191

MVS update

August 2002

In this issue

© Xephon plc 2002

MVS Update

Published by

Xephon
27-35 London Road
Newbury
Berkshire RG14 1JL
England
Telephone: 01635 38342
From USA: 01144 1635 38342
E-mail: trevore@xephon.com

North American office

Xephon
PO Box 350100
Westminster, CO 80035-0100
USA
Telephone: 303 410 9344

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 1999 issue, are available separately to subscribers for £29.00 ($43.50) each including postage.

MVS Update on-line

Code from *MVS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/mvs; you will need to supply a word from the printed issue.

Editor

Trevor Eddolls

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.

Contributions

When Xephon is given copyright, articles published in *MVS Update* are paid for at the rate of £170 ($260) per 1000 words and £100 ($160) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of £50 ($80) per 100 lines. In addition, there is a flat fee of £30 ($50) per article. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

© Xephon plc 2002. 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.

Copying files and members between HFS directories and PDSs

After working with Unix Systems Services for a while, I found that I needed a way to copy PDS members to HFS directories and files from HFS directories to a PDS. Unix Systems Services comes with the OCOPY, OGETX, and OPUTX commands to perform this function and the ISHELL ISPF application also provides ways to accomplish this. This is more than sufficient when one or two items need to be copied.

As I began to exploit USS more, I found that it was a convenient place to create software distribution packages for software that would be distributed on floppy disk or CD. Traditional Unix commands, like TAR and PAX, made this an easy proposition. I found that I regularly needed to copy entire PDSs to an HFS directory when the distribution contained traditional MVS components. Occasionally, I also had to refresh only certain subsets of files from the source PDSs. Using existing tools this was cumbersome for multiple copies and I wanted a batch equivalent to run unattended.

I first thought that I would write a REXX wrapper for OCOPY/OGETX/OPUTX that would satisfy my needs. This would have worked but became more difficult when I wanted to build in some bells and whistles using wildcards. Since I also wanted to move things back from the HFS directory to a PDS, my utility using OCOPY/OGETX/OPUTX became even more convoluted.

Then it struck me. I had abandoned the use of tools like IND$FILE long ago when I discovered the flexibility of MVS FTP using the MGET and MPUT commands. For those unfamiliar with FTP MGET and MPUT, this is a way to move the entire contents of a directory (or a wildcarded subset) from one host to another. When OS/390 is one of those hosts a couple of nice things happen for you. First, ASCII to EBCDIC translation can occur by using the default FTP ASCII transfer. Second, if a PDS is involved in an MGET or MPUT, the PDS members are moved to files or files are moved to PDS members (depending on which direction you are going). The only restriction is the eight-character

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

limitation on PDS member names. Since my primary direction was from PDS to HFS, this was not a problem, but would be for an MPUT from an HFS directory using file names longer than eight characters.

When using FTP with MGET or MPUT, FTP will default to prompt mode. If you are moving many files or members, it is possible to invoke FTP using the **-i** option to ignore prompting.

So, I decided to wrap this concept in a REXX EXEC for a nice self-contained utility. MVS FTP will recognize whether you are on the MVS side or USS side based on your first **cd** command. If there is a '/' in the target of the **cd** command, you are placed in the USS side; if not, you are placed in the MVS side.

Normally, you would FTP from your host to or from another host. This stumped me for a while. What am I FTPing to or from? Here is the beauty of this solution. We are FTPing to ourselves. We can accomplish everything we need using the standard TCP/IP Loopback address (127.0.0.1). Once I realized this, my XCOPY utility was born.

XCOPY is a REXX EXEC that will read an input directory or PDS and copy its contents to an output directory or PDS. This means it can copy PDS to HFS, HFS to PDS, HFS to HFS, and PDS to PDS. XCOPY can run interactively or in batch.

XCOPY parameters are:

- FROM – the source PDS or HFS directory.
- TO – the target PDS or HFS directory.
- PW – the password for the user performing the XCOPY. If run interactively, the user will be prompted if PW is missing.
- PATTERN – the wildcard pattern conforming to FTP MGET and MPUT rules. Defaulted to '*' for all members (case sensitive from HFS).
- REPLACE – 'Y' or 'N'. Defaults to 'Y'.

One thing to remember (as always) – Unix is case sensitive. Therefore, be sure to enter all HFS directory names in lower case when appropriate. If running XCOPY in batch, you may want to use CAPS OFF in your

 © 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

ISPF JCL edit session to retain lower case in the PARM string.

If XCOPY has a problem (usually a non-zero return code from FTP), it will list all the debugging information. If running under TSO/ISPF foreground, all FTP INPUT and OUTPUT dataset contents will be **say**'d to the screen. If running in background, SYSTSPRT will contain all the FTP INPUT and OUTPUT contents.

When XCOPY runs successfully under ISPF, all processing occurs and a simple ISPF message is issued with a success message and a count of the items that were copied. Pressing PF1/HELP will display the long message text, which will contain the list of members copied (until truncated, if a long list).

If you are processing a large number of items, batch is usually more appropriate. If XCOPY is running in background a report will be generated in SYSTSPRT identifying all items that were copied.

XCOPY

```
/*********************************************************************/
/*                               REXX                                */
/*********************************************************************/
/* Purpose: Copy HFS-->PDS PDS-->HFS PDS-->PDS HFS-->HFS             */
/*-------------------------------------------------------------------*/
/* Syntax:  XCOPY from to password pattern                           */
/*-------------------------------------------------------------------*/
/* Parms: from       - The 'from' directory or PDS                   */
/*        to         - The 'to' directory or PDS                     */
/*        pw         - Your password for FTP                         */
/*        pattern    - The name pattern (defaults to '*' - all)      */
/*        replace    - Replace existing files (Y or N - default Y)   */
/*                                                                   */
/* Notes: Uses FTP with the loopback address 127.0.0.1 so assumes    */
/*        you will use your ID and password.  Can also be run in     */
/*        batch.  All directories and/or PDS's must already exist.   */
/*********************************************************************/
/*                      Change Log                                   */
/*                                                                   */
/*********************************************************************/
/* Standard entry                                                    */
/*********************************************************************/
 parse upper source execenv . execname . execdsn .
 signal on syntax name trap
 signal on failure name trap
 signal on novalue name trap
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
/*********************************************************************/
/* Initialize base values for required variables                     */
/*********************************************************************/
 FTPRC = Ø
 EXITRC = Ø
 ftpcount = Ø
 ftplist = ''
 ftpenv = sysvar('SYSENV')
 x = time('r')
/*********************************************************************/
/* Start-up message                                                  */
/*********************************************************************/
 startmsg = execname 'started' date() time()
 if ftpenv = 'BACK' then
    do
     say center(' 'startmsg' ',78,'-')
     say
    end
/*********************************************************************/
/* Accept parms                                                      */
/*********************************************************************/
 parse arg from to pw pattern replace .
 if from = '' then call rcexit 999 'Source Directory or PDS is missing'
 if to = '' then call rcexit 999 'Target Directory or PDS is missing'
 if pw = '' then pw = getpw()
 if pattern = '' then pattern = '*'
 if replace = '' then replace = 'Y'
 if replace = 'Y' then replace = '(REPLACE'
/*********************************************************************/
/* Format the input and output to avoid double quoting               */
/*********************************************************************/
 from = strip(from,'B',"'")
 to = strip(to,'B',"'")
/*********************************************************************/
/* Make sure the from and to exist                                   */
/*********************************************************************/
 if pos('/',from) then
    do
    "ALLOC F($FROM) PATH('"from"')"
     call rcexit RC 'Source' from 'does not exist'
    "FREE F($FROM)"
    end
 else
    do
     from = "'"from"'"
     if sysdsn(from) <> 'OK' then
        call rcexit 12 'Source' from 'does not exist'
    end
 if pos('/',to) then
    do
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
    "ALLOC F($TO) PATH('"to"')"
     call rcexit RC 'Target' to 'does not exist'
    "FREE F($TO)"
    end
 else
    do
     to = "'"to"'"
     if sysdsn(to) <> 'OK' then
        call rcexit 12 'Target' to 'does not exist'
    end
/*********************************************************************/
/* Echo the requested action if running in background                */
/*********************************************************************/
 if ftpenv = 'BACK' then
    do
     say execname 'will copy from' from 'to' to 'using pattern' pattern
     say
    end
/*********************************************************************/
/* Setup FTP subcommands for the copy                                */
/*********************************************************************/
 input.0 = 6
 input.1 = userid() pw
 input.2 = 'cd' from
 input.3 = 'lcd' to
 input.4 = 'ascii'
 input.5 = 'mget' pattern replace
 input.6 = 'quit'
/*********************************************************************/
/* Allocate the FTP INPUT dataset                                    */
/*********************************************************************/
"ALLOC F(INPUT) NEW CATALOG TRACKS DSORG(PS) UNIT(VIO)",
"LRECL(80) BLKSIZE(0) RECFM(F B) SPACE(1 1)"
 call rcexit RC 'ALLOCATE error on VIO INPUT'
/*********************************************************************/
/* Allocate the FTP OUTPUT dataset                                   */
/*********************************************************************/
"ALLOC F(OUTPUT) NEW CATALOG CYLINDERS DSORG(PS) UNIT(VIO)",
"LRECL(80) BLKSIZE(0) RECFM(F B) SPACE(1 1)"
 call rcexit RC 'ALLOCATE error on VIO OUTPUT'
/*********************************************************************/
/* Write FTP subcommands to INPUT file                               */
/*********************************************************************/
"EXECIO * DISKW INPUT (STEM INPUT. FINIS"
 call rcexit RC 'EXECIO DISKW error in FTP INPUT'
/*********************************************************************/
/* Invoke FTP on the loopback address 127.0.0.1                      */
/*********************************************************************/
"FTP -i 127.0.0.1 (EXIT"
 FTPRC = RC
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
 call rcexit FTPRC 'FTP error on 127.0.0.1'
/*********************************************************************/
/* Read the FTP OUTPUT and tell the user what happened               */
/*********************************************************************/
"EXECIO * DISKR OUTPUT (STEM OUTPUT. FINIS"
 call rcexit RC 'EXECIO error on FTP OUTPUT reporting but FTP RC='FTPRC
/*********************************************************************/
/* Produce a listing of all items copied if running in background    */
/*********************************************************************/
 if ftpenv = 'BACK' then
    do
     say 'The following items were successfully copied:'
     say
    end
/*********************************************************************/
/* Loop through all output and parse out the names and count         */
/*********************************************************************/
 do i=1 to output.0
    parse var output.i ftpmsg . ftpcmd file .
    if ftpcmd = 'RETR' then
       do
        ftpcount = ftpcount + 1
/*********************************************************************/
/* If this is running in background, list all the members copied     */
/*********************************************************************/
        if ftpenv = 'BACK' then
           say file
        else
           ftplist = ftplist file
       end
 end
/*********************************************************************/
/* Summary completion message                                        */
/*********************************************************************/
 if ftpenv = 'FORE' then
    do
     zedsmsg = execname ftpcount 'items RC='FTPRC
     zedlmsg = execname 'from' from 'to' to 'pattern' pattern
     zedlmsg = zedlmsg ftpcount 'items RC='FTPRC 'copied:' ftplist
     address ISPEXEC "SETMSG MSG(ISRZ000)"
    end
 else
    do
     say
     say ftpcount 'items copied'
    end
/*********************************************************************/
/* Shutdown                                                          */
/*********************************************************************/
 shutdown: nop
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
/*********************************************************************/
/* If errors print the contents of the FTP INPUT (FTP Commands)      */
/*********************************************************************/
           if FTPRC <> 0 then
              do
               say
               say 'FTP input statements:'
               say
               do i=1 to input.0
                  say input.i
               end
               say
               say 'FTP output messages:'
               say
/*********************************************************************/
/* If errors print the contents of the FTP OUTPUT (FTP Messages)     */
/*********************************************************************/
              "EXECIO * DISKR OUTPUT (STEM OUTPUT. FINIS"
               do i=1 to output.0
                  say output.i
               end
              end
/*********************************************************************/
/* Cleanup the "temporary" datasets                                  */
/*********************************************************************/
           call outtrap "output", 0
          "FREE F(INPUT)"
          "FREE F(OUTPUT)"
/*********************************************************************/
/* Shutdown message                                                  */
/*********************************************************************/
           endmsg = execname 'ended' date() time() time('e') 'RC='EXITRC
           if ftpenv = 'BACK' then
              do
               say
               say center(' 'endmsg' ',78,'-')
              end
           exit(EXITRC)
/*********************************************************************/
/* Subroutines                                                       */
/*********************************************************************/
/* GETPW - Ask the user for their password (if foreground)           */
/*********************************************************************/
 getpw: if sysvar('SYSENV') = 'BACK' then
           call rcexit 913 'Missing Password'
        do while pw = '' | substr(pw,1,1) = ' '
           say 'Please enter your password'
           parse pull pw
        end
        return pw
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
/*********************************************************************/
/* RCEXIT - Exit on non-zero return codes                            */
/*********************************************************************/
 rcexit: parse arg EXITRC zedlmsg
         if EXITRC <> Ø then
            do
/*********************************************************************/
/* If execution environment is ISPF then VPUT ZISPFRC                */
/*********************************************************************/
             if execenv <> 'OMVS' then
                do
                "ISPQRY"
                 if RC = Ø then
                    do
                     zispfrc = EXITRC
                     address ISPEXEC "VPUT (ZISPFRC)"
                    end
                end
/*********************************************************************/
/* If a message is provided, wrap it in date, time and EXITRC        */
/*********************************************************************/
             if zedlmsg <> '' then
                do
                 zedlmsg = date() time() execname zedlmsg 'RC='EXITRC
/*********************************************************************/
/* If execution environment is ISPF SETMSG.  If not, say the message */
/*********************************************************************/
                 if sysvar('SYSENV') = 'BACK' | execenv = 'OMVS' then
                    say zedlmsg
                 else
                    do
                    "ISPQRY"
                     if RC = Ø then
                        address ISPEXEC "SETMSG MSG(ISRZØØØ)"
                     else
                        say zedlmsg
                    end
                end
/*********************************************************************/
/* Signal SHUTDOWN.  SHUTDOWN label MUST exist in the program        */
/*********************************************************************/
             signal shutdown
            end
         else
            return
/*********************************************************************/
/* Issue a common trap error message using rcexit                    */
/*********************************************************************/
 trap: traptype = condition('C')
       if traptype = 'SYNTAX' then
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
      msg = errortext(RC)
   else
      msg = condition('D')
   trapline = strip(sourceline(sigl))
   msg = traptype 'TRAP:' msg', Line:' sigl '"'trapline'"'
   call rcexit 666 msg
```

XCOPY JCL

```
//XCOPYJOB  JOB ......
/*JOBPARM SYSAFF=*
//*************************************************************
//* TSO BATCH JOB                                             *
//*************************************************************
//XCOPY    EXEC PGM=IKJEFT01,
//         PARM='XCOPY MY.PDS /u/my/pds mypass'
//SYSEXEC  DD   DSN=MY.EXEC.PDS,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DUMMY
```

Robert Zenuk
Systems Programmer (USA)
© Xephon 2002

Online batch

THE PROBLEM

As we all know, the main differences between batch and online programs are:

- Duration
- Complexity
- Amount of data
- Parameter dependency
- Schedule dependency.

Batch programs run much longer than online ones because of their larger complexity and demands for scheduling (not only processing but

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

data too). Also online programs deal with a limited set of data in one moment and do it at random moments of time, while batch programs deal with huge amounts of data, most often sequentially, in a limited period of time.

The aim of this article is to look at the fourth difference in the list – parameter dependency. Batch programs mostly deal with predefined parameters that are known before submitting a batch job. This can cause problems in situations where we need to change such parameters very often. If we leave the changing of the parameters to the operators there is a danger of mistakes being made, especially if there are dates or some long strings. So, a better approach is to create separate JCL, one for each set of parameters, and then schedule them appropriately. Or there is another solution, which I will describe here.

A SOLUTION

I want to show you how to create batch programs that are aware of parameter values online, which means at the time they are running, not from an operator or as predefined values. I found that the best way is to create parameter values directly in my programs. So I will need to use two passes. First I need to find out what my parameter values are, and in the next pass I will use those values for processing the data. But how will my JCL be aware of those values? Well, I will create my JCL on-the-fly, directly in the program.

The only restriction will be that I need to have several JCL. The first one is static and it has a step that will create the content of the next one, with real parameter values inside. A good approach would be to put these dynamic JCL in a separate PDS library, to avoid any accidental updates, because I must have permission to update that library from my programs.

The next technique that we can use is the usage of JCL procedures, separate members that we can call in an EXEC statement – something like a JCL module. PROCs can have parameters of their own, so we can call one PROC with different parameter values.

As I can create JCL on-the-fly, I can also create any other necessary parameter sources and some of them you will see in my examples.

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

EXAMPLES

Where do we need to change parameters for batch programs in an online manner? Well, I found the following situations.

Working with an unknown number of datasets

As a matter of fact this is the example where I used this 'online' batch technique for the first time. I had to divide the input dataset into an unknown number of datasets, named by the value of one field from the dataset record. The problem was to design a solution to work with an unknown number of output datasets, not knowing the names of these datasets before running the program. So parameters in this situation were values taken from fields in the input record.

I needed to divide this problem into two parts. First I wrote a program to process the input dataset and to produce a JCL member having as many steps as the input dataset demanded to have output datasets. Every step will be a call to a general PROC member, which will process the input dataset for one value of the parameter. The JCL looks like this:

```
//MANYOUTD JOB MSGCLASS=Z,CLASS=B,NOTIFY=&SYSUID
//*
//ESYLIB JCLLIB ORDER=(APPHLQ.UNIT.PROCLIB)
//*
//STEP1   EXEC PRONEVAL,VALUE='VALUE1',PRI=5,SEC=3
//STEP2   EXEC PRONEVAL,VALUE='VALUE2',PRI=25,SEC=5
//STEP3   EXEC PRONEVAL,VALUE='VALUE3',PRI=500,SEC=50
//...
//STEPn   EXEC PRONEVAL,VALUE='VALUEn',PRI=5,SEC=3
//...
//STEP125 EXEC PRONEVAL,VALUE='VALUE125',PRI=15,SEC=5
```

VALUE1, VALUE2, etc are synonyms for actual values of parameter VALUE.

So my first program processes the input dataset and, for every different value in a specific field, saves that value, calculates the necessary size of the output dataset, and writes one line in the JCL member which is the output for that program.

So there are as many steps as possible values in the input dataset. This can be different every time we run the program, and that's why it is online batch. Every step calls a procedure with three PROC parameters:

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

VALUE, which will be the value that we are currently workng on, and PRI and SEC as values for number of units for allocation. PRI and SEC are a helpful side effect in this example because we now know exactly what space we need for our output datasets.

In the PDS library APPHLQ.UNIT.PROCLIB, I have the following member with the name PRONEVAL (PRocess ONE VALue). It is actually my PROC, which will process the input dataset for the value of the field in the input record, given in the PROC parameter VALUE:

```
//* PROC DATE Ø111Ø1
//*****************************************************************
//* PROC FOR PROCESSING INPUT DATASET FOR ONE POSSIBLE FIELD VALUE
//*****************************************************************
//PRONEVAL PROC VALUE=,PRI=,SEC=
//*
//STEPPGM  EXEC PGM=EXTRACT,PARM='&VALUE.'
//STEPLIB  DD DSN=APPHLQ.UNIT.LOADLIB,DISP=SHR
//INPUT    DD DSN=APPHLQ.INPUT.FILE,DISP=SHR
//OUTPUT   DD DSN=APPHLQ.&VALUE..FILE,DISP=(NEW,CATLG,DELETE),
//            DCB=(LRECL=6ØØ,RECFM=FB,BLKSIZE=Ø),
//            SPACE=(CYL,(&PRI,&SEC),RLSE)
//SYSPRINT DD *
//*
// PEND
```

Program EXTRACT must have the input parameter defined through the PARM option, because this is the only way that we can pass parameters to programs using PROCs.

