One is the set of utilities used to maintain system control data at an organizational or system level. The other is the set of utilities used to reorganize, change, or compare data at the data set level and at the record level.

The utility programs are probably the most frequently used programs to gather information about the system when problems develop, to make modifications to the system, to recover lost data, to recreate lost data sets and to correct problems. Consider the example of an installation that reserves a disk pack to accommodate a partitioned data set whose members are the load modules of customer programs. If on some occasion an unusually large number of programs start abending with a system 213 completion code, the first thing to do is to run the utility program IEHLIST to list the VTØC on the disk pack on which the partitioned data set resides. Then for instance, if the data set were scratched (a strong possibility since the operator reply for a normal update to this data set and for scratching it altogether are the same), one would have to use the IEHDASDR or IEHMOVE utility data set to recover the customer partitioned data set.

The job control language for executing the utility programs is illustrated by the following:

```
//JOBNAME (JOB statement)

//STEPNAME EXEC PGM=PROGNAME
```

//SYSPRINT (DD statement for message output data set.)

DD statements for device allocation.

//SYSIN (DD statement for data set containing utility control statements.)

/*

Two of these JCL statements should be discussed. The first are the DD statements for device allocation. For all of the utilities the number of these DD statements remains constant when performing a singular application of the given utility program. It should also be noted that through the proper use of symbolic parameters, these DD statements can be made flexible enough to retain just one copy of the JCL for a singular application of a utility on secondary storage and to use that copy to execute the utility for any device by simply overriding the chosen symbolic parameters. The sysin card also requires consideration since it defines the data set containing the control statements to be used by the utility program. This must be sequential data set, e.g., it can be specified after the card

```
//SYSIN DD *
```

in the job stream or as a member of a partitioned data set pointed to by the sysin card. This data set is vital to the utility to designate the operation the utility is to perform and the data set and possibly the member on which the operation is to be performed.

Since the utility programs are the main source of information about the system (i.e., with the exception of a core dump and the hard copy), it would seem to be a good idea to make these programs and any others which the system personnel deem necessary more readily available at times when system problems arise and multiple console support offers this capability to the systems programmer. One of the operator commands offered the user in multiple console support is the start command. OS/360 allows the operator to start certain qualified tasks from the operator's console and to pass parameters to these tasks by the start command itself. There must be a procedure in the system procedure library to point to the program to be started, the program must exist in one of the system linkage libraries and the user must update the table (IEEVLNKT) used by the Post Scan Exit Routine to include the program name.

The procedure corresponding to the program is:

```
//SUFFER EXEC PGM=SUFFER
```

```
//PROCLIB DD DSN=SYS2.SUFFER.PROCLIB,DISP=SHR

//CONTROL DD DSN=SYS2.SUFFER.CONTROL,DISP=SHR

//JOB1 DD DSN=SYS2.SUFFER.JOB1,DISP=SHR

//JOB2 DD DSN=SYS2.SUFFER.JOB2,DISP=SHR
```

The PROCLIB and CONTROL DD cards refer to the libraries created to contain SUFFER routines. The JOB1 and JOB2 DD cards are for the data sets to contain the job streams created at the operator's console.

Program SUFFER started in this manner is a system task, operates in protect key 0 and is itself capable of issuing SVC's or performing any other function normally reserved for system tasks.

Since SUFFER is in protect key zero, one of the reserved operations it can perform is issuing the communications Supervisor Call, i.e. SVC 34. In this manner, SUFFER can issue any command which can be entered at the master console. All that is necessary, is that SUFFER point register zero to the message text and specify the text length in the first four bytes. Therefore, SUFFER can start any procedure, system task or, as we will need, its own reader. The standard IBM reader procedure requires three DD statements to be defined. The first named IEFRDER, describes the input stream to the reader/interpreter. The second, named IEFPDSI, describes the procedure library which is to be used by the invoked reader and the last, IEFDATA, describes the intermediate storage of input stream data. Hence, SUFFER, given its own reader procedure, can start programs, in this case utilities and other essential routines, by pointing the IEFPDSI data definition card to its own procedure library containing the necessary procedures to execute the utilities, and by pointing the IEFRDER data definition statement to this same utility procedure library but specifying the member or individual procedure symbolically. Then SUFFER maintains the ability to override this data definition card and can execute any utility program requested by the system programmer working with SUFFER for which a member has been created in this utility procedure library or for which SUFFER has created an input stream for the reader.

In order to initialize utility programs, SUFFER must use several data sets which have been mentioned already. The first data set is the data set to contain symbolic procedures for the utility programs. We have discussed the job control language necessary for executing a utility program but we have not seen any examples of how the use of symbolic parameters can be used to reduce the space necessary for this procedure data set and to facilitate the execution of the utilities. Let us consider two of the most frequently used utility programs, IEHDASDR

and IEHPROGM. IEHPROGM's functions include scratching, renaming, cataloging, and uncataloging data sets. Since data sets can be inadvertently uncataloged, cataloged incorrectly, or scratched, it would be convenient to have this utility available immediately. The symbolic procedure for this utility is as follows:

```
//DEFAULT PROC VOL=M2SYS4

//CATLG EXEC PGM=IEHPROGM

//SYSPRINT DD SYSOUT=A

//DD1 DD VOL=REF=SYSRES, DISP=OLD

//DD2 DD VOL=REF=&VOL, DISP=OLD
```

Since IEHPROGM requires a DD card for each volume referred to in the job step and since the catalog is on the system residence device, it is most favorable in this case to have a second card which can be used to describe any other pack which might be used. In this example, the DD2 card will point to the volume M2SYS4 unless overridden. SUFFER allows for this condition with its code option which is illustrated later. IEHPROGM, however, has a different control card for each of the functions mentioned. IEHPROGM by nature requires a data set name specified in the control card and therefore a method of just typing the control card at the console or editing it on secondary storage is necessary. SUFFER provides the option of replacing the sysin card by one specifying an in stream data set and then following it with the desired control cards.

The IEHDASDR utility can be used to analyze and initialize direct access storage devices, and to dump or restore these devices onto or from back up copies kept by the individual user. The procedure for executing the IEHDASDR can be:

```
//DEFAULT PROC VOL=M2DRM1,T=SYSTPE,MEM=DUMP

//DASDR EXEC PGM=IEHDASDR

//SYSPRINT DD SYSOUT=A

//DD1 DD VOL=REF=&VOL,DISP=OLD

//TO1 DD VOL=SER=&T,UNIT=(TAPE,DEFER),DSN=&VOL,DISP=
(,KEEP)

//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=OLD
```

SUFFER uses a second data set in which to store a control statement for each function as a member of this partitioned data set (SYS2.SUFFER.CONTROL) and then the appropriate member can be selected by just specifying the MEM parameter in the exec card. Here, the two most frequently used functions, the dump and the restore have so little variation in their control statements that it is unnecessary to code them specifically. In general, it is only necessary to include the members DUMP and RESTORE which are:

```
DUMP FROMDD=DD1,TODD=TO1
```

and

```
RESTORE TODD=DD1,FROMDD=TO1
```

The program SUFFER also utilizes two sequential data sets as temporary storage for the job stream it creates. SUFFER will prepare the final JCL for the procedure together with the proper control cards in one of these data sets to avoid timing problems incurred by trying to create new JCL for starting a second job while RDRSUFR is still processing the JCL for the first one.

