



231

MVS

December 2005

In this issue

- [3 Automatic jobcard](#)
 - [6 REXX conversion to/from packed data](#)
 - [13 JES2 RJE and NJE reporting](#)
 - [29 Catalog activity reporting with ICETOOL](#)
 - [41 Using SPLICE, a DFSORT feature, to group information together using a large group of key values](#)
 - [49 A glimpse at the WLM's predictions – part 2](#)
 - [73 MVS news](#)
-

update

© Xephon Inc 2005

MVS Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *MVS Update*, comprising twelve monthly issues, costs \$505.00 in the USA and Canada; £340.00 in the UK; £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 2001 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.

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 \$160 (£100 outside North America) per 1000 words and \$80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of \$32 (£20) per 100 lines. 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 Inc 2005. 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.

Printed in England.

Automatic jobcard

PROGRAM AIM

This program is designed to automatically insert a jobcard during JCL creation.

The ISPF macro is written using REXX, and is fast and easy to use.

THE ISPF MACRO

I have called the macro JC, but you can change the name to suit your standards.

```
/* REXX */
/* ONLY FOR DEBUG
CJNUM = CNTJNUM
ADDRESS ISPEXEC "VPUT (CJNUM) PROFILE"
*/
STMT2="//          NOTIFY="USERID()",REGION=8M,CLASS=A"
STMTC="//*** JOB CARD GENERATED AUTOMATICALLY ***      "
CNTJNUM = 0
ADDRESS ISPEXEC
"ISREDIT MACRO (PGMRNAME)" /* PGMRNAME: PARAMETER PASSED TO THE MACRO */
"ISREDIT (MBR) = MEMBER"
IF LENGTH(PGMRNAME) < 1 THEN DO          /* IF PGMRNAME NOT SPECIFIED */
    CALL GET_UNAME /* GET PGMRNAME */
    END
    ELSE DO
        IF LENGTH(PGMRNAME) > 18 THEN ,
            PGMRNAME = SUBSTR(PGMRNAME,1,18)
        END
    "VGET (CJNUM) PROFILE"
    CVAR = CJNUM
    IF DATATYPE(CVAR) <> "NUM" THEN
        DO
            CJNUM = CNTJNUM
            "VPUT (CJNUM) PROFILE"
        END
    CJNUM = CJNUM + 1
    IF CJNUM > 9 THEN DO
        CJNUM = 1
    END
```

```

STMT1 = "//"USERID()||CJNUM||,
        " JOB (ACCNT-INFO),'||PGMRNAME||'",MSGCLASS=R,"
"VPUT (CJNUM) PROFILE"
"ISREDIT LOCATE .ZFIRST"          /* TEST IF THE MEMBER IS EMPTY      */
IF RC <> 8 THEN "ISREDIT UP MAX" /* TOP OF THE SCREEN                */
'ISREDIT LINE_AFTER 0 =(STMT1)' /*PUT IN THE MBR FIRST JOBCARD PIECE */
'ISREDIT LINE_AFTER 1 =(STMT2)' /*PUT IN THE MBR SECOND JOBCARD PIECE*/
'ISREDIT LINE_AFTER 2 =(STMT3)' /*PUT IN THE MBR JOBCARD COMMENT    */
"ISREDIT HI JCL"                 /* HILITE JCL CONTROL STATEMENT     */
EXIT

GET_UNAME:
PSAOLD = STORAGE(D2X(548),4)      /* PTR PSA                          */
ASCBASXB = STORAGE(D2X(C2D(PSAOLD)+108),4) /* PTR ASXB                        */
ACEEADDR = STORAGE(D2X(C2D(ASCBASXB)+200),4) /* PTR ACEE                         */
USERID = STORAGE(D2X(C2D(ACEEADDR)+21),8) /* USER NAME                       */
LOGONGRP = STORAGE(D2X(C2D(ACEEADDR)+30),8) /* LOGON GROUP                      */
ACEEUNAM = STORAGE(D2X(C2D(ACEEADDR)+100),4) /* PTR USER NAME                   */
PGMRNAME = STORAGE(D2X(C2D(ACEEUNAM)+1),20) /* PROGRAMMER NAME                 */
PGMRNAME = STRIP(PGMRNAME)
RETURN

```

INSTALLATION

Copy the JC macro to the user SYSPROC library.