Sort conditions

There are few programmers who like writing report programs. From a client's point of view it is a very important part of the application – it is the way they see the data. But not all of them like to see data in the same way. Different people like the data in a different order. Some like to have data ordered by invoice number, some by product name, or in some more complex order. We satisfy different order types by sorting our datasets. But if we have many orders we will have lots of JCL with different conditions in the SORT step. Or, as I stated before, someone will need to change that every time the program runs. But what happens if we wish to give our clients the freedom to choose the sorting order themselves? They can ask for a different order every time.

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

One of the solutions is to include online parameters in our batch application. We can ask our client to define sort order in some way, which must be easy for them. Then we will have a program which will process that order, and we will produce standard SORT order statements in some dataset or member in a PDS library. Then we will use it in the SORT step. Something like this:

```
//REPORTS  JOB MSGCLASS=Z,CLASS=B,NOTIFY=&SYSUID
//************************************************
//* PRODUCING REPORTS WITH DIFFERENT ORDERS
//************************************************
//REPORT1  EXEC PGM=PRODCOND
//INPUT    DD DSN=APPHLQ.ORDER.FILE,DISP=SHR
//OUTPUT   DD DSN=APPHLQ.SORTORD.FILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
//REPORT2  EXEC SORT
//SORTIN   DD DSN=APPHLQ.INPUT.FILE,DISP=SHR
//SORTOUT  DD DSN=APPHLQ.SORTINP.FILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD DSN=APPHLQ.SORTORD.FILE,DISP=SHR
//*
//REPORT3  EXEC PGM=PRODREP
//INPUT    DD DSN=APPHLQ.SORTINP.FILE,DISP=SHR
//OUTPUT   DD DSN=APPHLQ.REPORT.FILE,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
```

In this JCL:

- PRODCOND is the program that processes the sort requirements from the client.
- PRODREP is the program that produces the report on the sorted data as the client asked.

If APPHLQ.ORDER.FILE is empty, PRODCOND produces the report using the most common order for sorting.

SENDING E-MAILS

Sending e-mails from a host can be a very useful tool for informing someone about events on the system. It can also be a standard requirement from the client to have some kind of automatic communication about ‘where’ the processing of his data is at any moment. But the recipient

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

of our e-mail can be unknown before the actual data comes to us. Actually, the e-mail address of the recipient can be part of the data that is sent by the client. The situation with our program is the same as for an online program when it gets data from an operator: it happens while the program runs, and there is no way of finding it before.

The simplest way to send e-mails from a host is to use IEBGENER:

```
//SENDMAIL JOB MSGCLASS=Z,CLASS=B,NOTIFY=&SYSUID
//************************************************
//* SENDING MAIL FROM HOST USING IEBGENER
//************************************************
//SENDMAIL EXEC PGM=IEBGENER
//SYSUT1   DD DSN=APPHLQ.HEADER.MSG,DISP=SHR
//         DD DSN=APPHLQ.BODY.MSG,DISP=SHR
//SYSUT2   DD SYSOUT=(B,SMTP)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
```

The content of APPHLQ.HEADER.MSG is something like this:

```
HELO SMTP_SERVER
MAIL FROM:<senderid@company.com>
RCPT TO:  <emailid@client.com>
DATA
SUBJECT:  MESSAGE TITLE
```

So, we can produce dataset APPHLQ.HEADER.MSG in our program for sending an e-mail, with all the relevant data. We know the SMTP_server and who the sender is, we can define a message title in the program, and we can enter the recipient's address from our input data. Our batch program will work in an online mode.

CONCLUSION

These were just some examples of how we can make batch behave as if it were online, to receive or produce input parameters, and then change the results, as the program works or just before it starts to work. Certainly there are many others and maybe you already use some. I hope this article will help you to think in a different and new way.

Predrag Jovanovic
Project Developer
Pinkerton Computer Consultants Inc (USA)

© Xephon 2002

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

A simple DFHSM report writer

PROBLEM

As is commonly known, DFHSM is a rather complex software package responsible for performing automated space management and availability in a storage media hierarchy.

In order to make sure that the DFHSM system at a given installation is working as desired, one must be aware that there is simply no substitute for frequent analysis of detailed information about the functions DFHSM performs. However, extracting, processing, and analysing this information requires experienced personnel as well as a great deal of time and effort, and there is a real danger that the DFHSM system will not receive the attention it deserves. Not having a full and comprehensive overview of DFHSM's operation could put an installation's data at risk of loss, to put it mildly.

It should be noted that there are just a few elementary reports available from within DFHSM itself, which can hardly provide any insight into its functioning. In the past few years several DFHSM add-on products have appeared on the market trying to bridge this gap. These products vary greatly in complexity, functionality, and reporting capabilities, as well as in flexibility.

SOLUTION

In order to find out what DFHSM was doing, a simple report writer was produced. This report writer tries to solve the problems mentioned above with its ability to access and to process the wealth of information DFHSM writes to SMF records. I chose to process the SMF records because the DFHSM log file is not formatted in a user-friendly way, and analysis of control datasets (CDS) is a bit difficult with the standard IBM tool set currently available. Prior to writing this tool, our storage administrator had to spend hours looking through huge DFHSM logs, in an attempt to track and document the functions and completion status of work performed by DFHSM.

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

The amount of data DFHSM collects and writes to SMF is enormous, and this report writer is an easy way to access what one needs from that data. The sets of reports it provides to technical staff give information and analysis needed to monitor and correct DFHSM operations. These reports quickly identify work that has been completed and, more importantly, not completed by DFHSM. Some of the information reported on is where the data was and where it was moved to, along with much more pertinent information.

As already said, this report writer uses SMF records produced by DFHSM. However, before using this program, one has to determine the SMF identification number related to DFHSM functional statistical records (FSR), which is the only source utilized. In a typical situation, DFHSM writes two types of SMF records: default SMF IDs are 240 for daily and volume statistic records (DSR, VSR), and 241 is reserved for functional statistics records (FSR and WWFSR). If you are not certain which ID is assigned to DFHSM, a simple **QUERY SETSYS** will tell you that:

```
ARC0150I JOURNAL={NONE | SPEED | RECOVERY}, LOG={YES | NO | HELD},
TRACE={YES | NO}, SMFID={smfid | NONE}, DEBUG={YES | NO}
```

Note that SMFID=NONE means there are no records being collected!

In general, there are twenty functional subtypes of FSR that DFHSM can write to an SMF record. What is really written depends on a particular function being performed on one dataset, so, if DFHSM's function is not utilized, one will not encounter its subtype record. When a DFHSM function is executed, only selected fields within the FSR record are set. Which fields are actually set depends on the function being performed and the method used to request the function. Function subtypes 1-14 and 17-20 are FSR records, while function subtypes 15/16 are ABARS WWFSR records and are beyond the scope of this report writer.

A detailed description of the layout of an FSR record and its fields can be obtained from the DFSMS manual *DFSMShsm Implementation and Customization Guide*.

Eleven sets of reports are produced by this report writer, each providing in-depth information on functions DFHSM performed:

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

- FSRSTAT is a DFHSM functions overall report providing information such as number of functions completed (with and without errors), various timings and timeframes, the functions performed by various parameters (by age of datasets – which is function specific but very useful for identifying thrashing conditions, by request, by date, or by one hour slot; this can help to identify tasking level or drives needed), extent reduction, expiration of datasets statistics, and thrashing analysis.
- FUNSUMM contains a set of summary reports on each function performed – as one drills down to more specific functions, statistics get more interesting and revealing.
- ERRORS is a detailed report of functions ending with a non-zero return code.
- MGMTCLAS reports describe activity of DFHSM against management classes, datasets managed by each class (by age criteria), and thrashing.
- BKRCVY provides a detailed report on backed up and recovered datasets.
- BKDELEX is a detailed report on expired and deleted incremental back-up datasets.
- MIGREC provides detailed reports on migrated and recalled datasets.
- MIGDELEX reports provide a list of deleted and expired migrated (or primary) datasets as well as a report on the partial release of unused space.
- RECYCLE reports lists of recycled tapes and datasets.
- DUMPS report is a volume dump list.
- TAPES is a report on tapes used by DFHSM's functions.

It should be noted that this report writer is not comprehensive but nevertheless is an open-ended program that allows users to modify and customize reports generated so as to meet an installation's needs or requirements. Reporting can be done on both current and historical

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

information provided that an appropriate database was created.

The report writer was written in SAS language and was kept very simple in order to maintain compatibility across various versions of SAS software. The code was tested on SAS Version 5.16 and SAS Version 6.06, using only functions of the BASE component.