There are several reasons to justify a program such as SUFFER. As was mentioned, the information which becomes more readily available is sufficient in itself. But then, SUFFER also can cure problems when the operator is using it while talking to the systems programmer by telephone after working hours. Errors are reduced when operating personnel do not have to code cards. SUFFER also increases the speed in which the utilities can be entered in the system, since all the cards need not be punched up or even modified. SUFFER also decreases the errors which can be made in keypunching by the system programmer.

There are also several improvements which should be made to SUFFER. As it is, SUFFER lacks the ability to format control blocks on the display console. Displaying a given TCB or UCB might prove to be a very handy feature when certain jobs are in the system for long periods of time. Finally, SUFFER does not provide for any output from programs to be returned to the console In many cases, the utilities output is limited to a few lines and would not clutter up the display, yet it would yield quick response to the utility. In any case, at least the condition code of the utility at completion could be very helpful in determining if DASDI,DUMP/RESTORE performed satisfactorily.

HOW TO USE SUFFER

The following section discusses the operation of SUFFER. Appendix A is a reproduction of system log for one SUFFER session.

SUFFER is initiated with a "S SUFFER" command. The user is then queried for the name of the routine in SYS2.SUFFER.PROCLIB he wishes to use. The specification of the routine name at this point if for display purposes only. He also specifies the display options: all lines of the procedure, the PROC statement, containing default values for parameters, or no display. The abbreviations A, D and N are allowed. The default option is to show the entire procedure. If the user types in a routine name not in the SUFFER library, he is requested to try again. Three consecutive incorrect routine names causes the termination of the SUFFER routine.

After the JCL desired has been displayed, SUFFER asks if the default utility control statements associated with the procedure should be shown. SUFFER then requests the job name to be used. Again there are three options: to specify all eight characters, to specify a three character suffix to K3SYS or to use K3SYSSUF by specifying 'U'. These defaults may be modified as explained in "Installation of SUFFER."

After the job name is specified, the user specifies all JCL and control statements that will make up his job. The first line is an EXEC, generally to invoke the catalogued procedure displayed above. At this point the user modifies any of the default parameters specified in the routine. Following the EXEC statement, any additional DD cards and utility control statements required are entered. After the last line, the user replies 'END' to show completion and to automatically start a reader to the job he has created. SUFFER then returns to the start and requests another routine name. At this point or at any time, the user may reply 'EXIT' to immediately terminate the operation of SUFFER.

SUFFER is simple to use, as the requests cue the user as to the form of the reply desired. A knowledge of the contents of SYS2.SUFFER.PROCLIB is helpful. It is recommended that at least a listing of the member names be available at the operators console. For simplicity in coding, SUFFER does not allow correction of lines already entered. In the case of errors in typing of JCL, 'EXIT' and restart is the best course.

SUFFER LOGIC FLOW

This section of the paper contains a brief narrative of the internal logic flow of SUFFER. The source listings can be found in Appendix B.

SUFFER determines from the start command which console is in use and routes all WTO's and WTOR's to it. After opening SYS2.SUFFER.PROCLIB and SYS2.SUFFER.CONTROL, it requests a routine name and display option. If the routine name cannot be found in the directory, another try is requested. Three

consecutive incorrect names causes termination of SUFFER. If the routine is not to be shown at all, the job preparation code is entered. Otherwise FIND locates the desired member and a BPAM READ is issued for the first block. Logical record counting and subsequent READs are built into the program. Each record is examined to determine if it is part of the PROC statement. If so, the default SYS2.SUFFER.CONTROL member name is searched for and the entire line displayed with a WTO. After the PROC statement, no search is done and the line displayed only if option ALL was chosen.

A WTOR then asks if the control statements are to be shown. If so, the same sequence of FIND, READ's and WTO's as above is used to display the control statements. Two alternating data sets are used to contain the job now to be built. A pair of sets is used to guarantee that in the case of consecutive jobs being built by SUFFER, the job data set is not written over before it has been read to the job queue. An one byte flag, JOBNO, determines which data set, SYS2.SUFFER.JOB1 or JOB2 is currently being used. After the correct data set is opened, the user is queried for a job name. He is given the options of eight characters, a three character suffix or allowed to use a default name. His reply is incorporated into a job card fixed at assembly and the result written to the JOB data set. Subsequent WTOR's request card images to be entered into the data set.

When the user is through entering card images, he replies 'END'. This causes the JOB data set to be closed. An SVC 34 is used to start RDRSUFR to the correct job data set, which is subsequently treated as a normal job by the system. A new routine name is requested and the process begins anew. In reply to any of the WTOR's, the user may type "EXIT', which causes all data sets to be closed and a RETURN to be issued.

INSTALLATION OF SUFFER

The following steps must be performed to make SUFFER operational:

1. Assemble SUFFER source code (See Appendix B). Before assembly constants DEFJOBID and JOBREST should be modified to contain the desired default job id and job card format.

2. The assembly output is then link-edited into SYS1.LINKLIB or any link library concatenated to it in SYS1.PARMLIB(LNKLSTØØ).

3. Place the catalogued procedure SUFFER (Appendix C) in SYS1.PROCLIB.

4. Modify SYS1.LINKLIB(IEEVLNKT) to allow a START SUFFER command to be valid.

5. Place the catalogued procedure RDRSUFR (Appendix C) in SYS1.PROCLIB.

6. Create and catalog the two SUFFER library data sets:

   SYS2.SUFFER.PROCLIB

   SYS2.SUFFER.CONTROL

7. Allocate space for and catalog the two SUFFER job stream data sets:

   SYS2.SUFFER.JOB1

   SYS2.SUFFER.JOB2

A few tracks for each should be sufficient.

Copies of the source code may be obtained by contacting the authors. Further improvements to SUFFER are contemplated and will be incorporated into future decks.

ACKNOWLEDGEMENT

The authors wish to acknowledge the valuable assistance of William J. Bradford in the coding and testing of SUFFER.

