



154

MVS

July 1999

In this issue

- 3 Year 2000 compliance issues
 - 4 System symbols in batch jobs
 - 7 A disk space storage manager monitor
 - 42 Writing a user SMF record
 - 49 Extracting DDname information
 - 52 Y2K, SVC screening update
 - 58 An IPL subsystem (part 2)
 - 72 MVS news
-

© Xephon plc 1999

update

MVS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 33598
From USA: 01144 1635 33598
E-mail: xephon@compuserve.com

North American office

Xephon/QNA
1301 West Highway 407, Suite 201-405
Lewisville, TX 75067
USA
Telephone: 940 455 7050

Contributions

If you have anything original to say about MVS, or any interesting experience to recount, why not spend an hour or two putting it on paper? The article need not be very long – two or three paragraphs could be sufficient. Not only will you be actively helping the free exchange of information, which benefits all MVS users, but you will also gain professional recognition for your expertise, and the expertise of your colleagues, as well as some material reward in the form of a publication fee – we pay at the rate of £170 (\$250) per 1000 words for all original material published in *MVS Update*. If you would like to know a bit more before starting on an article, write to us at one of the above addresses, and we'll send you full details, without any obligation on your part.

Editor

Jaime Kaminski

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, EXECs, and other contents of this journal before making any use of it.

MVS Update on-line

Code from *MVS Update* can be downloaded from our Web site at <http://www.xephon.com>; you will need the user-id shown on your address label.

Subscriptions and back-issues

A year's subscription to *MVS Update*, comprising twelve monthly issues, costs £340.00 in the UK; \$505.00 in the USA and Canada; £346.00 in Europe; £352.00 in Australasia and Japan; and £350.00 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1992 issue, are available separately to subscribers for £29.00 (\$43.00) each including postage.

© Xephon plc 1999. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Year 2000 compliance issues

Only a few months remain until the millennium. By now most systems will be in the final stages of testing, and will be running on the Y2K versions of their software, as detailed in supplier letters that were most probably obtained back when the project started.

Rather than assuming all is well, now is the time to re-contact your suppliers just to be sure. Although this may seem overkill and many suppliers will simply direct you to their Web sites for the relevant information, it can still be a useful exercise.

POTENTIAL PROBLEMS

In the majority of cases you will simply receive re-assurances that all is well, but be especially careful that you check what is said. For example, do any of the following apply to your site?

- References to products going out of service before the year-end despite being compliant.
- Statements that a release higher than the current one is now regarded as the 'officially' compliant level.

Such statements should be cause for concern, and my personal experience suggests there is a good chance that you may get some such responses. This could lead to some difficult decisions that may require the assistance of your auditors or legal department.

The following questions need to be considered: do you rely on your testing as a guarantee of Y2K compliance, or do you have a site requirement that you run on the currently official version? If the software is going out of service, can you risk going through the year-end on unsupported software? Is there enough time to reinstall and re-check software?

Hopefully your site will not be subject to such problems, but as always throughout the Y2K project, the key phrase has been 'never assume'.

System symbols in batch jobs

INTRODUCTION

With the advent of sysplex and shared PARMLIB, PROCLIB, and LINKLIB, it has become necessary to differentiate between the participants of a sysplex. One example is the ISPF profile dataset, which cannot be shared. At one of my previous sites, we used an ISPF profile DSN of ISP.&SYSNAME.ISPPROF. During JCL conversion, the &SYSNAME is replaced with the MVS SYSID.

However, there is a catch – this conversion applies only to started tasks (STC) and TSO user-ids (TSU). With the IEFUJV exit I developed, this is now also available for batch jobs.

The use of IEFUJV was necessary, since it gets control after PROC expansion, but before conversion.

The whole routine centers around the use of the IBM-supplied routine ASASYMBM, which does the actual replacement. A check is first done to see if any ‘&’ characters occur on the JCL image. No checking is done to see whether the caller is a JOB, STC, or TSU, since it is installed as a dynamic exit. With dynamic exits, one can activate exits STC.IEFUJV, TSU.IEFUJV, and SYS.IEFUJV. By defining this exit for SYS.IEFUJV, no further testing is required.

To activate, ensure that SMFPRMxx member has IEFUJV in the SYS(EXITS) list. This activates the exit point for IEFUJV. It is no longer required that member IEFUJV reside in the LPA, or that the module be named IEFUJV. Next, alter the PROGxx member as follows:

```
EXIT ADD
  EXITNAME(SYS.IEFUJV)
  MODNAME(PXFUJVS0)
  DSNNAME(your.load.library)
```

To activate immediately, issue operator command ‘SET SMF=xx’ (if IEFUJV is not yet active). Then, use the command:

```
SETPROG EXIT,ADD,EN=SYS.IEFUJV,MOD=PSXUJVS0,D
```

SOURCE CODE

```

*****
* SOME TECHNIQUES (EG RELATIVE USINGS) ARE USED WHICH REQUIRE THE      *
* HIGH-LEVEL ASSEMBLER. ALSO, THE "PUNCH ' SETOPT PARM(REUS(RENT))"    *
* STATEMENT WILL WORK ONLY FOR THE BINDER OF DFSMS 1.3 OR LATER.      *
* THIS PUNCHED BINDER STATEMENT WILL OVERRIDE ANY JCL PARMS,          *
* PREVENTING PROBLEMS DUE TO INCORRECT LINKAGE JCL.                    *
*****

        EJECT
        PRINT OFF,NOPRINT
* LKDPARM=RENT
* STDUSE=NO
        PRINT ON,NOPRINT
        PUNCH ' SETOPT PARM(REUS(RENT)) '
        PUNCH ' MODE AMODE(31),RMODE(ANY) '
        PUNCH ' ENTRY PSXUJVSØ '
PSXUJVSØ RSECT
PSXUJVSØ AMODE 31
PSXUJVSØ RMODE ANY
        USING PSXUJVSØ,R15
        B      CODE_START
        DC     AL1(L'EP_LITERAL)
EP_LITERAL      DC  C'PSXUJVSØ..DATE=&SYSDATC..TIME=&SYSTIME..UJV EXIT T+
                O REPLACE SYSTEM SYMBOLIC PARMS'
CODE_START      DS  ØH
                BAKR  R14,Ø                .SAVE REGS USING H/W STACK
                LR    R12,R15
                DROP  R15
                USING PSXUJVSØ,R12
                LM    R8,R11,Ø(R1)
                USING JMR,R8
                TM    Ø(R1Ø),X'1Ø'+X'2Ø'    .POST CONVERSION/INTERPRETATION
                BNZ   DONE                  .IF EITHER, GET OUT
                TRT   Ø(71,R9),TEST_TABLE   .ANY '&' IN CARD?
                BZ    DONE                  .NO - DUCK OUT
                STORAGE OBTAIN,LENGTH=WLEN,LOC=RES
                LR    R13,R1
                LR    R14,R1                .CLEAR OBTAINED AREA
                LR    R15,RØ
                LR    RØ,R14
                XR    R15,R15
                MVCL  R14,RØ
                MVC   Ø(4,R14),=C'F1SA'    .INDICATE USE OF REG.STACK
                USING WRKSTOR,R13
                USING SYMBP,MYSYMBP
                XC    SYMBP(SYMBP_LEN),SYMBP INITIALIZE TO ZERO
                ST    R9,SYMBPPATTERN@     INPUT PATTERN
                MVC   SYMBPPATTERNLENGTH,=A(71)
                LA    R1,ALTERED_CARD      ADDRESS OF TARGET
                ST    1,SYMBPTARGET@       SAVE IN SYMBP AREA
                MVC   TARGETLENGTH,=A(256) SET LENGTH OF TARGET
                LA    1,TARGETLENGTH        ADDRESS OF TARGET LENGTH

```

```

        ST      1,SYMBPTARGETLENGTH@      SAVE IN SYMBP AREA
        LA      1,RETURNCODE              ADDRESS OF RETURN CODE
        ST      1,SYMBPRETURNCODE@        SAVE IN SYMBP AREA
        LINK    EP=ASASYMBM,MF=(E,MYSYMBP)
        OC      RETURNCODE,RETURNCODE
        BZ      COPY_RESULT
CONVERT_ERROR DS  0H
        MVC     WT01+4(47),=C'PSXUJVS0: RETURN CODE FROM ASASYMBP=X''...+
            .1...''
        UNPK   WT01+4+38(9),RETURNCODE(5)
        MVC     WT01+4+38+8,C''''
        TR      WT01+4+38(8),HEX-C'0'
        MVC     WT01(4),=Y(51,0)
        WTO     MF=(E,WT01)
        B       RETURN
COPY_RESULT   DS  0H
        MVC     0(71,R9),ALTERED_CARD
RETURN        DS  0H
        LR      R1,R13
        STORAGE RELEASE,LENGTH=WSLEN,ADDR=(1)
DONE          DS  0H
        XR      15,15
        PR
        EJECT
TEST_TABLE    DC  XL256'0'
            ORG  TEST_TABLE+C'&&'
            DC  X'FF'
            ORG
HEX           DC  C'0123456789ABCDEF'
            EJECT
            LTORG
            EJECT
WRKSTOR       DSECT
MYSAVE        DS  9D
DUB           DS  D
MYSYMBP       DS  CL(SYMBP_LEN)          SYMBP AREA
RETURNCODE    DS  F                      RETURN CODE
TARGETLENGTH  DS  F                      LENGTH OF TARGET
ALTERED_CARD  DS  CL256
WT01          WTO  '
                                                    +
                                                    +
                                                    ' ,MF=L
WSLEN         EQU  *-WRKSTOR
            IEFJMR
            ASASYMBP
R0            EQU  0
            etc ...
R15          EQU  15
            END

```

Pieter Wiid
Systems Programmer (South Africa)

© Xephon 1999

A disk space storage manager monitor

INTRODUCTION

The DiskSpace Storage Manager Monitor (DSSMM) was designed to enable the billing and the control of the disk-space usage by groups of users/customers. The ownership of the datasets is described in the input diagram dataset. The sample structure of the diagram dataset is as follows:

Dataset name	Account (Group Owner)
CAT?????.ICF*	SOFTWARE
ADR.BKWEK.ID???????	IDEAL
*.BKUPD.???ØØ2	DATADICT

The wildcard characters (*, ?) are allowed. The structure of the diagram dataset is a fixed one:

- The dataset name must start in column 1 (up to 44 characters).
- The account name must start in column 46 (up to 15 characters).

This file is allocated by the '//DIAGRADD DD' with the following DCB characteristics: DSORG=PS, RECFM=FB, LRECL=80.

The following reports are produced and displayed on the '//REPORTDD DD' statement:

- 1 The datasets by volume report – dataset name, VOLSER, DSORG, date last referenced, number of extensions, space (allocated in KB, used in KB, %used, unusable (wasted due to non-optimal block size) in KB, %unusable), date created.
- 2 The datasets by accounts (groups) report – the same as above plus the total space allocated (in KB and in MB) per account (group).
- 3 The total space allocated by accounts (groups) – account (group) name, space allocated in KB and in MB.
- 4 The datasets allocated on the previous day – yesterday's allocated datasets.

- 5 The not catalogued datasets report – the same as item 2 but for the uncatalogued datasets.

A sequential file is created. This file contains the data from the report number 3 and it can be used as an input to the PC-based graphical utility.

This file is allocated on the ‘//PCFILE DD’ statement with the DCB: RECFM=FB, LRECL=80, DSORG=PS.

This file can be imported to a Microsoft Access database and a Microsoft Access report can be generated. It can also be processed using Microsoft Visual Basic.

The dataset defined by the ‘//MODDIAGR DD’ statement must be preallocated as an empty dataset with the following DCB characteristics: RECFM=FB, DSORG=PS, LRECL=240.

SPACECHK must be APF authorized (linked with AC=1 into the APF authorized library). The UCBSCAN macro is invoked to create a list of all existing disk volumes. Then DCOLLECT IDCAMS service is invoked from the program. The output from DCOLLECT is processed and sorted.

OPERATIONAL ENVIRONMENT

SPACECHK was tested in the following DFSMS environments: MVS 5.2.2, OS/390 Version 1 Release 3, and OS/390 Version 2 Release 6. The procedure SUTIME was described in *MVS Update*, Issue 102, page 70, which is downloadable from the Xephon Web site.

A copy of the date conversion subroutine SUDATE has not been supplied because most shops will have different requirements. Any date handling procedure can be used to perform the conversion of the date from TIME macro format to a specific format. The procedure CONVFICL was published in *RACF Update*, Issue 3, page 21. The procedure IEFSD095 (block character routine) is a standard MVS procedure.

To link the block character subroutine concatenate to the ‘//SYSLIB DD’ statement the following standard MVS library:

```
// DD DSN=SYS1.AOSB0,DISP=SHR for IEFSD095 block character rtne
```


SPACECHEK JCL

The following JCL should be used to run SPACECHEK:

```
//jobname JOB (acc,nt),'pl?grammer',CLASS=A,MSGCLASS=T,REGION=2M
//*-----
//ALLOCAT EXEC PGM=IEFBR14
//ALLO1 DD DISP=(NEW,PASS),DSN=&&DCOLL,UNIT=SYSDA,
// SPACE=(TRK,(30,30)),DCB=(RECFM=VB,LRECL=644)
//ALLO2 DD DISP=(NEW,PASS),DSN=&&TABCO,UNIT=SYSDA,
// SPACE=(TRK,(10,30)),DCB=(RECFM=FBA,LRECL=133)
//SPACECHK EXEC PGM=SPACECHK,REGION=2M,
// PARM='your company name and address (48 chars max) '
//STEPLIB DD DSN=your.apf.auth.LIB,DISP=SHR
//PRINTOUT DD SYSOUT=T
//SYSOUT DD DUMMY
//REPORTDD DD SYSOUT=A
//PCFILE DD DISP=SHR,DSN=your.PC.file
//MODDIAGR DD DISP=(OLD,KEEP),DSN=your.seq.MODDIAGR.output.file
//DIAGRADD DD DISP=SHR,DSN=your.sequential.diagram.input
//INPUTDDD DD DISP=(MOD,PASS),DSN=&&DCOLL
//INIDCAMS DD DISP=(NEW,PASS),SPACE=(TRK,(50,50)),UNIT=SYSDA,DSN=&&B
//OUIDCAMS DD DISP=(NEW,PASS),SPACE=(TRK,(50,50)),UNIT=SYSDA,DSN=&&C
//SORTIN DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(50,50))
//SORTOUT DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(50,50))
//YESTERDA DD DISP=(NEW,PASS),SPACE=(TRK,(50,50)),UNIT=SYSDA,DSN=&&Y
//NOTCATAL DD DISP=(NEW,PASS),SPACE=(TRK,(50,50)),UNIT=SYSDA,DSN=&&N
//MYSOIN DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(10,10))
//MYSOOUT DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(10,10))
//WORKFILA DD DISP=(MOD,PASS),DSN=&&TABCO
```

SPACECHEK

```
R0 EQU 0
.....
R15 EQU 15
CONVFICH CSECT
        USING *,R10,R11 ESTABLISH ADDRESSABILITY
        STM R14,R12,12(R13) SAVE3 REGISTERS
        LR R10,R15 SET FIRST BASE REGISTER
        LA R11,2048(R10) SET SECOND BASE REGISTER
        LA R11,2048(R11) AND INCREMENT TO PROPER VALUE
        LR R12,R13 STORE PREVIOUS SA ADDRESS
        LR R2,R1 (R2) = POINTER TO ADDRESS OF THE PARM LIST
        GETMAIN R, LV=500
        LR R9,R1 (R9)= ADDR. OF THE ALLOCATED VIRT. STORAGE AREA
        LTR 15,15
        BZ OKGETMAT
        B FINI
OKGETMAT EQU *
```

```

        USING SVC34DSE,R9          ESTABLISH ADDRESSABILITY
        LA    R13,SAVE3           LOAD ADDRESS OF THIS SAVE3 AREA
        ST    R12,SAVE3+4        CHAIN BACKWARDS
        ST    R13,8(R12)         CHAIN FORWARD
        L     R3,Ø(R2)           (R3) = ADDRESS OF THE FIRST PARAMETER
        LA    R1,FIXNUMBE
        MVC   Ø(4,R1),Ø(R3)      MOVE FIXNUMBE
        L     R1,FIXNUMBE
        CVD   R1,PACKFIEL
        MVC   COPYPATE(1Ø),PATTERN
        ED    COPYPATE(1Ø),PACKFIE2
        L     R3,4(R2)           (R3) = ADDRESS OF THE SECOND PARAMETER
        LA    R1,COPYPATE
        MVC   Ø(8,R3),2(R1)      MOVE RESULT
FINI    EQU    *
        FREEMAIN R,LV=5ØØ,A=(R9)
        L     R13,4(R13)
        LR    R15,R7
        RETURN (14,12),RC=(15)
PATTERN DC    XL1Ø'4Ø2Ø2Ø2Ø2Ø2Ø2Ø2Ø212Ø'
        LTORG
SVC34DSE DSECT
SAVE3    DS    18F
PACKFIEL DS    ØPL8
         DS    PL3
PACKFIE2 DS    PL5
FIXNUMBE DS    F
COPYPATE DS    CL1Ø
        END
RØ       EQU    Ø
.....
R15      EQU    15
SECATX   CSECT
        USING *,R1Ø,R11         ESTABLISH ADDRESSABILITY
        STM   R14,R12,12(R13)   SAVE3 REGISTERS
        LR    R1Ø,R15           SET FIRST BASE REGISTER
        LA    R11,2Ø48(R1Ø)     SET SECOND BASE REGISTER
        LA    R11,2Ø48(R11)     AND INCREMENT TO PROPER VALUE
        LR    R12,R13           STORE PREVIOUS SA ADDRESS
        LA    R13,SAVE3         LOAD ADDRESS OF THIS SAVE3 AREA
        ST    R12,SAVE3+4       CHAIN BACKWARDS
        ST    R13,8(R12)       CHAIN FORWARD
        LR    R2,R1             (R1) = POINTER TO ADDRESS OF THE PARAM. LIST
        L     R3,4(R2)          (R3) = ADDRESS OF THE SECOND PARAM.
        L     R2,Ø(R2)         (R2) = ADDRESS OF THE FIRST PARAM.
        LA    R1,DSNAME
        MVC   Ø(44,R1),Ø(R2)    MOVE DSNAME
        LOCATE CAM
        LR    R7,R15
        LTR   R7,R7
OKLOCATE EQU    *

```

```

        LA    R1,FIVOLSER
        MVC   Ø(6,R3),Ø(R1)
FINI    EQU   *
        LR    R15,R7
        L     R13,4(R13)
        RETURN (14,12),RC=(15)
* LOCATE - CAMLST VARIABLES
DSNAME DC    CL44'
CAM     CAMLST NAME,DSNAME,,LOCAREA
        DS    ØD
LOCAREA DS    ØCL265
COUNTNUM DS  CL2          ACCOUNT OF THE NUMBER OF VOLUMES IN THE LIST
FIRSTENT DS   ØCL12        FIRST ENTRY
FIDEVCOD DS   ØCL4         DEVICE CODE
FIDEBY16 DS   CL1
FIDEBY17 DS   CL1          OPTIONAL FEATURES
FIDEBY18 DS   CL1          DEVICE CLASS: X'8Ø'-MAGN. TAPE, X'2Ø'-DASD
FIDEBY19 DS   CL1          UNIT TYPE
FIVOLSER DS   CL6          VOLUME SERIAL NUMBER
FIDSNSEQ DS   CL2          DATASET SEQUENCE NUMBER
NEXTEXTE DS   CL251
SAVE3    DS   18F
        LTOrg
        END
RØ      EQU   Ø
.....
R15    EQU   15
SUBGETPA CSECT
        USING *,R1Ø,R11      ESTABLISH ADDRESSABILITY
        STM   R14,R12,12(R13) SAVE3 REGISTERS
        LR    R1Ø,R15        SET FIRST BASE REGISTER
        LA    R11,2Ø48(R1Ø)   SET SECOND BASE REGISTER
        LA    R11,2Ø48(R11)   AND INCREMENT TO PROPER VALUE
        LR    R12,R13        STORE PREVIOUS SA ADDRESS
        LR    R2,R1 (R2) = POINTER TO ADDRESS OF THE PARM LIST
GETMATD GETMAIN R,LV=5ØØ
        LR    R9,R1 (R9)= ADDR. OF THE ALLOCATED VIRT. STORAGE AREA
        LTR   15,15
        BZ    OKGETMAT
        B     FINI
OKGETMAT EQU   *
        USING SVC34DSE,R9     ESTABLISH ADDRESSABILITY
        LA    R13,SAVE3        LOAD ADDRESS OF THIS SAVE3 AREA
        ST    R12,SAVE3+4      CHAIN BACKWARDS
        ST    R13,8(R12)       CHAIN FORWARD
        L     R3,Ø(R2)         (R3) = ADDRESS OF THE FIRST PARAMETER
        LA    R1,PARMADDR
        MVC   Ø(4,R1),Ø(R3)    MOVE PARMADDR
        L     R3,4(R2)         (R3) = ADDRESS OF THE SECOND PARM
        ST    R3,OUTADDRE
        L     R8,PARMADDR
        L     R8,Ø(R8)         (R8) = FULLWORD