PROGRAM CODE

```
//FSR241   EXEC SAS
//SYSPRINT DD SYSOUT=X
//FSRSTAT  DD SYSOUT=X
//FUNSUMM  DD SYSOUT=X
//ERRORS   DD SYSOUT=X
//MGMTCLAS DD SYSOUT=X
//BKRCVY   DD SYSOUT=X
//BKDELEX  DD SYSOUT=X
//MIGREC   DD SYSOUT=X
//MIGDELEX DD SYSOUT=X
//RECYCLE  DD SYSOUT=X
//DUMPS    DD SYSOUT=X
//TAPES    DD SYSOUT=X
//SMF      DD DSN=your.smf.file,DISP=SHR
//SYSIN  DD *
PROC FORMAT;

 VALUE FSRFMT    1='MIGRATE PR->L1 '
                 2='MIGRATE L1->L2 '
                 3='MIGRATE PR->L2 '
                 4='RECALL L1->PR  '
                 5='RECALL L2->PR  '
                 6='DELETE MIG. DS '
                 7='DAILY BACKUP   '
                 8='SPILL BACKUP   '
                 9='RECOVERY       '
                10='RECYCLE BACKVOL'
                11='DELETE BY AGE  '
                12='RECYCLE ML2    '
                13='VOLUME DUMP    '
                14='RESTORE        '
                15='ABACKUP        '
                16='ARECOVER       '
                17='EXPIRE PR/L1-2 '
                18='PARTREL        '
                19='EXPIRE INCR. BK'
                20='DELETE INCR. BK' ;

 VALUE MIGRAT   1='95'
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
                2='95'
                3='95'
                4='94'
                5='94'
                7='93'
                8='93' ;

VALUE TPT       2='98'
                3='98'
                5='97'
                7='98'
                8='98'
                9='97'
               10='97'
               12='97' ;

VALUE TIMEFMT   0='00:00 -> 00:59'
                1='01:00 -> 01:59'
                2='02:00 -> 02:59'
                3='03:00 -> 03:59'
                4='04:00 -> 04:59'
                5='05:00 -> 05:59'
                6='06:00 -> 06:59'
                7='07:00 -> 07:59'
                8='08:00 -> 08:59'
                9='09:00 -> 09:59'
               10='10:00 -> 10:59'
               11='11:00 -> 11:59'
               12='12:00 -> 12:59'
               13='13:00 -> 13:59'
               14='14:00 -> 14:59'
               15='15:00 -> 15:59'
               16='16:00 -> 16:59'
               17='17:00 -> 17:59'
               18='18:00 -> 18:59'
               19='19:00 -> 19:59'
               20='20:00 -> 20:59'
               21='21:00 -> 21:59'
               22='22:00 -> 22:59'
               23='23:00 -> 23:59' ;

VALUE AGEFMT    0='0'
                1='1'
                2='2'
                3='3'
                4='4'
                5='5'
                6-HIGH='6+';

VALUE ELAPFMT (FUZZ=.5)
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
                Ø-9='Ø-9'
                1Ø-19='1Ø-19'
                2Ø-29='2Ø-29'
                3Ø-39='3Ø-39'
                4Ø-49='4Ø-49'
                5Ø-59='5Ø-59'
                6Ø-HIGH='6Ø+' ;

 VALUE EXPFMT   1='RECOVERY REQUEST         '
                2='ML1 DS EXPIRED           '
                3='ML2 DS EXPIRED           '
                4='BACKUP VER. BEING EXPIRED'
                5='TAPE BACKUP VER. DELETED '
                6='DELETED BY EXPDT/MGTCL   ' ;

 VALUE $FMTREQ  'ØØØ'='AUTOMATIC-NOWAIT'
                'ØØ1'='AUTOMATIC-WAIT  '
                '11Ø'='TSO-NOWAIT      '
                '111'='TSO-WAIT        '
                'Ø1Ø'='BATCH-NOWAIT    '
                'Ø11'='BATCH-WAIT      ' ;

OPTIONS NOCENTER SOURCE2 LINESIZE=132;

DATA FSR (DROP = ID       MVSXA    OFFSMF MVSXAFLG
                 SYSTEM   EOFSTRNG FSRSID FSRDEVT
                 FSRABCC  FSRDARC  FSRGRP FSRRACF FSRRQN
                 FSRDATR  FSRTIMRH FSRTIMRM FSRTIMRS
                 FSRTIMS  FSRTIME  FSRTIMA
                 FSRDATE  FSRTSO   FSRUSER  FSRWAIT
                 FSRTIMSH FSRTIMSM FSRTIMSS I J X
                 FSRTIMEH FSRTIMEM FSRTIMES FSRDAT1
                 FSRTIMAH FSRTIMAM FSRTIMAS FSRXXX
                 FSRDORG  FSRRECFM FSROPTCD
                 FSRDCL2  FSRDCL3  FSRDCL4  FSRDCL5 ) ;

  INFILE SMF STOPOVER LENGTH=LENGTH COL=COL RECFM=VBS LRECL=32756
             JFCB=SMFJFCB START=BEGINCPY;
  LENGTH ID MVSXA OFFSMF 2 ;
  FORMAT
         MVSXAFLG                   HEX2.
         SMFJFCB                 $HEX2ØØ.
         SYSTEM                       $4.
  ;
  IF OFFSMF=. THEN DO;
     IF SUBSTR(SMFJFCB,1ØØ,1)='....1...'B  THEN OFFSMF=4;
     ELSE OFFSMF=Ø;
     BEGINCPY=OFFSMF+1;
     RETAIN BEGINCPY OFFSMF SYSTEM;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
END;
IF OFFSMF=4 THEN DO;
  INPUT @5 EOFSTRNG $CHAR7. @;
  IF EOFSTRNG=:'SMF EOF'
  OR EOFSTRNG=:'SMFEOF' THEN STOP;
END;
INPUT @1+OFFSMF MVSXAFLG     PIB1.
      @2+OFFSMF ID           PIB1.
     @11+OFFSMF SYSTEM         $4.
@;
MVSXA=Ø;
IF MVSXAFLG='.....1..'B THEN MVSXA=1;
IF ID=241;

INPUT @Ø11 FSRSID      $4.
      @Ø15 FSRJBN      $8.
      @Ø31 FSRUID      $8.
      @Ø39 FSRTYPE     PIB1.
      @Ø4Ø FSRFLAGS    $1.
      @Ø41 FSRDSN      $44.
      @Ø85 FSRTVOL     $6.
      @Ø91 FSRDEVT     PIB4.
      @Ø95 FSRFVOL     $6.
      @1Ø5 FSRRC       PIB4.
      @1Ø9 FSRREAS     PIB4.
      @113 FSRABCC     PIB4.
      @117 FSRDARC     PIB2.
      @119 FSRGRP      $8.
      @127 FSRRACF     IB1.
      @129 FSRRQN      IB4.
      @133 FSRDAT1     PIB1.
      @133 FSRDATR     PD4.
      @137 FSRTIMRH    PK1.
      @138 FSRTIMRM    PK1.
      @139 FSRTIMRS    PK2.2
      @141 FSRTIMSH    PK1.
      @142 FSRTIMSM    PK1.
      @143 FSRTIMSS    PK2.2
      @145 FSRTIMEH    PK1.
      @146 FSRTIMEM    PK1.
      @147 FSRTIMES    PK2.2
      @149 FSRTIMAH    PK1.
      @15Ø FSRTIMAM    PK1.
      @151 FSRTIMAS    PK2.2
      @153 FSRDLU   ?? PD4.
      @157 FSRDLM1     PIB1.
      @157 FSRDLM      PD4.
      @161 FSRBYTR     IB4.
      @165 FSRBYTW     IB4.
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
       @169 FSRTRKR     IB2.
       @171 FSRTRKW     IB2.
       @173 FSRDORG     PIB2.
       @175 FSRFLG2     PIB2.
       @176 FSRXXX      PIB2.
       @177 FSRCPU      PIB4.2
       @181 FSRAGE      IB2.
       @183 FSRRECFM    IB1.
       @184 FSROPTCD    IB1.
       @211 FSRMGTCL     $8.
       @219 FSRFLG3     PIB1.
       @237 FSRNENT1    IB2.
       @239 FSRNENT2    IB2.
       @241 FSRDCOPR   PIB2.
       @243 FSRDCOPF   PIB2.
       @245 FSRDCL1  $CHAR8.
       @253 FSRDCL2  $CHAR8.
       @261 FSRDCL3  $CHAR8.
       @269 FSRDCL4  $CHAR8.
       @277 FSRDCL5  $CHAR8.   @ ;

IF FSRTYPE LT 1 THEN DELETE;
IF FSRDAT1 < 2 THEN FSRDATR = FSRDATR + 19ØØØØØ;

IF FSRFLG2='....1...........'B
THEN INPUT
       @1Ø1 FSRDATE  ?? PD4.     @ ;
ELSE INPUT
       @1Ø1 FSRGEN      PIB4.    @ ;

IF FSRFLG2= '.1.............'B THEN FSRTSO ='1';
   ELSE                            FSRTSO ='Ø';
IF FSRFLG2= '..1............'B THEN FSRUSER='1';
   ELSE                            FSRUSER='Ø';
IF FSRFLG2= '...1...........'B THEN FSRWAIT='1';
   ELSE                            FSRWAIT='Ø';

   MATRIX= FSRTSO !! FSRUSER !! FSRWAIT;

IF FSRXXX = '...1...........'B THEN EXREDU ='YES';
   ELSE                            EXREDU ='NO';
IF FSRXXX = '....1..........'B THEN CONVER ='YES';
   ELSE                            CONVER ='NO';

IF FSRFLG3='1.......'B THEN FSREX=1;
IF FSRFLG3='.1......'B THEN FSREX=2;
IF FSRFLG3='..1.....'B THEN FSREX=3;
IF FSRFLG3='...1....'B THEN FSREX=4;
IF FSRFLG3='....1...'B THEN FSREX=5;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
IF FSRFLG3='.....1..'B THEN FSREX=6;

IF FSRNENT1 GT Ø THEN  DO I = 1 TO FSRNENT1;
   INPUT @297 TAPEVOL       $6.
         @3Ø3 FSRTFLGS     PIB1.
         @3Ø5 FSRTBYBK     PIB4.     @;
IF FSRNENT2 GT Ø AND FSRTYPE =1Ø OR FSRTYPE=12
THEN  DO J = 1 TO FSRNENT2;
   INPUT @297 TAPEVOL2      $6.
         @3Ø5 FSRTBYB2     PIB4.     @;
         FSRTVOL = TAPEVOL2 ;
         OUTBLKS=  FSRTBYB2 ;
    END;
  END ;

IF FSRDAT1 < 2 THEN FSRDATR = FSRDATR + 19ØØØØØ;

FUNCTION=PUT(FSRTYPE,FSRFMT.);
TIME    =PUT(FSRTIMRH,TIMEFMT.);
AGE     =PUT(FSRAGE,AGEFMT.);
REQ     =PUT(MATRIX,$FMTREQ.);
EXPIRE  =PUT(FSREX,$EXPFMT.);

DATE=INPUT(PUT(FSRDATR,7.),JULIAN7.);
FORMAT DATE DATE. ;

FSRTIMR=HMS(FSRTIMRH,FSRTIMRM,FSRTIMRS);
FORMAT FSRTIMR TIME8. ;
FSRTIMS=HMS(FSRTIMSH,FSRTIMSM,FSRTIMSS);
FORMAT FSRTIMS TIME8. ;
FSRTIME=HMS(FSRTIMEH,FSRTIMEM,FSRTIMES);
FORMAT FSRTIME TIME8. ;
FSRTIMA=HMS(FSRTIMAH,FSRTIMAM,FSRTIMAS);
FORMAT FSRTIMA TIME8. ;

IF FSRTIMS GE FSRTIMR THEN DELAY = FSRTIMS - FSRTIMR;
   ELSE DELAY = FSRTIMS + '24.ØØ.ØØ'T - FSRTIMR;
   FORMAT DELAY TIME8. ;

IF FSRTIMA GE FSRTIMS THEN MOUNT = FSRTIMA - FSRTIMS;
   ELSE MOUNT = FSRTIMA + '24.ØØ.ØØ'T - FSRTIMS;
   FORMAT MOUNT TIME8. ;

IF FSRTIME GE FSRTIMR THEN DURATION = FSRTIME - FSRTIMR;
   ELSE DURATION = FSRTIME + '24.ØØ.ØØ'T - FSRTIMR;
   ELAPS   =PUT(DURATION,ELAPFMT.);
   FORMAT DURATION TIME8. ;

IF FSRTIMA GE FSRTIMR THEN PENDING = FSRTIMA - FSRTIMR;
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
   ELSE PENDING  = FSRTIMA + '24.00.00'T - FSRTIMR;
   FORMAT PENDING TIME8. ;
   X = DURATION - PENDING;
   FORMAT X 5.;
   XP   =PUT(X,ELAPFMT.);

IF FSRRC EQ 0 THEN RETCODE='OK   ';
   ELSE RETCODE='ERROR' ;

IF FSRMGTCL='        ' THEN FSRMGTCL='NON SMS ';

LABEL
    FSRAGE  ='DAYS SINCE*LAST*REF'
    FSRDLM  ='TIME LAST*MOVED'
    FSRDLU  ='LAST REF*DATE'
    FSRRC   ='RETURN*CODE'
    FSRREAS ='REASON*CODE'
    FSRTIMR ='REQUEST*TIME'
    FSRFVOL ='FROM*VOLUME'
    FSRTVOL ='TO*VOLUME'
    FSRDSN  ='DATASET*NAME'
    FSRMGTCL='MANAGEMENT*CLASS'
    FSRCPU  ='CPU TIME*USED'
    FSRTIMR ='TIME OF*REQUEST'
    DATE    ='DATE OF*REQUEST'
    FSRDCL1 ='DUMP*CLASS'
    FSRDCOPF='DUMP*COPIES*FAILED'
    FSRDCOPR='DUMP*COPIES*REQUESTED'
    FSRGEN  ='BACKUP*GEN*NUMBER'
    TAPEVOL ='TAPE VOLUME*SERIAL'
    FSRNENT1='NUMBER TAPES*USED'
    FUNCTION='DFHSM*FUNCTION'
    DATE    ='DATE OF*REQUEST'
    DURATION='ELAPSED*TIME'
    RETCODE ='RETURN*CODE'  ;

PROC SORT DATA=FSR; BY FSRTYPE RETCODE;

 *----  OVERALL DFHSM FUNCTION SUMMARY  ----;

PROC SUMMARY DATA=FSR NWAY;
  CLASS RETCODE;
  BY FSRTYPE;
  VAR DURATION PENDING FSRCPU;
  OUTPUT OUT=K
  MEAN(DURATION PENDING FSRCPU)= AVGRESP AVGPEND AVGCPU
  N=COUNT ;

DATA SUMMARY; SET K;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
  FUNCTION=PUT(FSRTYPE,FSRFMT.);
LABEL COUNT   ='TIMES*FUNCTION*EXECUTED'
      AVGRESP ='AVG*ELAPSED*TIME'
      AVGPEND ='AVG*PENDING*TIME'
      FUNCTION='DFHSM*FUNCTION'
      AVGCPU  ='AVG*CPU*TIME';

PROC PRINTTO PRINT=FSRSTAT; OPTIONS PAGENO=1;

PROC PRINT DATA=SUMMARY UNIFORM NOOBS SPLIT="*";
       VAR FUNCTION RETCODE COUNT AVGRESP AVGPEND AVGCPU;
       FORMAT AVGRESP AVGPEND AVGCPU TIME8. ;
       TITLE1"DFHSM FUNCTION OVERALL REPORT";

 *----- FUNCTIONS IN ERROR: RC > Ø----;

DATA ERROR(KEEP=DATE  FSRDSN  FSRTYPE FUNCTION FSRAGE
               FSRRC FSRREAS FSRTVOL FSRFVOL  FSRTIMR);
SET FSR; IF FSRRC NE Ø;

PROC SORT DATA=ERROR NODUP; BY DATE FSRTIMR FSRTYPE;
PROC PRINTTO PRINT=ERRORS; OPTIONS PAGENO=1;

PROC PRINT DATA=ERROR UNIFORM NOOBS SPLIT="*";
TITLE1"DFHSM FUNCTION ERRORS DETAILS REPORT";
VAR FSRTIMR FUNCTION FSRFVOL FSRTVOL
    FSRRC   FSRREAS  FSRDSN ;
    BY DATE;

DATA OK; SET FSR; IF FSRRC GT Ø THEN DELETE;

 *---------- MIGRATION ------------;

DATA MIG; SET OK;
MIG=PUT(FSRTYPE,MIGRAT.);
IF MIG='95';
TP=PUT(FSRTYPE,TPT.);
IF TP='98' THEN FSRTVOL=TAPEVOL ;

IF FSRFLAGS = '...1....'B THEN KBREAD = FSRBYTR;
   ELSE KBREAD = (FSRBYTR / 1Ø24);
IF FSRTFLGS =  '1.......'B THEN KBWRITE = FSRTBYBK;
   ELSE KBWRITE = (FSRTBYBK / 1Ø24);

IF FSRTYPE = 1 THEN DO;
  IF FSRFLAGS = '...1....'B THEN KBWRITE= FSRBYTW;
     ELSE KBWRITE= (FSRBYTW / 1Ø24);
  END;

MBREAD=(KBREAD/1Ø24);  MBWRITE=(KBWRITE/1Ø24);
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
TOTKB=KBREAD+KBWRITE;  COMPR=KBREAD/KBWRITE;
XFERTIME=DURATION - PENDING;
KBSEC=TOTKB/XFERTIME;  TOTMB=(KBREAD+KBWRITE)/1024;

IF FSRDLM1 < 2 THEN FSRDLM = FSRDLM + 19000000;
   DLM  = INPUT(PUT(FSRDLM,7.),JULIAN7.);

LABEL FSRAGE ='DAYS*ON*PRIMARY/ML1'
      DLM    ='MIGRATE*DATE'
      FSRTIMR='MIGRATE*TIME'
      KBREAD ='DS SIZE*IN KB'
      KBSEC  ='TRANSFER*RATE'
      COMPR  ='COMPRESS*RATE';

 *----  MIGRATION DATAIL REPORT  ----;

PROC SORT DATA=MIG NODUP;
     BY DATE FSRTIMR FUNCTION;
PROC PRINTTO PRINT=MIGREC;  OPTIONS PAGENO=1;

PROC PRINT DATA=MIG UNIFORM NOOBS SPLIT="*";
TITLE1"DFHSM MIGRATION ACTIVITY";
BY DLM;
FORMAT DLM DATE.;
VAR FSRTIMR FUNCTION FSRAGE  FSRMGTCL
    KBREAD  COMPR    FSRFVOL FSRTVOL FSRDSN;
FORMAT FSRCPU TIME11.;

 *----  MIGRATION SUMMARY  REPORT ----;

PROC SUMMARY DATA=MIG NWAY;
     CLASS FUNCTION DATE;
VAR DURATION PENDING TOTMB MBREAD COMPR KBSEC FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING TOTMB MBREAD COMPR KBSEC FSRCPU)=
     AVGRESP AVGPEND AVGMB AVGSIZE AVGCOMP AVGKBSEC AVGCPU
N=COUNT ;

PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC SORT DATA=S; BY DATE FUNCTION;

PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
TITLE1"DFHSM MIGRATION ACTIVITY - SUMMARY";
ID DATE FUNCTION;
VAR COUNT AVGRESP AVGPEND AVGCPU
    AVGMB AVGSIZE AVGCOMP AVGKBSEC;
FORMAT AVGRESP AVGPEND AVGCPU 6.2  ;
 LABEL  AVGMB   ='AVG MB*XFERED*(READ + WRITE)'
        FUNCTION='DFHSM*FUNCTION'
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
        DATE    ='DATE OF*REQUEST'
        AVGSIZE ='AVG DS*SIZE*IN MB'
        AVGCOMP ='AVG*COMPRESS*RATIO'
        AVGCPU  ='AVG*CPU*TIME(SEC)'
        AVGRESP ='AVG*ELAPSED*TIME(SEC)'
        AVGPEND ='AVG*PENDING*TIME(SEC)'
        AVGKBSEC='AVG*TRANSFER RATE*(KB/SEC)'
        FUNCTION='DFHSM*FUNCTION'
        COUNT   ='DATASETS*PROCESSED' ;

 *--- DELETION OF MIGRATED DATASETS ---;

DATA DELMIG; SET OK;
IF FSRTYPE =6 ;
IF FSRNENT1 GT Ø THEN  FSRFVOL =TAPEVOL;
IF FSRTFLGS =  '1.......'B THEN KBFREE = FSRTBYBK;
   ELSE KBFREE = (FSRTBYBK / 1Ø24);

IF FSRFLAGS = '...1....'B THEN KFREE= FSRBYTR;
   ELSE KFREE = (FSRBYTR / 1Ø24);
IF KFREE EQ Ø THEN KFREE=KBFREE;
 LABEL  KFREE ='KB FREED' ;

 *--- DELETION OF MIGRATED DATASETS DETAIL REPORT ---;

PROC SORT DATA=DELMIG NODUP; BY DATE FSRTIMR FSRTYPE;
PROC PRINTTO PRINT=MIGDELEX; OPTIONS PAGENO=1;
PROC PRINT DATA=DELMIG UNIFORM NOOBS SPLIT="*";
VAR FSRTIMR FUNCTION  FSRFVOL  KFREE
    FSRAGE  FSRMGTCL  FSRDSN ;
BY DATE;
TITLE1"DELETE MIGRATED DATASETS  - DETAIL";

 *--- DELETION OF MIGRATED DATASETS SUMMARY REPORT ---;

PROC SUMMARY DATA=DELMIG NWAY;
     CLASS FUNCTION DATE;
VAR DURATION PENDING KFREE FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING KFREE FSRCPU)=
     AVGRESP AVGPEND AVGFR AVGCPU
SUM(KFREE)= TOTFR
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
TITLE1"DELETE MIGRATED DATASETS  - SUMMARY";
VAR DATE FUNCTION COUNT AVGFR TOTFR AVGRESP AVGPEND AVGCPU;
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
FORMAT AVGRESP AVGPEND AVGCPU 5.2 ;
LABEL  AVGRESP ='AVG*ELAPSED*TIME(SEC)'
       AVGPEND ='AVG*PENDING*TIME(SEC)'
       AVGCPU  ='AVG*CPU*TIME(SEC)    '
       FUNCTION='DFHSM*FUNCTION'
       DATE    ='DATE OF*REQUEST'
       COUNT   ='DATASETS*PROCESSED'
       TOTFR   ='TOTAL KB*FREED'
       AVGFR   ='AVG*KB FREED' ;

 *---- EXPIRATION OF PRIMARY OR MIGRATED DATASETS ---;

DATA EXPMIG; SET OK;
IF FSRTYPE =17;
IF FSRNENT1 GT Ø THEN  FSRFVOL =TAPEVOL;

IF FSRTFLGS =  '1.......'B THEN KBFREE = FSRTBYBK;
   ELSE KBFREE = (FSRTBYBK / 1Ø24);
IF FSRFLAGS = '...1....'B THEN KFREE= FSRBYTR;
   ELSE KFREE = (FSRBYTR / 1Ø24);
K = 55;        /* DASD TRACK SIZE IN KB - DEVICE DEPENDENT */
IF FSRTRKR GE Ø THEN KFREE=(FSRTRKR*K);
IF KFREE EQ Ø THEN KFREE=KBFREE;
 LABEL  KFREE  ='KB FREED'
        EXPIRE ='DATASET*TYPE' ;

 *--- EXPIRATION OF PRIMARY/MIGRATED DATASETS DETAIL REPORT ---;

PROC SORT DATA=EXPMIG NODUP; BY DATE FSRTIMR EXPIRE;
PROC PRINTTO PRINT=MIGDELEX; OPTIONS PAGENO=1;
PROC PRINT DATA=EXPMIG UNIFORM NOOBS SPLIT="*";
     VAR FSRTIMR EXPIRE  FSRFVOL
         KFREE FSRAGE  FSRMGTCL FSRDSN ;
BY DATE;
TITLE1"EXPIRED PRIMARY/MIGRATED DATASETS  - DETAIL";

PROC SUMMARY DATA=EXPMIG NWAY;
     CLASS EXPIRE DATE;
VAR DURATION PENDING FSRAGE KFREE FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING FSRAGE KFREE FSRCPU)=
     AVGRESP AVGPEND AVGAGE AVGFR AVGCPU
SUM(KFREE)= TOTFR
N=COUNT ;

 *--- EXPIRATION OF PRIMARY/MIGRATED DATASETS SUMMARY REPORT ---;

PROC SORT DATA=S; BY DATE EXPIRE;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
```

 © 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
PROC PRINT DATA=S UNIFORM NOOBS SPLIT="*";
TITLE1"EXPIRED PRIMARY/MIGRATED DATASETS - SUMMARY";
VAR DATE EXPIRE COUNT AVGFR TOTFR AVGAGE AVGRESP AVGPEND AVGCPU;
 FORMAT AVGRESP AVGPEND AVGCPU 5.2 ;
 FORMAT AVGAGE 3. ;
 LABEL EXPIRE ='DATASET*TYPE'
       COUNT  ='DATASETS*PROCESSED'
       TOTFR  ='TOTAL KB*FREED'
       AVGFR  ='AVG*KB FREED'
       AVGAGE ='AVG AGE*OF DS'
       AVGCPU ='AVG*CPU*TIME'
       DATE   ='DATE OF*REQUEST'
       AVGRESP='AVG*ELAPSED*TIME(SEC)'
       AVGPEND='AVG*PENDING*TIME(SEC)';

 *----- PARTIAL RELEASE -------;

DATA PAR; SET OK;
IF FSRTYPE =18;
LABEL FSRTRKR  ='TRACKS*FREED';

PROC SORT DATA=PAR; BY DATE FUNCTION;

 *----- PARTIAL RELEASE DETAILED REPORT -------;

PROC PRINTTO PRINT=MIGDELEX; OPTIONS PAGENO=1;
PROC PRINT DATA=PAR UNIFORM NOOBS SPLIT="*";
 VAR FSRTIMR FUNCTION FSRAGE
     FSRTRKR FSRMGTCL FSRFVOL FSRDSN ;
 BY DATE;
TITLE1"PARTIAL RELEASE OF UNUSED SPACE - DETAILED" ;

PROC SUMMARY DATA=PAR  NWAY;
     CLASS FUNCTION DATE;
VAR FSRTRKR;
OUTPUT OUT=S MIN(FSRTRKR) = MINTRK
             MEAN(FSRTRKR) = AVGTRK
              MAX(FSRTRKR) = MAXTRK
              SUM(FSRTRKR) = SUMTRK
N=COUNT ;

 *----- PARTIAL RELEASE SUMMARY REPORT -------;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
 VAR DATE FUNCTION COUNT SUMTRK MINTRK AVGTRK MAXTRK;
TITLE1"PARTIAL RELEASE OF UNUSED SPACE - SUMMARY";
 LABEL COUNT  ='DATASETS*PROCESSED'
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
       SUMTRK ='TOTAL TRACKS*FREED'
       DATE   ='DATE OF*REQUEST'
       FUNCTION='DFHSM*FUNCTION'
       MINTRK  ='MIN TRACKS*FREED'
       AVGTRK  ='AVG TRACKS*FREED'
       MAXTRK  ='MAX TRACKS*FREED';

 *----- RECALLING MIGRATED DATASETS -----;

DATA RECALL; SET OK;
MIG=PUT(FSRTYPE,MIGRAT.);
IF MIG='94';
TP=PUT(FSRTYPE,TPT.);
IF TP='97' THEN FSRFVOL=TAPEVOL ;

IF FSRFLAGS = '...1....'B THEN KBWRITE = FSRBYTW;
   ELSE KBWRITE = (FSRBYTW / 1024);

IF FSRTYPE = 4 THEN DO;   /* L1--PR  */
   IF FSRFLAGS = '...1....'B THEN KBREAD = FSRBYTR;
   ELSE KBREAD = (FSRBYTR / 1024);
   END;

IF FSRTYPE = 5 THEN DO;   /* L2---PR */
   IF FSRTFLGS = '1.......'B THEN KBREAD = FSRTBYBK;
   ELSE KBREAD = (FSRTBYBK / 1024);
   END;

IF FSRDLM1 < 2 THEN FSRDLM = FSRDLM + 1900000;
   DLM  = INPUT(PUT(FSRDLM,7.),JULIAN7.);

MBREAD=(KBREAD/1024);   MBWRITE=(KBWRITE/1024);
TOTKB=KBREAD+KBWRITE;   COMPR=KBWRITE/KBREAD;
XFERTIME=DURATION - PENDING;
KBSEC=TOTKB/XFERTIME;   TOTMB=(KBREAD+KBWRITE)/1024;
LABEL KBWRITE='DS SIZE*IN KB'
      KBSEC  ='TRANSFER*RATE'
      FSRTIMR='RECALL*TIME'
      FSRAGE ='DAYS ON*ML1/ML2';

PROC SORT DATA=RECALL NODUP; BY DATE FUNCTION;

 *----  RECALL DETAIL   ----;

PROC PRINTTO PRINT=MIGREC; OPTIONS PAGENO=1;
PROC PRINT DATA=RECALL UNIFORM NOOBS SPLIT="*" ;
TITLE1"DFHSM RECALL ACTIVITY";
VAR FSRTIMR FUNCTION FSRAGE  FSRMGTCL
    KBWRITE KBSEC    FSRFVOL FSRTVOL FSRDSN ;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
BY DATE;

PROC SORT DATA=RECALL NODUP;
     BY DATE FSRTIMR FUNCTION;

 *----  RECALL SUMMARY  ----;