APPENDIX A

A SAMPLE SUFFER SESSION

```
*IEF4291 INITIATOR 'INIT' WAITING FOR WORK
su¢ suffer      (¢ is a backspace)
 ENTER 'EXIT' AT ANY TIME TO STOP SUFFERING.
*08 ENTER ROUTINE NAME AND DISPLAY OPTIONS: ALL, DEFAULTS OR NOTHING
r 08,'suflist,n'
 IEE600I ACCEPTED REPLY TO MSG 08 IS 'suflist,n'
*09 INCORRECT ROUTINE NAME SPECIFIED. TRY AGAIN.
r 09,'SUFLIST,N'
 IEE600I ACCEPTED REPLY TO MSG 09 IS 'SUFLIST,N'
*10 ENTER JOBNAME DESIRED OR SUFFIX TO K3SYS OR 'U' TO USE K3SYSSUF.
r 10,'U'
 IEE600I ACCEPTED REPLY TO MSG 10 IS 'U'
*11 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 11,'// EXEC SUFLIST'
 IEE600I ACCEPTED REPLY TO MSG 11 IS '// EXEC SUFLIST'
*12 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 12,'END'
 IEE600I ACCEPTED REPLY TO MSG 12 IS 'END'
*13 ENTER ROUTINE NAME AND DISPLAY OPTIONS: ALL, DEFAULTS OR NOTHING
   RRRR JOB D0001 K3SYSSUF ON 330 BY RDRSUFR
r 13,'DUMP,A'
 IEE600I ACCEPTED REPLY TO MSG 13 IS 'DUMP,A'
 //DEFAULT PROC VOL=M2DRM1,T=SYSTPE,MEM=DUMP                          0
 //DMPRES  EXEC  PGM=IEHDASDR,REGION=100K                             0
 //SYSPRINT DD SYSOUT=A                                               0
 //DD1 DD VOL=REF=&VOL,DISP=SHR                                       0
 //TO1 DD UNIT=2400-9,VOL=SER=&T,DISP=(,KEEP)                         0
 //SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR                    0
*14 DISPLAY CONTROL STATEMENTS?  Y OR N.
r 14,'Y'
 IEE600I ACCEPTED REPLY TO MSG 14 IS 'Y'
   DUMP  FROMDD=DD1,TODD=TO1                                          0
*15 ENTER JOBNAME DESIRED OR SUFFIX TO K3SYS OR 'U' TO USE K3SYSSUF.
r 15,'DMP'
 IEE600I ACCEPTED REPLY TO MSG 15 IS 'DMP'
*16 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
    SSS JOB D0001 K3SYSSUF AT 09.12.12 BY INIT
r 16,'// EXEC DUMP,T=SYS)¢073'
    EEE JOB D0001 K3SYSSUF AT 09.12.20 JOBT=0001
   WWWW JOB D0001 K3SYSSUF ON 00F, BOX=SYS
 IEE600I ACCEPTED REPLY TO MSG 16 IS '// EXEC DUMP,T=SYS073'
*18 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 18,'END'
 IEE600I ACCEPTED REPLY TO MSG 18 IS 'END'
*19 ENTER ROUTINE NAME AND DISPLAY OPTIONS: ALL, DEFAULTS OR NOTHING
   RRRR JOB D0002 K3SYSDMP ON 330 BY RDRSUFR
    SSS JOB D0002 K3SYSDMP AT 09.14.07 BY INIT
 IEF234E R 0D3,,K3SYSDMP
*IEF233A M 0D3,SYS073,,K3SYSDMP,DMPRES
*IEA000A  0D3,INT REQ,C3,0200,4024000400,,SYS073,
r 19,'LISTVTOC,D'
 IEE600I ACCEPTED REPLY TO MSG 19 IS 'LISTVTOC,D'
 //DEFAULT PROC MEM=LISTVTOC,CVOL=M2DRM1,VOL=M2SYS4                   0
 //IEHLIST EXEC PGM=IEHLIST                                           0
*20 DISPLAY CONTROL STATEMENTS?  Y OR N.
r 20,'N'
 IEC705I TAPE ON 0D3,SYS073 IS SL, 1600BPI
 IEE600I ACCEPTED REPLY TO MSG 20 IS 'N'
*21 ENTER JOBNAME DESIRED OR SUFFIX TO K3SYS OR 'U' TO USE K3SYSSUF.
 IEF280E K 0D3,SYS073,K3SYSDMP,DMPRES
    EEE JOB D0002 K3SYSDMP AT 09.16.38 JOBT=0002
   WWWW JOB D0002 K3SYSDMP ON 00F, BOX=SYS
```

(Capitals required at alternate console)

```
r 21,'K3JSCVTC'
 IEE6001 ACCEPTED REPLY TO MSG 21 IS 'K3JSCVTC'
*22 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 22,'// EXEC LISTVTOC,VOL=M2SYS2'
 IEE6001 ACCEPTED REPLY TO MSG 22 IS '// EXEC LISTVTOC,VOL=M2SYS2'
*23 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 23,'//SYSIN DD *'
 IEE6001 ACCEPTED REPLY TO MSG 23 IS '//SYSIN DD *'
*24 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 24,' LISTVTOC FORMAT,VOL=2314=M2SYS2'
 IEE6001 ACCEPTED REPLY TO MSG 24 IS ' LISTVTOC FORMAT,VOL=2314=M2S
*25 ENTER JCL OR CONTROL STATEMENTS.  ENTER 'END' TO EXECUTE YOUR JOB.
r 25,'e¢END'
 IEE6001 ACCEPTED REPLY TO MSG 25 IS 'END'
*26 ENTER ROUTINE NAME AND DISPLAY OPTIONS: ALL, DEFAULTS OR NOTHING
r 26,'EXIT'
   RRRR JOB D0003 K3JSCVTC ON 330 BY RDRSUFR
    SSS JOB D0003 K3JSCVTC AT 09.19.36 BY INIT
 IEE6001 ACCEPTED REPLY TO MSG 26 IS 'EXIT'
 YOUR SUFFERING IS OVER!
    EEE JOB D0003 K3JSCVTC AT 09.20.05 JOBT=0003
  WWWW JOB D0003 K3JSCVTC ON 00F, BOX=SYS
```

APPENDIX B

SUFFER SOURCE LISTING

```
SUFFER   CSECT                                                          00000100
 TITLE 'S U F F E R  -  SYSTEM UTILITY FACILITY FOR EASY RECOVERY'       00000150
*        THE FUNCTION OF THIS PROGRAM IS TO ALLOW THE EXECUTION         00000200
*        OF SYSTEM UTILITY PROGRAMS FROM EITHER OPERATOR'S              00000300
*        CONSOLE.                                                       00000400
*                                                                       00000500
*        TWO LIBRARIES ARE ASSOCIATED WITH THE PROGRAM:                 00000600
*              SYS2.SUFFER.PROCLIB - UTILITY JCL LIBRARY                00000700
*              SYS2.SUFFER.CONTROL - CONTROL STATEMENT LIBRARY          00000800
*        MEMBERS OF THESE LIBRARIES ARE DISPLAYED AT THE                00000900
*        OPERATOR'S REQUEST.                                            00001000
*                                                                       00001100
*        UTILITY PROGRAMS ARE CREATED BY REPLYING TO SUFFER'S           00001200
*        PROMPTING WITH JCL AND CONTROL STATEMENTS.  THE RESULTING      00001300
*        JOB IS PLACED IN ONE OF TWO DATA SETS:                         00001400
*              SYS2.SUFFER.JOB1 OR JOB2.                                00001500
*        A READER IS THEN STARTED TO THE CORRECT JOB DATA SET.          00001510
*                                                                       00001700
*                                                                       00001800
*                              WRITTEN BY:                              00001900
*                                    DENNIS GIBLIN                      00002000
*                                    JOHN CAVALLINI                     00002100
*                                    G.S.F.C. SFSD COMPUTING CENTER     00002200
*                                    JANUARY, 1971                      00002300
         EJECT                                                         00002400
R0       EQU   0                                                       00002500
R1       EQU   1                                                       00002600
R2       EQU   2                                                       00002700
R3       EQU   3                                                       00002800
R4       EQU   4                                                       00002900
R5       EQU   5                                                       00003000
R6       EQU   6                                                       00003100
R7       EQU   7                                                       00003200
R8       EQU   8                                                       00003300
R9       EQU   9                                                       00003400
R10      EQU   10                                                      00003500
R11      EQU   11                                                      00003600
R13      EQU   13                                                      00003700
R12      EQU   12                                                      00003800
R14      EQU   14                                                      00003900
R15      EQU   15                                                      00004000
RW       EQU   11            WTOR WORK REGISTER                        00004010
RM       EQU   9             POINTER TO WTO/WTOR MESSAGE               00004020
         SPACE 5                                                       00004100
         SAVE  (14,12)                                                 00004200
         BALR  12,0                                                    00004300
         USING *,12                                                    00004400
         ST    R13,SAVE+4                                              00004500
         LA    R4,SAVE                                                 00004600
         ST    R4,8(R13)                                               00004700
         LR    R13,R4                                                  00004800
         LA    R1,8(R1)      CIB POINTER 3RD WORD IN SPL               00004850
         USING JSCCIB,R1                                               00004875
         EJECT                                                         00004900
SETUP    EQU   *                                                       00005000
         MVC   MESSAGE3+37(5),DEEJOBID                          WJR    00005004
         MVC   MESSAGE3+57(8),DEEJOBID                          WJR    00005008
         TM    CIBCONID,X'01'      IS THIS THE MASTER CONSOLE          00005010
*                                   CIB PTED TO BY R1 AFTER START COMMAND 00005020
         BO    MASTER                                                  00005030
```