```

```

LH      R12,Ø(R8)          (R12) = LENGTH OF THE PARM FIELD
LA      R1,Ø
CR      R1,R12             TEST IF LENGTH = Ø
BNE     TESLE48           LENGTH NE Ø
MVC     PARMLIST(48),BLANK LENGTH= Ø
B       FREEM             END
TESLE48 EQU *
LA      R1,48
CR      R12,R1            TEST IF LENGTH OF THE PARM FIELD GT 48
BNH     FIADPAFI         LENGTH LE 48 - OK
XR      R12,R12
LA      R12,48           LOAD MAX PERMITTED LENGTH
FIADPAFI EQU *
MVC     PARMLIST(48),BLANK
A       R8,=F'2'
XR      R3,R3
LA      R3,1
LA      R4,PARMLIST
LOOPLOAD EQU *
MVC     Ø(1,R4),Ø(R8)
AR      R4,R3
AR      R8,R3
BCT     R12,LOOPLOAD
L       R8,OUTADDRE
LA      R1,PARMLIST
MVC     Ø(48,R8),Ø(R1)   MOVE PARMADDR
FREEM   EQU *
FREEMAIN R,LV=5ØØ,A=(R9)
FINI    EQU *
L       R13,4(R13)
LR      R15,R7
RETURN (14,12),RC=(15)
SAVE3   DS 18F
PARMADDR DS F
OUTADDRE DS F
NUMBER  DS F
RESULT8 DS CL8
CHARACTE DS CL16
PARMLIST DS CL48
BLANK   DC CL133' '     BLANK
        LTORG
SVC34DSE DSECT
        END
RØ      EQU Ø
.....
R15     EQU 15
SPACDIAG CSECT
        USING *,R1Ø,R11   ESTABLISH ADDRESSABILITY
        STM R14,R12,12(R13) SAVE3 REGISTERS
        LR R1Ø,R15       SET FIRST BASE REGISTER
        LA R11,2Ø48(R1Ø) SET SECOND BASE REGISTER
        LA R11,2Ø48(R11) AND INCREMENT TO PROPER VALUE

```

```

LR      R12,R13          STORE PREVIOUS SA ADDRESS
GETMATD GETMAIN R,LV=3000
LR      R9,R1  (R9)= ADDR. OF THE ALLOCATED VIRT. STORAGE AREA
LTR     15,15
BZ     OKGETMAT
B      FINI
OKGETMAT EQU *
LA     R13,SAVE3        LOAD ADDRESS OF THIS SAVE3 AREA
ST     R12,SAVE3+4      CHAIN BACKWARDS
ST     R13,8(R12)       CHAIN FORWARD
ST     R9,R9SAVE
USING  DUVBSNEW,R9     ESTABLISH ADDRESSABILITY
OPEN   (PRINTDCB,(OUTPUT))
OPEN   (DIAGRDCB,(INPUT))
OPEN   (MODDIDCB,(OUTPUT))
LOOPGDIA EQU *
GET     DIAGRDCB,DIAGRAM
MVI    TABDIAGR,C' '
MVC    TABDIAGR+1(L'TABDIAGR-1),TABDIAGR
MVC    TABACCOU(15),PATTACCO
LA     R7,44
LA     R2,PATTDSN
LA     R3,TABDIAGR
LR     R4,R3
LOOPTABD EQU *
MVC    Ø(1,R4),Ø(R2)
CLC    Ø(1,R2),=C'.'
BE     DOTFOUND
CLC    Ø(1,R2),BLANK
BE     BLANKFOU
B      PARMINCR
DOTFOUND EQU *
MVI    Ø(R4),C' '      REPLACE DOT BY BLANK
XR     R1,R1
LA     R1,9
AR     R3,R1
LR     R4,R3
B      PARMINC1
BLANKFOU EQU *
B      EOFPADSN
PARMINCR EQU *
XR     R1,R1
LA     R1,1
AR     R4,R1
PARMINC1 EQU *
XR     R1,R1
LA     R1,1
AR     R2,R1
BCT    R7,LOOPTABD
EOFPADSN EQU *
PUT    MODDIDCB,TABDIAGR
B      LOOPGDIA

```

```

ENDDIAGR EQU *
          CLOSE (DIAGRDCB)
          CLOSE (MODDIDCB)
FREEM    EQU *
          L      R9,R9SAVE
          FREEMAIN R, LV=3000, A=(R9)
          CLOSE (PRINTDCB)
FINI     EQU *
          L      R13,4(R13)
          LA     R15,0
          RETURN (14,12),RC=(15)

SAVE3    DS    18F
NUMBER   DS    F
R9SAVE   DS    F
PRINT    DS    CL133
BLANK    DC    CL133' '
          DS    0D
PRINTDCB DCB   MACRF=PT,RECFM=FBA,LRECL=133,BLKSIZE=133,DSORG=PS, *
          DDNAME=PRINTOUT
DIAGRDCB DCB   MACRF=GM,DSORG=PS,RECFM=FB,LRECL=80, *
          DDNAME=DIAGRADD,EODAD=ENDDIAGR
MODDIDCB DCB   MACRF=(GM,PM),DSORG=PS,RECFM=FB,LRECL=240, *
          DDNAME=MODDIAGR

          LTORG
DUVBSNEW DSECT          MAPPING MACRO
          DS    0D
DIAGRAM  DS    0CL80
PATTDSN  DS    CL44,CL1  PATTERN OR DSNAME
PATTACCO DS    CL15,CL20 ACCOUNT NAME
TABDIAGR DS    0CL240
TABEIG01 DS    CL8,CL1   1ST
TABEIG02 DS    CL8,CL1   2
TABEIG03 DS    CL8,CL1   3
TABEIG04 DS    CL8,CL1   4
TABEIG05 DS    CL8,CL1   5
TABEIG06 DS    CL8,CL1   6
TABEIG07 DS    CL8,CL1   7
TABEIG08 DS    CL8,CL1   8
TABEIG09 DS    CL8,CL1   9
TABEIG10 DS    CL8,CL1  10
TABEIG11 DS    CL8,CL1  11
TABEIG12 DS    CL8,CL1  12
TABEIG13 DS    CL8,CL1  13
TABEIG14 DS    CL8,CL1  14
TABEIG15 DS    CL8,CL1  15
TABEIG16 DS    CL8,CL1  16
TABEIG17 DS    CL8,CL1  17
TABEIG18 DS    CL8,CL1  18
TABEIG19 DS    CL8,CL1  19
TABEIG20 DS    CL8,CL1  20
TABEIG21 DS    CL8,CL1  21
TABEIG22 DS    CL8,CL1  22

```

```

TABACCOU DS    CL15,CL35          ACCOUNT NAME
          END
R0       EQU   0
.....
R15     EQU   15
SPACUCBS CSECT
          USING *,R10,R11          ESTABLISH ADDRESSABILITY
          STM  R14,R12,12(R13)     SAVE3 REGISTERS
          LR   R10,R15             SET FIRST BASE REGISTER
          LA   R11,2048(R10)       SET SECOND BASE REGISTER
          LA   R11,2048(R11)       AND INCREMENT TO PROPER VALUE
          LR   R12,R13            STORE PREVIOUS SA ADDRESS
GETMATD  GETMAIN R,LV=3000
          LR   R9,R1 (R9)= ADDR. OF THE ALLOCATED VIRT. STORAGE AREA
          LTR  15,15
          BZ   OKGETMAT
          B    FINI
OKGETMAT EQU   *
          LA   R13,SAVE3           LOAD ADDRESS OF THIS SAVE3 AREA
          ST   R12,SAVE3+4        CHAIN BACKWARDS
          ST   R13,8(R12)         CHAIN FORWARD
          ST   R9,R9SAVE
          USING DUVBSNEW,R9       ESTABLISH ADDRESSABILITY
          OPEN (PRINTDCB,(OUTPUT))
          OPEN (MYSOIN,(OUTPUT))
          MVC  DIAGRAM(80),BLANK
          MVC  DIAGRAM(36),=C' DCOLLECT OUTFILE(INPUTDDD) VOLUMES('
          MVI  DIAGRAM+42,C')'
*          EXECUTE UCB SCAN SERVICE
          XC   SCANWORK,SCANWORK   CLEAR WORK AREA
          LA   R2,UCBAREA
          USING UCBOB,R2          SET UP ADDRESSABILITY TO UCB
SEARCH   EQU   *
          UCBSCAN COPY,WORKAREA=SCANWORK,UCBAREA=(2),          *
          DEVCCLASS=DASD,DYNAMIC=YES,CMXTAREA=CMXTOKEN,RANGE=ALL
          LTR  R15,R15            HAS A UCB BEEN RETURNED?
          BNZ  DONE
          CLC  UCBVOLI(6),=X'00000000000000'
          BE   SEARCH
          MVC  PRINT,BLANK
          MVC  PRINT+1(6),VOLSER
          PUT  PRINTDCB,PRINT
          MVC  VOLSER(6),UCBVOLI
          MVC  DIAGRAM+36(6),VOLSER
          PUT  MYSOIN,DIAGRAM
          B    SEARCH
DONE     EQU   *
          CLOSE (MYSOIN)
          LA   R1,MYSOSORT        LOAD PARAMETER LIST
          LINK EP=ICEMAN
          OPEN (MYSOOUT,(INPUT))
          OPEN (IDCINDCB,(OUTPUT))

```

```

GETLOOP EQU *
GET     MYSOOUT, SORTPRIN
PUT     IDCINDCB, SORTPRIN
B       GETLOOP
ENDODATA EQU *
CLOSE  (MYSOOUT)
CLOSE  (IDCINDCB)
FREEM  EQU *
L      R9, R9SAVE
FREEMAIN R, LV=3000, A=(R9)
CLOSE  (PRINTDCB)
FINI   EQU *
L      R13, 4(R13)
LR     R15, R7
RETURN (14, 12), RC=(15)

SAVE3  DS 18F
NUMBER DS F
R9SAVE DS F
WORKAREA DS CL100
SORTPRIN DS CL80
PRINT   DS CL133
BLANK   DC CL133' '
        DS 0D
MYSOSORT DC X'80', AL3(MYSOLST)
        CNOP 2, 4
MYSOLST DC AL2(MYSOEND-MYSOBEG)
MYSOBEG DC A(CACSORTA)          STARTING ADDRESS OF SORT STMT
        DC A(CACSORTZ)          ENDING ADDRESS OF SORT STMT
        DC A(CACRECA)          STARTING ADDRESS OF RECORD STMT
        DC A(CACRECB)          ENDING ADDRESS OF RECORD STMT
        DC A(0)                NO E15 EXIT
        DC A(0)                NO E35 EXIT
        DC C'MYSO'

MYSOEND EQU *
CACSORTA DC C' SORT FIELDS=(37,6,CH,A)'
CACSORTZ DC C' '
CACRECA  DC C' RECORD TYPE=F, LENGTH=80 '
CACRECB  DC C' '
UCBPTR   DS F                  UCB COPY FROM SCAN
CMXTOKEN DS CL32              PIN TOKEN
SCANWORK DS CL100             WORK AREA FOR UCBSCAN
TEXTSCAN DC CL58'PIN TEXT FOR UCBSCAN'
UCBAREA  DS CL48
        DS 0D
IDCINDCB DCB MACRF=PM, RECFM=FB, DSORG=PS, LRECL=80, BLKSIZE=3120, *
          DDNAME=INIDCAMS
PRINTDCB DCB MACRF=PT, RECFM=FBA, LRECL=133, BLKSIZE=133, DSORG=PS, *
          DDNAME=PRINTOUT
MYSOIN   DCB MACRF=(GM, PM), RECFM=FB, LRECL=80, DSORG=PS, *
          DDNAME=MYSOIN
MYSOOUT  DCB MACRF=(GM, PM), RECFM=FB, DSORG=PS, LRECL=80, *
          DDNAME=MYSOOUT, EODAD=ENDODATA

```



```

LTORG
DUVBSNEW DSECT          MAPPING MACRO
          DS      ØD
DIAGRAM  DS      ØCL8Ø
PATTDSDN DS      CL44,CL1      PATTERN OR DSNAME
PATTACCO DS      CL15,CL2Ø     ACCOUNT NAME
          DS      ØD
OUTRECOR DS      ØCL13Ø
DSN      DS      CL44,CL1      DATASET NAME
VOLSER   DS      CL6,CL3       VOLSER
DSORG    DS      CL2,CL1       DATASET ORGANIZATION
DLASTREF DS      CL8,CL1       DATE
NEXTENTS DS      CL3,CL1       NUMBER OF EXTENTS USED
KBYTALLO DS      CL1Ø,CL1      NUMBER OF KBYTES ALLOCATED
KBYTUSED DS      CL1Ø,CL1      NUMBER OF KBYTES USED
PERCALL1 DS      CL3           PERCENTAGE: (SPACEUNUSED/SPALLOCATED)*1ØØ%
PERCADOT DS      CL1           DECIMAL DOT
PERCALL2 DS      CL2,CL1       PERCENTAGE
KBUNUSAB DS      CL1Ø,CL1     NUMBER OF KBYTES UNUSABLE IN BLOCKS
PERCUNU1 DS      CL3           PERCENTAGE: (SPACEUNUSABLE/SPALLOCATED)*1ØØ%
PERCUDOT DS      CL1           DECIMAL DOT
PERCUNU2 DS      CL2,CL5       PERCENTAGE
DCREATED DS      CL8,CL1Ø     CREATION DATE
          DS      ØD
DSECTIEF DSECT
          IEFUCBOB          UCB MACRO ID
          CVT  DSECT=YES
          END
RØ      EQU  Ø
.....
R15     EQU  15
          PRINT GEN
SPACECHK CSECT
          USING *,R1Ø,R11,R12  ESTABLISH ADDRESSABILITY
          LR   R1Ø,R15         SET FIRST BASE REGISTER
          LA   R11,2Ø48(R1Ø)    SET SECOND BASE REGISTER
          LA   R11,2Ø48(R11)    AND INCREMENT TO PROPER VALUE
          LA   R12,2Ø48(R11)    SET SECOND BASE REGISTER
          LA   R12,2Ø48(R12)    AND INCREMENT TO PROPER VALUE
          STM  R14,R12,12(R13)  SAVE3 REGISTERS
          LR   R2,R13          STORE PREVIOUS SA ADDRESS
          LA   R13,SAVE3        LOAD ADDRESS OF THIS SAVE3 AREA
          ST   R2,SAVE3+4      CHAIN BACKWARDS
          ST   R13,8(R2)       CHAIN FORWARD
          LR   R2,R1 (R2) = POINTER TO ADDRESS OF THE PARM LIST
          ST   R2,PARMADDR
GETMATD GETMAIN R,LV=4ØØØ
          LR   R9,R1 (R9)= ADDR. OF THE ALLOCATED VIRT. STORAGE AREA
          LTR  15,15
          BZ   OKGETMAT
          B    FINI

```

```

OKGETMAT EQU *
ST R9,R9SAVE
USING DUVBSNEW,R9 ESTABLISH ADDRESSABILITY
CALL SUBGETPA,(PARMADDR,YOURCOMP),VL
OPEN (PRINTDCB,(OUTPUT))
MVI PRINT,C' '
MVC PRINT+1(L'PRINT-1),PRINT
MVC PRINT+1(5),=C'START'
MVC PRINT+7(8),TEXTSPAC
PUT PRINTDCB,PRINT
CALL SUDATE,(DATE),VL
MVC STATDATE(3),DATENAME
MVC STATDATE+4(2),DATEDAY
MVC STATDATE+7(3),DATEMONT
MVC STATDATE+11(4),DATEYEAR
TIME BIN
ST R0,NUMBER
CALL SUTIME,(NUMBER,TIMESTAM),VL
MVC STATTIME(8),TIMESTAM
BAL R8,CONVDATE
CALL SPACDIAG PROCESS DIAGRAM FILE
CALL SPACUCBS PROCESS UCBS
LA R7,0
STH R7,OPTILIST
LOAD EP=IDCAMS
LR R15,R0
CALL (15),(OPTILIST,DNAMELIS),VL
LR R7,R15
LTR R7,R7
BNZ IDCAMSER
B OKIDCAMS
IDCAMSER EQU *
ST R7,NUMBER
CALL CONVFIH,(NUMBER,RESULT),VL
MVC PRINT(133),BLANK
MVC PRINT+1(4),=C'IDRC'
MVC PRINT+41(8),RESULT
PUT PRINTDCB,PRINT
OPEN (IDCOUDCB,(INPUT))
GETOUTPU EQU *
MVC IDCAMREC(130),BLANK
GET IDCOUDCB,IDCAMREC
MVC PRINT(133),BLANK
MVC PRINT+1(130),IDCAMREC
PUT PRINTDCB,PRINT
B GETOUTPU
ENDOUTPU EQU *
CLOSE (IDCOUDCB)
XR R1,R1
LA R1,4
CR R7,R1
BE OKIDCAMS

```

```

OKIDCAMS B      FREEM
          EQU    *
          OPEN  (TABCODECB,(OUTPUT))
          LA    R7,Ø
          OPEN  (INDCB,(INPUT))
          OPEN  (REPORDCB,(OUTPUT))
          OPEN  (STATSDCB,(OUTPUT))
          MVC   BLOCK1(9),TEXTSPAC
          MVC   BLOCK2(9),BLANK
          MVC   BLOCK2+1(7),STORAGE
          MVC   BLOCK3(9),MANAGER
          MVC   BLOCK4(9),MONITOR
          MVI   IYOURNAM,C'1'      PRINT OF YOURNAME REQUIRED
          BAL   R8,BLOCKLET        WRITE HEADER
          MVC   NUMBER(4),BLANK
          MVC   BLOCK1(9),TEXTSPAC
          MVC   BLOCK2(9),BLANK
          MVC   BLOCK2(4),DATASETS
          MVC   BLOCK2+5(4),DATASETS+4
          MVC   BLOCK3(9),BLANK
          MVC   BLOCK3(2),BY
          MVC   BLOCK3+3(6),VOLSERS
          MVC   BLOCK4(9),BLANK
          MVC   BLOCK4+1(6),REPORT
          BAL   R8,BLOCKLET        WRITE HEADER
          MVC   NEWTEX1A(38),NEWPAT1A
          MVC   NEWTEX1B(38),NEWPAT1B
          MVC   NEWTEX1C(38),NEWPAT1C
          MVC   NEWTEX1D(16),NEWPAT1D
          MVC   NEWTEX1Z(38),BLANK
          MVC   NEWTEX1E(38),NEWPAT1E
          MVC   NEWTEX1F(38),NEWPAT1F
          MVC   NEWTEX1G(17),NEWPAT1G
          MVC   NEWTEX1H(38),NEWPAT1H
          MVC   NEWTEX1I(38),NEWPAT1I
          XR    R5,R5
          LA    R5,5Ø              EST. LINES PER PAGE NUMBER
          XR    R4,R4
          LA    R4,1              EST. LINES PER PAGE COUNTER
          ST    R4,PAGENUM
          MVI   PRINT,C' '
          MVC   PRINT+1(L'PRINT-1),PRINT
          MVC   PRINT+2Ø(8),DATASETS
          MVC   PRINT+29(2),BY
          MVC   PRINT+32(7),VOLSERS
          MVC   PRINT+4Ø(6),REPORT
          CALL  CONVFICH,(PAGENUM,RESULT),VL
          MVC   PRINT+54(4),RESULT+4
          PUT   TABCODECB,PRINT
          MVC   PREVVOLU(6),=C'$$$$$$'
          LA    R2,RDATA
          USING DCUOUTH-4,R2      ESTABLISH ADDRESSABILITY

```

```

LOOPGET  EQU      *
         GET      INDCB,RDATA
         CLI      DCURCTYP,C'D'          TEST THE RECORD TYPE
         BNE      LOOPGET                NOT A DATA-TYPE RECORD
         MVC      OUTRECOR(130),BLANK
         MVC      STATSREC(150),BLANK
         TM       DCDDSORG,B'10000000'
         BNO      NOTINDEX
         MVC      DSORG(2),=C'IS'
         B        OKDSORG
NOTINDEX EQU      *
         TM       DCDDSORG,B'01000000'
         BNO      NOTSEQUE
         MVC      DSORG(2),=C'PS'
         B        OKDSORG
NOTSEQUE EQU      *
         TM       DCDDSORG,B'00100000'
         BNO      NOTDIREC
         MVC      DSORG(2),=C'DA'
         B        OKDSORG
NOTDIREC EQU      *
         TM       DCDDSORG,B'00000010'
         BNO      NOTPARTI
         MVC      DSORG(2),=C'PO'
         B        OKDSORG
NOTPARTI EQU      *
         TM       DCDDSORG,B'00000001'
         BNO      NOTUNMOV
         MVC      DSORG(2),=C'U'
         B        OKDSORG
NOTUNMOV EQU      *
         TM       DCDDSORG+1,B'00001000'
         BNO      NOTVSAM
         MVC      DSORG(2),=C'VS'
         B        OKDSORG
NOTVSAM  EQU      *
         MVC      DSORG(2),=C'??'
OKDSORG  EQU      *
         MVC      SRTYPE(2),DCURCTYP
         MVC      SDSORG(2),DSORG
         MVC      DSN(44),DCDDSNAM
         MVC      SDSN(44),DCDDSNAM
         MVC      VOLSER(6),DCDVOLSR
         MVC      SVOLSER(6),DCDVOLSR
         MVC      SPACALLO(4),DCDALLSP
         MVC      NUMBER(4),DCDALLSP
         CALL     CONVFI CL, (NUMBER,RESULT10),VL
         MVC      KBYTALLO(10),RESULT10
         MVC      SPACUSED(4),DCDUSESP
         MVC      SUNUSABL(4),DCDNMBLK
         CLC      DSORG(2),=C'VS'
         BNE      COMPSPAC