PROC SUMMARY DATA=RECALL NWAY;
     CLASS FUNCTION DATE;
VAR DURATION PENDING TOTMB MBWRITE COMPR KBSEC FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING TOTMB MBWRITE COMPR KBSEC FSRCPU)=
     AVGRESP AVGPEND AVGMB AVGSIZE AVGCOMP AVGKBSEC AVGCPU
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
TITLE1"DFHSM RECALL ACTIVITY - SUMMARY";
VAR DATE  FUNCTION COUNT   AVGRESP AVGPEND AVGCPU
    AVGMB AVGSIZE  AVGCOMP AVGKBSEC;
 FORMAT AVGRESP AVGPEND AVGCPU 6.2 ;
 LABEL  AVGMB   ='AVG MB*XFERED*(READ + WRITE)'
        AVGSIZE ='AVG DS*SIZE*IN MB           '
        AVGCOMP ='AVG*COMPRESS*RATIO          '
        AVGCPU  ='AVG*CPU*TIME                '
        DATE    ='DATE OF*REQUEST'
        FUNCTION='DFHSM*FUNCTION'
        COUNT   ='DATASETS*PROCESSED'
        AVGKBSEC='AVG*TRANSFER RATE*(KB/SEC)'
        AVGRESP='AVG*ELAPSED*TIME(SEC)'
        AVGPEND='AVG*PENDING*TIME(SEC)';

 *---------- FSR STATISTICS ---- -----;

PROC PRINTTO PRINT=FSRSTAT; OPTIONS PAGENO=1;

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES ELAPS/NOCUM;
TITLE1"FSR RECORDS BY TIME COMPLETED (SECONDS)";
PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES FUNCTION*ELAPS/NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY FUNCTION TYPE AND COMPLETED TIME (SECONDS)";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES XP/NOCUM;
TITLE1"FSR RECORDS BY TIME WITHOUT DELAY (SECONDS)";
PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES FUNCTION*XP/NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY FUNCTION TYPE AND EFFECTIVE TIME (SECONDS)";
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES REQ/NOCUM;
TITLE1"FSR RECORDS BY REQUEST";
PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES FUNCTION*REQ/NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY FUNCTION - REQUEST";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES AGE/NOCUM;
TITLE1"FSR RECORDS BY DATASET AGE";
PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES FUNCTION*AGE /NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY FUNCTION AND DATASET AGE";

PROC FREQ DATA=OK ;
     TABLES DATE / NOCUM;
FORMAT DATE WEEKDATE.;
TITLE1"FSR RECORDS BY DATE";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES TIME/NOCUM;
TITLE1"FSR RECORDS BY 1 HOUR SLOT";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES EXREDU/NOCUM;
TITLE1"FSR RECORDS BY EXTENT REDUCTION";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES EXPIRE/NOCUM;
TITLE1"FSR RECORDS BY EXPIRE";

PROC PRINTTO PRINT=MGMTCLAS; OPTIONS PAGENO=1;
PROC FREQ DATA=OK;
     TABLES FSRMGTCL/MISSING NOCUM;
TITLE1"FSR RECORDS BY MGMT CLASS";

PROC FREQ DATA=OK;
     TABLES FUNCTION*FSRMGTCL/MISSING NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY MGMT - FSRTYPE";

PROC FREQ DATA=OK ORDER=FORMATTED;
     TABLES FSRMGTCL*AGE/MISSING NOFREQ NOPERCENT NOCUM;
TITLE1"FSR RECORDS BY MGMT CLASS - DATASET AGE";

 *------ RECYCLING ----------;

DATA RECY; SET OK;
IF FSRTYPE =1Ø OR FSRTYPE =12 ;
   XFERTM  =DURATION - PENDING;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
   FSRFVOL =TAPEVOL;       TOTKB =FSRTBYBK*16;
   MBREAD  =(TOTKB/1Ø24);  KBSEC   =TOTKB/XFERTM;

PROC SUMMARY DATA=RECY NWAY;
     CLASS FUNCTION DATE;
VAR DURATION PENDING MBREAD KBSEC FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING MBREAD KBSEC FSRCPU)=
     AVGRESP AVGPEND AVGSIZE AVGKBSEC AVGCPU
N=COUNT ;

 *----  RECYCLE SUMMARY  ----;

PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
TITLE1"DFHSM RECYCLE ACTIVITY - SUMMARY";
VAR DATE FUNCTION COUNT AVGRESP AVGPEND AVGCPU
    AVGSIZE AVGKBSEC;
 FORMAT AVGRESP AVGPEND AVGCPU 7.2 ;
 LABEL  AVGSIZE ='AVG DS*SIZE*IN MB'
        AVGCPU  ='AVG*CPU*TIME'
        COUNT   ='DATASETS*PROCESSED'
        AVGKBSEC='AVG*TRANSFER RATE*(KB/SEC)'
        AVGRESP ='AVG*ELAPSED*TIME(SEC)'
        AVGPEND ='AVG*PENDING*TIME(SEC)';

 *----  RECYCLE DETAIL   ----;

PROC SORT DATA=RECY; BY DATE FSRTIMR;
PROC PRINTTO PRINT=RECYCLE; OPTIONS PAGENO=1;
PROC PRINT DATA=RECY UNIFORM NOOBS SPLIT="*";
TITLE1"DFHSM RECYCLE ACTIVITY - DETAIL";
VAR FSRTIMR FUNCTION FSRCPU
    TOTKB XFERTM  TAPEVOL FSRAGE FSRDSN ;
BY DATE;
LABEL DATE   ='RECYCLE*DATE '
      FSRTIMR='RECYCLE*TIME '
      FSRCPU ='CPU TIME*USED'
      TAPEVOL='RECYCLED*VOLUME'
      TOTKB  ='KB READ'
      XFERTM ='XFER TIME*(IN SEC)'
      KBSEC  ='XFER RATE*(KB/SEC)' ;

 *------ BACKUP: DAILY & SPILL ------;

DATA BACKUP; SET OK;
MIG=PUT(FSRTYPE,MIGRAT.);
IF MIG='93';                        /* BACKUP: DAILY, SPILL */
TP=PUT(FSRTYPE,TPT.);
IF TP='98' THEN FSRTVOL=TAPEVOL ;   /* BACKUP TAPEOUT */
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
IF FSRFLAGS = '...1....'B THEN KBREAD = FSRBYTR;
   ELSE KBREAD = (FSRBYTR / 1Ø24);
IF FSRTFLGS =  '1.......'B THEN KBWRITE = FSRTBYBK;
   ELSE KBWRITE = (FSRTBYBK / 1Ø24);

MBREAD=(KBREAD/1Ø24);   MBWRITE=(KBWRITE/1Ø24);
TOTKB=KBREAD+KBWRITE;   COMPR=KBREAD/KBWRITE;
XFERTIME=  DURATION - PENDING;
KBSEC=TOTKB/XFERTIME;   TOTMB=(KBREAD+KBWRITE)/1Ø24;

PROC SORT DATA=BACKUP NODUP ; BY DATE FSRTIMR FUNCTION;

DATA NOTBK; SET BACKUP; IF KBWRITE GT Ø THEN DELETE;
DATA BACK; SET BACKUP; IF KBWRITE=.    THEN DELETE;
IF KBWRITE GT KBREAD THEN
   KBWASTE = KBWRITE-KBREAD;
   BLOCKS=KBWRITE/16;
LABEL DATE   ='BACKUP*DATE'
      FSRTIMR='BACKUP*TIME '
      FSRCPU ='CPU TIME*USED'
      KBREAD ='DATASET*SIZE(KB)'
      KBWASTE='KB WASTED'
      BLOCKS ='16K OUTPUT*BLOCKS*WRITTEN'
      KBSEC  ='XFER RATE*(KB/SEC)' ;

 *----  BACKUP DETAIL    ----;

PROC SORT DATA=BACK; BY DATE FSRTIMR;
PROC PRINTTO PRINT=BKRCVY; OPTIONS PAGENO=1;
PROC PRINT DATA=BACK UNIFORM NOOBS SPLIT="*";
     VAR FSRTIMR FUNCTION FSRGEN FSRFVOL FSRTVOL
         KBREAD  KBWASTE  BLOCKS KBSEC   FSRDSN ;
     BY DATE;
TITLE1"BACKED UP DATASETS - DETAILED REPORT ";

 *----  BACKUP  SUMMARY  ----;

PROC SUMMARY DATA=BACK NWAY;
     CLASS FUNCTION DATE;
VAR DURATION KBWASTE TOTMB BLOCKS COMPR KBSEC FSRCPU;
OUTPUT OUT=S
MEAN(DURATION TOTMB BLOCKS COMPR KBSEC FSRCPU)=
     AVGRESP AVGMB  AVGBLK AVGCOMP AVGKBSEC AVGCPU
SUM(KBWASTE)=TOTWAST
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
     VAR DATE FUNCTION COUNT AVGRESP AVGCPU
         AVGMB TOTWAST AVGBLK AVGCOMP AVGKBSEC;
     FORMAT AVGRESP AVGCPU 7.2 ;
     FORMAT AVGMB 5.1;
LABEL  AVGMB   ='AVG MB*XFERED*(READ + WRITE)'
       COUNT   ='DATASETS*PROCESSED'
       AVGRESP ='AVG*ELAPSED*TIME(SEC)'
       AVGCPU  ='AVG*CPU*TIME'
       TOTWAST ='TOTAL*KB WASTED'
       AVGBLK  ='AVG NO.OF*16 KB BLOCKS*WRITTEN'
       AVGCOMP ='AVG*COMPRESS*RATIO'
       AVGKBSEC='AVG*TRANSFER RATE*(KB/SEC)' ;
TITLE1"DFHSM BACKUP ACTIVITY - SUMMARY";

PROC SORT DATA=NOTBK; BY DATE FSRTIMR;
PROC PRINTTO PRINT=BKRCVY; OPTIONS PAGENO=1;
PROC PRINT DATA=NOTBK UNIFORM NOOBS SPLIT="*";
     VAR FSRTIMR FUNCTION FSRAGE FSRDSN ;
     BY DATE;
TITLE1"DATASETS NOT BACKED UP ";

 *--- EXPIRED INCREMENTAL BACKUP DATASETS ----;

DATA EXPBK; SET OK;
IF FSRTYPE =19;
IF FSRNENT1 GT Ø THEN  FSRFVOL =TAPEVOL;

 *--- EXPIRED INCREMENTAL BACKUP DATASETS DATAIL ----;

PROC SORT DATA=EXPBK  NODUP; BY DATE FSRTIMR EXPIRE;
PROC PRINTTO PRINT=BKDELEX; OPTIONS PAGENO=1;
PROC PRINT DATA=EXPBK UNIFORM NOOBS SPLIT="*";
   VAR FSRTIMR FSRFVOL   FSRGEN
       FSRAGE  FSRMGTCL  FSRDSN  ;
   BY DATE;
TITLE1"EXPIRED INCREMENTAL BACKUP DATASETS  - DETAIL";

PROC SUMMARY DATA=EXPBK  NWAY;
     CLASS FUNCTION DATE;
VAR FSRAGE ;
OUTPUT OUT=S  MIN(FSRAGE) = MINAGE
             MEAN(FSRAGE) = AVGAGE
              MAX(FSRAGE) = MAXAGE
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
 VAR DATE FUNCTION COUNT MINAGE AVGAGE MAXAGE;
 TITLE1"EXPIRED INCLEMENTAL BACKUP DATASETS - SUMMARY";
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
 LABEL COUNT   ='DATASETS*PROCESSED'
       MINAGE  ='MINIMAL*DS AGE'
       AVGAGE  ='AVG AGE'
       MAXAGE  ='MAXIMAL*DS AGE';

 *---- DELETED INCREMENTAL BACKUP DATASETS ----;

DATA DELBK; SET OK;
IF FSRTYPE =2Ø;
IF FSRNENT1 GT Ø THEN  FSRFVOL =TAPEVOL;

PROC SORT DATA=DELBK  NODUP; BY DATE FSRTIMR EXPIRE;
PROC PRINTTO PRINT=BKDELEX; OPTIONS PAGENO=1;
PROC PRINT DATA=DELBK UNIFORM NOOBS SPLIT="*";
     VAR FSRTIMR FSRGEN FSRAGE FSRFVOL FSRMGTCL FSRDSN ;
     BY DATE;
TITLE1"DELETED INCREMENTAL BACKUP DATASETS  - DETAIL";

PROC SUMMARY DATA=DELBK  NWAY;
     CLASS FUNCTION DATE;
VAR FSRAGE  ;
OUTPUT OUT=S    MIN(FSRAGE) = MINAGE
                MEAN(FSRAGE) = AVGAGE
                MAX(FSRAGE) = MAXAGE
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
 VAR DATE FUNCTION COUNT MINAGE AVGAGE MAXAGE;
 LABEL COUNT   ='DATASETS*PROCESSED'
       MINAGE  ='MINIMAL*DS AGE'
       AVGAGE  ='AVG AGE'
       MAXAGE  ='MAXIMAL*DS AGE';
TITLE1"DELETED INCREMENTAL BACKUP DATASETS - SUMMARY";

 *-- RECOVERING BACKUP DATASETS -----;

DATA RECOV; SET OK;
IF FSRTYPE = 9;                      /* RECOVERY */
TP=PUT(FSRTYPE,TPT.);
IF TP='97' THEN FSRFVOL=TAPEVOL ;    /* RECOVERY TAPEIN */

IF FSRFLAGS = '...1....'B THEN KBWRITE= FSRBYTW;
   ELSE KBWRITE = (FSRBYTW / 1Ø24);
IF FSRTFLGS =  '1.......'B THEN KBREAD = FSRTBYBK;
   ELSE KBREAD = (FSRTBYBK / 1Ø24);

MBREAD=(KBREAD/1Ø24);  MBWRITE=(KBWRITE/1Ø24);
TOTKB=KBREAD+KBWRITE;  COMPR=KBREAD/KBWRITE;
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
XFERTIME= DURATION - PENDING;
KBSEC=TOTKB/XFERTIME;  TOTMB=(KBREAD+KBWRITE)/1Ø24;

PROC SORT DATA=RECOV NODUP ; BY DATE FSRTIMR FUNCTION;

PROC PRINTTO PRINT=BKRCVY; OPTIONS PAGENO=1;
PROC PRINT DATA=RECOV UNIFORM NOOBS SPLIT="*";
     VAR FSRTIMR  FSRAGE  FSRFVOL FSRTVOL
         FSRMGTCL KBWRITE KBSEC   FSRDSN ;
     BY DATE;
     LABEL  KBWRITE ='DATASET*SIZE(KB)'
            KBSEC   ='XFER RATE*(KB/SEC)' ;
TITLE1"RECOVERED DATASETS - DETAILED REPORT ";

 *---- RECOVERY SUMMARY  ----;

PROC SUMMARY DATA=RECOV NWAY;
     CLASS FUNCTION DATE;
VAR DURATION PENDING TOTMB MBWRITE COMPR KBSEC FSRCPU;
OUTPUT OUT=S
MEAN(DURATION PENDING TOTMB MBWRITE COMPR KBSEC FSRCPU)=
     AVGRESP AVGPEND AVGMB AVGSIZE AVGCOMP AVGKBSEC AVGCPU
N=COUNT ;

PROC SORT DATA=S; BY DATE FUNCTION;
PROC PRINTTO PRINT=FUNSUMM; OPTIONS PAGENO=1;
PROC PRINT DATA=S UNIFORM NOOBS SPLIT='*';
TITLE1"DFHSM RECOVERY ACTIVITY - SUMMARY";
VAR DATE FUNCTION COUNT AVGRESP AVGCPU
    AVGMB AVGSIZE AVGCOMP AVGKBSEC;
 FORMAT AVGRESP AVGPEND AVGCPU 8.2 ;
 LABEL  AVGMB   ='AVG MB*XFERED*(READ + WRITE)'
        AVGSIZE ='AVG DS*SIZE*IN MB '
        AVGCOMP ='AVG*COMPRESS*RATIO '
        COUNT   ='DATASETS*PROCESSED'
        AVGRESP ='AVG*ELAPSED*TIME(SEC)'
        AVGCPU  ='AVG*CPU*TIME(SEC)'
        AVGKBSEC='AVG*TRANSFER RATE*(KB/SEC)' ;

 *----- TAPE USAGE SUMMARY -----;

DATA TAPES; SET OK;
IF FSRNENT1 LT 1 THEN DELETE;

PROC SUMMARY DATA=TAPES NWAY;
  CLASS TAPEVOL DATE;
  BY FSRTYPE;
  VAR DURATION PENDING FSRCPU;
  OUTPUT OUT=K
  MEAN(DURATION PENDING FSRCPU)= AVGRESP AVGPEND AVGCPU
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
  N=COUNT ;

DATA SUMMARY; SET K;
  FUNCTION=PUT(FSRTYPE,FSRFMT.);

PROC SORT DATA=SUMMARY; BY DATE FUNCTION TAPEVOL;
PROC PRINTTO PRINT=TAPES; OPTIONS PAGENO=1;
PROC PRINT DATA=SUMMARY UNIFORM NOOBS SPLIT="*";
     VAR TAPEVOL FUNCTION COUNT AVGRESP AVGPEND AVGCPU;
     BY DATE;
LABEL  TAPEVOL ='TAPE*VOLSER '
       AVGRESP ='AVG*ELAPSED*TIME(SEC)'
       AVGPEND ='AVG*PENDING*TIME(SEC)'
       FUNCTION='DFHSM*FUNCTION'
       COUNT   ='DATASETS*PROCESSED'
       AVGCPU  ='AVG CPU*TIME(SEC)' ;
       TITLE1"TAPES USED REPORT";

 *----- DUMP VOLUME SUMMARY -----;

DATA DUMP; SET OK;
IF FSRTYPE =13;
PROC PRINTTO PRINT=DUMPS; OPTIONS PAGENO=1;
PROC PRINT DATA=DUMP UNIFORM NOOBS SPLIT="*";
TITLE1"DUMP VOLUME REPORT";
VAR DATE FSRTIMR FSRFVOL TAPEVOL
    DURATION FSRDCOPR FSRDCOPF FSRDCL1 ;

 *---- THRASHING ANALYSIS --------;

DATA TRAS; SET OK;
MIG=PUT(FSRTYPE,MIGRAT.);
IF MIG='95' OR MIG='94';
IF MIG='95' THEN FUNCTION='MIGRATION';
   ELSE          FUNCTION='RECALL';
PROC SORT DATA=TRAS; BY DATE FSRTIMR;
PROC PRINTTO PRINT=FSRSTAT; OPTIONS PAGENO=1;
PROC FREQ DATA=TRAS ORDER=FORMATTED;
  TABLES FUNCTION;
  TITLE1"THRASHING ANALYSIS";
PROC FREQ DATA=TRAS ORDER=FORMATTED;
  TABLES DATE*FUNCTION/NOFREQ NOPERCENT NOCUM NOCOL;
  TITLE1"THRASHING ANALYSIS BY DATE ";
PROC PRINTTO PRINT=MGMTCLAS; OPTIONS PAGENO=1;
PROC FREQ DATA=TRAS ORDER=FORMATTED;
  TABLES FSRMGTCL*FUNCTION/NOFREQ NOPERCENT NOCUM NOCOL;
  TITLE1"THRASHING ANALYSIS BY MGT. CLASS ";
```

Mile Pekic
Systems Programmer (Yugoslavia)
© Xephon 2002

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

Editing a new member when in Option 3.4

Personally I very seldom use Option 3.1 (Library Utility) to access the members of a dataset, preferring to use Option 3.4 (Data Set List Utility) instead. For several years, because I didn't know of another way and there is no menu option available under 3.4, I have used a strange method to edit an empty member.

The method I used was to go into an existing member and use the line command **c** (copy) together with the main command **CREATE newname** to create a new member. This new member was then selected with another 3.4 session and the extraneous text in the member deleted.

In a recent seminar, I learnt that it is possible to edit a new member directly from the Data Set List by typing:

```
e /(newname)
```

in the line command position for the selected dataset – see Figure 1.

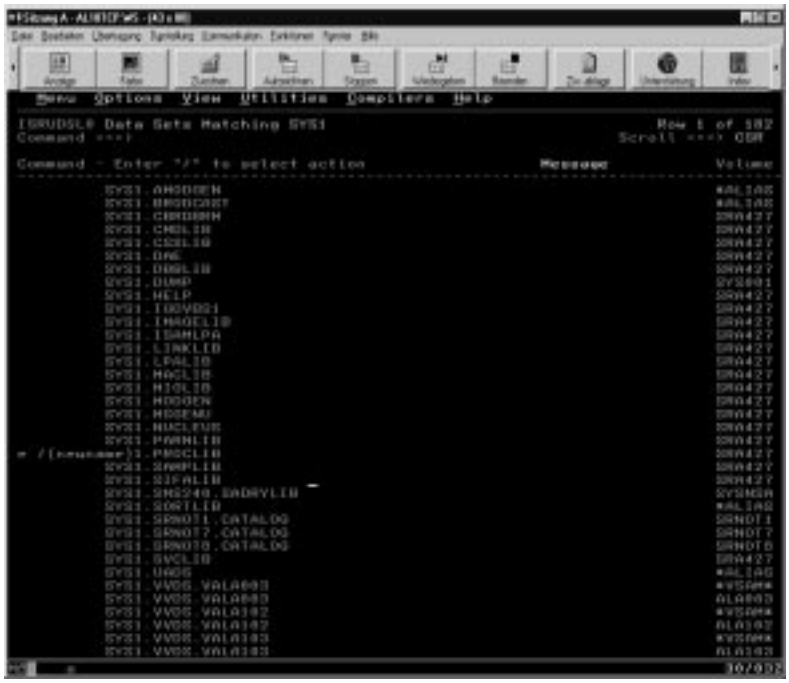

Figure 1: Edit a new member

Rolf Parker
Systems Programmer (Germany)
© Xephon 2002

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

Get current operating system version

PROBLEM ADDRESSED

In situations where operating systems are being installed and tested, the problem sometimes arises of the current operating system version having to be identified in order to adapt jobs appropriately. This functionality can also be required when an application has to adapt its processing depending on the operating system currently running.

SOLUTION

The GETOSVR program obtains the current operating system version and release from the Extended Communications Vector Table (ECVT). The program displays this information (for simplicity the TPUT service is used; this means that the display is made only in TSO – the WTO service with routing code 11 could be used to output the display on the job log). The program returns a hash code formed from the operating system name, the version, and the release. The application can use this returned hash code to perform the appropriate processing.

EXAMPLES

The following list shows the output for some recent operating systems:

- Hash code for OS/390 V2.8: 1819 (X'71B'):

```
OS/390          020800
```

- Hash code for OS/390 V2.10: 1555 (X'613'):

```
OS/390          021000
```

- Hash code for z/OS V1.2: 3968 (X'F80'):

```
z/OS            010200
```

The hash code is a numeric value within the range 0-4095 that the program sets as a return code.

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

PROGRAM CODE

```
         TITLE 'Get OS Name and Version'
* Display operating system name and version, eg OS/390  021000
* Return hash code formed from product name (bytes 0-3), version, and
* release
GETOSVR  CSECT
GETOSVR  AMODE 31
GETOSVR  RMODE 24               RMODE(24) required for TPUT
         BAKR  14,0             Save registers and return address
         BASR  12,0             Set base register
         USING *,12
         SPACE
         L     1,16             A(CVT)
         USING CVT,1
* R1: Address of Communications Vector Table
         L     2,CVTECVT        A(ECVT)
         USING ECVT,2
* R2: Address of Extended Communications Vector Table
         MVC   PNAMEVRL,ECVTPNAM
         LA    1,PNAMEVRL
         LA    0,L'PNAMEVRL
         TPUT  (1),(0)
* Form hash code from product name (bytes 0-3), version and release
         XC    HASH,HASH  Clear <hash>
         XC    HASH,PRODNAME+0  Bytes 0-1
         XC    HASH,PRODNAME+2  Bytes 2-3
         XC    HASH,PRODVERS    Version
         XC    HASH,PRODREL     Release
         NC    HASH,=X'FFF'     Modulo(4096)
         LH    15,HASH
         PR    ,                Program return
         SPACE
HASH     DS   H
         SPACE
PNAMEVRL DS   0CL22
PRODNAME DS   CL16              Product (Operating System) Name
PRODVERS DS   CL2               Product Version
PRODREL  DS   CL2               Product Release
         DS   CL2               Product Modification Level
         SPACE
         CVT DSECT=YES
         IHAECVT
         END
```

PROGRAM INVOCATION

The program does not have any files or parameters.

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

SYSOUT Application Programming Interface

OS/390 (z/OS) provides a number of directed subsystem interface (SSI) function calls that are available with the IEFSSREQ macro call. SSI function code 1 (process SYSOUT datasets) has existed for many years and was/is used by external writer interfaces to allow a program to access JES SYSOUT datasets independently of traditional JES functions such as print. Although this is a powerful interface, it has some limitations:

- Certain functionality is only available if the external writer program is running as a started task.
- For JES2 environments, the external writer program is unable to select spooled SYSOUT data that is directed to held output classes.

The more recent SSI function code 79 (SYSOUT Application Programming Interface – SAPI) is a more robust implementation that overcomes many of the restrictions inherent in SSI function code 1. Unlike SSI function code 1, the SSI 79 interface supports multiple threads and allows for SYSOUT dataset selection that goes beyond the scope of SSI 1 external writers.

External writers serve many useful purposes. They are well suited for managing SYSOUT data not processed by JES directly. Many third-party vendor products use the external writer interface – particularly report management and distribution products. The SAPISSI program provided with this article functions very much like a traditional SSI function code 1 external writer except that it can run as a batch job (or started task) and still get automatic notification of new SYSOUT output that has been created that matches its selection criteria, and it can extract SYSOUT output that is directed to JES2 held output classes.

The SAPISSI program does nothing more than collect output datasets that have been directed to the spool in the output classes that the external writer has been directed to collect output for. To initialize the external writer, you need to start a batch job or started task that specifies the external writer program name, the parameter-supplied external writer name, and an optional output class list. Here are a few examples:

 © 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
//XWTR1    EXEC PGM=SAPISSI,PARM='TESTWTR'
//STEPLIB  DD   DSN=authorized.library,DISP=SHR


//XWTR2    EXEC PGM=SAPISSI,PARM='TESTWTR1,AGRYU49'
//STEPLIB  DD   DSN=authorized.library,DISP=SHR
```

The first example would start external writer TESTWTR, which will collect SYSOUT data for all output classes (no class list is specified in the program parameters so the default is to collect data for all classes). The second example would start external writer TESTWTR2, which will collect SYSOUT data for only classes AGRYU49.

You can direct output to an external writer using the SYSOUT parameter on a JCL DD statement. Here is an example:

```
//OUTDD    DD   SYSOUT=(A,TESTWTR)
```

The above example would direct its output to an external writer with a name of TESTWTR. The TESTWTR external writer would process the output providing it was currently set up to collect class A output.

The external writer provided with this article has an operator command interface that supports two commands. The first command is a modify command that can be used to modify the class list that is currently being monitored by the external writer. The command has the following format:

```
F xwtr,CLASSLIST=classlst
```

where ‘xwtr’ is the name of your external writer started task or batch job, and ‘classlst’ is the list of classes you want the external writer to be managing. The ‘classlst’ can consist of a list of up to 36 alphanumeric characters (A-Z and 0-9) or a single ‘*’. A single asterisk indicates that every class is eligible for processing by this external writer.

The second operator command that is supported by the external writer is the stop command. It has the following format:

```
P xwtr
```

where ‘xwtr’ is the name of your external writer started task or batch job. This command will cause the termination of the external writer.

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

The SAPISSI external writer program should be link-edited into an authorized load library using the following link-edit control cards:

```
 INCLUDE OBJECT(SAPISSI)
 ENTRY   SAPISSI
 SETCODE AC(1)
 NAME    SAPISSI(R)
```

A SAPISSI external writer batch job or started task can be initiated with the following sample JCL:

```
//SAPISSI  EXEC PGM=SAPISSI,PARM='TESTWTR1'
//STEPLIB  DD   DSN=authorized.library,DISP=SHR
//OUTPUT   DD   DSN=xwtr.output,DISP=SHR
//SYSPRINT DD   SYSOUT=*
```

The SAPISSI external writer expects an OUTPUT DD statement to be used as a collection dataset for the target data. Since you cannot be sure what the characteristics of the source data will be, I recommend that the OUTPUT dataset used by the external writer be defined as RECFM=VB, LRECL=32765, BLKSIZE=32760, although the external writer will support RECFM=FB files and SYSOUT as well. The SYSPRINT DD dataset is used for messages that are produced from the external writer itself.

All output collected by the external writer is dumped to the OUTPUT dataset. Each independent output dataset is identified in the collection OUTPUT dataset by a separator record that has the following format:

```
JBN: jobname  JB#: jobnum  PRC: prcname   STP: stpnam   DDN: ddname
CLS: c  DSN: dsn
```

This allows you to determine the various different output datasets that have been dumped to the collector OUTPUT dataset. If you choose, additional information could be included in the separator record such as the system name that the job ran on, the JES member the job ran on, the time of day the job ran, etc. Have some fun deciding what your separator record will look like.

The SAPISSI program is meant to provide a very simple example of an external writer and some of its capabilities. I hope this article provides some interesting insights.