PRECEDING PAGE BLANK NOT FILMED.

```
         MVC   ROUT(2),ROUT13     NO. WORKING AT ALTERNATE          00005040
         B     PROCOPEN                                            00005050
MASTER   MVC   ROUT(2),ROUT1 WORKING AT MASTER CONSOLE             00005060
PROCOPEN EQU   *                                                   00005070
         OPEN  (PROC,(INPUT),CONTROL,(INPUT))                      00005100
         LA    RM,MESSAGE4        TELL HOW TO EXIT                 00005140
         BAL   R10,EXWTO                                           00005180
*                                                                  00005200
*        ISSUE FIRST WTOR BY LINKING TO EXWTOR                     00005300
*        REG. RM MUST POINT TO 2 BYTE LENGTH CODE AND MESSAGE      00005400
*        THE REPLY IS PLACED IN REPLY AND PADDED WITH BLANKS       00005500
WTOR1    LA    RM,MESSAGE1    POINT RM AT FIRST MESSAGE           00005600
         BAL   R10,EXWTOR     LINK TO WTOR ROUTINE                00005700
         SR    RR,RR          ERROR COUNT                         00005800
*                                                                  00005900
*        ANALYZE FIRST REPLY & DETERMINE WHAT TO DISPLAY           00006000
*        OR WHETHER TO TERMINATE.                                  00006100
*                                                                  00006200
REPLANAL EQU   *                                                   00006300
         SPACE                                                     00006500
*        LOOK FOR OPTION - IF NONE, DEFAULT WILL BE ALL            00006600
         SPACE                                                     00006700
         LA    R2,REPLY                                            00006800
         LA    R3,9           LIMIT REGISTER                       00006900
         SR    R4,R4                                               00007000
COMCOMP  CLC   0(1,R2),COMMA  IS THIS A COMMA?                     00007100
         BE    FOUNDCOM       YES, BRANCH OUT                     00007200
         CLC   0(1,R2),BLANK  IS IT A BLANK?                       00007300
         BE    SHOWALL        YES, WE WILL SHOW ALL OF PROC       00007400
         LA    R4,1(R4)       INCREMENT COUNT                     00007500
         LA    R2,1(R2)       INCREMENT POINTER                   00007600
         CR    R4,R3          HAVE WE LOOKED AT 9 CHARACTERS?     00007700
         BNL   ERROR1         ERROR IF NO BLANK OR COMMA IN       00007800
*                              FIRST 9 CHARACTERS.                 00007900
         B     COMCOMP        OTHERWISE, GO LOOK AT NEXT CHARACTER 00008000
FOUNDCOM EQU   *              NEXT CHARACTER SHOULD BE A,D OR N    00000100
         BAL   R14,MEMBMOVE                                        00000200
         CLC   1(1,R2),N      NONE TO BE SHOWN?                    00000300
         BE    NOSHOW                                              00000400
         CLC   1(1,R2),D      SHOW DEFAULTS ONLY?                  00000500
         BE    SHOWDEFL                                            00000600
         MVI   DEFAULT,X'00'    TURN DEFAULT BYTE OFF             00000625
         EJECT                                                     00000650
SHOWPROC EQU   *              SHOW ENTIRE PROC                     00000700
*                             FALL THROUGH FROM ABOVE OR FROM SHOWALL 00000800
         MVI   MEMBER1,X'FF'  SET SWITCH TO SHOW MEMBER NAME  WJB 00000840
*                              NOT FOUND YET                  WJB 00000880
         FIND  PROC,MEMBNAME,D     LOCATE MEMBER                   00000900
         LTR   R15,R15             ZERO RETURN CODE?               00000940
         BP    ERROR1              NO, GO TRY FOR THE MEMBER NAME AGAIN 00000980
READPROC L     R5,BLKCOUNT    NUMBER OF BYTES IN BLOCK             0000'000
         READ  PROCDECB,SF,PROC,DATA                               00001100
         CHECK PROCDECB                                            00001200
         L     R1,PROCDECB+16      LOAD A(IOB)                     00001300
         SH    R5,14(R1)      14 INTO IOB IS THE NUMBER OF BYTES   00001400
*                              IN BLOCK NOT READ                   00001500
         SR    R11,R11        CLEAR CARD COUNT REGISTER            00001600
         LA    R6,DATA        START OF DATA                        00001700
MOVEPROC MVC   WTOAREA+2(80),0(R6)     MOVE INTO WORK AREA         00001800
         TM    MEMBER1,X'FF'                              WJB 00001801
```

```
         BNO   DEECHECK       IF MEMBER NAME FOUND, BR TO DEECHECK   WJR 00001802
         LA    R3,WTOAREA+5                                         WJR 00001803
         LA    R9,WTOAREA+72                                        WJR 00001804
         LH    R8,ONE         LOAD INCREMENT INTO R8                WJR 00001805
         SR    R4,R4          CLEAR OUT FOR INCREM. WHEN NAME FOUND WJR 00001806
NAMECOMP CLC   0(4,R3),MEM    BEGIN SEARCH FOR MEMBER NAME          WJR 00001807
         BE    FOUNDMEM                                             WJR 00001808
         CLC   0(5,R3),MEMB                                         WJR 00001809
         BE    FOUNMEMB                                             WJR 00001810
         CLC   0(7,R3),MEMBER                                       WJR 00001811
         BE    FOMEMBER                                             WJR 00001812
         CLC   0(5,R3),EXEC2                                        WJR 00001813
         BE    EXECFOUN                                             WJR 00001814
         LA    R3,1(R3)                                             WJR 00001815
         BXH   R3,R8,DEECHECK  IF NOT FOUND AFTER 70 CHARACTERS,    WJR 00001816
*                               BRANCH TO DEECHECK                  WJR 00001817
         BNE   NAMECOMP       GO BACK TO KEEP SEARCHING             WJR 00001818
         SPACE 3                                                    00001819
FOUNDMEM AH    R3,FOUR                                              WJR 00001820
         B     COMPARE1                                             WJR 00001821
FOUNMEMB AH    R3,FIVE                                              WJR 00001822
         B     COMPARE1                                             WJR 00001823
FOMEMBER AH    R3,SEVEN                                             WJR 00001824
         B     COMPARE1                                             WJR 00001825
         SPACE 3                                                    00001826
COMPARE1 LR    R8,R3                                                WJR 00001827
COMPARE2 CLC   0(1,R3),BLANK                                       WJR 00001828
         BE    READYMOV                                             WJR 00001829
         CLC   0(1,R3),COMMA                                        WJR 00001830
         BE    READYMOV                                             WJR 00001831
         LA    R4,1(R4)       INCREMENT COUNT                       WJR 00001832
         LA    R3,1(R3)       INCREMENT ADDRESS                     WJR 00001833
         BNE   COMPARE2                                             WJR 00001834
         SPACE                                                      00001835
EXECFOUN MVI   MEMBER1,X'00'  TURN SWITCH TO SHOW EXEC CARD FOUND  WJR 00001836
         B     DEECHECK                                             WJR 00001837
         SPACE                                                      00001838
READYMOV BAL   R14,MOVENAME                                        WJR 00001839
SHOWNAME MVI   MEMBER1,X'00'  TURN SWITCH TO SHOW MEMBER NAME FOUND WJR 00001840
         B     PROCWTO                                              WJR 00001841
         SPACE 5                                                    00001842
DUMMOVE2 MVC   MEMBNAME(0),0(R8)                                    WJR 00001843
         SPACE                                                      00001844
MOVENAME SH    R4,ONE                                               WJR 00001845
         MVC   MEMBNAME,BLANK                                       WJR 00001846
         EX    R4,DUMMOVE2                                          WJR 00001847
         BR    R14                                                  WJR 00001848