You can obtain the SYSPROC library list by typing this command from the TSO command line:

```
tso isrddn
```

Press F8 until you find the DDname SYSPROC.

On the right is the dataset name list allocated to the DDname SYSPROC.

Copy the JC macro into one of these libraries (User.lib.exec).

SET-UP

To follow your jobcard standards you must modify the following statements:

```
Line # 6 ->STMT2="//          NOTIFY="USERID()",REGION=8M,CLASS=A"
```

Modify the *REGION* and *CLASS* parameters appropriately:

```
Line # 30 and 31 -> STMT1 = "//"USERID()||CJNUM||,
```

```
" JOB (ACCNT-INFO),'||PGMRNAME||',MSGCLASS=R,"
```

Modify the *ACCNT-INFO* and the *MSGCLASS* parameter.

USAGE

The JC macro has only one parameter, the programmer name, and can be used with or without the parameter.

Edit a dataset, which can be full or empty, and type JC in the TSO command line. The jobcard will be inserted automatically.

JC without parameters:

```
EDIT          USER.LIB.TEST(mem01) - 01.73          Columns 00001 00072
Command ==> JC                                     Scroll ==> HALF
***** ***** Top of Data *****
.....
.....
.....
```

After you have pressed the *Enter* key, the following screen will be shown:

```
EDIT          USER.LIB.TEST(mem01) - 01.74          Columns 00001 00072
Command ==>                                         Scroll ==> HALF
***** ***** Top of Data *****
000001 //USERID01 JOB (ACCNT-INFO),'PGMRNAME',MSGCLASS=R,
000002 //          NOTIFY=USERID,REGION=8M,CLASS=A
000003 //*** JOB CARD GENERATED AUTOMATICALLY ***
```

JC with parameters:

```
EDIT          USER.LIB.TEST(mem01) - 01.73          Columns 00001 00072
Command ==> JC MY NAME                             Scroll ==> HALF
***** ***** Top of Data *****
.....
.....
.....
```

The parameter can have a maximum length of 18 characters.

If the length exceeds 18 characters, the macro automatically selects only 18.

After you have pressed the *Enter* key, the following screen will be shown:

```

EDIT          USER.LIB.TEST(mem01) - 01.74          Columns 00001 00072
Command ==>          Scroll ==> HALF
***** ***** Top of Data *****
000001 //USERID01 JOB (ACCNT-INFO),'MY NAME',MSGCLASS=R,
000002 //          NOTIFY=USERID,REGION=8M,CLASS=A
000003 //*** JOB CARD GENERATED AUTOMATICALLY ***

```

Every time that the job card is created, the number associated with the jobname increases by 1, ie:

- First time jobname – USERID01
- Second time jobname – USERID02
- Third time jobname – USERID03.

After nine times the counter is reset and the jobname number restarts from 01.

Magni Mauro
Systems Engineer (Italy)

© Xephon 2005

REXX conversion to/from packed data

Among the many built-in functions offered by REXX to manipulate data in different formats (eg B2C, X2C, D2X etc), there isn't one to convert packed decimal strings, in either direction.

Below are a couple of generalized REXX procedures that perform these tasks:

- P2D converts a string containing packed data to decimal readable form.
- D2P converts, conversely, a numeric character string to packed.

The procedures can be called either internally or externally and request two parameters (one of which is optional) to be passed during the call, as you can see in the following:

P2D

Packed string to be converted

Number of digits after the decimal point (optional, default zero)

D2P

Numeric character string to be converted

Length of resulting packed string (optional, default exact length without leading zeros, truncated on the left, if too short)

Here is an example of an external call to procedure P2D (the value in the packed string is X'0234567D'):

```
/* REXX program */
Trace 0
stringa = " 1'"
posiz = 5
dec_string = 0
call "P2D" stringa posiz
parse VAR RESULT dec01 retc1
say "Resulting string: " dec01
say "RetCode: " retc1
exit ;
```

And this is the result:

```
Resulting string: -2.34567
RetCode: 0
```

An example call to D2P:

```
/* REXX program */
Trace 0
stringa_dec = "-0.0000234"
numb = 3
call "D2P" stringa_dec numb
parse VAR RESULT pack01 retc1
if retc1 = 0 then
do;
  queue pack01
  numline = queued()
  "EXECIO "numline" DISKW DECSTR (OPEN FINIS)"
  if rc ^= 0 then
  do
    say "D2P: Error in writing output: rc="rc
    retc1 = rc
  end;
end;
exit retc1;
```

The result is obviously unreadable; in the output file you will find a packed string of three bytes, containing X'00234D'.