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

SAPISSI ASM

```
SAPISSI  CSECT
SAPISSI  AMODE 31
SAPISSI  RMODE 24
*********************************************************************
*   THIS PROGRAM DEMONSTRATES THE USE OF THE SYSOUT APPLICATION     *
*   PROGRAM INTERFACE (SAPI) SUBSYSTEM FUNCTION CALL (SSI 79) AND   *
*   PROVIDES A VERY BASIC EXAMPLE OF AN OS/390 EXTERNAL WRITER.     *
*                                                                   *
*   THE SAPI SSI 79 FUNCTION CALL CAN BE USED TO EXTRACT SYSOUT DATA*
*   FROM THE JES SPOOL.  THE INTERFACE PROVIDES FOR A NUMBER OF     *
*   DIFFERENT SELECTION CRITERIA OPTIONS OF WHICH ONLY A FEW ARE USED*
*   BY THIS PARTICULAR PROGRAM.  A COMPLETE DESCRIPTION OF THE SAPI *
*   SELECTION CRITERIA CAN BE FOUND IN 'OS/390 MVS Using the        *
*   Subsystem Interface' OR BY EXAMINING THE IAZSSS2 CONTROL BLOCK  *
*   MAPPING MACRO.                                                  *
*                                                                   *
*   THIS PROGRAM SHOULD BE LINK-EDITED INTO AN AUTHORIZED DATASET    *
*   USING THE FOLLOWING LINK-EDIT CONTROL CARDS:                     *
*                                                                   *
*        INCLUDE OBJECT(SAPISSI)                                    *
*        ENTRY   SAPISSI                                            *
*        SETCODE AC(1)                                              *
*        NAME    SAPISSI(R)                                         *
*                                                                   *
*   THE PROGRAM CAN BE RUN AS FOLLOWS:                              *
*                                                                   *
*   //STEP1    EXEC PGM=SAPISSI,PARM='wtrnm,classlst'               *
*   //STEPLIB  DD   DSN=authorized.library,DISP=SHR                 *
*   //OUTPUT   DD   SYSOUT=*                                        *
*   //SYSPRINT DD   SYSOUT=*                                        *
*                                                                   *
*   WHERE 'wtrnm' IS THE EXTERNAL WRITER NAME THAT WILL BE USED BY  *
*   THIS PROGRAM AND 'classlst' IS THE LIST OF JES OUTPUT CLASSES   *
*   THAT WILL BE PROCESSED BY THIS EXTERNAL WRITER.  'wtrnm' IS     *
*   REQUIRED BUT THE 'classlst' PORTION OF THE PARM IS OPTIONAL.  IF*
*   'classlst' IS OMITTED, ALL OUTPUT CLASSES ARE ELIGIBLE FOR      *
*   SELECTION.  HERE ARE SOME VALID EXAMPLES:                       *
*                                                                   *
*   //STEP1    EXEC PGM=SAPISSI,PARM='TESTWTR1'                     *
*                                                                   *
*     THIS WILL START EXTERNAL WRITER TESTWTR1.  THIS EXTERNAL      *
*     WRITER WILL PROCESS SYSOUT OUTPUT DIRECTED TO IT ON ANY       *
*     OUTPUT QUEUE AND ANY OUTPUT CLASS.                            *
*                                                                   *
*   //STEP1    EXEC PGM=SAPISSI,PARM='TESTWTR1,ADFM259'             *
*                                                                   *
*     THIS WILL START EXTERNAL WRITER TESTWTR1.  THIS EXTERNAL      *
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
*     WRITER WILL PROCESS SYSOUT OUTPUT DIRECTED TO IT ON ANY        *
*     OUTPUT QUEUE AND ONLY OUTPUT CLASSES "ADFM259".                *
*                                                                    *
*   IF YOU WANT TO SEND OUTPUT TO THIS EXTERNAL WRITER, RUN A JOB    *
*   THAT DIRECTS OUTPUT TO A SYSOUT DATASET:                         *
*                                                                    *
*   //SYSPRINT DD    SYSOUT=(F,TESTWTR1)                             *
*                                                                    *
*   THIS EXTERNAL WRITER WILL PROCESS THIS SYSOUT DATASET WHEN THE   *
*   CREATING PROGRAM TERMINATES.                                     *
*                                                                    *
*   TO DYNAMICALLY MODIFY THE LIST OF CLASSES THE EXTERNAL WRITER    *
*   IS ELIGIBLE TO PROCESS, ENTER THE MODIFY OPERATOR COMMAND:       *
*                                                                    *
*   F sapissi,CLASSLIST=classlst                                     *
*                                                                    *
*   WHERE 'sapissi' IS THE JOBNAME OF THE EXTERNAL WRITER JOB AND    *
*   'classlst' IS THE NEW LIST OF VALID CLASSES.                     *
*                                                                    *
*   TO REACTIVATE ALL 36 OUTPUT CLASSES, THE FOLLOWING MODIFY        *
*   OPERATOR COMMAND CAN BE USED:                                    *
*                                                                    *
*   F sapissi,CLASSLIST=*                                            *
*                                                                    *
*   TO STOP THIS EXTERNAL WRITER SIMPLY ENTER THE STOP OPERATOR      *
*   COMMAND:                                                         *
*                                                                    *
*   P sapissi                                                        *
*                                                                    *
*   WHERE 'sapissi' IS THE JOBNAME OF THE EXTERNAL WRITER JOB.       *
**********************************************************************
         STM   R14,R12,12(R13)          SAVE INCOMING REGISTER VALUES
         LR    R12,R15                  COPY MODULE ADDRESS
         LA    R11,4Ø95(,R12)           SET SECOND BASE...
         LA    R11,1(,R11)                 REGISTER ADDRESS
         USING SAPISSI,R12,R11          SET MODULE ADDRESSABILITY
         LR    R1Ø,R1                   SAVE PARM ADDRESS
         LR    R3,R13                   COPY OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)           GET STORAGE LENGTH
         STORAGE OBTAIN,LENGTH=(R9)     GET DYNAMIC STORAGE
         LR    R13,R1                   SAVE STORAGE ADDRESS
         LR    RØ,R1                    COPY IT
         LR    R14,R1                   AGAIN
         L     R1,=A(WORKLEN)           GET STORAGE LENGTH
         XR    R15,R15                  SET FILL BYTE
         MVCL  RØ,R14                   CLEAR THE DYNAMIC STORAGE
         ST    R3,4(,R13)               SAVE OLD SAVEAREA ADDRESS
         USING WORKAREA,R13             SET WORKAREA ADDRESSABILITY
**********************************************************************
*   VALID PARM FORMATS ARE:                                          *
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
*                                                                     *
*   'WRTNM'                                                           *
*   'WRTNM,CLASSLIST'                                                 *
*                                                                     *
*   WHERE 'WRTNM' IS A MAXIMUM OF EIGHT CHARACTERS                    *
*         'CLASSLIST' IS A MAXIMUM OF 36 CHARACTERS                   *
*                                                                     *
*   IF 'CLASSLIST' IS SPECIFIED, THE 'WRTNM' AND 'CLASSLIST'          *
*   PARAMETERS MUST BE COMMA SEPARATED.                               *
***********************************************************************
         OPEN  (SYSPRINT,OUTPUT),MODE=31
         LR    R1,R1Ø                    GET PARM ADDRESS
         L     R9,Ø(,R1)                 GET ADDRESS OF PARM
         CLC   Ø(2,R9),=H'45'            MAX LENGTH OK?
         BH    RETURNØ8                  NO - ISSUE MESSAGE AND END
         CLC   Ø(2,R9),=H'Ø'             A PARM?
         BNH   RETURNØ8                  NO - ISSUE MESSAGE AND END
         B     WTRNMOK                   WRITER NAME SHOULD BE OK
WTRNMOK  EQU   *
         XR    R14,R14                   CLEAR R14
         ICM   R14,B'ØØ11',Ø(R9)         COPY THE LENGTH
         XR    R1,R1                     CLEAR LENGTH COUNTER
         MVC   WTRNM(8),=8C' '           CLEAR THE AREA
         LA    R2,WTRNM                  GET TARGET ADDRESS
         LA    R9,2(,R9)                 SKIP PAST LENGTH
WTRNMLP  EQU   *
         LTR   R14,R14                   PARM DONE?
         BZ    WTRNMCHK                  YES - CHECK LENGTH
         C     R1,=F'8'                  MAX LENGTH?
         BH    RETURNØ8                  YES - ISSUE MESSAGE AND END
         CLI   Ø(R9),C','                END OF PARM?
         BNE   WTRNMMV                   NO - COPY THE DATA
         BCTR  R14,Ø                     REDUCE LENGTH BY ONE
         LA    R9,1(,R9)                 POINT TO NEXT DATA BYTE
         LTR   R14,R14                   END OF DATA?
         BZ    RETURNØ8                  YES - ISSUE MESSAGE AND END
         B     WTRNMCHK                  CHECK LENGTH
WTRNMMV  EQU   *
         ST    R14,R14SAVE               SAVE R14
         BAL   R14,CHARCHK               CHECK FOR ALPHANUMERIC
         LTR   R15,R15                   ALPHANUMERIC?
         BNZ   RETURNØ8                  NO - ALL DONE
         L     R14,R14SAVE               LOAD R14
         MVC   Ø(1,R2),Ø(R9)             MOVE DATA
         BCTR  R14,Ø                     REDUCE LENGTH BY ONE
         LA    R9,1(,R9)                 POINT TO NEXT DATA BYTE
         LA    R2,1(,R2)                 POINT TO NEXT TARGET BYTE
         LA    R1,1(,R1)                 ADD ONE TO LENGTH
         B     WTRNMLP                   CHECK NEXT BYTE
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
WTRNMCHK EQU   *
         LTR   R1,R1                    LENGTH OK?
         BZ    RETURNØ8                 NO - ISSUE MESSAGE AND END
         MVC   CLASSLST(36),=36C' '     CLEAR THE AREA
         LA    R2,CLASSLST              GET CLASS LIST AREA ADDRESS
         XR    R1,R1                    CLEAR LENGTH COUNTER
CLLSTLP  EQU   *
         LTR   R14,R14                  PARM DONE?
         BZ    CLLSTDON                 YES - WE'RE DONE
         C     R1,=F'36'                MAX LENGTH?
         BNL   RETURNØ8                 YES - ISSUE MESSAGE AND END
         ST    R14,R14SAVE              SAVE R14
         BAL   R14,CHARCHK              CHECK FOR ALPHANUMERIC
         LTR   R15,R15                  ALPHANUMERIC?
         BNZ   RETURNØ8                 NO - ALL DONE
         L     R14,R14SAVE              LOAD R14
         MVC   Ø(1,R2),Ø(R9)            MOVE DATA
         BCTR  R14,Ø                    REDUCE LENGTH BY ONE
         LA    R9,1(,R9)                POINT TO NEXT DATA BYTE
         LA    R2,1(,R2)                POINT TO NEXT TARGET BYTE
         LA    R1,1(,R1)                ADD ONE TO LENGTH
         B     CLLSTLP                  CHECK NEXT BYTE
CLLSTDON EQU   *
*********************************************************************
*   DETERMINE THE ACTIVE SUBSYSTEM NAME.                            *
*********************************************************************
         L     R15,16                   GET CVT ADDRESS
         L     R15,Ø(,R15)              GET TCB/ASCB AREA ADDRESS
         L     R15,4(,R15)              GET CURRENT TCB
         L     R15,TCBJSCB-TCB(,R15)    GET JSCB ADDRESS?
         L     R15,JSCBSSIB-IEZJSCB(,R15) GET SSIB ADDRESS?
         MVC   SSNMSAVE(8),=8C' '       INITIALIZE THE AREA
         MVC   SSNMSAVE(4),SSIBSSNM-SSIB(R15) MV SSNAME
*********************************************************************
GETCIB   LA    R5,COMMADDR              ADDR OF RESPONSE AREA FOR QEDIT
         MVC   EXTWRK(EXTLN),EXTLST     MOVE IN THE MODEL
         EXTRACT (5),FIELDS=COMM,MF=(E,EXTWRK)  GET ADDR OF COMM AREA
         L     R5,COMMADDR              LOAD ADDR OF COMM AREA
         USING COMLIST,R5
         ST    R5,COMMADDR              SAVE COMM AREA ADDRESS
         L     R6,COMCIBPT              GET ADDRESS OF CIB
         USING CIBNEXT,R6
         C     R6,=F'Ø'                 CIB EXIST ?
         BE    SETCOUNT                 NO - GO SET COUNT
         QEDIT ORIGIN=COMCIBPT,BLOCK=(R6) YES - FREE IT
         LTR   R15,R15                  GO O.K. ?
         BZ    SETCOUNT                 YES - GO SET COUNT
         ABEND Ø9                       ERROR
SETCOUNT EQU   *
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
*********************************************************************
*   SET LIMIT ON MODIFY COMMANDS                                    *
*********************************************************************
         QEDIT ORIGIN=COMCIBPT,CIBCTR=1 ONE OPERATOR CMD AT A TIME
         WTO   'XWTRØ8ØI - COMMAND INTERFACE ACTIVATED.',              X
               ROUTCDE=(1),DESC=(6)
         LA    R7,TIOTADDR               GET ADDRESS OF TARGET AREA
         EXTRACT (R7),FIELDS=TIOT        GET THE TIOT ADDRESS
         L     R7,TIOTADDR               COPY THE TIOT ADDRESS
         LA    R7,24(,R7)                POINT TO TIOT ENTRY AREA
         USING TIOENTRY,R7               SET ADDRESSABILITY
DDLOOP1  EQU   *
         CLI   TIOELNGH,X'ØØ'            END OF LIST?
         BE    OPENOUT                   YES - NOT GOOD - LET OPEN FAIL
         CLC   TIOEDDNM(8),=C'OUTPUT  ' THE 'OUTPUT' DD?
         BE    DDMATCH1                  YES - CHECK DATASET TYPE
         XR    R1,R1                     CLEAR R1
         IC    R1,TIOELNGH               GET TIOT ENTRY LENGTH
         LA    R7,Ø(R1,R7)               POINT TO NEXT ENTRY
         B     DDLOOP1                   GO CHECK NEXT ENTRY
DDMATCH1 EQU   *
         LA    R8,EPA                    GET ADDRESS OF THE EPA
         ST    R8,SWEPAPTR               INITIALIZE EPA POINTER
         USING ZB5Ø5,R8                  ADDRESSABILITY TO EPA
         XC    EPA,EPA                   CLEAR THE WORK AREA
         MVC   SWVA,TIOEJFCB             MV SVA OF JFCB INTO EPA
         SWAREQ FCODE=RL,                                            X
               EPA=SWEPAPTR,                                         X
               UNAUTH=YES,                                           X
               MF=(E,SWAPARMS)
         LTR   R15,R15                   LOCATE WAS SUCCESSFUL?
         BNZ   RETURN2Ø                  NO - TERMINATE
         L     R8,SWBLKPTR               SET POINTER TO THE JFCB
         DROP  R8
         USING INFMJFCB,R8               SET JFCB ADDRESSABILITY
         TM    JFCBTSDM,JFCSDS           A SYSOUT DATASET?
         BNO   OPENOUT                   NO - OPEN THE OUTPUT DATASET
         MVC   OUTPUT+82(2),=H'32768Ø'   MOVE IN THE LRECL
         DROP  R7,R8
OPENOUT  EQU   *
         OPEN  (OUTPUT,OUTPUT),MODE=31
*********************************************************************
         MVI   RECBUFF,C' '              SET FILL CHARACTER
         MVC   RECBUFF+1(132),RECBUFF    CLEAR OUT RECORD AREA
         MVC   RECBUFF(L'MSG98),MSG98    MOVE IN THE MESSAGE
         MVC   RECBUFF+27(8),WTRNM       MOVE IN THE WRITER NAME
         PUT   SYSPRINT,RECBUFF          WRITE THE MESSAGE
*********************************************************************
SSICALL  EQU   *
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
*********************************************************************
         LA    R3,SSOBAREA              GET SSOB ADDRESS
         USING SSOBEGIN,R3              SET ADDRESSABILITY
         LA    R4,SSOBGN                GET SSS2 ADDRESS
         USING SSS2,R4                  SET ADDRESSABILITY
         XC    SSOB(SSOBHSIZ),SSOB      CLEAR THE SSOB
         MVC   SSOBID(4),=C'SSOB'       SET SSOB ID
         MVC   SSOBFUNC(2),=AL2(SSOBSOU2) SET FUNCTION ID
         MVC   SSOBLEN(2),=AL2(SSOBHSIZ) SET SSOB HEADER SIZE
         ST    R4,SSOBINDV              SET SSS2 ADDRESS
*********************************************************************
         LA    R14,SSS2                 GET AREA ADDRESS
         L     R1,=A(SSS2SIZE)          GET LENGTH
         LR    RØ,R14                   COPY ADDRESS
         XR    R15,R15                  SET FILL BYTE
         MVCL  RØ,R14                   CLEAR THE AREA
         MVC   SSS2EYE(4),=C'SSS2'      SET EYECATCHER
         MVC   SSS2LEN,=AL2(SSS2SIZE)   SET SSS2 SIZE
         MVI   SSS2VER,SSS2CVER         SET THE VERSION
*********************************************************************
*   DETERMINE THE JOB SELECTION CRITERIA THAT WILL BE USED BY THE     *
*   EXTERNAL WRITER.  THE IAZSSS2 MACRO PROVIDES INSIGHT INTO WHAT    *
*   CRITERIA MAY BE USED.                                             *
*                                                                     *
*   THIS PARTICULAR EXTERNAL WRITER SELECTS SYSOUT INDEPENDENTLY OF   *
*   CLASS DISPOSITION (SSS2SAWT), IT SELECTS BY CLASS LIST (SSS2SCLS  *
*   AND SSS2CLSL) IF A CLASS LIST HAS BEEN SPECIFIED ON THE PROGRAM   *
*   PARM, AND IT SELECTS BY WRITER NAME (SSS2SPGM AND SSS2PGMN).      *
*                                                                     *
*   AN ECB ADDRESS HAS BEEN INCLUDED IN SSS2ECBP.  THIS ALLOWS THE    *
*   EXTERNAL WRITER TO WAIT ON AN ECB THAT WILL GET POSTED WHEN NEW   *
*   SYSOUT OUTPUT BECOMES AVAILABLE ON SPOOL MATCHING THE SELECTION   *
*   CRITERIA SPECIFIED.                                               *
*********************************************************************
         MVI   SSS2SEL1,SSS2SAWT        SELECT ANY DISP
         OI    SSS2SEL2,SSS2SPGM        SELECT BY WRITER NAME
         MVC   SSS2PGMN(8),WTRNM        SET THE WRITER NAME
         OI    SSS2TYPE,SSS2PUGE        SET TYPE TO PUT/GET
SSICALL2 EQU   *
         OI    SSS2DSP1,SSS2RNPR        DON'T RETURN THIS D/S AGAIN
         NI    SSS2SEL1,255-SSS2SCLS    TURN CLASS SELECTION FLAG OFF
         CLC   CLASSLST(36),=36C' '     ANY CLASS SELECTION CRITERIA?
         BE    NOCLASS                  NO - DON'T SET CLASS FLAGS
         OI    SSS2SEL1,SSS2SCLS        SET CLASS SELECTION FLAG
         MVC   SSS2CLSL(36),CLASSLST    MOVE IN CLASS LIST
NOCLASS  EQU   *
         XC    SELECB(4),SELECB         CLEAR ECB
         LA    R1,SELECB                GET ECB ADDRESS
         ST    R1,SSS2ECBP              SAVE ECB ADDRESS
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
*************************************************************************
         MODESET MODE=SUP
         LA    R1,SSOBAREA              GET SSOB ADDRESS
         O     R1,=X'80000000'          TURN ON X'80' BIT
         ST    R1,SSOBPTR               SAVE SSOB PTR
         LA    R1,SSOBPTR               POINT TO SSOB PTR
         IEFSSREQ ,                     MAKE SUBSYSTEM REQUEST
         LR    R9,R15                   SAVE THE RETURN CODE
         MODESET MODE=PROB
         LTR   R9,R9                    SUBSYSTEM REQUEST OK?
         BNZ   SSREQAB                  NO - LET'S END
         XR    R8,R8                    CLEAR R8
         ICM   R8,B'0001',SSOBRETN+3    GET RETURN CODE
         C     R8,=F'44'                SSOBRETN TOO HIGH?
         BH    ERRUNKWN                 YES - UNKNOWN
         B     BRTBL1(R8)               PROCESS ACCORDINGLY
BRTBL1   EQU   *
         B     REQOK                    SSOBRETN=00
         B     REQEODS                  SSOBRETN=04
         B     REQINVA                  SSOBRETN=08
         B     REQUNAV                  SSOBRETN=12
         B     REQDUPJ                  SSOBRETN=16
         B     REQIDST                  SSOBRETN=20
         B     REQAUTH                  SSOBRETN=24
         B     REQTKNM                  SSOBRETN=28
         B     REQLERR                  SSOBRETN=32
         B     REQICLS                  SSOBRETN=36
         B     REQBDIS                  SSOBRETN=40
         B     REQCLON                  SSOBRETN=44
ERRUNKWN EQU   *
         MVC   DBL2(4),SSOBRETN         COPY RETURN CODE
         UNPK  DBL1(9),DBL2(5)          UNPACK IT
         NC    DBL1(8),=8X'0F'          TURN OFF HIGH NIBBLE
         TR    DBL1(8),=C'0123456789ABCDEF' MAKE IT READABLE
         MVI   RECBUFF,C' '             SET FILL CHARACTER
         MVC   RECBUFF+1(132),RECBUFF   CLEAR OUT RECORD AREA
         MVC   RECBUFF(MSG18L),MSG18    MOVE IN THE MESSAGE
         MVC   RECBUFF+33(8),DBL1       MOVE IN THE SSOBRETN
         PUT   SYSPRINT,RECBUFF         WRITE THE MESSAGE
         B     RETURN12                 RETURN
SSREQAB  EQU   *
         LA    R1,MSG17                 GET MESSAGE ADDRESS
         LA    R15,L'MSG17              GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     RETURN12                 RETURN
REQEODS  EQU   *
         L     R8,SSS2ECBP              GET SAPI ECB ADDRESS?
         ST    R8,ECBLIST+0             SAVE IN THE ECB LIST
         L     R8,COMECBPT              GET ADDR OF COMMUNICATION ECB
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
         ST    R8,ECBLIST+4            SAVE IN THE ECB LIST
         OI    ECBLIST+4,X'8Ø'         SET LAST ECB INDICATOR
         LA    R8,ECBLIST              GET ECB LIST ADDRESS
         WAIT  1,ECBLIST=(R8)          WAIT FOR AN EVENT TO COMPLETE
         L     R8,ECBLIST+Ø            GET FIRST ECB ADDRESS
         TM    Ø(R8),X'4Ø'             A SAPI EVENT?
         BNO   CHKECB2                 NO - CHECK OTHER ECB
         XC    SELECB(4),SELECB        CLEAR ECB FOR NEXT TIME
         B     SSICALL2                MAKE SSI CALL AGAIN
CHKECB2  EQU   *
         L     R8,ECBLIST+4            GET SECOND ECB ADDRESS
         TM    Ø(R8),X'4Ø'             A CONSOLE EVENT?
         BNO   REQEODS                 NO - LET'S GO WAIT
         L     R5,COMMADDR             LOAD ADDR OF COMMUNICATION A
         L     R6,COMCIBPT             GET ADDRESS OF CIB
         LTR   R6,R6                   VALID POINTER?
         BZ    REQEODS                 NO - RETURN
         CLI   CIBVERB,CIBMODFY        IS IT A MODIFY COMMAND ?
         BE    CMODIFY                 YES - GO PROCESS
         CLI   CIBVERB,CIBSTOP         IS IT A STOP COMMAND ?
         BNE   NOSTOP                  NO - JUST CONTINUE
         B     CSTOP                   YES - GO SHUT THINGS DOWN
CMODIFY  EQU   *
         XR    R14,R14                 CLEAR R15
         ICM   R14,B'ØØ11',CIBDATLN    GET COMMAND LENGTH
         C     R14,=F'11'              MINIMUM LENGTH OK?
         BL    CMDERRØ1                NO - ISSUE AN ERROR
         CLC   CIBDATA(1Ø),=C'CLASSLIST=' CLASSLIST= OPERAND?
         BNE   CMDERRØ1                NO - ISSUE AN ERROR
         S     R14,=F'1Ø'              REDUCE BUFFER LENGTH
         C     R14,=F'36'              TOO MUCH DATA?
         BH    CMDERRØ2                YES - ISSUE AN ERROR
         MVC   CLASSBUF(36),=36C' '    INITIALIZE BUFFER AREA
         CLI   CIBDATA+1Ø,C'*'         TURN ALL CLASSES BACK ON?
         BE    MDLSTDON                YES - THEY'LL BE ACTIVE AGAIN
         BCTR  R14,Ø                   REDUCE LENGTH BY ONE FOR EX
         EX    R14,CLASSMVC            COPY THE CLASS LIST
         LA    R9,CLASSBUF             GET ADDRESS OF CLASS LIST
         LA    R14,1(,R14)             ADD ONE BACK TO LENGTH
MDLSTLP  EQU   *
         LTR   R14,R14                 CLASS LIST DONE?
         BZ    MDLSTDON                YES - GO ON
         ST    R14,R14SAVE             SAVE R14
         BAL   R14,CHARCHK             CHECK FOR ALPHANUMERIC
         LTR   R15,R15                 ALPHANUMERIC?
         BNZ   CMDERRØ2                NO - ISSUE AN ERROR
         L     R14,R14SAVE             LOAD R14
         BCTR  R14,Ø                   REDUCE LENGTH BY ONE
         LA    R9,1(,R9)               POINT TO NEXT DATA BYTE
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
         B     MDLSTLP                  CHECK NEXT BYTE
MDLSTDON EQU   *
         MVC   CLASSLST(36),CLASSBUF    REPLACE THE ACTIVE CLASS LIST
         QEDIT ORIGIN=COMCIBPT,BLOCK=(R6) YES - FREE IT
         WTO   'XWTR081I - CLASSLIST UPDATE SUCCESSFUL.',              X
               ROUTCDE=(1),DESC=(6)
         B     REQEODS                  JUST GO WAIT
CMDERR01 EQU   *
         WTO   'XWTR089I - UNRECOGNIZED MODIFY COMMAND.',              X
               ROUTCDE=(1),DESC=(6)
         B     NOSTOP                   GO RELEASE CIB
CMDERR02 EQU   *
         WTO   'XWTR088I - INVALID CLASSLIST FORMAT.  CLASSLIST MUST NOX
               T EXCEED 36 ALPHANUMERIC CHARACTERS.',                  X
               ROUTCDE=(1),DESC=(6)
         B     NOSTOP                   GO RELEASE CIB
NOSTOP   EQU   *
         QEDIT ORIGIN=COMCIBPT,BLOCK=(R6) YES - FREE IT
         B     REQEODS                  JUST GO WAIT
CSTOP    EQU   *
         QEDIT ORIGIN=COMCIBPT,BLOCK=(R6) YES - FREE IT
         B     SAPIDONE                 WE'RE DONE
REQINVA  EQU   *
         LA    R1,MSG3                  GET MESSAGE ADDRESS
         LA    R15,L'MSG3               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQUNAV  EQU   *
         LA    R1,MSG4                  GET MESSAGE ADDRESS
         LA    R15,L'MSG4               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQDUPJ  EQU   *