```

```

MVC SPACUSED(4),DCDALLSP SPACE USED = SPACE ALLOCATED
XR R1,R1
ST R1,SUNUSABL SPACE UNUSABLE = 0
COMPSPAC EQU *
MVC NUMBER(4),SPACUSED
CALL CONVFI CL,(NUMBER,RESULT10),VL
MVC KBYTUSED(10),RESULT10
MVC NUMBER(4),SUNUSABL
CALL CONVFI CL,(NUMBER,RESULT10),VL
MVC KBUNUSAB(10),RESULT10
MVC NUMBER(4),DCDCREDIT
BAL R8,CONTOHEX
MVC DCREATED(4),RESULT10+2
MVC DCREATED+5(3),RESULT10+6
MVC SDACREAT(7),RESULT10
MVC NUMBER(4),DCDLSTRF
BAL R8,CONTOHEX
MVC DLASTREF(4),RESULT10+2
MVC DLASTREF+5(3),RESULT10+6
MVC SDALREF(7),RESULT10+2
MVI SCATALOG,C'Y'
CALL SECATX,(SDSN,RESULT),VL TEST IF CATALOGED?
LTR R15,R15
BNZ NOTCTLGD
B CARRYSUC
NOTCTLGD EQU *
MVI SCATALOG,C'N'
CARRYSUC EQU *
XR R7,R7
IC R7,DCDNMEXT
ST R7,NUMBER
CALL CONVFI CH,(NUMBER,RESULT),VL
MVC SNEXTENT(3),RESULT+5
MVC NEXTENTS(3),RESULT+5
* CALCULATE PERCENTAGE: (SPACE USED)/(SPACE ALLOCATED)
L R7,SPACALLO (R7) = KILOBYTES ALLOCATED
SR R1,R1 (R1) = 0
CR R7,R1 TEST IF ZERO
BE ENDPERCA
L R7,SPACUSED (R7) = KILOBYTES USED
SR R6,R6 (R6) = 0
SR R1,R1
LA R1,100
MR R6,R1
L R1,SPACALLO (R1) = KILOBYTES ALLOCATED
DR R6,R1
ST R7,NUMBER
CALL CONVFI CL,(NUMBER,RESULT10),VL
MVC PERCALL1(3),RESULT10+7
LR R7,R6 (R7) = REMAINS
SR R6,R6 (R6) = 0
SR R1,R1

```

```

LA    R1,100
MR    R6,R1
L     R1,SPACALLO      (R1) = KILOBYTES ALLOCATED
DR    R6,R1
ST    R7,NUMBER
CALL  CONVFI CL,(NUMBER,RESULT10),VL
MVI   PERCADOT,C'.'
MVC   PERCALL2(2),RESULT10+8
CLI   PERCALL2,C' '    TEST IF BLANK
BNE   ENDPERCA
MVI   PERCALL2,C'0'    FILL WITH BLANK
ENDPERCA EQU *
MVC   SPERUSED(5),PERCALL1
*    CALCULATE PERCENTAGE: UNUSABLE/ALLOCATED
L     R7,SPACALLO      (R7) = KILOBYTES ALLOCATED
SR    R1,R1
CR    R7,R1            TEST IF ZERO
BE    ENDPERCU
L     R7,SUNUSABL      (R7) = KILOBYTES UNUSABLE IN BLOCKS
SR    R6,R6            (R6) = 0
SR    R1,R1
LA    R1,100
MR    R6,R1
L     R1,SPACALLO      (R1) = KILOBYTES ALLOCATED
DR    R6,R1
ST    R7,NUMBER
CALL  CONVFI CL,(NUMBER,RESULT10),VL
MVC   PERCUNU1(3),RESULT10+7
LR    R7,R6            (R7) = REMAINS
SR    R6,R6            (R6) = 0
SR    R1,R1
LA    R1,100
MR    R6,R1
L     R1,SPACALLO      (R1) = KILOBYTES ALLOCATED
DR    R6,R1
ST    R7,NUMBER
CALL  CONVFI CL,(NUMBER,RESULT10),VL
MVI   PERCUDOT,C'.'
MVC   PERCUNU2(2),RESULT10+8
CLI   PERCUNU2,C' '    TEST IF BLANK
BNE   ENDPERCU
MVI   PERCUNU2,C'0'    FILL WITH BLANK
ENDPERCU EQU *
MVC   SPERUNUS(5),PERCUNU1
CLC   PREVVOLU(6),VOLSER
BE    OLDVOLSE
MVC   PREVVOLU(6),VOLSER
MVC   BLOCK1(9),BLANK
MVC   BLOCK1+1(6),VOLSER
MVC   BLOCK2(9),BLANK
MVC   BLOCK2(4),DATASETS
MVC   BLOCK2+5(4),DATASETS+4

```

```

MVC    BLOCK3(9),BLANK
MVC    BLOCK3+1(7),DISPLAY
MVC    BLOCK4(9),BLANK
MVC    BLOCK4+1(6),REPORT
BAL    R8,BLOCKLET          WRITE HEADER
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+24(6),VOLSEERS
MVC    PRINT+36(6),VOLSER
CALL   CONVFICH,(PAGENUM,RESULT),VL
MVC    PRINT+54(4),RESULT+4
PUT    TABCODCB,PRINT
MVC    TEXTVAR(28),BLANK
MVC    TEXTVAR(18),=C'DATASETS ON VOLUME'
MVC    TEXTVAR+19(6),VOLSER
XR     R5,R5
LA     R5,50                EST. LINES PER PAGE NUMBER
XR     R4,R4
BAL    R8,NEWPAGE
OLDVOLSE EQU *
MVC    PRINT,BLANK
MVC    PRINT+1(130),OUTRECOR
PUT    REPORDCB,PRINT
CR     R4,R5                TEST IF PAGE IS FULL
BNE    PAGENFUL            NOT
XR     R4,R4
BAL    R8,NEWPAGE
PAGENFUL EQU *
A      R4,=F'1'
ST     R2,R2SAVE           SAVE REGISTER
ST     R4,R4SAVE           SAVE REGISTER
ST     R5,R5SAVE           SAVE REGISTER
BAL    R8,COMPARED        ESTABLISH ACCOUNT #
L      R2,R2SAVE           RESTORE REGISTER
L      R4,R4SAVE           RESTORE REGISTER
L      R5,R5SAVE           RESTORE REGISTER
PUT    STATSDCB,STATSREC
B      LOOPGET
ENDATA EQU *
CLOSE (INDCB)
CLOSE (STATSDCB)
MVI    PRINT,C'-'
MVC    PRINT+1(L'PRINT-1),PRINT
MVI    PRINT,C' '
PUT    REPORDCB,PRINT
LA     R1,PARMSORT         LOAD PARAMETER LIST
LINK   EP=ICEMAN          LINK DFSORT
MVC    BLOCK1(9),TEXTSPAC
MVC    BLOCK2(9),BLANK
MVC    BLOCK2+1(6),REPORT
MVC    BLOCK3(9),BLANK
MVC    BLOCK3+3(2),BY

```

```

MVC    BLOCK4(9),BLANK
MVC    BLOCK4(8),ACCOUNTS
BAL    R8,BLOCKLET          WRITE ACCOUNTS REPORT
MVC    NEWTEX1A(38),NEWPAT2A
MVC    NEWPAT2B(38),BLANK
MVC    NEWPAT2B(8),SETNAME
MVC    NEWTEX1B(38),NEWPAT2B
MVC    NEWTEX1C(38),NEWPAT2C
MVC    NEWTEX1D(16),NEWPAT2D
MVC    NEWTEX1Z(38),NEWPAT2Z
MVC    NEWTEX1E(38),BLANK
MVC    NEWTEX1F(38),NEWPAT2F
MVC    NEWTEX1G(17),NEWPAT2G
MVC    NEWTEX1H(38),BLANK
MVC    NEWTEX1I(38),BLANK
MVC    TEXTVAR(28),BLANK
MVC    TEXTVAR(7),ACCOUNTS
XR     R5,R5
LA     R5,5Ø                EST. LINES PER PAGE NUMBER
XR     R4,R4
LA     R4,1                 EST. LINES PER PAGE COUNTER
BAL    R8,NEWPAGE
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+2Ø(6),REPORT
MVC    PRINT+27(2),BY
MVC    PRINT+3Ø(8),ACCOUNTS
CALL   CONVFIH,(PAGENUM,RESULT),VL
MVC    PRINT+54(4),RESULT+4
PUT    TABCODCB,PRINT
OPEN   (SORTODCB,(INPUT))
OPEN   (PCFILE,(OUTPUT))
OPEN   (YESTERDA,(OUTPUT))
OPEN   (NOTCATAL,(OUTPUT))
MVC    PREVGROU(15),BLANK
XR     R1,R1
ST     R1,TOTALLOC         CLEAN THE COUNTER
A      R1,=F'1'
ST     R1,INDPCFIL        STORE PCFIL ON/OFF INDICATOR
LOOPGESO EQU *
GET    SORTODCB,STATSREC
BAL    R8,PROCRECO        PROCESS THE RECORD
CLC    ADATCREA(4),DWORD+1 CHECK IF CREATED YESTERDAY?
BNE    NOTYESTE           NOT
CLC    ADATCREA+5(3),DWORD+5 CHECK IF CREATED YESTERDAY?
BNE    NOTYESTE           NOT
PUT    YESTERDA,STATSREC
NOTYESTE EQU *
CLI    SCATALOG,C'N'
BNE    OKCATALO
PUT    NOTCATAL,STATSREC
OKCATALO EQU *

```



```

B      LOOPGESO
ENDSOROU EQU  *
CLOSE (SORTODCB)
CLOSE (YESTERDA)
CLOSE (NOTCATAL)
MVC   PRINT, BLANK
MVC   PRINT+1(23), TOTKB
CALL  CONVFICL, (TOTALLOC, RESULT10), VL
MVC   PRINT+24(10), RESULT10
PUT   REPORDCB, PRINT
MVC   DIAGRAM(80), BLANK
MVC   DIAGRAM(15), SACCOUNT
MVC   DIAGRAM+16(10), RESULT10
MVC   PRINT, BLANK
MVC   PRINT+1(23), TOTMEGS
MVC   KBYTES(4), TOTALLOC
BAL   R8, CONVMEGS          CONVERT KILOBYTES TO MEGABYTES
MVC   PRINT+23(11), MEGS1
PUT   REPORDCB, PRINT
MVC   DIAGRAM+27(11), MEGS1
MVC   PCFILREC(80), BLANK
MVI   PCFILREC, X'7F'
MVC   PCFILREC+1(15), PREVGROU
MVI   PCFILREC+16, X'7F'
MVI   PCFILREC+17, C', '
MVI   PCFILREC+18, X'7F'
MVC   PCFILREC+19(11), MEGS1
MVI   PCFILREC+30, X'7F'
PUT   PCFILE, PCFILREC
CLOSE (PCFILE)
MVC   BLOCK1(9), YESTTEXT
MVC   BLOCK2(9), ALLOCATE
MVC   BLOCK3(9), BLANK
MVC   BLOCK3+2(4), DATASETS
MVC   BLOCK4(9), BLANK
MVC   BLOCK4+2(4), SETS
BAL   R8, BLOCKLET          WRITE TRAILER
MVC   TEXTVAR(28), BLANK
MVC   TEXTVAR(9), YESTTEXT
MVC   TEXTVAR+10(9), ALLOCATE
MVC   NEWTEX1A(38), NEWPAT2A
MVC   NEWPAT2B(38), BLANK
MVC   NEWPAT2B(8), SETNAME
MVC   NEWTEX1B(38), NEWPAT2B
MVC   NEWTEX1C(38), NEWPAT2C
MVC   NEWTEX1D(16), NEWPAT2D
MVC   NEWTEX1Z(38), NEWPAT2Z
MVC   NEWTEX1E(38), BLANK
MVC   NEWTEX1F(38), NEWPAT2F
MVC   NEWTEX1G(17), NEWPAT2G
MVC   NEWTEX1H(38), BLANK
MVC   NEWTEX1I(38), BLANK

```

```

MVI PRINT,C' '
MVC PRINT+1(L'PRINT-1),PRINT
MVC PRINT+20(9),YESTTEXT
MVC PRINT+30(9),ALLOCATE
MVC PRINT+40(4),DATASETS
MVC PRINT+45(4),SETS
CALL CONVFICH,(PAGENUM,RESULT),VL
MVC PRINT+54(4),RESULT+4
PUT TABCODCB,PRINT
XR R5,R5
LA R5,50 EST. LINES PER PAGE NUMBER
XR R4,R4
LA R4,1 EST. LINES PER PAGE COUNTER
BAL R8,NEWPAGE
MVC PREVGROU(15),BLANK
XR R1,R1
ST R1,TOTALLOC CLEAN THE COUNTER
ST R1,INDPCFIL STORE PCFIL ON/OFF INDICATOR
OPEN (YESTERDA,(INPUT))
GETYESTE EQU *
GET YESTERDA,STATSREC
BAL R8,PROCRCO PROCESS THE RECORD
B GETYESTE
ENDYESTE EQU *
CLOSE (YESTERDA)
BAL R8,LASTTOTA CALCULATE TOTAL FOR THE LAST GROUP
MVC BLOCK1(9),BLANK
MVC BLOCK1+3(3),NOTTEXT
MVC BLOCK2(9),CATALOG
MVC BLOCK3(9),BLANK
MVC BLOCK3+2(4),DATASETS
MVC BLOCK4(9),BLANK
MVC BLOCK4+2(4),SETS
BAL R8,BLOCKLET WRITE TRAILER
MVC TEXTVAR(28),BLANK
MVC TEXTVAR(3),NOTTEXT
MVC TEXTVAR+4(9),CATALOG
MVI PRINT,C' '
MVC PRINT+1(L'PRINT-1),PRINT
MVC PRINT+20(23),=C'NOT CATALOGED DATASETS'
CALL CONVFICH,(PAGENUM,RESULT),VL
MVC PRINT+54(4),RESULT+4
PUT TABCODCB,PRINT
XR R5,R5
LA R5,50 EST. LINES PER PAGE NUMBER
XR R4,R4
LA R4,1 EST. LINES PER PAGE COUNTER
BAL R8,NEWPAGE
MVC PREVGROU(15),BLANK
XR R1,R1
ST R1,TOTALLOC CLEAN THE COUNTER
ST R1,INDPCFIL STORE PCFIL ON/OFF INDICATOR

```

```

OPEN (NOTCATAL,(INPUT))
GETNOTCA EQU *
GET NOTCATAL,STATSREC
BAL R8,PROCRECO PROCESS THE RECORD
B GETNOTCA
ENDNOTCA EQU *
CLOSE (NOTCATAL)
BAL R8,LASTTOTA CALCULATE TOTAL FOR THE LAST GROUP
MVC BLOCK1(9),=C'T O T A L'
MVC BLOCK2(9),BLANK
MVC BLOCK2+1(7),STORAGE
MVC BLOCK3(9),BLANK
MVC BLOCK3(9),ALLOCATE
MVC BLOCK4(9),BLANK
MVC BLOCK4(2),BY
MVC BLOCK4+3(5),GROUPS
BAL R8,BLOCKLET WRITE BLOCK PAGE
MVC NEWTEX1A(38),BLANK
MVC NEWTEX1A(15),=C'..GROUP OWNER..'
MVC NEWTEX1A+21(9),=C'MEGABYTES'
MVC NEWTEX1B(38),BLANK
MVC NEWTEX1C(38),BLANK
MVC NEWTEX1D(16),BLANK
MVC NEWTEX1Z(38),BLANK
MVC NEWTEX1E(38),BLANK
MVC NEWTEX1F(38),BLANK
MVC NEWTEX1G(17),BLANK
MVC NEWTEX1H(38),BLANK
MVC NEWTEX1I(38),BLANK
MVC TEXTVAR(28),BLANK
MVC TEXTVAR(5),TOTAL
MVC TEXTVAR+6(2),BY
MVC TEXTVAR+9(6),GROUPS
MVI PRINT,C' '
MVC PRINT+1(L'PRINT-1),PRINT
MVC PRINT+20(5),TOTAL
MVC PRINT+26(7),STORAGE
MVC PRINT+34(9),ALLOCATE
MVC PRINT+44(2),BY
MVC PRINT+47(5),GROUPS
CALL CONVFICH,(PAGENUM,RESULT),VL
MVC PRINT+54(4),RESULT+4
PUT TABCODCB,PRINT
XR R5,R5
LA R5,50 EST. LINES PER PAGE NUMBER
XR R4,R4
LA R4,1 EST. LINES PER PAGE COUNTER
BAL R8,NEWPAGE
OPEN (PCFILE,(INPUT))
GETLOTKI EQU *
GET PCFILE,DIAGRAM
MVC PRINT,BLANK

```

```

MVC PRINT+1(80),DIAGRAM
PUT REPORDCB,PRINT
B GETLOTKI
ENDPCFIL EQU *
CLOSE (PCFILE)
CLOSE (TABCODECB)
OPEN (TABCODECB,(INPUT))
MVC PRINT,BLANK
MVC PRINT(20),=C'1 TABLE OF CONTENTS'
PUT REPORDCB,PRINT
GETTABCO EQU *
GET TABCODECB,PRINT
PUT REPORDCB,PRINT
B GETTABCO
ENDTABCO EQU *
CLOSE (TABCODECB)
MVC BLOCK1(9),TEXTSPAC
MVC BLOCK2(9),BLANK
MVC BLOCK2+1(7),STORAGE
MVC BLOCK3(9),MANAGER
MVC BLOCK4(9),=C' END '
BAL R8,BLOCKLET WRITE TRAILER
CLOSE (REPORDCB)
MVC PRINT,BLANK
MVC PRINT+1(3),=C'END'
PUT PRINTDCB,PRINT
FREEM EQU *
L R9,R9SAVE
FREEMAIN R,LV=4000,A=(R9)
CLOSE (PRINTDCB)
FINI EQU *
L R13,4(R13)
LR R15,R7
RETURN (14,12),RC=(15)
LASTTOTA EQU *
ST R8,R8SAVE
MVC PRINT,BLANK
MVC PRINT+1(23),TOTKB
CALL CONVFICL,(TOTALLOC,RESULT10),VL
MVC PRINT+24(10),RESULT10
PUT REPORDCB,PRINT
MVC DIAGRAM(80),BLANK
MVC DIAGRAM(15),SACCOUNT
MVC DIAGRAM+16(10),RESULT10
MVC PRINT,BLANK
MVC PRINT+1(23),TOTMEGS
MVC KBYTES(4),TOTALLOC
BAL R8,CONVMEGS CONVERT KILOBYTES TO MEGABYTES
MVC PRINT+23(11),MEGS1
PUT REPORDCB,PRINT
L R8,R8SAVE
BR R8

```

```

COMPARER EQU *
MVC NUMBER(4),SPACALLO
CALL CONVFI CL, (NUMBER,RESULT10),VL
MVC SALPRINT(10),RESULT10
MVC NUMBER(4),SPACUSED
CALL CONVFI CL, (NUMBER,RESULT10),VL
MVC SUSPRINT(10),RESULT10
MVI TABSTATS,C' '
MVC TABSTATS+1(L'TABSTATS-1),TABSTATS
LA R7,44
LA R2,SDSN
LA R3,TABSTATS
LR R4,R3
LOOPCOM1 EQU *
MVC 0(1,R4),0(R2)
CLC 0(1,R2),=C'.'
BE DOTFOUN2
CLC 0(1,R2),=C' '
BE BLANKF02
B PARMINC2
DOTFOUN2 EQU *
MVI 0(R4),C' ' REPLACE DOT BY BLANK
XR R1,R1
LA R1,9
AR R3,R1
LR R4,R3
B PARMINC3
BLANKF02 EQU *
B EOFPADS1
PARMINC2 EQU *
XR R1,R1
LA R1,1
AR R4,R1
PARMINC3 EQU *
XR R1,R1
LA R1,1
AR R2,R1
BCT R7,LOOPCOM1
EOFPADS1 EQU *
LA R4,TABSTATS
LR R6,R4
OPEN (MODDIDCB,(INPUT))
LOOPMODI EQU *
GET MODDIDCB,TABDIAGR
LA R3,TABDIAGR
XR R7,R7
LA R7,5 EST. NUMBER OF 8 BYTE FIELDS
LOOPFIVE EQU *
LR R5,R3 (R5) = ADDRESS OF THE TABDIAGR 8 BYTES
LR R6,R4 (R6) = ADDRESS OF THE TABSTATS 8 BYTES
XR R2,R2
LA R2,8 EST. 8 BYTE FIELD COUNTER