Here are the two REXX procedures.

P2D

```
/*          REXX          */
/* P2D procedure: converts a packed decimal string to printable form */
/* Input.: Packed data,  number of decimal digits (optional)      */
/* Output: Character string, return code                          */
/*          */
/* Personalization: set dec_symb to "," or to "." as needed      */
/*          */
/* When called internally, the following must be the first line: */
/* P2D: procedure expose dec_string P2rc;                        */
/* When called externally, the calling procedure must parse      */
/* a variable called "RESULT" to get the values of dec_string & P2rc */
/*          */
/* eg: Call "P2D"  packed_string numdecpos                       */
/*      Parse VAR RESULT dec01 retc1                             */
/*          */
Trace 0
P2rc = 0;
msg_err   = "Error!"
dec_symb  = ".";
dec_string = 0;
dec_numb  = 0;
Parse ARG pk_string comma_pos

/* analyse the input string */

if comma_pos = "" then
  comm_pos = 0;
if comma_pos < 0 | comma_pos > 31 then
do;
  say "P2D: comma_pos parameter invalid " comma_pos
  P2rc = 8
  return msg_err P2rc
end;
if pk_string = "" then
do;
  say "P2D: Input string missing "
  P2rc = 16
  return msg_err P2rc
end;
pk_length = length(pk_string);
if pk_length > 16 then
do;
  say "P2D: Packed data too long "
  P2rc = 12
  return msg_err P2rc
end;

/* load the packed digit table */
```



```

num_digits = (pk_length * 2) - 1;
do i = 1 to pk_length;
    x = pk_length - i + 1
    tabp.i = substr(pk_string,x,1)
end;
tabp.Ø = pk_length;

/* extract the single digits */
/*   in hexadecimal form   */

do i = 1 to Tabp.Ø;
    semi_left = BITAND(tabp.i,'FØ'x)
    Tab_hexl.i = C2D(semi_left)
    if i > 1 then
        do;
            semi_right = BITAND(tabp.i,'ØF'x)
            Tab_hexr.i = C2D(semi_right)
        end;
    end;
end;

/* calculate the decimal digits */

exp = Ø
do i = 1 to Tabp.Ø;
    if i > 1 then
        do;
            Tab_decr.i = Tab_hexr.i * (1Ø ** exp)
            exp = exp + 1
        end;
    else
        do;
            Tab_decr.i = Ø
        end;
    Tab_decl.i = (Tab_hexl.i / 16) * (1Ø ** exp)
    exp = exp + 1
    dec_num = dec_num + Tab_decr.i + Tab_decl.i
end;
dec_num = right(dec_num,num_digits,Ø)
/* set the sign and the comma position */

first_digit = C2X(tabp.1)
if right(first_digit,1) <> "D" then
    num_sign = "+"
else
    num_sign = "-" ;
select;
when comma_pos = Ø then
    do;
        dec_string = strip(dec_num,L,Ø);
    end;
when comma_pos < num_digits then
    do;
        left_digits = num_digits - comma_pos
        dec_string = left(dec_num,left_digits) || dec_symb || ,
                    right(dec_num,comma_pos)
    end;
end;

```

```

        dec_string = strip(dec_string,L,0);
    end;
    otherwise
    do;
        right_appo = right(dec_numb,comma_pos,0)
        dec_string = "0" || dec_symb || right(dec_numb,comma_pos,0)
    end;
end;
if num_sign = "-" then
    dec_string = num_sign || dec_string;
return dec_string P2rc;