DEECHECK TM    DEFAULT,X'FF' SHOW DEFAULTS ONLY?                    00001900
         BO    JCLCHECK      YES, GO ANALYZE                        00002000
PROCWTO  EQU   *                                                    00002100
         LA    RM,WTOAREA    PREPARE FOR WTO                        00002200
         BAL   R10,EXWTO                                            00002300
         AH    R6,EIGHTY     POINT AT NEXT CARD                     00002400
         SH    R5,EIGHTY     NUMBER OF BYTES LEFT                   00002500
         BP    MOVEPROC      IF ANYMORE CARDS ARE LEFT,GO PROCESS   00002600
         B     READPROC      IF NO MORE, GO READ ANOTHER BLOCK      00002700
         SPACE 5                                                    00002800
JCLCHECK EQU   *             ANALYZE JCL TO FIND PROC CARDS ONLY    00002900
         LTR   R7,R7         IS THIS THE FIRST CARD - IF SO MUST BE 00003000
*                             A PROC CARD(MY ASSUMPTION)            00003100
```

```
         LA    R7,1(R7)        BUMP CARD COUNTER                        00003200
         BZ    PROCWTO         IF FIRST CARD WTO IT                     00003300
         LA    R2,WTOAREA+4    POINT AT FIRST NON-SLASH                 00003400
*                              (FIRST 2 BYTES OF AREA ARE LENGTH CODE)  00003500
         CLC   0(1,R2),BLANK   CHARACTER A BLANK?                       00003600
         BNE   PROCEND         IF NOT MUST BE THE EXEC CARD             00003700
EXECSCAN LA    R2,1(R2)        LOOK AT NEXT CHAR.                       00003800
         CLC   0(1,R2),BLANK   BLANK?                                   00003900
         BE    EXECSCAN        YES,LOOK AT NEXT                         00004000
         CLC   0(5,R2),EXEC2   IF NON-BLANK, IS IT 'EXEC'?              00004100
         BE    PROCEND         YES, WE'VE SHOWN ALL DEFAULTS            00004200
         B     PROCWTO         NO,WTO THIS CARD                         00004300
         EJECT                                                          00004400
PROCEND  EQU   *               THE PROCEDURE HAS BEEN WRITTEN OUT,AS    00004500
*                              MUCH AS REQUESTED.  NOW WE WILL REPEAT   00004600
*                              THE ABOVE FOR THE CONTROL STATEMENTS.    00004700
*                              WE WILL ASSUME FOR NOW THAT THE          00004800
*                              CONTROL MEMBER IS NAMED THE SAME AS      00004900
*                              THE PROC MEMBER.  W. BRADFORD WILL       00005000
*                              GENERALIZE                               00005100
         SPACE 5                                                        00005200
         LA    RM,MESSAGE2     PREPARE FOR WTOR                         00005400
         BAL   R10,EXWTOR      ASK IF SHOW CONTROL STATEMENTS           00005500
         CLC   REPLY(1),N      NO SHOW?                                 00005600
         BE    CONTREOD        IF SO, GO TO NEXT PHASE                  00005700
         SPACE 5                                                        00005800
         FIND  CONTROL,MEMBNAME,D LOCATE MEMBER                         00005900
         LTR   R15,R15                 GOOD RETURN?                     00005910
         BZ    READCONT                YES, CONTINUE                    00005920
         LA    RM,MESSAGE9             TELL USER WE WONT LIST CONTROLS  00005930
         BAL   R10,EXWTO                                                00005940
         B     CONTREOD                                                 00005950
READCONT L     R5,BLKCOUNT    NO. OF BYTES IN BLOCK                    00006000
         READ  CONTDECB,SF,CONTROL,DATA                                 00006100
         CHECK CONTDECB                                                 00006200
         L     R1,CONTDECB+16 LOAD A(IOB)                               00006300
         SH    R5,14(R1)       SEE PROCREAD                             00006400
         LA    R6,DATA         START OF DATA                            00006500
MOVECONT MVC   WTOAREA+2(80),0(R6)   MOVE INTO WORK AREA                00006600
         LA    RM,WTOAREA                                               00006700
         BAL   R10,EXWTO                                                00006800
         AH    R6,EIGHTY       POINT AT NEXT CARD                       00006900
         SH    R5,EIGHTY       NUMBER OF BYTES LEFT                     00007000
         BP    MOVECONT        IF ANY MORE GO PROCESS                   00007100
         B     READCONT        OTHERWISE, READ ANOTHER BLOCK            00007200
         EJECT                                                          00007300
CONTREOD EQU   *               WE ARE NOW READY TO CREATE A JOB.        00000100
NOSHOW   EQU   *               COME HERE FOR NO SHOW OF PROC, CONTROL   00000150
         SPACE 5                                                        00000200
*        FIRST, REQUEST THE JOB NAME.  THE REPLY WILL BE 8, 3 OR 1       00000300
*        CHARACTERS LONG.  IF 8 THEN THE REPLY IS THE JOB NAME.          00000400
*        IF 3 THEN THE REPLY IS A SUFFIX TO K3SYS.                       00000500
*        IF REPLY IS 'U' THEN K3SYSSUF WILL BE USED                      00000600
         SPACE 5                                                        00000700
*        FIRST DETERMINE WHICH JOB DATA SET TO BE USED                   00000800
         TM    JOBNO,X'FF'     FLIP JOBNO AS SOON AS WE TEST IT.        00000900
*                               AFTER FLIP '00' MEANS SET 1.             00001000
*                               'FF' MEANS SET2.                         00001100
         BNO   FLIP                                                     00001200
         MVC   JOBNO,SET1      MOVE IN '00'                             00001300
```

```
         OPEN  (JOB1,(OUTPUT))                                          00001400
         R     JOBNAME                                                  00001450
FLIP     MVC   JOBNO,SET2      MOVE IN 'FF'                             00001500
         OPEN  (JOB2,(OUTPUT))                                          00001600
JOBNAME  EQU   *                                                       00001700
         LA    RM,MESSAGE3     REQUEST JOBNAME                          00001800
         BAL   R10,EXWTOR                                               00001900
         CLC   REPLY(2),U      DEFAULT?                                 00002000
         BE    DEFALJOB        YES, WILL USE K3SYSSUF                   00002100
         CLC   REPLY+7(1),BLANK   IS EIGHTH CHAR, PRESENT               00002200
         BE    THREECHR        IF IT'S NOT THERE,THEN USE K3SYS         00002300
         MVC   JOBID(8),REPLY      OTHERWISE, USE ALL EIGHT             00002400
         B     PUTJOB                                                   00002500
THREECHR MVC   JOBID(5),DEFJOBID  K3SYS                                 00002600
         MVC   JOBID+5(3),REPLY   XYZ                                   00002700
         B     PUTJOB                                                   00002800
DEFALJOB MVC   JOBID(8),DEFJOBID  K3SYSSUF                              00002900
         EJECT                                                          00003000
PUTJOB   EQU   *               WRITE OUT JOBCARD                        00003100
         TM    JOBNO,X'FF'     SET 2?                                   00003200
         BO    PUTJOB2                                                  00003300
         PUT   JOB1,JOBOUT                                              00003400
         B     INPUTASK                                                 00003500
PUTJOB2  PUT   JOB2,JOBOUT                                              00003600
         SPACE 5                                                        00003700
INPUTASK EQU   *                                                       00003800
*        WE ARE NOW READY TO QUERY THE OPERATOR FOR THE                 00003900
*        JOB STREAM TO BE EXECUTED. WE ALREADY HAVE THE                 00004000
*        JOB CARD.  ANY FURTHER CARDS WILL BE TYPED IN NOW              00004100
*        UP TO AN 'END' CARD.                                           00004200
         SPACE 5                                                        00004300
         LA    RM,MESSAGE5       REQUEST CARD OR 'END'                  00004400