```

```

LOOPEIGH EQU *
          CLI  Ø(R5),C'*'
          BNE  COMPARE
          CLI  1(R5),C' '   TEST IF BLANK IS AFTER THE *?
          BE   NOCOMEIG     DO NOT COMPARE THE REST OF THE FIELD
          CLI  1(R5),C'*'   TEST IF * IS AFTER THE *?
          BE   EQUALFOU     DO NOT COMPARE THE REST OF THE DATA
COMPARE  EQU *
          CLI  Ø(R5),C'?'
          BE   NOCOMPAR
          CLC  Ø(1,R5),Ø(R6) COMPARE ONE BYTE
          BE   NOCOMPAR
          LA  R4,TABSTATS   NOT EQUAL
          B   LOOPMODI
NOCOMPAR EQU *
          A   R5,=F'1'      INCREASE TABDIAGR COUNTER
          A   R6,=F'1'      INCREASE TABSTATS COUNTER
          BCT R2,LOOPEIGH
NOCOMEIG EQU *
          A   R3,=F'9'      INCREASE TABDIAGR COUNTER
          A   R4,=F'9'      INCREASE TABSTATS COUNTER
          CLI  Ø(R3),C' '   TEST IF FIRST CHAR OF DIAGRAM IS BLANK?
          BNE  GOTOBCT
EQUALFOU EQU *
          MVC  SACCOUNT(15),TABACCOU
          B   MODICLOS
GOTOBCT  EQU *
          BCT  R7,LOOPFIVE
          B   LOOPMODI
ENDMODDI EQU *
          MVC  SACCOUNT(15),=C'— UNKNOWN —'
MODICLOS EQU *
          CLOSE (MODDIDCB)
          BR   R8
NEWPAGE  EQU *
          MVI  PRINT,C' '
          MVC  PRINT+1(L'PRINT-1),PRINT
          MVI  PRINT,C'1'
          PUT  REPORDCB,PRINT
          MVI  PRINT,C' '
          MVC  PRINT+1(L'PRINT-1),PRINT
          MVC  PRINT+1(9),TEXTSPAC
          MVC  PRINT+11(28),TEXTVAR
          MVC  PRINT+41(15),STATDATE
          MVC  PRINT+57(8),STATTIME
          MVC  PRINT+7Ø(4),=C'PAGE'
          CALL CONVFIH,(PAGENUM,RESULT),VL
          MVC  PRINT+75(4),RESULT+4
          L   R1,PAGENUM    (R1) = CURRENT PAGE NUMBER
          A   R1,=F'1'
          ST  R1,PAGENUM
          MVC  PRINT+83(48),YOURCOMP

```

```

PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVI    PRINT,C' '
PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+2(38),NEWTEX1A
MVC    PRINT+40(38),NEWTEX1B
MVC    PRINT+78(38),NEWTEX1C
MVC    PRINT+116(16),NEWTEX1D      '
PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+2(38),NEWTEX1Z
MVC    PRINT+40(38),NEWTEX1E
MVC    PRINT+78(38),NEWTEX1F
MVC    PRINT+116(17),NEWTEX1G    '
PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+40(38),NEWTEX1H
MVC    PRINT+78(38),NEWTEX1I
PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVI    PRINT,C' '
PUT    REPORDCB,PRINT
BR     R8
BLOCKLET EQU *
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVI    PRINT,C'1'
PUT    REPORDCB,PRINT
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
PUT    REPORDCB,PRINT
LA     R5,1
MVC    CHARFIEL(9),BLOCK1
NEWLOOPS EQU *
LA     R7,12
LA     R6,1
ST     R6,LINECOUN
LOOPIEFS EQU *
MVI    CONSAREA,C' '
MVC    CONSAREA+1(L'CONSAREA-1),CONSAREA
LA     R1,PARMBLOC
CALL   IEFSD095
MVI    PRINT,C' '
MVC    PRINT+1(L'PRINT-1),PRINT
MVC    PRINT+1(132),CONSAREA
PUT    REPORDCB,PRINT

```

```

LA      R1,1
AR      R6,R1
ST      R6,LINECOUN
BCT     R7,LOOPIEFS
LA      R1,1
AR      R5,R1
LA      R1,5
CR      R5,R1
BE      ENDBLOCK
LA      R1,2
CR      R5,R1
BE      SECONLIN
LA      R1,3
CR      R5,R1
BE      THIRDLIN
LA      R1,4
CR      R5,R1
BE      FOURTLIN
SECONLIN EQU *
MVI     PRINT,C' '
MVC     PRINT+1(L'PRINT-1),PRINT
MVI     PRINT,C'Ø'
PUT     REPORDCB,PRINT
MVC     CHARFIEL(9),BLOCK2
B       NEWLOOPS
THIRDLIN EQU *
MVI     PRINT,C' '
MVC     PRINT+1(L'PRINT-1),PRINT
MVI     PRINT,C'Ø'
PUT     REPORDCB,PRINT
MVC     CHARFIEL(9),BLOCK3
B       NEWLOOPS
FOURTLIN EQU *
MVI     PRINT,C' '
MVC     PRINT+1(L'PRINT-1),PRINT
MVI     PRINT,C'Ø'
PUT     REPORDCB,PRINT
MVC     CHARFIEL(9),BLOCK4
B       NEWLOOPS
ENDBLOCK EQU *
MVI     PRINT,C' '
MVC     PRINT+1(L'PRINT-1),PRINT
MVI     PRINT,C'Ø'
CLI     IYOURNAM,C'1'      TEST IF PRINT OF YOURNAME REQUIRED
BNE     NOYOURNA
MVC     PRINT+1(14),YOURNAME
MVC     PRINT+2Ø(31),PERMCOMP
B       PRINNEXT
NOYOURNA EQU *
MVC     PRINT+1(48),YOURCOMP
PRINNEXT EQU *
MVC     PRINT+6Ø(15),STATDATE

```



```

MVC PRINT+76(8),STATIME
MVC PRINT+92(13),=C'1MB = 1024 KB'
MVC PRINT+112(18),=C'MVS TOOLBOX: SPACE'
PUT  REPORDCB,PRINT
MVI  IYOURNAM,C'Ø'          PRINT NOT REQUIRED
BR   R8
CALCTRKS EQU *
*   CONVERT KBYTES TO TRACKS
    L   R7,NUMBER          (R7) = KILOBYTES
    SR  R1,R1
    CR  R7,R1              TEST IF ZERO
    BE  ENDKTCON
    SR  R6,R6              (R6) = Ø
    SR  R1,R1
    LA  R1,128
    MR  R6,R1
    SR  R1,R1
    L   R1,OCTBYTTR        (R1) = 7083 (56664/8)
    DR  R6,R1
    SR  R1,R1              (R1) = Ø
    CR  R6,R1              REMAINS = Ø?
    BE  ENDKTCON          YES
    SR  R1,R1              (R1) = Ø
    LA  R1,1
    AR  R7,R1              INCREASE NUMBER OF TRACKS
ENDKTCON EQU *
    ST  R7,NUMBER
    CALL CONVFIH,(NUMBER,RESULT),VL
    MVC TRACKS(8),RESULT
    BR  R8
CONVMEGS EQU *
*   CONVERT KBYTES TO MEGABYTES
    MVC MEGS1(11),BLANK
    L   R7,KBYTES          (R7) = KILOBYTES
    SR  R1,R1
    CR  R7,R1              TEST IF ZERO
    BE  ENDCMEGS
    SR  R6,R6              (R6) = Ø
    SR  R1,R1
    L   R1,KILO
    DR  R6,R1
    ST  R7,NUMBER
    CALL CONVFIH,(NUMBER,RESULT10),VL
    MVC MEGS1(8),RESULT10+2
    LR  R7,R6              (R7) = REMAINS
    SR  R6,R6              (R6) = Ø
    SR  R1,R1
    LA  R1,100
    MR  R6,R1
    L   R1,KILO
    DR  R6,R1
    ST  R7,NUMBER

```

```

CALL CONVFI CL, (NUMBER, RESULT10), VL
MVI MEGSDOT, C' .'
MVC MEGS2(2), RESULT10+8
CLI MEGS2, C' ' TEST IF BLANK
BNE ENDCMEGS
MVI MEGS2, C'0' FILL WITH BLANK
ENDCMEGS EQU *
BR R8
CONVDATE EQU *
TIME BIN
LR R7, R1
ST R7, PACKDAT
SLL R7, 16
SRL R7, 16
XR R1, R1
ST R1, DWORD
ST R7, DWORD+4
CVB R7, DWORD (R7) = DAY NUMBER
S R7, =F'1' FIND THE PREVIOUS DAY NUMBER
XR R1, R1
CR R7, R1 TEST IF ZERO
BNZ NZERODAY
MVI PRINT, C' '
MVC PRINT+1(L'PRINT-1), PRINT
MVC PRINT+1(31), =C'DAY=000, DECREASE THE YEAR VALUE'
PUT PRINTDCB, PRINT
NZERODAY EQU *
UNPK DWORD(8), PACKDAT
CLI DWORD+2, C'0'
BE TWENTYCE
MVC DWORD+1(2), =C'20' 21ST CENTURY
B OKCENTUR
TWENTYCE EQU *
MVC DWORD+1(2), =C'19' 21ST CENTURY
OKCENTUR EQU *
ST R7, NUMBER
CALL CONVFI CL, (NUMBER, RESULT10), VL
MVC DWORD+5(3), RESULT10+7
CLI DWORD+5, C' '
BNE TSECBLAN
MVI DWORD+5, C'0'
TSECBLAN EQU *
CLI DWORD+6, C' '
BNE TTHIBLAN
MVI DWORD+6, C'0'
TTHIBLAN EQU *
CLI DWORD+7, C' '
BNE OKBLANKS
MVI DWORD+7, C'0'
OKBLANKS EQU *
BR R8
PROCRECO EQU *

```

```

ST      R8,R8SAVE
MVC     ACCORECO(130),BLANK
MVC     ADSN(44),SDSN
MVC     AVOLSER(6),SVOLSER
MVC     AKBTALLO(10),SALPRINT
MVC     AKBTUSED(10),SUSPRINT
MVC     APERUSED(4),SPERUSED
MVC     APERUNUS(4),SPERUNUS
MVC     AACCOUNT(15),SACCOUNT
MVC     ADSORG(2),SDSORG
MVC     ADATLREF(4),SDALREF
MVC     ADATLREF+5(3),SDALREF+4
MVC     ADATCREA(4),SDACREAT
MVC     ADATCREA+5(3),SDACREAT+4
MVC     NUMBER(4),SPACALLO
BAL     R8,CALCTRKS           CONVERT KBYTES TO TRKS
MVC     ATRKALLO(8),TRACKS
MVC     NUMBER(4),SPACUSED
BAL     R8,CALCTRKS           CONVERT KBYTES TO TRKS
MVC     ATRKUSED(8),TRACKS
CLC     PREVGROU(15),SACCOUNT
BE      SAMEGROU             THE SAME GROUP
CLC     PREVGROU(15),BLANK
BE      SAMEGROU             BLANK ACCOUNT
*      NEW ACCOUNT GROUP
MVI     PRINT,C' '
MVC     PRINT+1(L'PRINT-1),PRINT
MVC     PRINT+24(15),SACCOUNT
CALL    CONVFICH,(PAGENUM,RESULT),VL
MVC     PRINT+54(4),RESULT+4
PUT     TABCODCB,PRINT
MVC     PRINT,BLANK
MVC     PRINT+1(23),TOTKB
CALL    CONVFICL,(TOTALLOC,RESULT10),VL
MVC     PRINT+24(10),RESULT10
PUT     REPORDCB,PRINT
MVC     DIAGRAM(80),BLANK
MVC     DIAGRAM(15),PREVGROU
MVC     DIAGRAM+16(10),RESULT10
MVC     PRINT,BLANK
MVC     PRINT+1(23),TOTMEGS
MVC     KBYTES(4),TOTALLOC
BAL     R8,CONVMEGS           CONVERT KILOBYTES TO MEGABYTES
MVC     PRINT+23(11),MEGS1
PUT     REPORDCB,PRINT
L       R7,INDPCFIL           (R7) = PCFIL INDICATOR
XR      R1,R1
CR      R7,R1
BE      NOTPCFIL
MVC     DIAGRAM+27(11),MEGS1
MVC     PCFILREC(80),BLANK
MVI     PCFILREC,X'7F'

```

```

MVC PCFILREC+1(15),PREVGROU
MVI PCFILREC+16,X'7F'
MVI PCFILREC+17,C', '
MVI PCFILREC+18,X'7F'
MVC PCFILREC+19(11),MEGS1
MVI PCFILREC+30,X'7F'
PUT PCFILE,PCFILREC
NOTPCFIL EQU *
XR R1,R1
ST R1,TOTALLOC CLEAN THE COUNTER
XR R4,R4
BAL R8,NEWPAGE
SAMEGROU EQU *
L R1,TOTALLOC (R1) = TOTAL ALLOCATED
L R2,SPACALLO (R2) = CURRENT ALLOCATED
AR R1,R2 INCREASE COUNTER
ST R1,TOTALLOC STORE TOTAL ALLOCATED
MVC PREVGROU(15),SACCOUNT
MVC PRINT,BLANK
MVC PRINT+1(132),ACCORECO
PUT REPORDCB,PRINT
CR R4,R5 TEST IF PAGE IS FULL
BNE PAGENFU2 NOT
XR R4,R4
BAL R8,NEWPAGE
PAGENFU2 EQU *
A R4,=F'1'
L R8,R8SAVE
BR R8
*-----
CONTOHEX EQU *
XC CHARACTE,CHARACTE
MVC CHARACTE(4),NUMBER
UNPK CHARACTE,CHARACTE(5)
TR CHARACTE+7(8),HEXTABLE
MVC RESULT10+2(8),CHARACTE+7
BR R8
HEXTABLE DC 240CL1'?',C'0123456789ABCDEF'
CHARACTE DS CL16
SAVE3 DS 18F
NUMBER DS F
R8SAVE DS F
R9SAVE DS F
TOTALLOC DS F TOTAL SPACE ALLOCATED IN KB
PAGENUM DS F CURRENT PAGE NUMBER
PARMADDR DS F ADDRESS OF THE PARM LIST FROM THE EXEC STMT
OCTBYTTR DC F'7083' = 56664/8 (56664 = BYTES PER TRACK)
KILO DC F'1024'
PACKDAT DS F
RESULT DS CL8
STATTIME DS CL8 STATISTICS TAKEN ON TIME
RESULT10 DS CL10

```

STORAGE	DC	CL9'STORAGE'	
MANAGER	DC	CL9' MANAGER '	
MONITOR	DC	CL9'DASD TOOL'	
SETS	DC	CL4'SETS'	
TOTMEGS	DC	CL23'TOTAL IN MEGABYTES: ...'	
TOTKB	DC	CL23'TOTAL:'	
TEXTSPAC	DC	CL9'SPACECHEK'	
BLANK	DC	CL133' '	
STATDATE	DS	CL15	STATISTICS TAKEN ON DATE
YOURNAME	DC	CL14'YOUR NAME.....'	
YOURCOMP	DC	CL48'THE JOHANNESBURG STOCK EXCHANGE'	
PERMCOMP	DC	CL31'YOUR PERMANENT TEXT TO DISPLAY '	
DATASETS	DC	CL8'DATASETS'	
BY	DC	CL2'BY'	
VOLSERS	DC	CL7'VOLSERS'	
REPORT	DC	CL6'REPORT'	
DISPLAY	DC	CL7'DISPLAY'	
ACCOUNTS	DC	CL8'ACCOUNTS'	
TOTAL	DC	CL5'TOTAL'	
ALLOCATE	DC	CL9'ALLOCATED'	
CATALOGE	DC	CL9'CATALOGED'	
NOTTEXT	DC	CL3'NOT'	
YESTTEXT	DC	CL9'YESTERDAY'	
GROUPS	DC	CL6'GROUPS'	
PRINT	DS	CL133	
DATE	DS	ØCL12	
DATENAME	DS	CL3	
DATEDAY	DS	CL2	
DATEMONT	DS	CL3	
DATEYEAR	DS	CL4	
TIMESTAM	DS	ØCL11	
HH	DS	CL2,CL1	BLANK
MM	DS	CL2,CL1	BLANK
SS	DS	CL2,CL1	BLANK
DD	DS	CL2	BLANK
PARMBLOC	DS	ØD	
WORDB1	DC	A(CHARFIEL)	ADDRESS OF THE FIELD CONT. CHARS STRING
WORDB2	DC	A(LINECOUN)	ADDRESS OF LINE COUNT FIELD
WORDB3	DC	A(CONSAREA)	ADDRESS OF A CONSTRUCION AREA
WORDB4	DC	A(NUMCHARA)	ADDRESS OF THE NUMBER OF CHARACTERS
CHARFIEL	DS	CL9	
LINECOUN	DC	F'1'	FOR THE FIRST ENTRY TO IEFSDØ95
CONSAREA	DC	CL132' '	CONSTRUCTION AREA
NUMCHARA	DC	F'9'	NUMBER OF CHARACTERS IN THE STRING
	DS	ØD	
PARMSORT	DC	X'8Ø',AL3(ADLST)	
	CNOP	2,4	
ADLST	DC	AL2(LISTEND-LISTBEG)	
LISTBEG	DC	A(SORTA)	STARTING ADDRESS OF SORT STMT
	DC	A(SORTZ)	ENDING ADDRESS OF SORT STMT
	DC	A(RECA)	STARTING ADDRESS OF RECORD STMT
	DC	A(RECB)	ENDING ADDRESS OF RECORD STMT

NEWPAT2Z	DC	CL38'	ALLOCATED	'
NEWPAT2F	DC	CL38'	ALLOCATED USED	USED SABLE '
NEWPAT2G	DC	CL17'	LAST REF CREATED	'
	DS	ØD		
IDCINDCB	DCB	MACRF=PM, RECFM=FB, DSORG=PS, LRECL=8Ø, BLKSIZE=312Ø,		*
		DDNAME=INIDCAMS		
IDCOUDCB	DCB	MACRF=GM, RECFM=VB, DSORG=PS, LRECL=644,		*
		DDNAME=OUIDCAMS, EODAD=ENDOUTPU		
INDCB	DCB	MACRF=GM, DSORG=PS, RECFM=VB, LRECL=644,		*
		DDNAME=INPUTDDD, EODAD=ENDATA		
MODDIDCB	DCB	MACRF=(GM, PM), DSORG=PS, RECFM=FB, LRECL=24Ø,		*
		DDNAME=MODDIAGR, EODAD=ENDMODDI		
NOTCATAL	DCB	MACRF=(GM, PM), RECFM=FB, LRECL=15Ø, DSORG=PS,		*
		DDNAME=NOTCATAL, EODAD=ENDNOTCA		
PCFILE	DCB	MACRF=(GM, PM), RECFM=FB, DSORG=PS, LRECL=8Ø,		*
		DDNAME=PCFILE, EODAD=ENDPCFIL		
PRINTDCB	DCB	MACRF=PT, RECFM=FBA, LRECL=133, BLKSIZE=133, DSORG=PS,		*
		DDNAME=PRINTOUT		
REPORDCB	DCB	MACRF=PT, RECFM=FBA, LRECL=133, BLKSIZE=133, DSORG=PS,		*
		DDNAME=REPORTDD		
SORTODCB	DCB	MACRF=GM, DSORG=PS, RECFM=FB, LRECL=15Ø,		*
		DDNAME=SORTOUT, EODAD=ENDSOROU		
STATSDCB	DCB	MACRF=PM, DSORG=PS, RECFM=FB, LRECL=15Ø,		*
		DDNAME=SORTIN		
TABCODECB	DCB	MACRF=(GM, PM), RECFM=FBA, LRECL=133, DSORG=PS,		*
		DDNAME=WORKFILA, EODAD=ENDTABCO		
YESTERDA	DCB	MACRF=(GM, PM), RECFM=FB, LRECL=15Ø, DSORG=PS,		*
		DDNAME=YESTERDA, EODAD=ENDYESTE		
	LTORG			
DUVBSNEW	DSECT		MAPPING MACRO	
	DS	ØD		
DIAGRAM	DS	ØCL8Ø		
PATTDSN	DS	CL44, CL1	PATTERN OR DSNAME	
PATTACCO	DS	CL15, CL2Ø	ACCOUNT NAME	
PCFILREC	DS	ØCL8Ø		
	DS	CL1	DOUBLE APOSTROPHE AREA	
	DS	CL79	ACCOUNT NAME, DOUBLE APOSTR + DATA	
IDCAMREC	DS	CL644	IDCAMS CONTROL CARD	
WORKAREA	DS	CL1ØØ	WORK AREA	
	DS	ØD		
OUTRECOR	DS	ØCL13Ø		
DSN	DS	CL44, CL1	DATASET NAME	
VOLSER	DS	CL6, CL3	VOLSER	
DSORG	DS	CL2, CL1	DATASET ORGANIZATION	
DLASTREF	DS	CL8, CL1	DATE	
NEXTENTS	DS	CL3, CL1	NUMBER OF EXTENTS USED	
KBYTALLO	DS	CL1Ø, CL1	NUMBER OF KBYTES ALLOCATED	
KBYTUSED	DS	CL1Ø, CL1	NUMBER OF KBYTES USED	
PERCALL1	DS	CL3	PERCENTAGE: (SPACEUNUSED/SPALLOCATED)*1ØØ%	
PERCADOT	DS	CL1	DECIMAL DOT	
PERCALL2	DS	CL2, CL1	PERCENTAGE	
KBUNUSAB	DS	CL1Ø, CL1	NUMBER OF KBYTES UNUSABLE IN BLOCKS	