```

D2P

```

/*          REXX          */
/* D2P procedure: converts a string to packed decimal format      */
/* Input: Numeric string - length of packed data string (optional) */
/* Output: Character string containing packed data, return code    */
/*          */
/* Personalization: set dec_symb to "," or to "." as needed        */
/*          */
/* When called internally, the following must be the first line:  */
/* D2P: procedure expose pak_string D2rc;                          */
/* When called externally, the calling procedure must Parse       */
/* a Variable called "RESULT" to get the values of pak_string & D2rc */
/*          */
/* eg: Call "D2P" Numeric_string pk_string_length                 */
/*       Parse VAR RESULT pak01 retc1                               */
/*          */
Trace 0
D2rc = 0;
zero_pak = x2c(B2X('0000 0000'))
msg_error = "Error!"
pak_len   = "0";
dec_symb  = ".";
Parse ARG dec_string pak_len .

/* analyse the input string */

if pak_len = "" then
    pak_len = 0;
if pak_len > 16 then
do;
    say "D2P: Output string length over maximum (16) "
    D2rc = 8
    return msg_error D2rc
end;
if dec_string = "" then
do;
    say "D2P: Input string missing "
    D2rc = 16

```

```

    return msg_error D2rc
end;
dec_length = length(dec_string);
/* remove dec. point and + or - signs */
select
  when left(dec_string,1,1) = '-' then
  do;
    num_sign = "D"
    dec_length = dec_length - 1
    dec_string = right(dec_string,dec_length)
  end;
  when left(dec_string,1,1) = '+' then
  do;
    num_sign = "C"
    dec_length = dec_length - 1
    dec_string = right(dec_string,dec_length)
  end;
  otherwise
  do;
    num_sign = "F"
  end;
end;
comma_pos = pos(dec_symb,dec_string);
if comma_pos > 0 then
do;
  dec_length = dec_length - 1
  dec_string = substr(dec_string,1,comma_pos-1) || ,
              substr(dec_string,comma_pos+1)
end;
if dec_length > 31 then
do;
  say "D2P: Input string too long "
  D2rc = 12
  return msg_error D2rc
end;

/* normalize the decimal string */
num_pk_tem = (dec_length +1) / 2;
num_packed = format(num_pk_tem,2,0)
if num_pk_tem <> num_packed then
do;
  dec_length = dec_length + 1
  dec_string = "0" || dec_string
end;

/* load the decimal digit table */
tabd.1 = 0
do i = 2 to dec_length + 1;
  x = dec_length - i + 2
  tabd.i = substr(dec_string,x,1)
end;
tabd.0 = dec_length + 1;

```

```

/* build the packed string */
select
  when num_sign = "C" then
    tabd.1 = 12
  when num_sign = "D" then
    tabd.1 = 13
  otherwise
    tabd.1 = 15
end;
do i = 2 to Tabd.0 by 2;
  k = i / 2
  tabd.i = tabd.i * 16
  j = i - 1
  tabp.k = tabd.i + tabd.j
end;
tabp.0 = Tabd.0 / 2
/* convert hexadecimal */
do i = 1 to tabp.0;
  Tab_cha.i = X2C(D2X(Tabp.i))
end;
/* create string */
stringa = ""
do i = tabp.0 to 1 by - 1;
  stringa = stringa || tab_cha.i
end;
if pak_len > 0 then
do;
  select;
  when pak_len < tabp.0 then
    pak_string = right(stringa,pak_len);
  when pak_len > tabp.0 then
    pak_string = right(stringa,pak_len,zero_pak)
  otherwise
    pak_string = stringa
  end;
end;
return pak_string D2rc;

```

Alberto Dallaturca
Systems Programmer
Phoenix Informatica Bancaria SpA (Italy)

© Xephon 2005

Please note that the correct contact address for Xephon Inc is PO Box 550547, Dallas, TX 75355, USA. The phone number is (214) 340 5690, the fax number is (214) 341 7081, and the e-mail address to use is info@xephon.com.

JES2 RJE and NJE reporting

As we all know too well, one can run z/OS (MVS) at one's installation in many processing configurations ranging from a single MVS image with a single JES2 that is completely isolated from other processing systems, to one that is a part of a worldwide processing network. The choice of configuration complexity is a dynamic one that grows as your business needs grow.