         BAL   R10,EXWTOR        GO WTOR                                00004500
         CLC   REPLY(3),END      END CARD?                              00004600
         BE    CLOSESET          YES,GO CLOSE DATA SET                  00004700
PUTOUT   TM    JOBNO,X'FF'       SET 2?                                 00004800
         BO    PUT2              YES, PUT INTO SET 2                    00004900
         PUT   JOB1,REPLY        OTHERWISE INTO SET 1                   00005000
         B     INPUTASK          GET ANOTHER CARD                       00005100
PUT2     PUT   JOB2,REPLY                                               00005200
         B     INPUTASK          GET ANOTHER CARD                       00005300
         SPACE 5                                                        00005400
CLOSESET EQU   *                                                       00005500
         TM    JOBNO,X'FF'       SET 2?                                 00005600
         BO    CLOSE2                                                   00005700
         CLOSE (JOB1)                                                   00005800
         B     SRDR              GO START A READER TO THE DATA SET      00005900
CLOSE2   CLOSE (JOB2)                                                   00006000
         B     SRDR              GO START A READER TO THE DATA SET      00006100
         EJECT                                                          00006150
EXITCODE EQU   *                                                       00006200
         CLOSE (PROC,,CONTROL)                                          00006300
         LA    RM,MESSAGE7       ANNOUNCE TERMINATION                   00006340
         BAL   R10,EXWTO                                                00006380
         L     R13,SAVE+4                                               00006400
         RETURN (14,12)                                                 00006500
         SPACE 5                                                        00006600
SHOWALL  EQU   *                                                       00006700
*        THE SECTION ENTERED IF ONLY THE MEMBER NAME IS ENTERED FOR THE 00006800
*        INITIAL QUERY.  THE DEFAULT IS TO SHOW ALL LINES OF THE MEMBER 00006900
```

```
         SPACE 3                                                        00007000
         MVI   DEFAULT,X'00'       TURN DEFAULT BYTE OFF                00007100
         BAL   R14,MEMBRMOVE       MOVE MEMBER NAME FOR FIND            00007200
         B     SHOWPROC                                                 00007300
         SPACE 5                                                        00007400
SHOWDEFI EQU   *                                                        00007500
*        THIS SECTION ENTERED IF ONLY THE PROC STATEMENT IS TO BE SHOWN 00007600
         SPACE 3                                                        00007700
         MVI   DEFAULT,X'FF'       TURN DEFAULT BYTE ON                 00007800
         SR    R7,R7                CLEAR CARD COUNTER                  00007850
         B     SHOWPROC                                                 00007900
         SPACE 5                                                        00008000
ERROR1   EQU   *                                                        00008100
*        INCORRECT REPLY TO FIRST QUERY                                 00008200
*        E.G. TOO LONG A MEMBER NAME                                    00008300
         LA    RB,1(RB)            BUMP ERROR COUNT                     00008400
         CH    RB,MAXERROR         EXCEEDED MAX?                        00008500
         BL    RETRY               LESS THAN 3 TRIES, TRY AGAIN         00008600
         LA    RM,MESSAGE6         OTHERWISE, TERMINATE                 00008700
         BAL   R10,EXWTO                                                00008800
         B     EXITCODE            GO TERMINATE                         00008900
         SPACE 5                                                        00009000
RETRY    EQU   *                                                        00009100
*        ENTERED IF ERROR COUNT FOR ERROR1 IS LESS THAN 3               00009200
         SPACE 3                                                        00009300
         LA    RM,MESSAGE8         REQUEST REPLY AGAIN                  00009400
         BAL   R10,EXWTOR                                               00009500
         B     REPLYANAL           ANALYZE RESPONSE                     00009600
         SPACE 5                                                        00009700
DUMMYMOV MVC   MEMBNAME(0),REPLY USED TO MOVE PROCLIB MEMBER           00009800
*                                    INTO PLACE FOR FIND WITH AN EX     00009900
         SPACE 5                                                        00010000
MEMBRMOVE SH   R4,ONE              DECREMENT FOR MVC CODE               00010100
         MVC   MEMBNAME,BLANK      CLEAR OUT MEMBNAME                   00010200
         EX    R4,DUMMYMOV         EXECUTE MVC ABOVE                    00010300
         BR    R14                 GO BACK                              00010400
         EJECT                                                          00010500
EXWTOR   MVI   REPLY,X'40'              BLANK OUT REPLY AREA            00000100
         MVC   REPLY+1(119),REPLY                                       00000200
         MVC   WTORMSG+8(2),0(RM)       MOVE IN MSG LENGTH              00000300
         MVC   WTORMSG+12(120),2(RM)    MOVE IN MSG                     00000400
         LA    RM,WTOLN                                                 00000500
         LH    RW,WTOLN                                                 00000600
         AR    RM,RW                                                    00000700
         MVC   0(4,RM),DESC             MOVE IN ROUT AND DESC CODES     00000800
         LH    RW,=H'120'               LENGTH OF REPLY                 00000900
         MVI   WTORECB,X'00'                              CLEAR ECB     00000925
         MVC   WTORECB+1(3),WTORECB                                     00000950
         WTOR  ,REPLY,(RW),WTORECB,MF=(E,WTORMSG)          ISSUE WTOR   00001000
         WAIT  ECB=WTORECB                                 WAIT         00001100
         CLC   REPLY(4),EXIT            DOES USER WISH TO EXIT?         00001140
         BE    EXITCODE                 IF SO, GET OUT                  00001180
         BR    R10                                        GET BACK JOJO 00001200
EXWTO    MVC   WTOLN(2),0(RM)                                           00001300
         MVC   WTOLN+4(120),2(RM)                                       00001400
         LA    RM,WTOLN                                                 00001500
         LH    RW,WTOLN                                                 00001600
         AR    RM,RW                                                    00001700
         MVC   0(4,RM),DESC                                             00001800
         WTO   MF=(E,WTOLN)                                             00001900
```

```
         BR    R10                                                  00002000
SRDR     CLI   JOBNO,X'FF'       SEE IF USING JOB2                  00002100
         BNE   ITSA1              NO                                 00002200
         MVI   NAMQUAL,C'2'                                          00002300
         B     ITSAOK                                               00002400
ITSA1    MVI   NAMQUAL,C'1'                                         00002500
ITSAOK   SR    0,0                SET UP REG 0                       00002600
         LA    1,NAMELEN          PUT ADDR OF COMMAND IN REG 1       00002700
         SVC   34                 ISSUE SVC FOR START                00002800
         B     WTOR1              GET BACK CORRETTO                  00002900
         EJECT                                                       00000010
         SPACE 5                                                     00000020
*****    SUFFER MESSAGES          *****                              00000030
         SPACE 5                                                     00000040
MESSAGE1 DC    H'68'                                                 00000100
         DC    C'ENTER ROUTINE NAME AND DISPLAY OPTIONS: ALL, DEFAULTS *00000200