PERCUNU1	DS	CL3	PERCENTAGE: (SPACEUNUSABLE/SPALLOCATED)*100%
PERCUDOT	DS	CL1	DECIMAL DOT
PERCUNU2	DS	CL2,CL5	PERCENTAGE
DCREATED	DS	CL8,CL10	CREATION DATE
TABDIAGR	DS	ØCL24Ø	
TABEIGØ1	DS	CL8,CL1	1ST
TABEIGØ2	DS	CL8,CL1	2
TABEIGØ3	DS	CL8,CL1	3
TABEIGØ4	DS	CL8,CL1	4
TABEIGØ5	DS	CL8,CL1	5
TABEIGØ6	DS	CL8,CL1	6
TABEIGØ7	DS	CL8,CL1	7
TABEIGØ8	DS	CL8,CL1	8
TABEIGØ9	DS	CL8,CL1	9
TABEIG1Ø	DS	CL8,CL1	1Ø
TABEIG11	DS	CL8,CL1	11
TABEIG12	DS	CL8,CL1	12
TABEIG13	DS	CL8,CL1	13
TABEIG14	DS	CL8,CL1	14
TABEIG15	DS	CL8,CL1	15
TABEIG16	DS	CL8,CL1	16
TABEIG17	DS	CL8,CL1	17
TABEIG18	DS	CL8,CL1	18
TABEIG19	DS	CL8,CL1	19
TABEIG2Ø	DS	CL8,CL1	2Ø
TABEIG21	DS	CL8,CL1	21
TABEIG22	DS	CL8,CL1	22
TABACCOU	DS	CL15,CL35	ACCOUNT NAME
TABSTATS	DS	ØCL24Ø	
TABSTAØ1	DS	CL8,CL1	1ST
TABSTAØ2	DS	CL8,CL1	2
TABSTAØ3	DS	CL8,CL1	3
TABSTAØ4	DS	CL8,CL1	4
TABSTAØ5	DS	CL8,CL1	5
TABSTAØ6	DS	CL8,CL1	6
TABSTAØ7	DS	CL8,CL1	7
TABSTAØ8	DS	CL8,CL1	8
TABSTAØ9	DS	CL8,CL1	9
TABSTA1Ø	DS	CL8,CL1	1Ø
TABSTA11	DS	CL8,CL1	11
TABSTA12	DS	CL8,CL1	12
TABSTA13	DS	CL8,CL1	13
TABSTA14	DS	CL8,CL1	14
TABSTA15	DS	CL8,CL1	15
TABSTA16	DS	CL8,CL1	16
TABSTA17	DS	CL8,CL1	17
TABSTA18	DS	CL8,CL1	18
TABSTA19	DS	CL8,CL1	19
TABSTA2Ø	DS	CL8,CL1	2Ø
TABSTA21	DS	CL8,CL1	21
TABSTA22	DS	CL8,CL1	22
TSTATACC	DS	CL15,CL35	ACCOUNT NAME

	DS	ØD	
ACCORECO	DS	ØCL132	
AACCOUNT	DS	CL15,CL1	ACCOUNT (GROUP OWNER)
AVOLSER	DS	CL6,CL1	VOLUME SERIAL
AKBTALLO	DS	CL1Ø,CL1	NUMBER OF KBYTES ALLOCATED
ADSN	DS	CL44,CL1	DATASET NAME
ADSORG	DS	CL4,CL1	DSORG
ATRKALLO	DS	CL8,CL1	NUMBER OF TRACKS ALLOCATED
ATRKUSED	DS	CL8,CL1	NUMBER OF TRACKS USED
APERUSED	DS	CL4,CL1	PERCENTAGE OF SPACE USED
APERUNUS	DS	CL4,CL4	PERCENTAGE OF SPACE UNUSABLE IN BLOCKS
ADATLREF	DS	CL8,CL1	DATE LAST REFERENCED
ADATCREA	DS	CL8,CL1	DATE CREATED
AKBTUSED	DS	CL1Ø,CL1Ø	NUMBER OF KBYTES USED
NEWTEX1A	DS	CL38	
NEWTEX1B	DS	CL38	
NEWTEX1C	DS	CL38	
NEWTEX1D	DS	CL16	
NEWTEX1Z	DS	CL38	
NEWTEX1E	DS	CL38	
NEWTEX1F	DS	CL38	
NEWTEX1G	DS	CL17	
NEWTEX1H	DS	CL38	
NEWTEX1I	DS	CL38	
DWORD	DS	D	
R2SAVE	DS	F	
R4SAVE	DS	F	
R5SAVE	DS	F	
KBYTES	DS	F	KBYTES WORK AREA
INDPCFIL	DS	F	PCFIL PUT ON/OFF INDICATOR
TRACKS	DS	CL8	TRACKS
BLOCK1	DS	CL9	1ST TEXT BLOCK
BLOCK2	DS	CL9	2ND TEXT BLOCK
BLOCK3	DS	CL9	3RD TEXT BLOCK
BLOCK4	DS	CL9	4TH TEXT BLOCK
RDATA	DS	CL644	
PREVGROU	DS	CL15	PREVIOUS ACCOUNT GROUP
PREVVOLU	DS	CL6	PREVIOUS VOLSER
MEGS1	DS	CL8	MEGABYTES
MEGSDOT	DS	CL1	DECIMAL DOT
MEGS2	DS	CL2	MEGABYTES DECIMAL PART
TEXTVAR	DS	CL28	
NEWPAT2B	DS	CL38	
IYOURNAM	DS	CL1	PRINT STEVEK SWITCH
	DS	ØD	
		IDCDOUT	
DSECTIEF	DSECT		
		END	

Writing a user SMF record

INTRODUCTION

It is often necessary to write user SMF records for different evaluations, for example which CLIST is called when and by whom. The program WRTUSMF can be used to write user SMF records uniformly. It can be called from different environments. You can pass a string as an argument (with a maximum of 100 characters), which is placed in the SMF record. In addition to the default part consisting of:

- RDW
- Flag byte
- Record type
- Time
- Date
- SMF id.

The SMF record contains the user part consisting of:

- Job name
- Step name
- JES job-id (JOBnnnnn,STCnnnnn,TSUnnnnn)
- RACF user
- Length of data
- Data (string passed as argument).

The record type (128 to 255) is defined in the program (see label USMFREC#) and can be altered by re-assembling the program.

For SMFEWTM as well as SMFWTM (SVC 83), normally APF authorization is necessary. At the shop, where WRTUSMF was written, APF authorization was not mandatory because of a usermod (zap of SVC table). The program determines itself whether it must be APF authorized (SVC 83 not modified) and in this case whether APF

authorization exists. If APF authorization is necessary for calling it as TSO command, then WRTUSMF must be defined in the SYS1.PARMLIB(IKJTSOxx) as an authorized TSO command:

```
AUTHCMD NAMES (... WRTUSMF...)
```

The program can be called in the following ways:

- Batch program in JCL EXEC card:

```
// EXEC PGM=WRTUSMF, PARM = 'ABC'
```

- Subroutine of a program (eg Assembler language):

```
LINK EP=WRTUSMF, PARAM=(DATA)
DATA DC H'3', C'ABC '
```

The program must be APF authorized if necessary.

- TSO command, directly called or in a CLIST:

```
WRTUSMF ABC
```

- TSO command in a REXX EXEC running in TSO environment:

```
ADDRESS TSO "WRTUSMF ABC"
```

- In a REXX EXEC in a batch environment (// EXEC PGM=IRXJCL, PARM=...):

```
ADDRESS LINK "WRTUSMF ABC" (as well as ADDRESS ATTACH "WRTUSMF ABC")
```

This is possible only if APF authorization is not necessary!

In case of error, the program outputs a proper WTO message and terminates with return code 8 (otherwise it is 0).

SOURCE CODE

```
TITLE '—— WRTUSMF —— WRITE USER SMF RECORD ——'
PRINT NOGEN MACRO EXPANSION INVISIBLE
YREGS , REGISTER SYMBOLS
WRTUSMF CSECT , REUS, RENT, REFR, AC(1)
WRTUSMF RMODE ANY
WRTUSMF AMODE 31
SAVE (14,12),, 'WRTUSMF &SYSDATC &SYSTIME '
LR R12, R15
USING WRTUSMF, R12
LR R9, R1 SAVE ADDRESS OF PARAMETER ADDRESSES
LA R0, VARSL LENGTH OF VARIABLE AREA
STORAGE OBTAIN, LENGTH=(0), LOC=ANY
LR R4, R1 SAVE ADDRESS OF VARIABLE AREA
LR R0, R1
```

```

LA    R1,VARSL
XR    R15,R15
MVCL R0,R14          CLEAR VARIABLE AREA
ST    R4,8(,R13)     CHAIN SAVE AREAS
ST    R13,4(,R4)
LR    R13,R4         ADDR OF OWN SAVE AREA = ADDR OF VARS
USING VARS,R13
EJECT

*****
*      CHECK APF                                           *
*****
MVC   TIME_,TIME     TIME PROTOTYPE TO VAR AREA
MVC   WTO_,WTO       WTO PROTOTYPE TO VAR AREA
SPACE
USING PSA,0
L     R1,PSATOLD     ADDR OF TCB
LTR   R11,R1        TCB EXISTING?
BNZ   TCBOK         YES, EXECUTING UNDER A TCB
* EXECUTING UNDER A SVRB, THEREFORE IN SUPERVISOR STATE;
* BECAUSE OF THIS, APF AUTHORIZATION IS UNIMPORTANT
L     R2,PSAAOLD     ADDR OF ASCB
USING ASCB,R2
L     R2,ASCBASXB    ADDR OF ASXB
USING ASXB,R2
L     R11,ASXBLTCB   ADDR OF LAST TCB
TCBOK DROP R2
DS    0H
SPACE
USING TCB,R11
L     R10,TCBJSCB    ADDR OF JSCB
SLL   R10,8
SRL   R10,8         IT'S A 24-BIT ADDRESS
USING IEZJSCB,R10
LTR   R1,R1         EXECUTING UNDER A SVRB?
BZ    APFOK         YES, IN SUPERVISOR STATE
TM    JSCBOPTS,JSCBAUTH APF AUTHORIZED?
BO    APFOK         YES
SPACE
NUCLKUP BYNAME,NAME='SVCTABLE',ADDR=(R3) SVC TABLE IN NUCLEUS
LTR   R15,R15       ADDRESS OF SVC TABLE FOUND?
BNZ   APFKO         NO
USING SVCENTRY,R3
TM    83*SVCENTL+SVCTP,SVCAPF APF NECESSARY FOR SVC 83?
BZ    APFOK         NO
DROP  R3
SPACE
APFKO DS    0H
MVI   RC,8          ERROR RETURN CODE
WTO   MF=(E,WTO_),TEXT=MAPF MESSAGE TO JOBLOG AND SYSLOG
B     RETURN
APFKO DS    0H
EJECT

```

```

*****
*          BUILD SMF RECORD          *
*****
LA      R0,SMFUDATA-SMFRCD2  MIN LENGTH OF SMF RECORD
STH     R0,SMF2LEN          RDW
MVI     SMF2FLG,B'00011110'  MVS/SP VERSION 4
MVI     SMF2RTY,USMFREC#    RECORD TYPE
SPACE
TIME    BIN,TMEDTE,LINKAGE=SYSTEM,MF=(E,TIME_)
MVI     DTE+4,X'C0'
LM      R0,R3,TMEDTE        TTTTTTTT,00000000,0YYYYDDD,C0000000
STCM    R0,B'1111',SMF2TME  TIME
SLDL    R2,4                YYYYDDDC
STCM    R2,B'1111',SMF2DTE  DATE
SP      SMF2DTE,=P'1900000'  0ZYYDDDC (Z=0: 19.. , Z=1: 20..)
OI      SMF2DTE+3,X'0F'     0ZYYDDDF
SPACE
L       R1,CVTPTR           ADDR OF CVT
USING   CVTMAP,R1
L       R1,CVTSMCA         ADDR OF SMCA
USING   SMCABASE,R1
MVC     SMF2SID,SMCASID    SYS ID TO SMF RECORD
DROP    R1
SPACE
L       R1,TCBTIO          ADDR OF TIOT
USING   TIOT1,R1
MVC     SMFUJOBN,TIOCJOB   JOB NAME TO SMF RECORD
MVC     SMFUSTPN,TIOCSTEP  STEP NAME TO SMF RECORD
DROP    R1
SPACE
L       R1,JSCBSSIB        ADDR OF SSIB
USING   SSIB,R1
MVC     SMFUJOBI,SSIBJBID  JES2 JOB ID TO SMF RECORD
CLC     =C'JES',SSIBSSNM   STARTED BY JES?
BE      JOBIDOK            YES
MVI     SMFUJOBI,C' '      NO JOB ID WHEN STARTED BY MSTR
MVC     SMFUJOBI+1(L'SMFUJOBI-1),SMFUJOBI
JOBIDOK DS 0H
DROP    R1
SPACE
L       R1,PSAAOLD         ADDR OF ASCB
USING   ASCB,R1
L       R1,ASCBASXB        ADDR OF ASXB
USING   ASXB,R1
L       R1,ASXBSENV        ADDR OF ACEE
USING   ACEE,R1
MVC     SMFUUSER,ACEEUSRI  RACF USERID TO SMF RECORD
DROP    R1
SPACE
LTR     R9,R9              NO PARAMETER?
BZ      PARERR              YES, CALLED THE WRONG WAY
TM      0(R9),X'80'        ONE PARAMETER?
BO      PGM                 YES, CALLED AS BATCH PROGRAM

```

	TM	4(R9),X'80'	TWO PARAMETERS?
	BO	REXXLINK	YES, CALLED AS PROGRAM BY REXX
	TM	8(R9),X'80'	THREE PARAMETERS?
	BO	PARERR	YES, CALLED THE WRONG WAY
	* WHEN 4 PARAMETERS THEN MAYBE IT'S A TSO COMMAND (R1->CPPL);		
	* IN THIS CASE THE LEFTMOST BIT OF THE 4TH PARAMETER ADDRESS IS ZERO;		
	* 3RD PARAMETER IS PSCB, VERIFY IT		
	USING CPPL,R9		
	CLC	JSCBPSCB,CPPLPSCB	PSCB AS 3RD PARAMETER?
	BE	CMD	YES, CALLED AS TSO COMMAND
PARERR	DS	ØH	
	MVI	RC,8	ERROR RETURN CODE
	WTO	MF=(E,WTO_),TEXT=MPAR	MESSAGE TO JOBLOG & SYSLOG
	B	RETURN	
	SPACE		
PGM	DS	ØH	CALLED AS BATCH PROGRAM
	L	R9,Ø(,R9)	ADDR OF PARAMETER
	LH	R7,Ø(,R9)	LENGTH OF PARAMETER STRING
	LA	R6,2(,R9)	ADDR OF PARAMETER STRING
	B	PAROK	
	SPACE		
REXXLINK	DS	ØH	CALLED BY REXX STMT 'ADDRESS LINK'
	LM	R6,R7,Ø(R9)	ADDRESS OF ADDR AND LENGTH
	L	R7,Ø(,R7)	LENGTH OF PARAMETER STRING
	L	R6,Ø(,R6)	ADDR OF PARAMETER STRING
	B	PAROK	
	SPACE		
CMD	DS	ØH	CALLED AS TSO COMMAND
	L	R9,CPPLCBUF	ADDR OF COMMAND BUFFER
	DROP	R9	
	LH	R7,Ø(,R9)	LENGTH OF COMMAND BUFFER
	LH	R6,2(,R9)	OFFSET TO COMMAND PARAMETER
	LA	R6,4(,R6)	OFFSET INCLUDING LENGTH FIELDS
	SLR	R7,R6	LENGTH OF PARAMETER OF COMMAND
	ALR	R6,R9	ADDR OF PARAMETER STRING
PAROK	DS	ØH	
	SPACE		
	LTR	R7,R7	LENGTH GREATER ZERO?
	BNP	DATOK	NO, NO DATA IN SMF RECORD
	LA	RØ,L'SMFUDATA	MAX LENGTH OF DATA IN SMF RECORD
	CLR	R7,RØ	LONGER?
	BNH	LNGOK	NO
	LR	R7,RØ	MAX ALLOWED LENGTH
LNGOK	DS	ØH	
	LR	RØ,R7	
	AH	RØ,SMF2LEN	CORRECT RDW
	STH	RØ,SMF2LEN	
	STC	R7,SMFUDATL	LENGTH OF DATA IN SMF RECORD
	BCTR	R7,Ø	LENGTH MINUS ONE BECAUSE OF EX
	EX	R7,EXMVC	DATA TO SMF RECORD
DATOK	DS	ØH	
	EJECT		

```

*****
*          WRITE SMF RECORD                                     *
*****
          LA      R3,SMFRCD2          ADDR OF SMF RECORD
          SMFEWTM (R3),BRANCH=NO      WRITE SMF RECORD
          LTR     R15,R15              OK?
          BZ      SMFOK                YES
          MVI     RC,8                 ERROR RETURN CODE
          MVC     MSMF_TXT,MSMF        MESSAGE TEXT TO VAR AREA
          CVD     R15,PACK
          OI      PACK+L'PACK-1,X'0F'
          UNPK    MSMF_R15,PACK+L'PACK-2(2) R15 TO MESSABE TEXT
          WTO     MF=(E,WTO_),TEXT=MSMF_ MESSAGE TO JOBLOG AND SYSLOG
SMFOK    DS      0H
          EJECT
*****
*          FINISH                                             *
*****
RETURN   DS      0H
          XR      R3,R3
          IC      R3,RC                RETURN CODE
          LR      R1,R13              ADDR OF VAR AREA
          L       R13,4(,R13)         R13 ADDR OF CALLER'S SAVE AREA AGAIN
          LA      R0,VARSL            LENGTH OF VAR AREA
          STORAGE RELEASE,LENGTH=(0),ADDR=(1)
          LR      R15,R3              RETURN CODE
          RETURN (14,12),,RC=(15)
          EJECT
*****
*          CONSTANTS                                         *
*****
USMFREC# EQU    239                  RECORD TYPE OF USER SMF RECORD
          SPACE
EXMVC    MVC     SMFUDATA(0),0(R6)    DATA IN SMF RECORD
          SPACE
TIME     TIME    LINKAGE=SYSTEM,MF=L  TIME PROTOTYPE
TIMEL    EQU     *-TIME              LENGTH OF TIME PROTOTYPE
WTO      WTO     TEXT=,ROUTCDE=11,MF=L WTO PROTOTYPE
WTOL     EQU     *-WTO              LENGTH OF WTO PROTOTYPE
          SPACE
MAPFT    DC      Y(L'MAPFT)
MAPF     EQU     MAPFT-2,*-MAPFT+2
          SPACE
MPART    DC      Y(L'MPART)
MPAR     EQU     MPART-2,*-MPART+2
          SPACE
MSMFT    DC      Y(L'MSMFT)
MSMF     EQU     MSMFT-2,*-MSMFT+2
          SPACE
          LTORG ,

```

DROP R13,R12,R11,R10 PERMANENT REGISTERS

EJECT

* VARIABLES *

* NOTE: BECAUSE R13 IS ALSO BASE REGISTER OF VARIABLE AREA,
* SAVE AREA HAS TO BE LOCATED AT START OF VARIABLE AREA

```

VARS      DSECT ,          VARIABLE AREA
          DS      18F      OWN SAVE AREA
          SPACE

PACK      DS      D
          DS      0F

TIME_     DS      XL(TIMEL)    TIME PARAMETER LIST
          DS      0F

WTO_     DS      XL(WTOL)     WTO PARAMETER LIST
RC        DS      X          RETURN CODE
          SPACE

TME      DS      F          TIME BINARY, 1/100 SECONDS
          DS      F          ZERO
DTE      DS      F          DATE PACKED 0YYYYDDD
          DS      F          ZERO
TMEDTE   EQU     TME,*-TME    TIME/DATE GIVEN BY TIME MACRO
          SPACE

MSMF_TXT DS      CL(L'MSMF)    VARIABLE MESSAGE TEXT
          ORG     *-2

MSMF_R15 DS      ZL2          VALUE OF R15 (2 DIGITS) IN MESSAGE
MSMF_    EQU     MSMF_TXT,*-MSMF_TXT
          SPACE 3
          IFASMFR 2          SMF RECORD, DEFAULT PART
          ORG     SMFRCD2+18 IT'S LENGTH IS ALWAYS 18
SMFUJOBN DS      CL8          JOB NAME
SMFUSTPN DS      CL8          STEP NAME (IF TSO: LOGON PROCEDURE)
SMFUJOBI DS      CL8          JES JOB ID
SMFUUSER DS      CL8          RACF USER
SMFUDATL DS      X          LENGTH OF VAR DATA
SMFUDATA DS      CL100       VAR DATA, MAX LENGTH IS 100
          SPACE 3
          DS      0D
VARSL    EQU     *-VARS      LENGTH OF VAR AREA
          EJECT
    
```

* DSECTS *

```

          IHAPSA LIST=NO      PSA, POINTING TO CVT, TCB, ASCB
          SPACE 3
          IHAASCB LIST=NO    ASCB, POINTING TO ASXB
          SPACE 3
          IHAASXB LIST=NO    ASXB, POINTING TO ACEE
          SPACE 3
          IHAACEE ,          ACEE, CONTAINING USER
          SPACE 3
          IKJTCB ,          TCB, POINTING TO JSCB, TIOT
          SPACE 3
    
```



```

        IEZJSCB ,                JSCB, POINTING TO SSIB, PSCB; APF
        SPACE 3
TIOT    DSECT ,
        IEFTIOT1 ,             TIOT, CONTAINING JOB NAME
        SPACE 3
        IEFJSSIB ,            SSIB, CONTAINING JOB ID
        SPACE 3
        CVT  DSECT=YES         CVT, POINTING TO SMCA
        SPACE 3
        IEESMCA ,             SMCA, CONTAINING SYS ID
        SPACE 3
        IKJCPPL ,             CPPL, POINTING TO PSCB
        SPACE 3
        IHASVC ,              ENTRY IN SVC TABLE
SVCENTL EQU  *-SVCENTRY      LENGTH OF ENTRY IN SVC TABLE
        SPACE
        END  WRTUSMF

```

Extracting DDname information

INTRODUCTION

DDINFO is a utility that can be used to extract information about the allocation of a DDname. The utility allows you to obtain the jobname, VOLSER, and the DSNNAME through the DDname. This information can be useful for applications that require verification for particular DD cards.