Remote Job Entry (RJE) and Network Job Entry (NJE) are IBM facilities permitting the extension of your local processing configuration by defining remote locations that can consist of job input terminals and output devices at a different physical site, connected to the MVS/JES2 image through telecommunication links. In this manner, an installation can provide input and output support throughout a large office building, to locations across town, or even in another city. But all the remote sites and their attached devices are defined to a single MVS/JES2 configuration. An RJE workstation is a workstation that is connected to a member by means of data transmission facilities. The workstation can be a single I/O device or group of I/O devices, or it can include a processor such as a z/Series machine. Generally, RJE workstations include a programmable workstation (such as a personal computer) connected to the MVS system through a telecommunication link. Such a link utilizes Synchronous Data Link Control (SDLC) or Binary Synchronous Communication (BSC) protocols for communicating between JES2 and remote devices. The remote device will be either a System Network Architecture (SNA) remote, which uses SDLC, or a BSC remote, which uses BSC. Remote jobs are processed in the same manner as those received from local readers, printers, and punches.

On the other hand, network job entry (NJE) allows individual MVS/JES2 site to be connected in a network of JES2 and non-

JES2 systems that can communicate, pass jobs, and route output to any attached output devices. Each system and its local and remote device make up a node. In its simplest terms, NJE is 'networking' between systems that interact as peers, whereas RJE is networking between JES2 and workstations. The main difference between them is one of overall computing power and processor location. Remember, RJE is an extension of a single computer system (that is, either a single-processor or multi-access spool complex) allowing jobs to be submitted from, and output routed back to, sites that are remote to the location of that system. NJE provides the ability to link many such single-processor systems or multi-access spool complexes into a processing network. Each system can be located on the same physical processor, side-by-side in a single room, or across the world in a network of thousands of nodes. The important difference is that a processor and its local and remote devices make up a node. Two or more attached nodes make up an NJE network. For example, an NJE interconnecting two or more JES2 spools allows an RJE site at location A, connected to JES2 at site B to send a job to be executed at the company's computer center C, where the database needed by the job is located. NJE picks up the job from site B's spool and sends it to site C's spool for execution. When the job completes at C, NJE sends its SYSOUT back to site B, where it resides on the spool until the A's RJE operator allows it to print. NJE is in common usage because enterprises have computer systems in different cities that process other cities' jobs, and, for many companies, NJE represents a fairly secure method of allowing access to a system without too complex security procedures.

With the expected large number of RJE users, NJE appears to offer flexibility in managing the remote user – therefore the need to analyse NJE traffic and performance data will also increase. Up to now, the analysis of these NJE systems may not have been required because there were probably more pressing problems, and the volume of traffic across the NJE lines has tended to be small. As a result I myself had no

extensive experience in analysis of NJE traffic and performance. However, it seems as if there are some drawbacks with NJE, one of which is that it either fails to work and jobs or output can appear to spend a long time on the queue before transmission occurs or that transmission is too slow.

What do we need in order to measure and report NJE traffic? When enabled by the SMFPRMxx TYPE parameter of your system parameter library, SMF creates several records, which I have used as the primary sources for the analysis of the NJE traffic:

- A type 55 SMF record is written at each node in a JES2 network when a start networking command is processed. The initial SIGNON is recorded at the node to which the SIGNON was sent, and the response SIGNON is recorded at the node that originated the initial SIGNON.
- A type 58 record is similarly written at each node when a networking session is terminated. The record contains the node name, member number, and line name.
- The record type 56 is written whenever an attempt to SIGNON contains an incorrect line or node password.
- When SYSOUT is transmitted between nodes, a type 57 SMF record is written by JES2, describing the transmission volume and duration. It contains transmitter start and stop times as well as a count of records transmitted, and is useful in tracking NJE usage. There is no accounting information from the job that causes the SYSOUT transmission except network account number. However, please note that there are two job identification pointers that could be used to identify the job initiated SYSOUT transmission. The first one is the original job identification, which is a JES2 job ID at the originating node of this job. The job identification provides the type of task (job/tsu/stc) and the JES2-assigned job number. The second one is the current job ID representing the JES2 job ID at the

current node, and this is the node at which this record was written. This value may have no relationship with the original job number assigned by JES2 when the job was created. However, if the original job number is not currently in use on this system, the original number is used by JES2 here. Furthermore, there is not enough information in the record to match the observation with the job since the job's read-in time and date are not provided.