               OR NOTHING.  '                                        00000300
MESSAGE2 DC    H'40'                                                 00000500
         DC    C'DISPLAY CONTROL STATEMENTS?  Y OR N.'               00000600
MESSAGE3 DC    H'70'                                                 00000700
         DC    C'ENTER JOBNAME DESIRED OR SUFFIX TO XXXXX OR ''U'' TO U*00000800
               SE XXXXXXXX.  '                                       00000900
MESSAGE4 DC    H'49'                                                 00000920
         DC    C'ENTER ''EXIT'' AT ANY TIME TO STOP SUFFERING.       00000940
MESSAGE5 DC    H'72'                                                 00001000
         DC    C'ENTER JCL OR CONTROL STATEMENTS.  ENTER ''END'' TO EXF*00001100
               CUTE YOUR JOB.  '                                     00001200
MESSAGE6 DC    H'66'                                                 00001300
         DC    C'INCORRECT ROUTINE NAME SPECIFIED THREE TIMES.  SUFFER *00001400
               NO MORE!!'                                            00001500
MESSAGE7 DC    H'27'                                                 00001600
         DC    C'YOUR SUFFERING IS OVER!'                            00001700
MESSAGE8 DC    H'48'                                                 00001800
         DC    C'INCORRECT ROUTINE NAME SPECIFIED. TRY AGAIN.'       00001900
MESSAGE9 DC    H'48'                                                 00002000
         DC    C'MEMBER NOT FOUND OR I/O ERROR. NO LISTING.          00002100
         EJECT                                                       00000050
         LTORG                    BEGINETH THE CONSTANT VARIABLES    00000100
*                                                                    00000200
*    AREA FOR WTO(R) LIST FORMS                                      00000300
*                                                                    00000400
WTORMSG  DS    0F                                                    00000500
         DS    F                  FOR REPLY LENGTH AND ADDR          00000600
         DS    F                  FOR ECB                            00000700
WTOLN    DC    X'0000'            --- THIS BEGINS WTO LIST F  M      00000800
         DC    X'8000'            WTO(R) LIST FORM                   00000900
         DC    CL120' '           MSG AREA                           00001000
DESC     DC    X'0400'            DESCRIPTOR CODE                    00001100
ROUT     DC    X'8000'            ROUTING CODE                       00001200
         DS    0F                                                    00001300
WTORECB  DC    F'0'               ECB FOR WTOR                       00001400
REPLY    DS    CL120              REPLY AREA                         00001500
*                                                                    00001600
*  AREA FOR DATA SET NAMES FOR RDRSUFR                               00001700
*                                                                    00001800
         DS    0F                                                    00001900
NAMEREG  EQU   *                                                     00002000
NAMELEN  DC    AL2(NAMELN,0)                                         00002100
NAMEDS   DC    C'S RDRSUFR,DSN=SYS2.SUFFER.JOB'                      00002200
NAMQUAL  DC    C'1 '                                                 00002300
```

```
NAMEND   EQU   *                                                      00002400
NAMELN   EQU   NAMEND-NAMEREG                                        00002500
SAVE     DS    18F                                                    00002600
JOBNO    DC    X'FF'                                                  00002700
ROUT13   DC    X'0008'              13TH BIT ON                       00002800
ROUT1    DC    X'8000'              1ST BIT ON                        00002900
BLANK    DC    CL8' '                                                 00003100
COMMA    DC    C','                                                   00003200
A        DC    C'A'                                                   00003300
D        DC    C'D'                 DEFAULTS                          00003400
N        DC    C'N'                 NOLISTING                         00003500
EXIT     DC    C'EXIT'                                                00003600
MAXERROR DC    H'3'                                                   00003700
ONE      DC    H'1'                                                   00003800
         EJECT                                                        00003850
         EJECT                                                        00004100
*                                                                     00004200
*  MORE CONSTANTS TO BE PRESENTED MONOTONICALLY                       00004300
*                                                                     00004400
BLKCOUNT DC    F'800'                                                 00004500
MEMBNAME DS    CL8                                                    00004600
DATA     DS    1000                                                   00004700
EIGHTY   DC    H'80'                                                  00004800
DEFAULT  DS    CL1                                                    00004900
EXEC     DC    C'EXEC'                                                00005000
SET1     DC    X'00'                                                  00005100
SET2     DC    X'FF'                                                  00005200
*                              JOB CARD TO BE BUILT                   00005250
JOBOUT   DS    0CL80                                                  00005300
SLASHES  DC    C'//'                                                  00005400
JOBID    DS    CL8                                                    00005500
JOBREST  DC    C' JOB (K30651311G,5,AAAAAA,N99200),SYS,MSGLEVEL=1'    00005600
JOBFILL  DC    CL22' '                                                00005700
DEFJOBID DC    C'K3SYSSUF'                                            00005800
*                                                                     00005900
END      DC    C'END'                                                 00006000
WTOAREA  DS    0CL82                                                  00006100
LENGTH   DC    X'0050'              80 BYTES LONG                     00006200
WORDS    DS    CL80                                                   00006300
U        DC    C'U '                                                  00006350
SYSIN    DC    C'SYSIN'                                               00006360
STAR     DC    C'*'                                                   00006370
NEWSUFF  DC    C'DSN=SYS2.SUFFER.UTIL,DISP=SHR'                       00006380
TWO      DC    C'2'                                                   00006390
         EJECT                                                        00006400
PROC     DCB   BLKSIZE=800,LRECL=80,DSORG=PO,DDNAME=PROCLIB,          *00006500
               EODAD=PROCEOD,MACRF=(R),RECFM=FB                      00006600
CONTROL  DCB   BLKSIZE=800,LRECL=80,DSORG=PO,DDNAME=CONTROL,         *00006700
               EODAD=CONTREOD,MACRF=(R),RECFM=FB                     00006800
JOB1     DCB   BLKSIZE=800,LRECL=80,RECFM=FB,DDNAME=JOB1,            *00006900
               DSORG=PS,MACRF=(PM)                                   00007000
JOB2     DCB   BLKSIZE=800,LRECL=80,RECFM=FB,DDNAME=JOB2,            *00007100
               DSORG=PS,MACRF=(PM)                                   00007200
JSCCIB   DSECT                                                       00000100
         DS    0D                                                    00000300
         DS    F                                                     00000500
         DS    XL1                                                   00000700
         DS    XL1                                                   00000900
         DS    XL6                                                   00001100
CIRCONID DS    XL1                                                   00001300
         DS    XL1                                                   00001500
         DS    H                                                     00001700
         END                                                         00000700
```