A program will, for example, know after a write to a tape, when the operating system requests the mounting of the next volume. Other applications will be able to verify if any DD card has the keywords DUMMY or NULLFILE.

OPERATIONAL ENVIRONMENT

The parameters of the routine are: pjob, pddn, pdsn, and pvls. A return code other than zero indicates an existing condition or anomaly. DDINFO can be called from any programming language that supports standard OS/370 linkage conventions.


```

L      R4, WORD          FCB AREA
MVC   PDSN+Ø(44),16(R4) DSNAME - JFCBDSNM
UCBØØ EQU *
LA    R4,41(R3)         TIOEFSRT - UCB ADDRESS
MVC   WORD+Ø(4),=X'ØØØØØØØØ'
MVC   WORD+1(3),Ø(R4)   UCB ADDRESS WORD
L     R4,WORD           UCB AREA
MVC   PVLS+Ø(6),28(R4)  VOLUME SERIAL - UCBVOLI
INCRO EQU *
XR    R4,R4             R4 = Ø
IC    R4,24(R3)        TIOENTRY VALUE
AR    R3,R4            ADDRESS NEW DD ENTRY
B     LOOPØØ
DDEND EQU *
CLI   SW,C'N'
BE    ERRD2
CLI   PDSN,C'? '
BE    ERRD3
CLI   PVLS,C'? '
BE    ERRVL
MVC   VOCE,=F'ØØØØ'    INFO OK
B     EXITØ
ERRVL MVC VOCE,=F'ØØ2Ø' INVALID VOLSER
B     EXITØ
ERRD1 MVC VOCE,=F'ØØ25' INVALID DDNAME
B     EXITØ
ERRD2 MVC VOCE,=F'ØØ3Ø' DDNAME NOT FOUND
B     EXITØ
ERRD3 MVC VOCE,=F'ØØ35' INVALID DSNAME
EXITØ EQU *
MVC   Ø(8,R6),RJOB
MVC   Ø(8,R7),PDDN
MVC   Ø(44,R8),PDSN
MVC   Ø(6,R9),PVLS
L     R15,VOCE
PR                                         <<<<<<<<<<<<<<<<<<<<<<<<<<<<
WKAREA EQU *
WORD  DC F'Ø'
PDSN  DC 44C' '
PDDN  DC 8C' '
PVLS  DC 6C' '
PJOB  DC 8C' '
VOCE  DC F'Ø'
MSGREC DC 44C' '
SW    DC C' '
END   DDINFO

```

Y2K, SVC screening update

INTRODUCTION

In Issue 127 of *MVS Update*, April 1997, a program of mine called MYDATE was published, which provided an SVC11 screening tool for date manipulation. In case some users are exploiting this code, you may like to see the latest version, which incorporates a couple of fixes that have recently been required. If you obtained an occasional abend C03 using DFSORT, then this version fixes that; and if you have SAS 6.09E to allow the use of SVC11 screening, then you will require this version of MYDATE. Also if you are using LE370, then you will be receiving C03 abends using MYDATE. The solution to this is to use a stub COBOL routine to call MYDATE in order that the LE environment is correctly set up. Anyway, here is the latest version of the code. Note that this module will require access to an SVC for dynamic APF authorization. This was documented in the earlier article.

MYDATE

```
*****
*      MYDATE LETS YOU SIMULATE PROGRAM EXECUTION ON ANY ARBITRARY
*      DATE.  THE DATE TO BE USED IS STORED IN PACKED FORMAT AT
*      LABEL NEWDATE.
*
*      INVOCATION JCL IS AS FOLLOWS:
*
*      //      EXEC  PGM=MYDATE
*      //      MYDATEP DD  DISP=SHR,DSN=USER.LOADLIB(PROGRAM)
*      //      MYDATED DD  DISP=SHR,DSN=USER.LOADLIB(DYYYYYDDD)
*      //
*
*      MYDATE MODULE MUST BE LINK-EDITED WITH AC=1 INTO AN APF
*      LIBRARY. DDNAME MYDATEP POINTS TO THE LIBRARY
*      AND MEMBER NAME OF THE PROGRAM TO BE EXECUTED. MYDATEP
*      CAN POINT TO AN AUTHORIZED OR NON-AUTHORIZED LIBRARY. ADD
*      ADDITIONAL DD STATEMENTS AS NECESSARY.  PARM INFORMATION
*      ON THE EXEC STATEMENT WILL BE PASSED NORMALLY TO THE INVOKED
*      PROGRAM.
*      MYDATED DD NEEDS TO POINT TO A VALID LIBRARY, BUT THE
*      MEMBER DOES NOT NEED TO EXIST. IT IS MERELY A MECHANISM
*      BY WHICH THE JULIAN DATE CAN BE SUPPLIED AND BE EASILY
```

```

*      SEEN IN THE JOB.
*
*      NOTE 1: FOLLOWING USER ABENDS ARE POSSIBLE;
*              U001, THE DATE IS NOT NUMERIC
*              U002, THE DAY NUMBER IS GT 366
*              U003, THE DATE IS ZERO
*              IT IS ASSUMED THAT THE USER OF MYDATE WILL CARRY OUT
*              HIS/HER OWN VAILDATIONS OF LEAP YEARS.
*
*      NOTE: MYDATE OPERATES BY INTERCEPTING THE "TIME" SVC
*      AND RETURNING A PHONEY DATE VALUE IN REGISTER 1.  PROGRAMS
*      THAT CHECK CVTDATE OR USE PC-TYPE LINKAGE WILL NOT BE
*      AFFECTED.
*
MYDATE   CSECT                ADDR
MYDATE   AMODE 31
MYDATE   RMODE 24
NEWSVC   EQU    235
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
        USING *,R6
        PUSH USING
        BAKR 14,0
        LR   R6,R15
*
        XR   R4,R4          * CLEAR R4
        USING PSA,R4       * AND MAP THE PSA
        L    R4,PSATOLD    * GET THE PSATOLD CONTENTS FROM THE PSA
        USING TCB,R4
*
*** SUPERVISOR STATE AND KEY ZERO REQUIRED TO SET APPROPRIATE FLAGS IN
*** TCB. FIRST OF ALL NEED TO KNOW IF WE ARE OPERATING IN AN APF
*** CONCATENATION OR NOT. IF WE ARE, THEN WE DON'T NEED TO WORRY ABOUT
*** THE APF AUTHORIZATION BIT. IF NOT THEN THIS WILL REQUIRE SETTING
*** OFF PRIOR TO CALLING THE ROUTINE TO BE EXECUTED. FURTHERMORE WE
*** WILL ALSO HAVE TO DYNAMICALLY PREPARE OURSELVES RE-AUTHORIZED
*** BY USING SVC NEWSVC IF AUTHORISATION IS NOT NATURALLY AVAILABLE.
*** THE EASIEST WAY TO CHECK THIS IS TO USE TSETAUTH

```

```

*** IF APF IS OK, THEN R15 WILL BE ZERO.
*
      TESTAUTH FCTN=1
      LTR  15,15          * IS IT AUTHORIZED
      BZ   GO_MODE       * APF OK?
      SVC NEWSVC         * NO SO APF SET
*
GO_MODE DS 0H
*
      MODESET MODE=SUP,          +
          KEY=ZERO
      EJECT
*
*** WE NOW NEED TO LOCATE THE RELEVANT PROGRAM NAME AND APPROPRIATE
*** DATE TO USE. THIS IS DONE BY ISSUING A RDJFCB FOR THE MYDATEP AND
*** MYDATED DD'S.
*
GET_THE_DATE DS 0H
*
      RDJFCB (MYDCB)
*
      LA 2,JFCB          * MAP THE JFCB
      USING MYJFCB,2
*
*** AND RETRIEVE THE DATE. NOTE THE DATE WILL BE PRECEDED BY AN
*** ALPHABETIC IN THE MEMBER NAME TO AVOID JCL ERRORS.
*
      MVC CHARDATE,JFCBELNM+1
*
*** VALIDATE THE DATE.
*
      MVZ  NUMFIELD,CHARDATE
      CLC  NUMFIELD,=C'00000000' CHECK FOR NUMERIC
      BE   ITS_NUMERIC
*
*** IF THE DATE IS NOT NUMERIC, ISSUE AN ABEND MACRO CODE 1.
*
      ABEND 1
*
ITS_NUMERIC DS 0H
*
*** ONCE WE KNOW IT IS NUMERIC, WE NEED TO BE SURE IT IS A VALID DATE.
*
      PACK NEWDATE,CHARDATE
      CP   NEWDATE+2(2),=P'366' * CHECK DAY RANGE
      BNH  CHECK_LOW          * GO CHECK NOT ZERO
*
*** DAYS TOO HIGH SO ABEND CODE 2
*
      ABEND 2
CHECK_LOW DS 0H
*

```

```

        CP    NEWDATE+2(2),=P'0'    * CHECK FOR ZERO
        BH    CHECK_YEAR            * ITS OK SO CHECK THE YEAR.
*
*** DAYS ZERO SO ABEND CODE 3
*
        ABEND 3
*
CHECK_YEAR DS 0H
*
        CLI    NEWDATE,X'20'        * IF ITS NOT A YEAR2000 DATE.
        BNE    SET_1900             * GO INDICATE 1900
        MVI    NEWDATE,X'01'        * ELSE SET FLAG FOR 2000.
        B      GET_PROGRAM          * NO GO SEE WHAT WE ARE TO CALL.
*
SET_1900 DS 0H
*
        MVI    NEWDATE,X'00'
*
GET_PROGRAM DS 0H
*
        RDJFCB (PROGDCB)
*
        LA    R2,JFCB
        USING MYJFCB,2
*
        MVC    PROGNAME,JFCBELNM
*
*** NOW INITIATE THE SCREENING PROCESS.
*
*** IN ORDER TO SET UP SCREENING CARRY OUT THE FOLLOWING:
*** SET TCBSVCS FLAG BIT ON IN TCBFLGS7
*** SET TCBSVCA2 TO THE ADDRESS OF THE SCREENING TABLE
*** SET TCBSVCSP BIT ON IF SCREENING IS TO APPLY TO ALL SUBSEQUENT
*** TCBS
*** THEN IN THE SCREENING TABLE:
*** SET BIT 1 TO ADDRESSING MODE OF THE SCREENING ROUTINE AS FOLLOWS:
*** IF 0 THEN IT'S ADDRESSING MODE 24. IF 1 THEN ITS 31 BIT.
*** THE REST OF BITS 1-31 CONTAIN THE ADDRESS OF OUR SVC ROUTINE
*** BYTE 4 OF THE SCREEN TABLE INDICATES THE TYPE OF OUR SVC. IN THIS
*** CASE WE USE X'C0' TO INDICATE A TYPE 4 SVC.
*** BYTE 5 INDICATES WHETHER OR NOT THE SVC CAN BE ISSUED IN AR MODE;
*** 0 SAYS IT CAN'T WHILE X'80' SAYS IT CAN.
*** BYTES 6&7 INDICATE THE LOCKS. IN OUR CASE 0 FOR LOCAL. (BIT 0)
*** THE SCREENING TABLE ITSELF CONSISTS OF BYTES 8-263 WHERE A X'80'
*** AT AN OFFSET EQUIVALENT TO THE SVC NUMBER MEANS THE SVC CAN BE
*** ISSUED WHILE A X'00' MEANS PASS CONTROL TO OUR SVC ROUTINE.
***
        STORAGE OBTAIN,LENGTH=280,SP=254,ADDR=(10)
*
        MVC    0(4,R10),=AL4(SCREEN) * ADDRESS OF SCREENING ROUTINE
        OI    0(R10),X'80'
        MVC    4(4,R10),=X'C0000000'
        MVI    8(R10),X'80'          * INDICATE A NO SCREEN BLOCK

```

```

MVC  9(255,R10),8(R10) * BY REPEATING THE X'80'
MVI  19(R10),X'00'      * THEN MAKE SVC 11 SCREENED.
ST   R10,TCBSVCA2      * SET THIS UP IN TCB.
OI   TCBFLGS7,X'28'    * AND SWITCH ON SCREENING.
*
*** NOW CALL THE TIME DIVERTED PROGRAM.
*** HAVING FIRST RESET THE CALLERS REGISTERS TO PRE-MYDATE FIDDLING.
*
      L R2,TCBJSCB      * OTHERWISE, DO RESET AND SET PROBLEM
      USING IEZJSCB,R2  * STATE
      NI JSCBOPTS,X'FF'-JSCBAUTH
      MODESET MODE=PROB,KEY=NZERO
*
*** NOW LOAD THE PROGRAM TO CHECK IF IT IS APF'ED
*
      LOAD EPLOC=PROGNAME
      SRL R1,24
      C   R1,=F'1'
      BNE GO_ASIS * THE PROGRAM ISN'T AUTHORISED
      SVC NEWSVC * RE-AUTHORISE
*
GO_ASIS DS 0H
      EREG 1,2
      EREG 13,14
      LINK EPLOC=PROGNAME
*
* RETURN TO CALLER
      PR
      EJECT
*
*** THE FOLLOWING IS THE SCREENING ROUTINE.
*
SCREEN DS 0H
*
*** ON ENTRY TO THIS SVC SCREENING ROUTINE, R4 WILL BE SET TO THE TCB
*** AND R6 WILL POINT TO THIS ROUTINE.
*** IF R1 CONTAINS X'04' THEN WE NEED TO DO SOME FANCY FOOTWORK TO
*** RETURN AN STCK VALUE IN R0.
*
      USING *,R6
      LR 9,14          * SAVE RETURN ADDRESS.
*
      NI   TCBFLGS7,X'D7' * SWITCH SCREENING OFF TO PREVENT
*
      ST   R1,MYREG1      * SAVE REG1
      ST   R0,MYREG0      * BETTER KEEP R0 AS WELL
      TM   MYREG1+3,X'04' * IS THIS AN STCK SVC?
      BC   1,FLIPREGS     * YES SO GO DO FOOTWORK
*
* A DOUBLE SCREENING ISSUE.
      SVC  11            * GET THE TIME
*
RESFLAG DS 0H

```



```

      OI    TCBFLGS7,X'28'   * AND RESET THE SCREENING.
*
      ICM   R1,15,NEWDATE   * FIDDLE DATE
      BR   9                 * AND RETURN TO CALLER
FLIPREGS DS  ØH
      LA    R1,2             * SWITCH TO APPROPRIATE TYPE OF SVC
      SVC   11
      ICM   R1,15,NEWDATE   * FIDDLE DATE
      XC    BYTES16,BYTES16
      ST    RØ,BYTES16+4
      ST    R1,BYTES16+8
      CONVTO CONVVAL=BYTES16,TODVAL=MYSTCK,DATETYPE=YYDDD
*
*** LOCATE THE CVT
*
      XR    3,3
      USING PSA,3
*
      L     3,FLCCVT
      USING CVT,3
*
      L     3,CVTEXT2
      USING CVTXTNT2,3
*
*** NOW REMOVE THE DATE OFFSET
*
      L     R1,MYSTCK
      S     R1,CVTLDTOL   * OFFSET REMOVED
      ST    R1,MYSTCK
      L     R1,MYREGØ
      MVC   Ø(8,R1),MYSTCK
      LA    R1,4
      L     RØ,MYREGØ
      B     RESFLAG
*
      DS    ØF
MYREGØ DS    F
MYREG1 DS    F
BYTES16 DS   2D
MYSTCK  DS    D
NEWDATE DS    F
MYDCB   DCB  DDNAME=MYDATED,EXLST=LIST,DSORG=PS,MACRF=R
PROGDCB DCB  DDNAME=MYDATEP,EXLST=LIST,DSORG=PS,MACRF=R
LIST    DS  ØF
          DC  X'Ø7'
          DC  AL3(JFCB)
JFCB    DS  ØF,18ØC
PROGNAME DS  CL8
CHARDATE DS   CL7
NUMFIELD DC  XL7'ØØØØØØØØØØØØØØØØ'
          LTORG

```

```

MYJFCB DSECT
    IEFJFCBN
    PRINT NOGEN
    IKJTBC
    IEZJSCB
    IHAPSA
    CVT DSECT=YES

```

An IPL subsystem (part 2)

This month we continue our look at the Initial Program Load Subsystem which reduces the errors inherent in the manual typing and entering of system commands required to activate on-line systems.

```

        B      CPWTOERR          GO TO ISSUE AN ERROR MESSAGE
        EJECT
*****
*      START NETWORK PRODUCTS ON TECHNOLOGY AFTER SHUTDOWN      *
*****
        SPACE 1
YMMUP   MVC   CMDPARM(16),CMDDIR29 SAVE PARM FOR BLDL(NET RELOAD)
        SPACE 1
        BAS   R10,READCMD5      BRANCH TO BUILD THIS COMMAND TABLE
        L     R3,CMDA1STC       LOAD ADDRESS OF 1ST COMMAND
        SPACE 1
YMMUP2  CLC   0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
        BE    YMMUP3            IF SO START TO TERMINATE
        CLI   41(R3),C'P'       IS IT A PRODUCT BEING STARTED
        BNE   YMMUP2A           SIMPLY ISSUE THIS NON-PRODUCT CMD
        BAS   R10,CPACTIVE      SCAN SYSTEM TO SEE IF ALREADY ACTIVE
        CLI   41(R3),C'A'       WAS IT ACTIVE THEN BYPASS ISSUANCE
        BE    YMMUP2B           OF THIS START COMMAND
        MVC   33(8,R3),=C'      ' BLANK OUT PRODUCT NAME AREA
        MVI   41(R3),C' '       BLANK OUT PRODUCT INDICATOR
YMMUP2A MVC   COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK & ISSUE
        LA    R1,50+4           LENGTH OF EACH COMMAND PLUS CONSTANT
        STH   R1,WTOMSG         SET LENGTH IN INTERNAL COMMAND
        BAS   R10,PATSV34       ISSUE START COMMAND
YMMUP2B L     R3,0(R3)          LOAD ADDRESS OF NEXT COMMAND
        B     YMMUP2            PROCESS NEXT COMMAND
        SPACE 1
YMMUP3  BAS   R10,CMDFREE       FREE ALL COMMAND AREAS ACQUIRED
        MVC   CMDPARM(16),CMDDIR31 SAVE PARM FOR BLDL DIRECTORY