Thus type 55, 56, and 58 records are important for accounting for the passage of the jobs through an NJE network.

What do we need to measure about RJE stations? The SMF records 52, 53, and 54 (for SDLC protocol) are the primary source for the analysis of the RJE station itself. There is a record for each sign-on (record type 52) and each sign-off (record type 53), and there is a security record if an invalid password is attempted (record type 54). The sign-on and sign-off observations from the same remote can be matched to create an observation for each RJE session:

- Record type 52 is written whenever an SNA line is started (**\$S LINE(n)** command) or restarted (**\$E LINE(n)** command) or an SNA RJE workstation user signs on. This record contains a record indicator, remote name, line name, password, and message text.
- Record type 53 is written whenever a **\$S LNE** command (to start a line) is issued or a **\$E LNE** command (to restart a line) is issued, and when a remote user signs on. This record contains a record indicator, a remote name, a line name, a password, a line adapter address, the number of execute channel programs (EXCP), negative acknowledgements to write text, data checks to read text, and timeouts to read text.
- Record type 54 is written when an SNA remote user attempts to sign on with an incorrect password. It contains a record indicator, remote name, line name, invalid password, and message text.

Please note that RJE SMF data can be analysed from several perspectives. Since each session contains the JES line number used, examination of the line being used by each remote can identify the unexpected use of dial-in lines. On the other hand, the error statistics for each session are also provided by RJE SMF records and can be used to adequately distinguish those lines that have high error rates from those that do not. Absolute error rates are not as important as comparing rates by type of line. By sharing this kind of information with telecommunication personnel, one aids their problem analysis since they generally have no knowledge of how much the performance analyst knows about the line usage.

Based on JES2 RJE and NJE SMF record descriptions obtained from the *MVS System Management Facilities (SMF)* manual, SA22-7630, a sample JES2 RJE and NJE event report writer was written. In order to extract JES2 RJE and NJE event information from SMF data, I have constructed a three-part job stream. In the first step (NETDMP) SMF records 52, 53, 54, 55, 56, 57, and 58 are extracted from SMF datasets to a file, which can be used as a base of archived records. In the second step (NETCOPY), previously extracted records (selection being defined by INCLUDE's condition) are sorted and copied to a file, which is the input to the analysis and reporting JESNET REXX EXEC invoked in the last (NETJES) step.

Three reports are generated by this REXX EXEC. The JES2 RJE report provides information about the RJE event/operation that caused JES2 to write to the SMF record. Also reported are line identification as well as the line event counters (number of VTAM requests processed, number of exception responses, number of LUSTATs received, number of bid rejects, and number of temporary errors). The second report, JES2 NJE report, provides information about the line and node name being used as well as the NJE event/operation (signoff/signon/network integrity) that caused JES2 to write to the SMF record. The JES2 network SYSOUT transmission report

is the last one in this set, and it provides information about transmitter start and stop times, the count of records transmitted, and transmit rate (TP records per second).