APPENDIX C

REQUIRED SUFFER PROCEDURES

71.083 11.09 RDRSUFF

```
//IEFPROC EXEC PGM=IEFIRC,REGION=50K,                              00000100
//     PARM='01203C020040249100112314     E00020'                  00000200
//IEFRDER DD DSN=SYS2.SUFFER.PRCCLIE(DUMP),DISP=SHR                00000300
//IEFPDSI  DD  DSN=SYS2.SUFFER.PROCLIB,DISP=SHR                    00000400
//    DD DSN=SYS1.PROCLIB,DISP=SHR                                 00000500
//IEFDATA DD UNIT=2314,SPACE=(TRK,(1,20),,CONTIG),                 00000600
//       DCB=(RECFM=F,LRECL=80,BLKSIZE=80,BUFNO=2,BUFL=80)         00000700
```

END OF MEMBER .... 7 RECORDS PROCESSED.

71.083 11.09 SUFFER

```
//SUFFER EXEC PGM=SUFFER                                           00000100
//PROCLIB DD DSN=SYS2.SUFFER.PROCLIB,DISP=SHR                      00000200
//CONTROL DD DSN=SYS2.SUFFER.CONTROL,DISP=SHR                      00000300
//JOB1 DD DSN=SYS2.SUFFER.JOB1,DISP=SHR                            00000400
//JOB2 DD DSN=SYS2.SUFFER.JOB2,DISP=SHR                            00000500
//SYSABEND DD SYSOUT=A,SPACE=(CYL,(5,2))                           00000600
```

END OF MEMBER .... 6 RECORDS PROCESSED.

APPENDIX D

SAMPLE MEMBERS OF SYS2.SUFFER.PROCLIB AND SYS2.SUFFER.CONTROL

71.077 11.10 DUMP

```
//DEFAULT PROC VOL=M2DRM1,T=SYSTPE,MEM=DUMP                    00000100
//DMPRES EXEC PGM=IEHDASDR,REGION=100K                         00000200
//SYSPRINT DD SYSOUT=A                                         00000300
//DD1 DD VOL=REF=&VOL,DISP=SHR                                 00000400
//T01 DD UNIT=2400-9,VOL=SER=&T,DISP=(,KEEP)                   00000500
//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR              00000600

END OF MEMBER ....     6 RECORDS PROCESSED.
```

71.077 11.10 IEHIOSUP

```
//DEFAULT PROC SYSRES=M2DRM1                                   00000100
//IOSUP EXEC PGM=IEHIOSUP                                      00000200
//SYSPRINT DD SYSOUT=A                                         00000300
//SYSUT1 DD DSN=SYS1.SVCLIB,DISP=OLD,VOL=REF=&SYSRES           00000400

END OF MEMBER ....     4 RECORDS PROCESSED.
```

71.077 11.10 IEHLIST

```
//DEFAULT PROC MEM=LISTPDS,CVOL=M2DRM1,VOL=M2SYS4              00000100
//IEHLIST EXEC PGM=IEHLIST                                     00000200
//SYSPRINT DD SYSOUT=A                                         00000300
//CVOLUME DD VOL=REF=&CVOL,DISP=SHR                            00000400
//VOLUME DD VOL=REF=&VOL,DISP=SHR                              00000500
//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR              00000600

END OF MEMBER ....     6 RECORDS PROCESSED.
```

71.077 11.10 LISTCTLG

```
//DEFAULT PROC MEM=LISTCTLG,CVOL=M2DRM1,VOL=M2SYS4             00000100
//IEHLIST EXEC PGM=IEHLIST                                     00000200
//SYSPRINT DD SYSOUT=A                                         00000300
//CVOLUME DD VOL=REF=&CVOL,DISP=SHR                            00000400
//VOLUME DD VOL=REF=&VOL,DISP=SHR                              00000500
//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR              00000600

END OF MEMBER ....     6 RECORDS PROCESSED.
```

71.077 11.10 LISTVTOC

```
//DEFAULT PROC MEM=LISTVTOC,CVOL=M2DRM1,VOL=M2SYS4                    00000100
//IEHLIST EXEC PGM=IEHLIST                                            00000200
//SYSPRINT DD SYSOUT=A                                                00000300
//CVOLUME DD VOL=REF=&CVOL,DISP=SHR                                   00000400
//VOLUME DD VOL=REF=&VOL,DISP=SHR                                     00000500
//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR                     00000600
```

END OF MEMBER .... 6 RECORDS PROCESSED.

71.077 11.10 RENAME

```
//DEFAULT PROC CVOL=M2DRM1                                            00000100
//CATLG EXEC PGM=IEHPROGM                                             00000200
//SYSPRINT DD SYSOUT=A                                                00000300
//CVOLUME DD VOL=REF=&CVOL,DISP=SHR                                   00000400
```

END OF MEMBER .... 4 RECORDS PROCESSED.

71.077 11.10 RESTORE

```
//DEFAULT PROC VOL=M2DRM1,T=SYSTPE,MEM=RESTORE                        00000100
//DMPRES EXEC PGM=IEHDASDR,REGION=100K                                00000200
//SYSPRINT DD SYSOUT=A                                                00000300
//DD1 DD VOL=REF=&VOL,DISP=SHR                                        00000400
//FROM1 DD UNIT=2400-9,VOL=SER=&T,DISP=(OLD,KEEP)                     00000500
//SYSIN DD DSN=SYS2.SUFFER.CONTROL(&MEM),DISP=SHR                     00000600
```

END OF MEMBER .... 6 RECORDS PROCESSED.

71.077 11.10 SUPERZAP

```
//DEFAULT PROC DSN='SYS1.LINKLIB'                                     00000100
//ZAP EXEC PGM=SUPERZAP                                               00000200
//SYSLIB DD DSN=&DSN,DISP=SHR                                         00000300
//SYSPRINT DD SYSOUT=A                                                00000400
//*      //SYSIN DD * NEEDED FOR INPUT                                00000500
```

END OF MEMBER .... 5 RECORDS PROCESSED.

```
1.077  11.10                                   DUMP

   DUMP   FROMDD=DD1,TODD=T01                                    00000100

 END OF MEMBER ....      1 RECORDS PROCESSED.

71.077  11.10                                   IEHLIST

   LISTPDS DSNAME=SYS1.SVCLIB                                    00000
   LISTPDS DSNAME=SYS1.LINKLIB1,VOL=2314=M2SYS4                  00000

 END OF MEMBER ....      2 RECORDS PROCESSED.

71.077  11.10                                   LISTCTLG

   LISTCTLG                                                      00000

 END OF MEMBER ....      1 RECORDS PROCESSED.

71.077  11.10                                   LISTVTOC

   LISTVTOC FORMAT,VOL=2314=M2SYS4                               00000001

 END OF MEMBER ....      1 RECORDS PROCESSED.

71.077  11.10                                   RENAME

   THIS IS A DUMMY MEMBER FOR RENAME.                            00000
   THE ACTUAL CONTROL STATEMENTS MUST BE ENTERED AT THE CONSOLE  00000
 THE FORM OF A RENAME STATEMENT IS:                              00000
   RENAME DSNAME=SYS6.DUMMY,VOL=2314=M2SYS8,NEWNAME=CUR(,MEMBER=TOTOT)  00000

 END OF MEMBER ....      4 RECORDS PROCESSED.

71.077  11.10                                   RESTORE

   RESTORE  TODD=DD1,FROMDD=FROM1                                00000

 END OF MEMBER ....      1 RECORDS PROCESSED.
```