```

	ST	R1,CMDR1SAV	SAVE R1 TILL AFTER READCMDS
	ST	R2,CMDR2SAV	SAVE R2 TILL AFTER READCMDS
	BAS	R10,READCMDS	BRANCH TO BUILD THIS COMMAND TABLE
	L	R1,CMDR1SAV	RESTORE R1
	L	R2,CMDR2SAV	RESTORE R2
	L	R3,CMDA1STC	LOAD ADDRESS OF FIRST COMMAND
	ST	R3,PATVTAB	STORE IT FOR ORE PROCESSING
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R8,PATDOORE	RESPOND TO WTOR
	BAS	R10,CMDFREE	FREE ALL COMMAND AREAS ACQUIRED
	SPACE	1	
YMMUP5	B	DCABORT	CLEAN UP AND TERMINATE
	EJECT		
	SPACE	1	
TECUP	MVC	CMDPARM(16),CMDDIR01	SAVE PARM FOR BLDL(NET RELOAD)
	SPACE	1	
	BAS	R10,READCMDS	BRANCH TO BUILD THIS COMMAND TABLE
	L	R3,CMDA1STC	LOAD ADDRESS OF 1ST COMMAND
	SPACE	1	
TECUP2	CLC	0(4,R3),=X'00000000'	CHECK FOR END OF CHAIN
	BE	TECUP3	IF SO START TO TERMINATE
	CLI	41(R3),C'P'	IS IT A PRODUCT BEING STARTED
	BNE	TECUP2A	SIMPLY ISSUE THIS NON-PRODUCT CMD
	BAS	R10,CPACTIVE	SCAN SYSTEM TO SEE IF ALREADY ACTIVE
	CLI	41(R3),C'A'	WAS IT ACTIVE THEN BYPASS ISSUANCE
	BE	TECUP2B	OF THIS START COMMAND
	MVC	33(8,R3),=C'	' BLANK OUT PRODUCT NAME AREA
	MVI	41(R3),C' '	BLANK OUT PRODUCT INDICATOR
TECUP2A	MVC	COMMNDWK(50),4(R3)	MOVE COMMAND TO COMMAND WORK & ISSUE
	LA	R1,50+4	LENGTH OF EACH COMMAND PLUS CONSTANT
	STH	R1,WTOMSG	SET LENGTH IN INTERNAL COMMAND
	BAS	R10,PATSVC34	ISSUE START COMMAND
TECUP2B	L	R3,0(R3)	LOAD ADDRESS OF NEXT COMMAND
	B	TECUP2	PROCESS NEXT COMMAND
	SPACE	1	
TECUP3	BAS	R10,CMDFREE	FREE ALL COMMAND AREAS ACQUIRED
	MVC	CMDPARM(16),CMDDIR21	SAVE PARM FOR BLDL DIRECTORY
	ST	R1,CMDR1SAV	SAVE R1 TILL AFTER READCMDS
	ST	R2,CMDR2SAV	SAVE R2 TILL AFTER READCMDS
	BAS	R10,READCMDS	BRANCH TO BUILD THIS COMMAND TABLE
	L	R1,CMDR1SAV	RESTORE R1
	L	R2,CMDR2SAV	RESTORE R2
	L	R3,CMDA1STC	LOAD ADDRESS OF FIRST COMMAND
	ST	R3,PATVTAB	STORE IT FOR ORE PROCESSING
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R10,PATREST	TAKE A BREAK
	BAS	R8,PATDOORE	RESPOND TO WTOR
	BAS	R10,CMDFREE	FREE ALL COMMAND AREAS ACQUIRED

```

SPACE 1
TECUP5  B   DCABORT          CLEAN UP AND TERMINATE
        EJECT
*****
*       START NETWORK PRODUCTS ON DEVELOPMENT AFTER SHUTDOWN       *
*****

SPACE 1
DEVUP   MVC  CMDPARM(16),CMDDIR04 SAVE PARM FOR BLDL(NET RELOAD)
        SPACE 1
        BAS  R10,READCMDS      BRANCH TO BUILD THIS COMMAND TABLE
        L    R3,CMDA1STC      LOAD ADDRESS OF 1ST COMMAND
        SPACE 1
DEVUP2  CLC  0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
        BE   DEVUP3          IF SO START TO TERMINATE
        CLI  41(R3),C'P'      IS IT A PRODUCT BEING STARTED
        BNE  DEVUP2A         SIMPLY ISSUE THIS NON-PRODUCT CMD
        BAS  R10,CPACTIVE     SCAN SYSTEM TO SEE IF ALREADY ACTIVE
        CLI  41(R3),C'A'      WAS IT ACTIVE THEN BYPASS ISSUANCE
        BE   DEVUP2B         OF THIS START COMMAND
        MVC  33(8,R3),=C'     ' BLANK OUT PRODUCT NAME AREA
        MVI  41(R3),C' '     BLANK OUT PRODUCT INDICATOR
DEVUP2A MVC  COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK & ISSUE
        LA   R1,50+4         LENGTH OF EACH COMMAND PLUS CONSTANT
        STH  R1,WTOMSG       SET LENGTH IN INTERNAL COMMAND
        BAS  R10,PATSV34     ISSUE START COMMAND
DEVUP2B L    R3,0(R3)        LOAD ADDRESS OF NEXT COMMAND
        B    DEVUP2         PROCESS NEXT COMMAND
        SPACE 1
DEVUP3  BAS  R10,CMDFREE     FREE ALL COMMAND AREAS ACQUIRED
        MVC  CMDPARM(16),CMDDIR21 SAVE PARM FOR BLDL DIRECTORY
        ST   R1,CMDR1SAV     SAVE R1 TILL AFTER READCMDS
        ST   R2,CMDR2SAV     SAVE R2 TILL AFTER READCMDS
        BAS  R10,READCMDS    BRANCH TO BUILD THIS COMMAND TABLE
        L    R1,CMDR1SAV     RESTORE R1
        L    R2,CMDR2SAV     RESTORE R2
        L    R3,CMDA1STC     LOAD ADDRESS OF FIRST COMMAND
        ST   R3,PATVTAB     STORE IT FOR ORE PROCESSING
        BAS  R10,PATREST     TAKE A BREAK
        BAS  R10,PATREST     TAKE A BREAK
        BAS  R8,PATDOORE     RESPOND TO WTOR
        BAS  R10,CMDFREE     FREE ALL COMMAND AREAS ACQUIRED
        SPACE 1
DEVUP5  B    DCABORT          CLEAN UP AND TERMINATE
        EJECT
*****
*       CONNECT PRODUCTS AFTER THEY ARE UP                           *
*****

SPACE 1
CONNECT CLC  CMDSYSID(4),=C'VS05' WHAT SYSTEM IS IT??
        BE   CONNEPRO        BRANCH TO PRODUCTION
        CLC  CMDSYSID(4),=C'VS04' WHAT SYSTEM IS IT??
        BE   CONNEACC        BRANCH TO ACCENT

```

```

CLC  CMDSYSID(4),=C'VS01' WHAT SYSTEM IS IT??
BE   CONNEDEV          BRANCH TO DEVELOPMENT
CLC  CMDSYSID(4),=C'VS03' WHAT SYSTEM IS IT??
BE   CONNE09          BRANCH TO TECH (NOTHING-TERMINATE)
CLC  CMDSYSID(4),=C'VS02' WHAT SYSTEM IS IT??
BE   CONNEYMM         BRANCH TO YMM
SPACE
CPNOCONN WTO 'DCIPL08E - SYSTEM ID NOT FOUND: CONNECTS ARE IMPOSSIBLE'
      WTO  'DCIPL09A - NOTIFY SYSTEM PROGRAMMER'
      B    CONNE09          TERMINATE DCIPLES
      EJECT
CONNEPRO MVC  CMDPARM(16),CMDDIR24 SAVE PARM FOR PRODUCTION
      B    CONNE01
      SPACE 1
CONNEACC MVC  CMDPARM(16),CMDDIR23 SAVE PARM FOR ACCENT
      B    CONNE01
      SPACE 1
CONNEYMM MVC  CMDPARM(16),CMDDIR32 SAVE PARM FOR MILLENNIUM
      B    CONNE01
      SPACE 1
CONNEDEV MVC  CMDPARM(16),CMDDIR22 SAVE PARM FOR DEVELOPMENT
      SPACE 1
CONNE01 BAS  R10,READCMDS          BRANCH TO BUILD THIS COMMAND TABLE
      L    R3,CMDA1STC          LOAD ADDRESS OF 1ST COMMAND
      SPACE 1
CONNE02 CLC  0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
      BE   CONNE06          IF SO START TO TERMINATE
CONNE03 MVC  COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK & ISSUE
      LA   R1,50+4          LENGTH OF EACH COMMAND PLUS CONSTANT
      STH  R1,WTOMSG          SET LENGTH IN INTERNAL COMMAND
      BAS  R10,PATSVC34          ISSUE START COMMAND
CONNE04 L    R3,0(R3)          LOAD ADDRESS OF NEXT COMMAND
      B    CONNE02          PROCESS NEXT COMMAND
      SPACE 1
CONNE06 BAS  R10,CMDFREE          FREE ALL COMMAND AREAS ACQUIRED
      MVC  CMDPARM(16),CMDDIR25 SAVE WTORS
      ST   R1,CMDR1SAV          SAVE R1 TILL AFTER READCMDS
      ST   R2,CMDR2SAV          SAVE R2 TILL AFTER READCMDS
      BAS  R10,READCMDS          BRANCH TO BUILD THIS COMMAND TABLE
      L    R1,CMDR1SAV          RESTORE R1
      L    R2,CMDR2SAV          RESTORE R2
      L    R3,CMDA1STC          LOAD ADDRESS OF FIRST COMMAND
      ST   R3,PATVTAB          STORE IT FOR ORE PROCESSING
      BAS  R8,PATDOORE          RESPOND TO WTOR
      BAS  R10,CMDFREE          FREE ALL COMMAND AREAS ACQUIRED
      SPACE 2
      CLC  CMDSYSID(4),=C'VS05' WHAT SYSTEM IS IT??
      BE   CONNEPR          BRANCH TO PRODUCTION
      CLC  CMDSYSID(4),=C'VS04' WHAT SYSTEM IS IT??
      BE   CONNEAC          BRANCH TO ACCENT
      CLC  CMDSYSID(4),=C'VS01' WHAT SYSTEM IS IT??
      BE   CONNEDE          BRANCH TO DEVELOPMENT
      CLC  CMDSYSID(4),=C'VS03' WHAT SYSTEM IS IT??

```

```

BE    CONNEØ9          BRANCH TO TECH(NOTHING FOR TECH)
CLC   CMDSYSID(4),=C'VSØ2' WHAT SYSTEM IS IT??
BE    CONNEYM          BRANCH TO YMM
B     CPNOCONN         TERMINATE DCIPLES
EJECT
CONNEPR MVC  CMDPARM(16),CMDDIR28 SAVE PARM FOR PRODUCTION
B     CONNEØ7
SPACE
CONNEAC MVC  CMDPARM(16),CMDDIR26 SAVE PARM FOR ACCENT
B     CONNEØ7
SPACE
CONNEDE MVC  CMDPARM(16),CMDDIR27 SAVE PARM FOR DEVELOPMENT
B     CONNEØ7
SPACE
CONNEYM MVC  CMDPARM(16),CMDDIR33 SAVE PARM FOR MILLENNIUM
SPACE 1
CONNEØ7 ST   R1,CMDR1SAV          SAVE R1 TILL AFTER READCMDS
ST   R2,CMDR2SAV          SAVE R2 TILL AFTER READCMDS
BAS  R1Ø,READCMDS        BRANCH TO BUILD THIS COMMAND TABLE
L    R1,CMDR1SAV          RESTORE R1
L    R2,CMDR2SAV          RESTORE R2
L    R3,CMDA1STC         LOAD ADDRESS OF FIRST COMMAND
BAS  R1Ø,PATREST         TAKE A BREAK
ST   R3,PATVTAB         STORE IT FOR ORE PROCESSING
BAS  R8,PATDOORE         RESPOND TO WTOR
BAS  R1Ø,CMDFREE         FREE ALL COMMAND AREAS ACQUIRED
SPACE 1
CONNEØ9 B     DCABORT          CLEAN UP AND TERMINATE
EJECT
*****
*      TERMINATE NETWORK PRODUCTS ON DEVELOPMENT FOR QUICK SHUTDOWN *
*****
SPACE 1
DEVDOWN CLC  CMDSYSID(4),=C'VSØ1' TEST IF ISSUED ON CORRECT SYSTEM
BE    CPDWNDEV          IF SO, PROCESS IT, ELSE
B     CPWTOERR          ISSUE INFORMATIVE ERROR MESSAGE
SPACE 3
*****
*      TERMINATE NETWORK PRODUCTS ON ACCENT FOR A QUICK SHUTDOWN *
*****
SPACE 1
ACCDOWN CLC  CMDSYSID(4),=C'VSØ4' TEST IF ON CORRECT DOMAIN
BNE   CPWTOERR          IF NOT, ISSUE AN ERROR MESSAGE
SPACE 1
MVC   CMDPARM(16),CMDDIRØ8 SAVE PARM FOR BLDL DIRECTORY
B     CPCOMMON          BRANCH TO BUILD THIS COMMAND TABLE
SPACE 3
*****
*      TERMINATE NETWORK PRODUCTS ON PRODUCTION FOR A QUICK SHUTDOWN*
*****
SPACE 1
PRODOWN CLC  CMDSYSID(4),=C'VSØ5' TEST IF ON CORRECT DOMAIN

```

```

BNE CPWTOERR          IF NOT, ISSUE AN ERROR MESSAGE
SPACE 1
MVC CMDPARM(16),CMDDIR11 SAVE PARM FOR BLDL DIRECTORY
B CPCOMMON           BRANCH TO BUILD THIS COMMAND TABLE
SPACE 3
*****
*      TERMINATE NETWORK PRODUCTS ON TECHNOLOGY FOR QUICK SHUTDOWN *
*****
SPACE 1
TECDOWN CLC  CMDSYSID(4),=C'VS03' TEST IF ISSUED ON CORRECT DOMAIN
BNE CPWTOERR          BRANCH IF NOT
SPACE 1
MVC CMDPARM(16),CMDDIR02 SAVE PARM FOR BLDL DIRECTORY
B CPCOMMON           BRANCH TO BUILD THIS COMMAND TABLE
EJECT
*****
*      TERMINATE NETWORK PRODUCTS ON MILLENNIUM FOR A QUICK SHUTDOWN*
*****
SPACE 1
YMMDOWN CLC  CMDSYSID(4),=C'VS02' TEST IF ON CORRECT DOMAIN
BNE CPWTOERR          IF NOT, ISSUE AN ERROR MESSAGE
SPACE 1
MVC CMDPARM(16),CMDDIR30 SAVE PARM FOR BLDL DIRECTORY
B CPCOMMON           BRANCH TO BUILD THIS COMMAND TABLE
SPACE 3
*****
*      UNIVERSAL CODE FOR THE TERMINATION OF PRODUCTS ON ALL SYSTEMS *
*****
SPACE 1
CPDWNDEV MVC  CMDPARM(16),CMDDIR05 SAVE PARM FOR BLDL DIRECTORY
SPACE 1
CPCOMMON BAS  R10,READCMDS          BRANCH TO BUILD THIS COMMAND TABLE
L          R3,CMDA1STC             LOAD ADDRESS OF 1ST COMMAND
SPACE 1
CPDWN2 CLC  0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
BE CPDWN3          IF SO START TO TERMINATE
MVC COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK
LA R1,50+4          LENGTH OF EACH COMMAND PLUS CONSTANT
STH R1,WTOMSG       SET LENGTH IN INTERNAL COMMAND
BAS R10,PATSVC34    ISSUE COMMAND
L R3,0(R3)          LOAD ADDRESS OF NEXT COMMAND
B CPDWN2           PROCESS NEXT COMMAND
EJECT
CPDWN3 BAS  R10,CMDFREE           FREE ALL COMMAND AREAS ACQUIRED
MVC CMDPARM(16),CMDDIR19 SAVE PARM FOR BLDL DIRECTORY
ST R1,CMDR1SAV     SAVE R1 TILL AFTER READCMDS
ST R2,CMDR2SAV     SAVE R2 TILL AFTER READCMDS
BAS R10,READCMDS   BRANCH TO BUILD THIS COMMAND TABLE
L R1,CMDR1SAV     RESTORE R1
L R2,CMDR2SAV     RESTORE R2
L R3,CMDA1STC     LOAD ADDRESS OF FIRST COMMAND
ST R3,PATVTAB     STORE IT FOR ORE PROCESSING

```

```

BAS R8,PATDOORE RESPOND TO WTOR
BAS R10,CMDFREE FREE ALL COMMAND AREAS ACQUIRED
BAS R10,PATREST TAKE A BREAK
MVC CMDPARM(16),CMDDIR20 SAVE PARM FOR BLDL DIRECTORY
ST R1,CMDR1SAV SAVE R1 TILL AFTER READCMDS
ST R2,CMDR2SAV SAVE R2 TILL AFTER READCMDS
BAS R10,READCMDS BRANCH TO BUILD THIS COMMAND TABLE
L R1,CMDR1SAV RESTORE R1
L R2,CMDR2SAV RESTORE R2
L R3,CMDA1STC LOAD ADDRESS OF FIRST COMMAND
ST R3,PATVTAB STORE IT FOR ORE PROCESSING
BAS R10,PATREST TAKE ONE FINAL BREAK
BAS R8,PATDOORE RESPOND TO WTOR
BAS R10,CMDFREE FREE ALL COMMAND AREAS ACQUIRED
B DCABORT BRANCH TO END UP
EJECT

```

```

*****
* START NETWORK PRODUCTS ON ACCENT AFTER A QUICK SHUTDOWN *
*****

```

```

SPACE 1
ACCUP MVC CMDPARM(16),CMDDIR07 SAVE PARM FOR BLDL(NET RELOAD)
SPACE 1
ACCUP1 BAS R10,READCMDS BRANCH TO BUILD THIS COMMAND TABLE
L R3,CMDA1STC LOAD ADDRESS OF 1ST COMMAND
SPACE 1
ACCUP2 CLC 0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
BE ACCUP3 IF SO START TO TERMINATE
CLI 41(R3),C'P' IS IT A PRODUCT BEING STARTED
BNE ACCUP2A SIMPLY ISSUE THIS NON-PRODUCT CMD
BAS R10,CACTIVE SCAN SYSTEM TO SEE IF ALREADY ACTIVE
CLI 41(R3),C'A' WAS IT ACTIVE THEN BYPASS ISSUANCE
BE ACCUP2B OF THIS START COMMAND
MVC 33(8,R3),=C' ' BLANK OUT PRODUCT NAME AREA
MVI 41(R3),C' ' BLANK OUT PRODUCT INDICATOR
ACCUP2A MVC COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK & ISSUE
LA R1,50+4 LENGTH OF EACH COMMAND PLUS CONSTANT
STH R1,WTOMSG SET LENGTH IN INTERNAL COMMAND
BAS R10,PATSVC34 ISSUE START COMMAND
ACCUP2B L R3,0(R3) LOAD ADDRESS OF NEXT COMMAND
B ACCUP2 PROCESS NEXT COMMAND
SPACE 1
ACCUP3 BAS R10,CMDFREE FREE ALL COMMAND AREAS ACQUIRED
MVC CMDPARM(16),CMDDIR21 SAVE PARM FOR BLDL DIRECTORY
ST R1,CMDR1SAV SAVE R1 TILL AFTER READCMDS
ST R2,CMDR2SAV SAVE R2 TILL AFTER READCMDS
BAS R10,READCMDS BRANCH TO BUILD THIS COMMAND TABLE
L R1,CMDR1SAV RESTORE R1
L R2,CMDR2SAV RESTORE R2
L R3,CMDA1STC LOAD ADDRESS OF FIRST COMMAND
ST R3,PATVTAB STORE IT FOR ORE PROCESSING
BAS R10,PATREST TAKE A BREAK
BAS R10,PATREST TAKE A BREAK
BAS R10,PATREST TAKE A BREAK

```



```

        BAS R8,PATDOORE           RESPOND TO WTOR
        BAS R10,CMDFREE           FREE ALL COMMAND AREAS ACQUIRED
        SPACE 1
ACCUP5  B DCABORT                 CLEAN UP AND TERMINATE
        EJECT
*****
*      START NETWORK PRODUCTS ON PRODUCTION AFTER QUICK SHUTDOWN      *
*      OR AN IPL.                                                         *
*****

        SPACE 1
PROUP   MVC CMDPARM(16),CMDDIR10 SAVE PROUPNET FOR BLDL(NET RELOAD)
        SPACE 1
PROUP1  BAS R10,READCMDS          BRANCH TO BUILD THIS COMMAND TABLE
        L   R3,CMDA1STC           LOAD ADDRESS OF 1ST COMMAND
        SPACE 1
PROUP2  CLC 0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
        BE  PROUP3                 IF SO START TO TERMINATE
        CLI 41(R3),C'P'           IS IT A PRODUCT BEING STARTED
        BNE PROUP2A              SIMPLY ISSUE THIS NON-PRODUCT CMD
        BAS R10,CPACTIVE          SCAN SYSTEM TO SEE IF ALREADY ACTIVE
        CLI 41(R3),C'A'           WAS IT ACTIVE THEN BYPASS ISSUANCE
        BE  PROUP2B              OF THIS START COMMAND
        MVC 33(8,R3),=C'         ' BLANK OUT PRODUCT NAME AREA
        MVI 41(R3),C' '          BLANK OUT PRODUCT INDICATOR
PROUP2A MVC COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK & ISSUE
        LA  R1,50+4              LENGTH OF EACH COMMAND PLUS CONSTANT
        STH R1,WTOMSG            SET LENGTH IN INTERNAL COMMAND
        BAS R10,PATSVC34         ISSUE START COMMAND
PROUP2B L   R3,0(R3)            LOAD ADDRESS OF NEXT COMMAND
        B   PROUP2              PROCESS NEXT COMMAND
        SPACE 1
PROUP3  BAS R10,CMDFREE           FREE ALL COMMAND AREAS ACQUIRED
        MVC CMDPARM(16),CMDDIR21 SAVE NETWTOR3 FOR BLDL DIRECTORY
        ST  R1,CMDR1SAV          SAVE R1 TILL AFTER READCMDS
        ST  R2,CMDR2SAV          SAVE R2 TILL AFTER READCMDS
        BAS R10,READCMDS        BRANCH TO BUILD THIS COMMAND TABLE
        L   R1,CMDR1SAV          RESTORE R1
        L   R2,CMDR2SAV          RESTORE R2
        L   R3,CMDA1STC          LOAD ADDRESS OF FIRST COMMAND
        ST  R3,PATVTAB          STORE IT FOR ORE PROCESSING
        BAS R10,PATREST          TAKE A BREAK
        BAS R10,PATREST          TAKE A BREAK
        BAS R10,PATREST          TAKE A BREAK
        BAS R8,PATDOORE          RESPOND TO WTOR
        BAS R10,CMDFREE          FREE ALL COMMAND AREAS ACQUIRED
        SPACE 1
PROUP5  B DCABORT                 CLEAN UP AND TERMINATE
        EJECT
*****
*      ISOLATE THE WARNING MESSAGE SUPPLIED BY AN OPERATOR,            *
*      ENSURE THAT IT IS NOT 'TOO LONG' FOR WTO, THEN SHIFT IT RIGHT  *
*      IN A WTO BUFFER AREA SO THAT IT CAN BE PREFIXED WITH MCS DATA  *
*****