         LA    R1,MSG5                  GET MESSAGE ADDRESS
         LA    R15,L'MSG5               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQIDST  EQU   *
         LA    R1,MSG6                  GET MESSAGE ADDRESS
         LA    R15,L'MSG6               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQAUTH  EQU   *
         LA    R1,MSG7                  GET MESSAGE ADDRESS
         LA    R15,L'MSG7               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQTKNM  EQU   *
         LA    R1,MSG8                  GET MESSAGE ADDRESS
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
         LA    R15,L'MSG8               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQLERR  EQU   *
         LA    R1,MSG9                  GET MESSAGE ADDRESS
         LA    R15,L'MSG9               GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQICLS  EQU   *
         LA    R1,MSG1Ø                 GET MESSAGE ADDRESS
         LA    R15,L'MSG1Ø              GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQBDIS  EQU   *
         LA    R1,MSG11                 GET MESSAGE ADDRESS
         LA    R15,L'MSG11              GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQCLON  EQU   *
         LA    R1,MSG12                 GET MESSAGE ADDRESS
         LA    R15,L'MSG12              GET MESSAGE LENGTH
         BAL   R14,MSGPUT               GO WRITE THE MESSAGE
         B     EXIT
REQOK    EQU   *
*************************************************************************
         LA    R1,SSS2BTOK              GET BLOCK TOKEN ADDRESS
         LA    RØ,DYNPRMS2              GET TARGET AREA ADDRESS
         L     R1,=A(S992LN)            GET THE LENGTH
         LA    R14,S992                 GET SOURCE AREA ADDRESS
         LR    R15,R1                   GET THE LENGTH
         MVCL  RØ,R14                   MOVE IN THE MODEL
         LA    R1,DYNPRMS2              GET PARM AREA ADDRESS
         LA    R2,S992RB-S992(,R1)      GET RELOCATED S992RB ADDRESS
         O     R2,=X'8ØØØØØØØ'          SET FLAG
         ST    R2,Ø(,R1)                SAVE RELOCATED ADDRESS IN PARMS
         LA    R2,S992TUPL-S992(,R1)    GET RELOCATED S992TUPL ADDRESS
         STCM  R2,B'1111',S992TXTP-S992(R1) SV RELOCATED S992TUPL ADR
         LA    R2,TU2ØØ1-S992(,R1)      GET RELOCATED TU2ØØ1 ADDRESS
         STCM  R2,B'1111',S992TUPL-S992(R1) SV RELOCATED TU2ØØ1 ADDR
         LA    R2,TU2ØØ2-S992(,R1)      GET RELOCATED TU2ØØ2 ADDRESS
         STCM  R2,B'1111',TU2Ø2-S992(R1) SV RELOCATED TU2ØØ2 ADDR
         LA    R2,TU2ØØ3-S992(,R1)      GET RELOCATED TU2ØØ3 ADDRESS
         MVC   6(4,R2),SSNMSAVE         MOVE IN THE SUBSYSTEM NAME
         STCM  R2,B'1111',TU2Ø3-S992(R1) SV RELOCATED TU2ØØ3 ADDR
         MVC   TU2Ø4-S992(4,R1),SSS2BTOK GET DALBRTKN ADDRESS
         OI    TU2Ø4-S992(R1),X'8Ø'     SET LAST TU FLAG
         MVC   DSN2-S992(44,R1),SSS2DSN MOVE IN THE DSNAME
         SVC   99                       ALLOCATE THE DATASET
         LTR   R15,R15                  ALLOCATE OK?
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
         BZ    ALLOC2OK                 YES - KEEP GOING
         LA    R1,DYNPRMS2              GET PARM AREA ADDRESS
         MVC   DBL2(4),S992ERR-S992(R1) COPY RETURN CODE
         UNPK  DBL1(9),DBL2(5)          UNPACK IT
         NC    DBL1(8),=8X'ØF'          TURN OFF HIGH NIBBLE
         TR    DBL1(8),=C'Ø123456789ABCDEF' MAKE IT READABLE
         MVC   ALLOCMSG+17(8),DBL1      MOVE IN RETURN CODE
         MVC   ALLOCMSG+41(44),SSS2DSN  COPY THE DSNAME
         PUT   SYSPRINT,ALLOCMSG        WRITE THE MESSAGE
         ABEND 1Ø                       ABEND
ALLOC2OK EQU   *
         LA    R1,DYNPRMS2              GET PARM AREA ADDRESS
         MVC   TEMPDDN(8),DDN2-S992(R1) SAVE THE DDNAME
         MVC   INPUT+4Ø(8),TEMPDDN      MOVE IN THE DDNAME
         CLC   SSS2MLRL(2),=H'Ø'        MAX LOGICAL RECORD LENGTH=Ø?
         BNH   INDONE                   NO - JUST SKIP THIS ONE
         MVC   INPUT+82(2),SSS2MLRL     MOVE IN THE LRECL
         OPEN  (INPUT,INPUT),MODE=31
*********************************************************************
*   BUILD AND WRITE OUT A SEPARATOR RECORD SO THAT THE OUTPUT FROM    *
*   DIFFERENT DD'S CAN BE IDENTIFIED IN THE EXTERNAL WRITER OUTPUT    *
*   FILE.  THE SEPARATOR RECORD (HDRREC1) USED BY THIS PROGRAM        *
*   USES JOBNAME, JOB NUMBER, PROC NAME, STEP NAME, DD NAME,          *
*   OUTPUT CLASS, AND DS NAME INFORMATION.                            *
*                                                                     *
*   THE SAPI SSI CALL RETURNS MUCH USEFUL INFORMATION IN THE SSS2.    *
*   OTHER POTENTIALLY USEFUL FIELDS THAT COULD BE USED FOR            *
*   SEPARATOR INFORMATION MIGHT INCLUDE THE SYSTEM THE JOB RAN        *
*   ON (SSS2SYS), THE JES MEMBER THE JOB RAN ON (SSS2MBR), AND THE    *
*   TIME OF DAY THE OUTPUT DATASET WAS MADE AVAILABLE TO SPOOL        *
*   (SSS2TOD).                                                        *
*********************************************************************
         LA    RØ,RECBUFF               GET BUFFER ADDRESS
         LA    R14,RECBUFF              GET BUFFER ADDRESS
         L     R1,=F'32760'             SET LENGTH
         XR    R15,R15                  SET FILL BYTE
         MVCL  RØ,R14                   CLEAR THE BUFFER
         MVC   RECBUFF(133),HDRREC1     MOVE IN THE SEPARATOR RECORD
         MVC   RECBUFF+JBNMOFF(8),SSS2JOBR MOVE IN THE JOBNAME
         MVC   RECBUFF+JB#OFF(8),SSS2JBIR  MOVE IN THE JOB#
         MVC   RECBUFF+PRCOFF(8),SSS2PRCD  MOVE IN THE PROCNAME
         MVC   RECBUFF+STPOFF(8),SSS2STPD  MOVE IN THE STEPNAME
         MVC   RECBUFF+DDNDOFF(8),SSS2DDND MOVE IN THE DDNAME
         MVC   RECBUFF+CLSOFF(1),SSS2CLAR  MOVE IN THE CLASS
         MVC   RECBUFF+DSNMOFF(44),SSS2DSN MOVE IN THE DSNAME
         XC    LRECL(4),LRECL           CLEAR LRECL AREA
         MVC   LRECL(2),=H'137'         SET LENGTH
         TM    OUTPUT+36,X'8Ø'          FIXED LENGTH RECORDS?
         BO    FIXED1                   YES - SET TO FIXED OUTPUT
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
         TM    OUTPUT+36,X'40'          VARIABLE LENGTH RECORDS?
         BO    VARIABL1                 YES - SET TO VARIABLE OUTPUT
         B     RETURN16                 SOMETHING'S NOT RIGHT
FIXED1   EQU   *
         LA    R8,RECBUFF               SET BUFFER ADDRESS
         B     OUTPUT1                  GO WRITE THE OUTPUT
VARIABL1 EQU   *
         LA    R8,LRECL                 SET BUFFER ADDRESS
OUTPUT1  EQU   *
         PUT   OUTPUT,(R8)
*********************************************************************
*   THIS SAMPLE PROGRAM HAS TAKEN A VERY SIMPLE APPROACH TO HANDLING  *
*   THE SYSOUT OUTPUT DIRECTED TO ITS EXTERNAL WRITER.  IT USES A     *
*   SEQUENTIAL OUTPUT DATASET (EITHER F(B) OR V(B)) THAT CAN BE       *
*   EITHER A REAL DATASET OR A SYSOUT DATASET ON SPOOL.  IF YOU USE   *
*   A REAL DATASET, DEFINE THE DATASET WITH AN LRECL AT LEAST AS      *
*   LARGE AS THE MAXIMUM RECORD SIZE THAT WILL BE PROCESSED BY THE    *
*   EXTERNAL WRITER.  IF YOU DIRECT THE OUTPUT DD TO SYSOUT, THE      *
*   SYSOUT DATASET WILL GET CREATED WITH AN LRECL OF 32760.           *
*                                                                     *
*   THERE ARE MANY POSSIBILITIES AVAILABLE.  AT THIS POINT IN THE     *
*   PROGRAM, THE INPUT DD IS OPENED TO A SYSOUT SPOOL DATASET THAT    *
*   HAS BEEN CREATED WITH OUTPUT DIRECTED TO THIS EXTERNAL WRITER.    *
*   HOW YOU CHOOSE TO PROCESS THIS INPUT DATASET IS LEFT TO YOUR      *
*   IMAGINATION.                                                      *
*********************************************************************
GETLP1   EQU   *
         LA    R0,RECBUFF               GET BUFFER ADDRESS
         LA    R14,RECBUFF              GET BUFFER ADDRESS
         L     R1,=F'32760'             SET LENGTH
         XR    R15,R15                  SET FILL BYTE
         MVCL  R0,R14                   CLEAR THE BUFFER
         GET   INPUT,RECBUFF            READ INPUT RECORD
         XC    LRECL(4),LRECL           CLEAR LRECL AREA
         XR    R1,R1                    CLEAR R1
         ICM   R1,B'0011',SSS2MLRL      GET RECORD LENGTH
         LA    R1,4(,R1)                ADD 4 FOR RDW
         STCM  R1,B'0011',LRECL         SAVE THE LENGTH
         CLC   LRECL(2),=H'32756'       LENGTH OK?
         BNH   LRECLOK                  YES - DON'T ADJUST
         MVC   LRECL(2),=H'32760'       SET TO MAX
LRECLOK  EQU   *
         TM    OUTPUT+36,X'80'          FIXED LENGTH RECORDS?
         BO    FIXED2                   YES - SET TO FIXED OUTPUT
         TM    OUTPUT+36,X'40'          VARIABLE LENGTH RECORDS?
         BO    VARIABL2                 YES - SET TO VARIABLE OUTPUT
         B     RETURN16                 SOMETHING'S NOT RIGHT
FIXED2   EQU   *
         LA    R8,RECBUFF               SET BUFFER ADDRESS
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
         B     OUTPUT2                  GO WRITE THE OUTPUT
VARIABL2 EQU   *
         LA    R8,LRECL                 SET BUFFER ADDRESS
OUTPUT2  EQU   *
         PUT   OUTPUT,(R8)
         B     GETLP1                   GET NEXT INPUT RECORD
INDONE   EQU   *
         CLOSE (INPUT),MODE=31
*********************************************************************
         LA    RØ,DYNPRMS4              GET TARGET AREA ADDRESS
         L     R1,=A(S994LN)            GET THE LENGTH
         LA    R14,S994                 GET SOURCE AREA ADDRESS
         LR    R15,R1                   GET THE LENGTH
         MVCL  RØ,R14                   MOVE IN THE MODEL
         LA    R1,DYNPRMS4              GET PARM AREA ADDRESS
         LA    R2,S994RB-S994(,R1)      GET RELOCATED S994RB ADDRESS
         O     R2,=X'8ØØØØØØØ'          SET FLAG
         ST    R2,Ø(,R1)                SAVE RELOCATED ADDRESS IN PARMS
         LA    R2,S994TUPL-S994(,R1)    GET RELOCATED S994TUPL ADDRESS
         STCM  R2,B'1111',S994TXTP-S994(R1) SV RELOCATED S994TUPL ADR
         LA    R2,TU4ØØ1-S994(,R1)      GET RELOCATED TU4ØØ1 ADDRESS
         O     R2,=X'8ØØØØØØØ'          SET LAST TU FLAG
         STCM  R2,B'1111',S994TUPL-S994(R1) SV RELOCATED TU4ØØ1 ADDR
         MVC   DDN4-S994(8,R1),TEMPDDN  COPY THE DDNAME
         SVC   99                       DEALLOCATE THE DATASET
         LTR   R15,R15                  DEALLOCATE OK?
         BZ    DALLOC2OK                YES - KEEP GOING
         LA    R1,DYNPRMS4              GET PARM AREA ADDRESS
         MVC   DBL2(4),S994ERR-S994(R1) COPY RETURN CODE
         UNPK  DBL1(9),DBL2(5)          UNPACK IT
         NC    DBL1(8),=8X'ØF'          TURN OFF HIGH NIBBLE
         TR    DBL1(8),=C'Ø123456789ABCDEF' MAKE IT READABLE
         MVC   DALOCMSG+17(8),DBL1      MOVE IN RETURN CODE
         MVC   DALOCMSG+43(44),SSS2DSN  COPY THE DSNAME
         PUT   SYSPRINT,DALOCMSG        WRITE THE MESSAGE
         ABEND 11                       ABEND
DALLOC2OK EQU   *
         B     SSICALL2                 GO TRY AGAIN
SAPIDONE EQU   *
         LA    R14,SSS2INPT             GET AREA ADDRESS
         L     R1,=A(SSS2DISP-SSS2INPT) GET LENGTH
         LR    RØ,R14                   COPY ADDRESS
         XR    R15,R15                  SET FILL BYTE
         MVCL  RØ,R14                   CLEAR THE AREA
         LA    R14,SSS2DISP             GET AREA ADDRESS
         L     R1,=A(SSS2OUTP-SSS2DISP) GET LENGTH
         LR    RØ,R14                   COPY ADDRESS
         XR    R15,R15                  SET FILL BYTE
         MVCL  RØ,R14                   CLEAR THE AREA
         OI    SSS2MSC1,SSS2CTRL        SET TERMINATION FLAG
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
         MODESET MODE=SUP
         LA    R1,SSOBPTR               POINT TO SSOB PTR
         IEFSSREQ ,                     MAKE SUBSYSTEM REQUEST
         LR    R9,R15                   SAVE THE RETURN CODE
         MODESET MODE=PROB
         LTR   R9,R9                    IEFSSREQ WAS OK?
         BZ    RETNCHK                  YES - CHECK SSOBRETN
         ST    R9,DBL2                  COPY THE RETURN CODE
         UNPK  DBL1(9),DBL2(5)          UNPACK IT
         NC    DBL1(8),=8X'ØF'          TURN OFF HIGH NIBBLE
         TR    DBL1(8),=C'Ø123456789ABCDEF' MAKE IT READABLE
         MVC   TERMMSG1+38(8),DBL1      MOVE IN THE RETURN CODE
         PUT   SYSPRINT,TERMMSG1        ISSUE MSG
         B     EXIT                     WE'RE DONE
RETNCHK  EQU   *
         CLC   SSOBRETN(4),=F'Ø'        IEFSSREQ WENT AS EXPECTED?
         BE    EXIT                     YES - WE'RE DONE
         MVC   DBL2(4),SSOBRETN         SAVE RETURN CODE
         UNPK  DBL1(9),DBL2(5)          UNPACK IT
         NC    DBL1(8),=8X'ØF'          TURN OFF HIGH NIBBLE
         TR    DBL1(8),=C'Ø123456789ABCDEF' MAKE IT READABLE
         CLI   SSOBRETN+3,X'ØØ'
         BE    EXIT
         MVC   TERMMSG2+44(8),DBL1      MOVE IN THE RETURN CODE
         PUT   SYSPRINT,TERMMSG1        ISSUE MSG
         B     EXIT                     WE'RE DONE
*********************************************************************
EXIT     EQU   *
*********************************************************************
         MVI   RECBUFF,C' '             SET FILL CHARACTER
         MVC   RECBUFF+1(132),RECBUFF   CLEAR OUT RECORD AREA
         MVC   RECBUFF(L'MSG99),MSG99   MOVE IN THE MESSAGE
         MVC   RECBUFF+27(8),WTRNM      MOVE IN THE WRITER NAME
         PUT   SYSPRINT,RECBUFF         WRITE THE MESSAGE
*********************************************************************
         CLOSE (OUTPUT),MODE=31
         CLOSE (SYSPRINT),MODE=31
         LR    R1,R13                   COPY WORKING STORAGE ADDRESS
         L     R1Ø,4(,R13)              SAVE OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)           GET STORAGE LENGTH
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1) RELEASE IT
         LR    R13,R1Ø                  RESTORE SAVEAREA ADDRESS
         LM    R14,R12,12(R13)          RESTORE REGISTERS
         XR    R15,R15                  SET RETURN CODE
         BR    R14                      RETURN
*********************************************************************
RETURNØ8 EQU   *
         PUT   SYSPRINT,PARMMSGØ        ISSUE BAD PARM MSG
         PUT   SYSPRINT,PARMMSG1        ISSUE BAD PARM MSG
         PUT   SYSPRINT,PARMMSG2        ISSUE BAD PARM MSG
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
         CLOSE (SYSPRINT),MODE=31
         LR    R1,R13                  COPY WORKING STORAGE ADDRESS
         L     R1Ø,4(,R13)             SAVE OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)          GET STORAGE LENGTH
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1) RELEASE IT
         LR    R13,R1Ø                 RESTORE SAVEAREA ADDRESS
         LM    R14,R12,12(R13)         RESTORE REGISTERS
         LA    R15,8                   SET RETURN CODE
         BR    R14                     RETURN
*********************************************************************
RETURN12 EQU   *
         CLOSE (SYSPRINT),MODE=31
         CLOSE (OUTPUT),MODE=31
         LR    R1,R13                  COPY WORKING STORAGE ADDRESS
         L     R1Ø,4(,R13)             SAVE OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)          GET STORAGE LENGTH
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1) RELEASE IT
         LR    R13,R1Ø                 RESTORE SAVEAREA ADDRESS
         LM    R14,R12,12(R13)         RESTORE REGISTERS
         LA    R15,12                  SET RETURN CODE
         BR    R14                     RETURN
*********************************************************************
RETURN16 EQU   *
         PUT   SYSPRINT,RECFMSG1       ISSUE RECORD FORMAT MESSAGE
         CLOSE (OUTPUT),MODE=31
         CLOSE (SYSPRINT),MODE=31
         LR    R1,R13                  COPY WORKING STORAGE ADDRESS
         L     R1Ø,4(,R13)             SAVE OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)          GET STORAGE LENGTH
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1) RELEASE IT
         LR    R13,R1Ø                 RESTORE SAVEAREA ADDRESS
         LM    R14,R12,12(R13)         RESTORE REGISTERS
         LA    R15,8                   SET RETURN CODE
         BR    R14                     RETURN
*********************************************************************
RETURN2Ø EQU   *
         PUT   SYSPRINT,JFCBMSG1       ISSUE JFCB LOCATE FAIL MESSAGE
         CLOSE (SYSPRINT),MODE=31
         LR    R1,R13                  COPY WORKING STORAGE ADDRESS
         L     R1Ø,4(,R13)             SAVE OLD SAVEAREA ADDRESS
         L     R9,=A(WORKLEN)          GET STORAGE LENGTH
         STORAGE RELEASE,LENGTH=(R9),ADDR=(R1) RELEASE IT
         LR    R13,R1Ø                 RESTORE SAVEAREA ADDRESS
         LM    R14,R12,12(R13)         RESTORE REGISTERS
         LA    R15,8                   SET RETURN CODE
         BR    R14                     RETURN
*********************************************************************
MSGPUT   EQU   *
         ST    R14,R14SAVE             SAVE THE RETURN ADDRESS
         MVI   RECBUFF,C' '            SET FILL CHARACTER
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
         MVC   RECBUFF+1(132),RECBUFF   INITIALIZE OUTPUT RECORD AREA
         BCTR  R15,Ø                    REDUCE LENGTH BY ONE FOR EX
         EX    R15,MSGMVC               MOVE THE OUTPUT DATA
         PUT   SYSPRINT,RECBUFF         WRITE THE OUTPUT RECORD
         L     R14,R14SAVE              LOAD RETURN ADDRESS
         BR    R14                      RETURN
MSGMVC   MVC   RECBUFF(*-*),Ø(R1)       MOVE OUTPUT DATA
*********************************************************************
CHARCHK  EQU   *
         CLI   Ø(R9),C'A'               ALPHANUMERIC?
         BL    CHARBAD                  NO - ISSUE MESSAGE AND END
         CLI   Ø(R9),C'I'               ALPHANUMERIC?
         BNH   CHAROK                   YES - SAVE IT
         CLI   Ø(R9),C'J'               ALPHANUMERIC?
         BL    CHARBAD                  NO - ISSUE MESSAGE AND END
         CLI   Ø(R9),C'R'               ALPHANUMERIC?
         BNH   CHAROK                   YES - SAVE IT
         CLI   Ø(R9),C'S'               ALPHANUMERIC?
         BL    CHARBAD                  NO - ISSUE MESSAGE AND END
         CLI   Ø(R9),C'Z'               ALPHANUMERIC?
         BNH   CHAROK                   YES - SAVE IT
         CLI   Ø(R9),C'Ø'               ALPHANUMERIC?
         BL    CHARBAD                  NO - ISSUE MESSAGE AND END
         CLI   Ø(R9),C'9'               ALPHANUMERIC?
         BH    CHARBAD                  NO - ISSUE MESSAGE AND END
CHAROK   EQU   *
         XR    R15,R15                  SET RETURN CODE
         BR    R14                      RETURN TO MAINLINE
CHARBAD  EQU   *
         LA    R15,8                    SET RETURN CODE
         BR    R14                      RETURN TO MAINLINE
*********************************************************************
CLASSMVC MVC   CLASSBUF(*-*),CIBDATA+1Ø COPY CLASS LIST
*********************************************************************
* REGISTER EQUATES
RØ       EQU   Ø
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
R1Ø      EQU   1Ø
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
R15      EQU   15
*************************************************************************
JBNMOFF  EQU   5
JB#OFF   EQU   2Ø
PRCOFF   EQU   35
STPOFF   EQU   5Ø
DDNDOFF  EQU   65
CLSOFF   EQU   8Ø
DSNMOFF  EQU   88
HDRREC1  DC    C'JBN: XXXXXXXX  JB#: XXXXXXXX  PRC: XXXXXXXX  '
HDRREC2  DC    C'STP: XXXXXXXX  DDN: XXXXXXXX  CLS: X  '
         DC    CL(133-(L'HDRREC1+L'HDRREC2))'DSN:  '
*************************************************************************
PARMMSGØ DC    CL133'XWTRØ00I - INVALID INPUT PARAMETER FORMAT'
PARMMSG1 DC    C'XWTRØ01I - VALID PARMS ARE WRTNAME OR '
         DC    CL(133-L'PARMMSG1)'WRTNAME,CLASSLIST'
PARMMSG2 DC    C'XWTRØ02I - A VALID WRTNAME IS 1-8 ALPHANUMERIC '
PARMM2B  DC    C'CHARACTERS.  CLASSLIST IS 1-36 CHARACTERS A-Z, Ø-9.'
         DC    CL(133-(L'PARMMSG2+L'PARMM2B))' '
*************************************************************************
ALLOCMSG DC    CL133'XWTRØ15I - ERROR XXXXXXXX ALLOCATING DSN '
DALOCMSG DC    CL133'XWTRØ16I - ERROR XXXXXXXX DEALLOCATING DSN '
*************************************************************************
TERMMSG1 DC    CL133'XWTRØ13I - SAPI TERMINATION FAILED RC=XXXXXXXX'
TERMMSG2 DC CL133'XWTRØ14I - SAPI TERMINATION FAILED SSOBRETN=XXXXXXXX'
*************************************************************************
RECFMSG1 DC    CL133'XWTRØ19I - OUTPUT DATASET WAS NOT F(B) OR V(B)'
*************************************************************************
JFCBMSG1 DC    CL133'XWTRØ2Ø I - NO OUTPUT DATASET JFCB LOCATED'
*************************************************************************
MSG3     DC    C'XWTRØ03I - SAPI INVALID SEARCH ARGUMENTS'
MSG4     DC    C'XWTRØ04I - SAPI UNABLE TO PROCESS NOW'
MSG5     DC    C'XWTRØ05I - SAPI DUPLICATE JOBNAMES'
MSG6     DC    C'XWTRØ06I - SAPI INVALID DESTINATION SPECIFIED'
MSG7     DC    C'XWTRØ07I - SAPI AUTHORIZATION FAILED'
MSG8     DC    C'XWTRØ08I - SAPI TOKEN MAP FAILED'
MSG9     DC    C'XWTRØ09I - SAPI LOGIC ERROR'
MSG1Ø    DC    C'XWTRØ1ØI - SAPI INVALID CLASS'
MSG11    DC    C'XWTRØ11I - SAPI BAD DISPOSITION SETTINGS'
MSG12    DC    C'XWTRØ12I - SAPI DATASET DISPOSITION NOT UNIFORM'
MSG17    DC    C'XWTRØ17I - SAPI REQUEST ABENDED.  XWTR TERMINATING.'
MSG18    DC    C'XWTRØ18I - SAPI UNKNOWN SSOBRETN XXXXXXXX.  '
         DC    C'XWTR TERMINATING.'
MSG18L   EQU   *-MSG18
MSG98    DC    C'XWTRØ98I - EXTERNAL WRITER XXXXXXXX INITIALIZED.'
MSG99    DC    C'XWTRØ99I - EXTERNAL WRITER XXXXXXXX TERMINATED.'
*************************************************************************
INPUT    DCB   MACRF=(GM),DDNAME=TEMP,LRECL=133,DSORG=PS,EODAD=INDONE
OUTPUT   DCB   MACRF=(PM),DDNAME=OUTPUT,DSORG=PS
SYSPRINT DCB   MACRF=(PM),DDNAME=SYSPRINT,LRECL=133,DSORG=PS
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
*********************************************************************
EXTLST   EXTRACT *-*,MF=L
EXTLN    EQU   *-EXTLST
*********************************************************************
*   THE STATIC COPY OF THE DYNAMIC ALLOCATION PARAMETER LIST
*   FOR ALLOCATING A JES DATASET THAT RESIDES ON SPOOL AND WHOSE
*   OUTPUT HAS BEEN DIRECTED THERE BY AN EXTERNAL WRITER
S992     DC   A(X'80000000'+S992RB)
S992RB   DC   X'14'
S992VERB DC   X'01'
S992FLG1 DC   X'0000'
S992ERR  DC   X'0000'
S992INFO DC   X'0000'
S992TXTP DC   AL4(S992TUPL)
         DC   XL4'00'
S992FLG2 DC   XL4'00'
S992TUPL DC   AL4(TU2001)
TU202    DC   AL4(TU2002)
TU203    DC   AL4(TU2003)
TU204    DC   AL4(0)
TU2001   DC   X'0055',X'0001',X'0008'          //DDN2     DD
DDN2     DC   CL8' '
TU2002   DC   X'0002',X'0001',X'002C'          DSN=SSS2DSN
DSN2     DC   CL44' '
TU2003   DC   X'005C',X'0001',X'0004',C'    '  INDICATE WHICH SUBSYS
S992LN   EQU  *-S992
*********************************************************************
*   THE STATIC COPY OF THE DYNAMIC UNALLOCATION PARAMETER LIST
*   FOR UNALLOCATING THE JES EXTERNAL WRITER DATASET
S994     DC   A(X'80000000'+S994RB)
S994RB   DC   X'14'
S994VERB DC   X'02'
S994FLG1 DC   X'0000'
S994ERR  DC   X'0000'
S994INFO DC   X'0000'
S994TXTP DC   AL4(S994TUPL)
         DC   XL4'00'
S994FLG2 DC   XL4'00'
S994TUPL DC   AL4(TU4001)
TU4001   DC   X'0001',X'0001',X'0008'          //DDN4     DD
DDN4     DC   CL8' '
S994LN   EQU  *-S994
*********************************************************************
         LTORG
* SAVEAREA AND OTHER PROGRAM STORAGE
WORKAREA DSECT
SAVEAREA DS    18F
R14SAVE  DS    F
DYNPRMS2 DS    0D,CL(S992LN)
DYNPRMS4 DS    0D,CL(S994LN)
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
TIOTADDR DS    F
SSOBPTR  DS    F
REPLYECB DS    F
SELECB   DS    F
DBL1     DS    2D
DBL2     DS    2D
SSNMSAVE DS    CL8
WTRNM    DS    CL8
CLASSLST DS    CL36
REPLY    DS    CL1
TEMPDDN  DS    CL8
SSOBAREA DS    ØD,CL(SSOBHSIZ+SSS2SIZE)
CLASSBUF DS    CL36
*************************************************************************
COMMADDR DS    F
ECBLIST  DS    2F
EXTWRK   DS    ØD,CL(EXTLN)
*************************************************************************
SWEPAPTR DS    F
EPA      DS    CL28
SWAPARMS SWAREQ MF=L
*************************************************************************
LRECL    DS    CL4
RECBUFF  DS    CL32760
WORKLEN  EQU   *-WORKAREA
*************************************************************************
         PRINT NOGEN
         IEFSSOBH
SSOBGN   EQU   *
         PRINT GEN
         IAZSSS2 DSECT=YES
         PRINT NOGEN
         IEFJESCT TYPE=DSECT
         CVT   DSECT=YES
         IKJTCB
         IEZJSCB
         IEFJSSIB
MAPCOM   DSECT
         IEZCOM
MAPCIB   DSECT
CIB      IEZCIB
MAPJFCB  DSECT
         IEFJFCBN LIST=YES
         IEFZB505
TIOT     DSECT
         IEFTIOT1
         END
```


© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

Matching a filename against a pattern

The following REXX program was created as a subroutine for several other procedures. It matches a filename against a generic pattern, according to the ISPF conventions. It needs two arguments, separated by spaces – one is the filename, the other is the pattern. The order in which they are specified is not important. The EXEC returns 0 if the filename fits the pattern, -1 otherwise. You can also execute the EXEC directly (not as a subroutine), in which case the return code is displayed on the screen.

A pattern is a string containing literal characters, single asterisks (*), double asterisks (**), and percentage signs (%). It can also contain dots (.) separating the dataset qualifiers. Their meaning is as follows:

- % – only one character.
- * – any number of characters, including none. If it stands alone on a qualifier (.*.), then it means a qualifier must exist in that position.
- ** – any number of qualifiers, including none, if it stands alone on a qualifier (.**.).

For example, the following are valid patterns. In front of each pattern I also put a string to check the filename and indicate whether they match (0) or not (-1):

```
TH*.IS.%.T%*          THIS.IS.A.TEST              Ø
TH%%.*.AL*.A*         THIS.IS.ALSO.A.TEST         Ø
THIS.**.GOOD          THIS.IS.ALSO.GOOD           Ø
A%%.HERE%%.*          AND.HERE.IS.A.FAILURE      -1
AND.*.*.*             AND.HERE                   -1
AND*NOT*ENAME         ANDTHISISNOTAFILENAME       Ø
```

The program does not check for valid filenames. The strings passed can be almost anything, as long as only one of them is a pattern, which means it contains '*' or '%'. Also, the strings need not be separated into qualifiers – a long string of characters will be accepted as well. The only restrictions that apply are that the string cannot contain spaces, '?', or ';'. This is because the EXEC uses them internally. When the string is dot-separated, as filenames are, each qualifier is matched independently.

© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

The algorithm used in the EXEC is probably not very conventional, but works very well. Basically, for each qualifier, it transforms the pattern of literals and wildcards into a string of literals, variables, and positions. This transformed pattern is used as a template for REXX's PARSE VAR instruction. After the execution, by means of an INTERPRET, each variable receives a value from the string being checked. If there is a match, replacing these variables by their values, in conjunction with any literals that exist, will equal the checked name.

This is just an outline of what I do; the real process is a little more complicated, because there are minimum lengths to take care of (%%%% must match exactly four characters, while %%%%* must match four or more), there are relative positions to be considered, etc. The only drawback with this approach is that the algorithm produced is not directly translatable to other programming languages that lack REXX facilities. But this is a REXX EXEC, created as support for other EXECs, and so I decided to exploit its potential.

```
/* REXX MVS *============================================================*/
/*                                                                      */
/* PATTERN - This program matches a filename to a generic pattern.      */
/* It should be used as a subroutine. The entry parameter is a          */
/* string containing 2 words, the filename to check and the pattern     */
/* separated by one or more spaces. Their order is irrelevant.          */
/*                                                                      */
/* Filename is a fully-qualified dataset name                           */
/* Pattern follows the ISPF style convention:                           */
/*    *   - any number of characters                                    */
/*    %   - one character                                              */
/*   .*.  - one qualifier                                               */
/*   .**. - any number of qualifiers (including none)                   */
/*                                                                      */
/* RC: Ø if there is a match, -1 otherwise.                             */
/* If this program is called as a subroutine, then RC is returned,      */
/* otherwise RC is "said".                                              */
/*                                                                      */
/*======================================================================*/

arg arg1 arg2 .

if pos("*",arg1) > Ø | pos("%",arg1) > Ø then do
   file = arg2
   pattern = arg1
end
else do
   file = arg1
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
   pattern = arg2
end
any = Ø
returncode = Ø

do alpha = Ø
   parse var pattern p1 "." pattern
   if p1 = "" then p1 = ";"
   if p1 = "**" then do
      if pattern = "" then do
         leave alpha
      end
      else do
         any = 1
         iterate alpha
      end
   end
   if right(p1,1) = "*" & pattern = "" & any = Ø then do
      any = 2
   end
   do beta = Ø
      parse var file f1 "." file
      if f1 = "" then f1 = ";"
      if f1 = ";" & p1 = ";" then leave alpha
      call check_qualifier p1 f1
      returncode = result
      if any = Ø then do
         if returncode = Ø then do
            iterate alpha
         end
         else do
            leave alpha
         end
      end
      if any = 1 then do
         if returncode = Ø then do
            if right(p1,1) = "*" & pattern = "" then do
               leave alpha
            end
            else do
               any = Ø
            end
            iterate alpha
         end
         else do
            if f1 = ";" then do
               leave alpha
            end
            else do
               iterate beta
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
            end
         end
      end
      if any = 2 then do
         leave alpha
      end
   end
end

parse source . calltype .
if calltype = "COMMAND" then say returncode
else return returncode

exit

/*====================================================================*/

check_qualifier: procedure

arg str1 str2

if str2 = ";" then return -1
if str1 = ";" then return -1
if str1 = "*" then return  Ø

v = Ø
str3 = ""
prv  = ""
no_pos = Ø

do forever
   p = pos("*%",str1)
   if p = Ø then leave
   str1 = overlay("%*",str1,p)
end

do forever
   p = pos("%*%",str1)
   if p = Ø then leave
   str1 = overlay("%%*",str1,p)
end

do forever
   p = pos("**",str1)
   if p = Ø then leave
   str1 = delstr(str1,p,1)
end

do k = 1 to length(str1)
  select
```

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

```
    when substr(str1,k,1) = '%' then do
       do k1 = k + 1 to length(str1)
          if substr(str1,k1,1) <> '%' then leave k1
       end
       v = v + 1
       if no_pos = 1 then do
          str3 = str3"?var."v
       end
       else do
          str3 = str3"?="k"?var."v
       end
       if substr(str1,k1,1) <> '*' then do
          if no_pos = 1 then do
             str3 = str3"?"
          end
          else do
             str3 = str3"?="k1"?"
          end
          len_eq.v = k1 - k
          len_ge.v = 0
          k = k1 - 1
       end
       else do
          len_ge.v = k1 - k
          len_eq.v = 0
          k = k1
       end
    end
    when substr(str1,k,1) = '*' then do
       no_pos = 1
       v = v + 1
       str3 = str3"?var."v"?"
       len_eq.v = 0
       len_ge.v = 0
    end
    otherwise do
       str3 = str3"?'"
       do k1 = k to length(str1)
          if substr(str1,k1,1) = '%' |,
             substr(str1,k1,1) = '*' then leave k1
          str3 = str3||substr(str1,k1,1)
       end
       str3 = str3"'"
       k = k1 - 1
    end
  end
end

str3 = space(str3,0)
str3 = translate(str3," ","?")
```


© 2002. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 438 0290.

```
interpret "parse var" str2 str3

ww = words(str3)
do w = 1 to ww
   if left(word(str3,w),1)="=" then,
   str3 = delword(str3,w,1)
end

str3 = space(str3,0)
interpret "string = value("str3")"
if string <> str2 then do
   answer = -1
end
else do
   answer = 0
   do v1 = 1 to v
      if len_ge.v1 > 0 then do
         if length(var.v1) <  len_ge.v1 then answer = -1
      end
      if len_eq.v1 > 0 then do
         if length(var.v1) <> len_eq.v1 then answer = -1
      end
   end
end

return answer
```

Systems Programmer
(Portugal)

© Xephon 2002

Why not share your expertise and earn money at the same time? *MVS Update* is looking for technical articles and hints and tips that experienced MVS users have written to make their life, or the lives of their users, easier. We would also be interested in articles about performance and tuning.

We will publish it (after vetting by our expert panel) and send you a cheque when the article is published. Articles can be of any length and can be sent or e-mailed to Trevor Eddolls at any of the addresses shown on page 2. Why not call or write now for a free copy of our *Notes for contributors*? Alternatively, you can now download a copy from our Web site. Point your browser at www.xephon.com/nfc.

© 2002. Reproduction prohibited. Please inform Xephon of any infringement.

Data 21 has announced JES2Mail and JES2FTP. JES2Mail is a mainframe-based tool that transforms print files into formats such as Adobe PDF, HTML, RTF, and comma-delimited, and 'pushes' them directly to end-users via e-mail. JES2FTP automatically publishes JES reports on a Web site as HTML or Adobe PDF documents. It also sends e-mail notifications (with links to the reports!) to end users.

For further information contact:
Data 21, 3510 Torrance Blvd, Suite 300, Torrance, CA 90503, USA.
Tel: (310) 792 1771.
URL: http://www.data21.com/products/jes2ftp.asp.

* * *

MacKinney Systems has announced VTAM Virtual Printer (VVP) Release 1.3, which is used to route print from CICS TS to the JES queue for OS/390 and z/OS or to the POWER Queue for VSE.

VVP is said to require no programming changes and it runs as a VTAM application. All printer maintenance may be done through on-line panels.

For further information contact:
Mackinney Systems, 2740 South Glenstone, Suite 103, Springfield, MI 65804, USA.
Tel: (417) 882 8012.
URL: http://www.mackinney.com/news.htm.

* * *

Compuware has announced general availability of its STROBE 2.5 and introduced iSTROBE, a new browser-based tool for analysing application performance information.

STROBE 2.5 introduces new functions that measure abnormal program behaviour. iSTROBE enables interactive analysis of enterprise applications based on WebSphere MQ, CICS/TS, Visual Age for Java, DB2, IMS, and Unix System Services.

STROBE enables sites to determine where and how applications use resources in OS/390 and z/OS environments. Incorporating STROBE measurement into key phases of the application life cycle such as development, test, and production is designed to help ensure that applications run efficiently and responsively and that no performance problems are unintentionally introduced.

Utilizing STROBE's performance profiles, iSTROBE is said to identify the performance characteristics of complex application problems. It augments STROBE by increasing and simplifying the identification of performance opportunities within the STROBE Performance Profile.

For further information contact:
Compuware, 31440 Northwestern Highway, Farmington Hills, MI 48334-2564, USA.
Tel: (248) 737 7300.
URL: http://www.compuware.com/products/strobe/istrobe.

* * *

For a limited time, the purchase of selected Tivoli NetView for z/OS Version 5.1 licences will qualify the buyer for a two-day on-site Tivoli NetView Migration and Planning Assistance Package that has an approximate value of $6,000.

For further information contact your local IBM representative.
URL: http://www.tivoli.com.