```

```

*****
        SPACE 1
PATWARN  LA    R15,COMMNDWK+ENTLEN-2 POINT TO END OF BUFFER
        CLI   Ø(R15),X'41'          TEST IF BUFFER IS FULL
        BNL   PAT2LONG              BRANCH IF MESSAGE IS TOO LENGTHY
        MVI   1(R15),C''''         SET ENDING QUOTE
        SPACE 1
        LA    R14,COMMNDWK+ENTLEN-3 POINT TO FIRST CHAR TO BE MOVED
        LA    R1,COMMNDWK+12        POINT TO START OF WARNING MESSAGE
        SPACE 1
PATDOWN  CR    R14,R1              TEST IF AT START OF COMMAND
        BL    DCNOWAIT             BRANCH IF ENTIRE MESSAGE WAS BLANKS
        BCTR  R14,RØ              POINT TO NEXT SOURCE CHARACTER
        CLI   Ø(R14),X'41'         TEST IF DATA PRESENT
        BL    PATDOWN              BRANCH IF NOT
        SPACE 1
PATLOOP  MVC   Ø(1,R15),Ø(R14)     SHIFT DATA TO THE RIGHT IN INPUT BUF
        BCTR  R14,RØ              POINT TO NEXT SOURCE CHARACTER
        BCTR  R15,RØ              POINT TO NEXT TARGET CHARACTER
        CR    R14,R1              TEST IF AT START OF COMMAND
        BNL   PATLOOP             BRANCH IF COMPLETE MSG WAS SHIFTED
        LR    R1,R15              SAVE POINTER TO START OF MESSAGE - 1
        SPACE 1
PATGETNB LA    R1,1(R1)            POINTER TO START OF MESSAGE
        CLI   Ø(R1),C' '          TEST IF BLANK
        BE    PATGETNB           IF SO, SCAN FOR NON-BLANK CHARACTER
        BCTR  R1,RØ              POINT TO PREVIOUS AVAILABLE LOCATION
        MVI   Ø(R1),C''''         SET BEGINNING QUOTE
        SPACE 1
        LA    R15,COMMNDWK+ENTLEN-1 POINT TO END OF BUFFER
        SR    R15,R1              COMPUTE LENGTH OF MESSAGE
        CH    R15,=AL2(ENTLEN-2Ø) TEST IF THERE IS ROOM FOR CMD + MSG
        BH    PAT2LONG           BRANCH IF NOT
        EX    R15,PATMOVE         SHIFT MESSAGE LEFT IN COMMAND BUFFER
        LA    R1,4+2Ø+1(R15)      LEN OF MSG PLUS MCS FLAGS + CON + 1
        SLL   R1,16              ALIGN LENGTH, CLEAR MCSFLAGS
        ST    R1,WTOMSG          SET LENGTH OF MESSAGE IN PARM LIST
        EJECT
*****
*        TRANSMIT A WARNING MESSAGE SUPPLIED BY AN OPERATOR          *
*        TO ALL ROSCOE AND TSO USERS.                                *
*****
        SPACE 1
        MVC   CMDPARM(16),CMDDIR13 SAVE PARM FOR BLDL DIRECTORY
        BAS   R1Ø,READCMD5        BRANCH TO BUILD THIS COMMAND TABLE
        BAS   R1Ø,CMRENQ          SERIALIZE ACCESS TO CSCB CHAIN
        L     R1,CVTPTR           POINT TO CVT
        USING CVT,R1              ESTABLISH CVT ADDRESSABILITY
        L     R2,CVTMSER          DATA AREA OF MSTR SCHD RES DATA AREA
        DROP  R1                  FORGET CVT
        SPACE 2
        USING CHAIN,R2           ESTABLISH CSCB ADDRESSABILITY

```

```

PATLOOP1 L    R3,CMDA1STC      LOAD ADDRESS OF 1ST COMMAND
        SPACE 1
        ICM  R2,15,CHPTR      POINTER TO NEXT CSCB
        BZ   PATDOTSO        DONE, NOW DO TSO
        SPACE 1
PATFIND  CLC  4(8,R3),CHCLS    TEST IF SHOULD JOB BE PROCESSED
        BNE  PATMISS          BRANCH IF NOT
        MVC  COMMNDWK(20),12(R3) SET SYSTEM-DEPENDENT BROADCAST COMND
        SPACE 1
        BAS  R10,PATSVC34     ISSUE WARNING MESSAGE
        B    PATLOOP1         POINT TO NEXT JOB'SNAME
        SPACE 1
PATMISS  CLC  0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
        BE   PATLOOP1        LOOP POWER
        L    R3,0(R3)        LOAD ADDRESS OF NEXT COMMAND
        B    PATFIND         POINT TO NEXT JOB'SNAME
        SPACE 1
PATDOTSO MVC  COMMNDWK(20),=CL20'          SE  '
        SPACE 1
        BAS  R10,CMRDEQ      REMOVE SERIALIZATION OF CSCB CHAIN
        SPACE 1
        BAS  R10,PATSVC34     TRANSMIT WARNING MSG TO TSO USERS
        BAS  R10,CMDFREE     FREE ALL COMMAND AREAS ACQUIRED
        B    DCNOWAIT        TRULY DONE
        EJECT
*****
*   W A R N I N G - ?PAP LOGIC - TO BE USED FOR IPL PROCESS ONLY!   *
*   PROCESS IMMEDIATE COMMANDS, ISSUE COMMANDS TO GRACEFULLY       *
*   TERMINATE ONLINE SYSTEMS, AND THEN TERMINATE A L L ACTIVE TASKS. *
*   DO NOT USE FOR QUICK NETWORK TAKEDOWNS...                       *
*****
        SPACE 1
*   DRAIN INITIATORS, PRINTERS, AND REMOTES
        SPACE 1
PATDRAIN MVC  CMDPARM(16),CMDDIR14 SAVE PARM FOR BLDL DIRECTORY
        BAS  R10,READCMDS     BRANCH TO BUILD THIS COMMAND TABLE
        L    R3,CMDA1STC     LOAD ADDRESS OF 1ST COMMAND
        SPACE 1
PATIMLUP CLC  0(4,R3),=X'00000000' CHECK FOR END OF CHAIN
        BE   PATIMEND        ISSUE COMMAND
        MVC  COMMNDWK(50),4(R3) MOVE COMMAND TO COMMAND WORK
        LA   R1,50+4         LENGTH OF EACH COMMAND PLUS CONSTANT
        STH  R1,WTOMSG       SET LENGTH IN INTERNAL COMMAND
        BAS  R10,PATSVC34     ISSUE COMMANDS
        L    R3,0(R3)        LOAD ADDRESS OF NEXT COMMAND
        B    PATIMLUP        PROCESS NEXT COMMAND
        SPACE 2
PATIMEND BAS  R10,CMDFREE     FREE ALL COMMAND AREAS ACQUIRED
        BAS  R10,PATREST     DODDLE SO REMOTES CAN DRAIN
        EJECT
*****

```

```

*          TERMINATE TASKS BY RESPONDING TO OUTSTANDING WTORS.          *
*****
SPACE 1
*          INITIATE TERMINATION OF IMS, NETVIEW, AND CA
SPACE 1
MVC  CMDPARM(16),CMDDIR18 SAVE PARM FOR BLDL DIRECTORY
ST   R1,CMDR1SAV          SAVE R1 TILL AFTER READCMDS
ST   R2,CMDR2SAV          SAVE R2 TILL AFTER READCMDS
BAS  R10,READCMDS        BRANCH TO BUILD THIS COMMAND TABLE
L    R1,CMDR1SAV          RESTORE R1
L    R2,CMDR2SAV          RESTORE R2
L    R3,CMDA1STC         LOAD ADDRESS OF FIRST COMMAND
ST   R3,PATVTAB          STORE IT FOR ORE PROCESSING
BAS  R8,PATDOORE         RESPOND TO OUTSTANDING WTORS
BAS  R10,CMDFREE         FREE ALL COMMAND AREAS ACQUIRED
SPACE 2
BAS  R10,PATREST         DODDLE SO THAT IMS
BAS  R10,PATREST         CAN INITIATE SHUTDOWN
BAS  R10,PATREST         HO HUM
EJECT
*****
*          GRACEFULLY TERMINATE TASKS BY PASSING APPROPRIATE PARAMETERS *
*          TO THEM VIA 'MODIFY' AND THE SVC 34 MECHANISM.          *
*          1          8          *
*          PAPMODIF HAS THE FORMAT: TASKNAMECOMMAND USED TO TERMINATE IT*
*          NAMES OF TASKS MUST BE COLLATED BY JOBNAME.          *
*****
SPACE 1
MVC  CMDPARM(16),CMDDIR15 SAVE PARM FOR BLDL DIRECTORY
BAS  R10,READCMDS        BRANCH TO BUILD THIS COMMAND TABLE
BAS  R10,CMRENQ          SERIALIZE ACCESS TO CSCB CHAIN
SPACE 2
L    R1,CVTPTR           POINT TO CVT
USING CVT,R1             ESTABLISH CVT ADDRESSABILITY
L    R2,CVTMSER          DATA AREA OF MSTR SCHD RES DATA AREA
DROP R1                 FORGET CVT
SPACE 1
PATLOOP2 L R3,CMDA1STC   LOAD ADDRESS OF 1ST COMMAND
SPACE 1
ICM  R2,15,CHPTR        POINTER TO NEXT CSCB
BZ   PATDOCM             DONE, NOW PLAY HARD BALL
PATGETSY CLC 4(8,R3),CHCLS TEST IF SHOULD JOB BE PROCESSED
BH   PATLOOP2           BRANCH IF NOT EVER POSSIBLE
BNE  PATMISSY           BRANCH IF NOT YET
MVC  COMMNDWK(30),12(R3) SET SYSTEM-DEPENDENT BROADCAST COMND
LA   R1,30+4            LENGTH OF EACH COMMAND PLUS CONSTANT
STH  R1,WTOMSG          SET LENGTH IN INTERNAL COMMAND
BAS  R10,PATSVC34       ISSUE WARNING MESSAGE
B    PATLOOP2           POINT TO NEXT JOB'S NAME
SPACE 1
PATMISSY CLC 0(4,R3),=X'00000000' CHECK FOR END OF CHAIN

```

```

BE      PATLOOP2          LOOP POWER
L       R3,0(R3)         LOAD ADDRESS OF NEXT COMMAND
B       PATGETSY         POINT TO NEXT JOB'SNAME
SPACE 1
PATDOCM BAS R10,CMRDEQ    REMOVE SERIALIZATION OF CSCB CHAIN
BAS R10,CMDFREE         FREE ALL COMMAND AREAS ACQUIRED
SPACE 1
BAS R10,PATREST        PAUSE FOR GRACEFUL TERMINATIONS
BAS R10,PATREST        ANOTHER PAUSE FOR THE CAUSE
EJECT

*****
*      ABRUPTLY TERMINATE TASKS BY ISSUING APPROPRIATE 'STOP'      *
*      OR 'CANCEL' COMMANDS VIA SVC 34 MECHANISM                    *
*                                                                    *
*      PAPPCANC HAS THE FORMAT: P PROCNAME (PROCNAME IS EQUIVALENT *
*                                C PROCNAME TO A TASK'S JOBNAME) *
*      NAMES OF TASKS MUST BE COLLATED.                            *
*****

SPACE 1
MVC CMDPARM(16),CMDDIR16 SET PAPPCANC PARM FOR BLDL COMMAND
BAS R10,READCMDS        BRANCH TO BUILD THIS COMMAND TABLE
SPACE 1
BAS R10,CMRENQ          SERIALIZE ACCESS TO CSCB CHAIN
SPACE 1
L       R1,CVTPTR        POINT TO CVT
USING CVT,R1            ESTABLISH CVT ADDRESSABILITY
L       R2,CVTMSER       DATA AREA OF MSTR SCHD RES DATA AREA
DROP R1                 FORGET CVT
SPACE 1
PATLOOP3 L R3,CMDA1STC    LOAD ADDRESS OF FIRST COMMAND
ICM R2,15,CHPTR         POINTER TO NEXT CSCB
BZ FINISHCM             CLEAN AND END
SPACE 1
PATGETCM CLC 6(8,R3),CHCLS TEST IF SHOULD JOB BE PROCESSED
BH PATLOOP3            BRANCH IF NOT EVER POSSIBLE
BNE PATMISSM           BRANCH IF NOT YET
MVI COMMNDWK,C' '      INITIALIZE WORK AREA
MVC COMMNDWK+1(49),COMMNDWK
MVC COMMNDWK(2),4(R3)  SET TERMINATION COMMAND
MVC COMMNDWK+2(8),CHKEY TASK'S STEPNAME IS OPERAND OF KOMAND
CLI COMMNDWK,C'C'      TEST IF COMMAND IS CANCEL
BNE PATD034            BRANCH IF NOT
SPACE 1
OI CHACT,CHCL          OTHERWISE, ENSURE CANCEL COMMAND OK
MVC COMMNDWK+10(3),=C',A=' AND SET ASID
UNPK COMMNDWK+13(5),CHASID(3) UNPACK ASID
TR COMMNDWK+13(4),HEXTRAN-240 AND CONVERT TO EBCDIC
MVI COMMNDWK+17,C' '   INITIALIZE TRAILING GARBAGE
SPACE
DROP R2                FORGET CSCB
EJECT

```

```

LA R10,COMMNDWK+2 SET UP DISPLACEMENT FOR COMPRESS
LA R6,2 SET UP COUNTER FOR COMPRESS1
SPACE 1
FINDEND LA R10,1(R10) POINT TO SECOND POSITION OF APPLID
LA R6,1(R6) SET COUNTER TO CORRESPOND TO APPLID
C R6,=F'11' IS APPLID AT MAXIMUM SIZE
BE PATD034 IF IT IS THEN ISSUE COMMAND
CLI Ø(R10),C' ' ELSE CHECK FOR END OF APPLID
BNE FINDEND IF NOT THEN CONTINUE LOOP
LR R6,R10 SET R6 TO END OF APPLID
SPACE 1
COMPRES2 LA R10,1(R10) UP R10 IN SEARCH OF ',A=ØØØØ'
CLI Ø(R10),C' ' IF BLANK THEN CONTINUE TO LOOK
BE COMPRES2
MVC Ø(3Ø,R6),Ø(R10) MOVE ',A=ØØØØ' TO END OF APPLID
SPACE 1
PATD034 LA R1,3Ø+4 LENGTH OF EACH COMMAND + CONSTANT
STH R1,WTOMSG SET LENGTH IN INTERNAL COMMAND
BAS R1Ø,PATSVC34 ISSUE COMMAND
B PATLOOP3
SPACE 1
PATMISSM CLC Ø(4,R3),=X'ØØØØØØØØ' CHECK FOR END OF CHAIN
BE PATLOOP3 LOOP POWER
L R3,Ø(R3) LOAD ADDRESS OF NEXT COMMAND
B PATGETCM POINT TO NEXT JOB'SNAME
SPACE 1
FINISHCM BAS R1Ø,CMRDEQ REMOVE SERIALIZATION OF CSCB CHAIN
BAS R1Ø,CMDFREE FREE ALL COMMAND AREAS ACQUIRED
EJECT

```

```

*****
* THIS SECTION BRINGS DOWN THE REMAINING APPLICATIONS, EXCLUDING JES2,*
* ENTERED IN PAP PROCESSING BY RESPONDING APPROPRIATELY TO OUTSTANDING*
* WTORS. THE FORMAT OF PAPWTOR2 IS: 1 9 *
* WTOR-MSGRESPONSE *
*****

```

```

SPACE 1
MVC CMDPARM(16),CMDDIR17 SAVE PARM FOR BLDL DIRECTORY
ST R1,CMDR1SAV SAVE R1 TILL AFTER READCMDS
ST R2,CMDR2SAV SAVE R2 TILL AFTER READCMDS
BAS R1Ø,READCMDS BRANCH TO BUILD THIS COMMAND TABLE
L R1,CMDR1SAV RESTORE R1
L R2,CMDR2SAV RESTORE R2
L R3,CMDA1STC LOAD ADDRESS OF FIRST COMMAND
ST R3,PATVTAB STORE IT FOR ORE PROCESSING
BAS R1Ø,PATREST PAUSE FOR GRACEFUL TERMINATIONS
BAS R1Ø,PATREST ANOTHER PAUSE FOR THE CAUSE
BAS R8,PATDOORE RESPOND TO OUTSTANDING WTORS
BAS R1Ø,CMDFREE FREE ALL COMMAND AREAS ACQUIRED
EJECT

```

```

*****
* THIS SECTION BRINGS DOWN APPLICATIONS IN THE PAP PROCESS *

```

```

*   THAT CAUSE OTHER APPLICATIONS PROBLEMS TERMINATING, NAMELY           *
*   SILO AND HSM.                                                         *
*   THESE ARE KILLED JUST PRIOR TO THE ISSUANCE OF A Z NET COMMAND.      *
*****
      SPACE 1
PROBLEMS CLC   SYSID,=C'VS01' CHECK SYSTEM ID
          BE   PROBVS01                AND PAUSE THE FINAL APPLS
          CLC   SYSID,=C'VS03' CHECK SYSTEM ID
          BE   PROBVS03                AND PAUSE THE FINAL APPLS
          CLC   SYSID,=C'VS04' CHECK SYSTEM ID
          BE   PROBVS04                AND PAUSE THE FINAL APPLS
          CLC   SYSID,=C'VS05' CHECK SYSTEM ID
          BE   PROBVS05                AND PAUSE THE FINAL APPLS
      SPACE 1
          MVC   COMMNDWK(20),HSMYMM PAUSE HSMY2K ON THE Y2K DOMAIN
          BAS   R10,PATSV34          ISSUE COMMAND
          B     ZNETCANC              ENTER COMMON CODE
      SPACE 1
PROBVS01 MVC   COMMNDWK(20),HSMA    PAUSE HSMA ON DEVELOPMENT
          B     CPDOSILO              ENTER COMMON CODE
      SPACE 1
PROBVS03 MVC   COMMNDWK(20),HSMB    PAUSE HSMB ON TECHNOLOGY
          BAS   R10,PATSV34
          B     ZNETCANC
      SPACE 1
PROBVS04 MVC   COMMNDWK(20),HSMD    PAUSE HSMD ON ACCENT
          B     CPDOSILO              ENTER COMMON CODE
      SPACE 1
PROBVS05 MVC   COMMNDWK(20),HSMC    PAUSE HSMC ON PRODUCTION
      SPACE 1
CPDOSILO BAS   R10,PATSV34          ISSUE COMMAND
          BAS   R10,PATREST          TAKE A BREAK
          BAS   R10,PATREST          TAKE A BREAK
      SPACE
          MVC   COMMNDWK(20),SILOS   PAUSE SILO ON PRODUCTION
          BAS   R10,PATSV34          ISSUE COMMAND
          EJECT
*****
*   ENTER CODE TO CANCEL NET AND TERMINATE LLA AND VLF - TM TOO        *
*****

```

Editor's note: this article will be continued in the next issue.

MVS news

Sterling Software has announced Release 1.5 of its VISION:Phaseshift date-masking tool for MVS and OS/390, that insulates legacy applications from Y2K date issues without making any changes to the code.

The software dynamically encapsulates programs and data so applications with two-digit date formats never see the century roll over and so operate unchanged from 1999 into 2000. It is said to avoid all logic problems by shifting the period of operation back in time, so that all dates to be processed fall within the same century.

Phaseshift supports MVS, QSAM, VSAM, BSAM, IMS/DB, DB2, CICS, IMS/DC, and TSO and installation requires no user customization. With the Data Mapper system, staff can tell the software which applications and jobs to work with and the format and locations of the dates held in external data sources. Improvements to installation allow the mass import of data definitions for DB2, IMS, and flat files, eliminating manual data entry. Data can now be aged in any increment, including months or days.

For further information contact:
Sterling Software, 300 Crescent Court, 1200
Dallas, TX 75201-7832, USA.
Tel: 214 981 1000
Fax: 214 981 1255

Sterling Software, Sterling Court, Eastworth
Road, Chertsey, Surrey, KT16 8DF, UK.
Tel: 01932 58 7000
Fax: 01932 58 7001
<http://www.sterling.com>

Tivoli has announced new e-business management software for OS/390, promising the means to use the S/390 as the management server with service level improvement and business process view capabilities.

The new products include Tivoli Manager for OS/390, Tivoli Service Desk for OS/390 Version 1.2, enhancements to Tivoli NetView for OS/390 and Tivoli Global Enterprise Manager, and an OS/390 version of Tivoli Enterprise.

The new Global Enterprise Manager models relationships and data flow between applications, providing a single point of management for interconnected application components, middleware, and databases on platforms spanning host and distributed systems. There is new direct support for OS/390 applications including CICS, VTAM, and MQ and an application toolkit for custom-built applications. There is now a capability to view S/390 environments and existing applications from a business view perspective. It is integrated with Global Enterprise Manager and provides a single view of all business applications.

For further information contact:
Tivoli Systems, 9442 Capital of Texas
Highway, North Austin, TX 78759, USA.
Tel: 512 436 8000
Fax: 512 794 0623

Tivoli Systems, Sefton Park, Bells Hills,
Buckinghamshire, SL2 4HD, UK.
Tel: 01753 896 896
Fax: 01753 896 899
<http://www.tivoli.com>



xephon