SAMPLE JCL

This will execute SMF type 52 to 58 data extraction and JES2 RJE and NJE reporting:

```

/**-----*
/** UNLOAD SMF 52:58 RECORDS FROM VSAM OR VBS TO VB          *
/** Note: change the DUMPIN DSN=your.smfdata to be the name of  *
/** the dataset where you currently have SMF data being        *
/** recorded. It may be either an SMF weekly dataset or an active *
/** dump dataset. If you chose the later, then prior to      *
/** executing this job, you need to terminate SMF recording  *
/** on the currently active dump dataset to allow the        *
/** unload of SMF records.                                   *
/** Also, change the DCB reference to match the name of your  *
/** weekly SMF dump dataset.                                 *
/**-----*
//NETDMP   EXEC   PGM=IFASMFDP
//DUMPIN   DD     DISP=SHR,DSN=your.smfdata
//DUMPOUT  DD     DISP=(NEW,PASS),DSN=&&SMFOUT,UNIT=SYSDA,
//          SPACE=(CYL,(2,2)),DCB=(your.smfweekly.dataset)
//SYSPRINT DD     SYSOUT=*
//SYSIN    DD     *
           INDD(DUMPIN,OPTIONS(DUMP))
           OUTDD(DUMPOUT,TYPE(52:58))
/*
/**-----*
/** COPY VBS TO VB, DROP HEADER/TRAILER RECORDS, SORT ON DATE/TIME *
/** Note: change the SMF DSN=h1q.SMF.DATA to the name of      *
/** the dataset you'll use in the last step.                  *
/**-----*
//NETCOPY  EXEC   PGM=ICETOOL
//TOOLMSG  DD     SYSOUT=*
//DFSMSG   DD     SYSOUT=*
//RAWSMF   DD     DSN=&&SMFOUT,DISP=SHR
//SMF      DD     DSN=h1q.SMF.DATA,SPACE=(CYL,(x,y)),UNIT=SYSDA,
//          DISP=(NEW,CATLG,KEEP),
//          DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//TOOLIN   DD     *
           SORT FROM(RAWSMF) TO(SMF) USING(SMFI)
//SMFICNTL DD    *
           OPTION SPANINC=RC4,VLSHRT
           INCLUDE COND=(6,1,BI,GT,51,AND,6,1,BI,LT,59)
           SORT FIELDS=(11,4,PD,A,7,4,BI,A)

```

```

/*
/**-----*
/** FORMAT JES2 RJE and NJE RECORDS *
/** Note: change the SYSEXEC DSN=your.rexx.library to be the name *
/** of the dataset where you have placed the JESNET REXX EXEC. *
/** Also, change the SMF DSN=h1q.SMF.DATA to the name of *
/** the dataset you have created in the previous step. *
/**-----*
//NETJES EXEC PGM=IKJEFT01,REGION=0M
//SYSEXEC DD DISP=SHR,DSN=your.rexx.library
//SMF DD DISP=SHR,DSN=h1q.SMF.DATA
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
prof nopref
%JESNET
/*

```

JESNET REPORTING EXEC

```

/* REXX RJE, NJE & XMIT */
/* This sample REXX is used in conjunction with the COPYJES */
/* job to read and format RJE, NJE, and JES2 Network SYSOUT */
/* transmission SMF records in a readable report. */
/*-----*/
Numeric digits 16
userid=SYSVAR(SYSUID)
/*-----*/
/* Part 1: handle report files allocation & datasets */
/* existence */
/*-----*/
r5xrje = userid||'.rje.rep' /* RJE report file dsn */
r5xnje = userid||'.nje.rep' /* NJE report file dsn */
r5xmit = userid||'.xmt.rep' /* SYSOUT Xmit report dsn */
If SYSDSN(r5xrje) = 'OK'
Then "DELETE "r5xrje" PURGE"
    "ALLOC FILE(RJERPT) DA("r5xrje")",
    "UNIT(SYSALLDA) NEW TRACKS SPACE(1,5) CATALOG",
    "REUSE LRECL(65) RECFM(F B) "
If SYSDSN(r5xnje) = 'OK'
Then "DELETE "r5xnje" PURGE"
    "ALLOC FILE(NJERPT) DA("r5xnje")",
    "UNIT(SYSALLDA) NEW TRACKS SPACE(1,5) CATALOG",
    "REUSE LRECL(65) RECFM(F B) "
If SYSDSN(r5xmit) = 'OK'
Then "DELETE "r5xmit" PURGE"
    "ALLOC FILE(XMITRPT) DA("r5xmit")",
    "UNIT(SYSALLDA) NEW TRACKS SPACE(1,5) CATALOG",
    "REUSE LRECL(100) RECFM(F B) "

```

```

Totrje = 0;          /* Count the total RJE records processed */
Totnje = 0;          /* Count the total NJE records processed */
Totxmt = 0;          /* Count the total SYSOUT Transmission    */
                      /* records processed                               */
t1 =0; t2 =0; t3 =0; t4 =0; t5 =0;
Ttime =0; TPrec =0;
/*-----*/
/* Print RJE report header                               */
/*-----*/
rje.1 = left('JES2 RJE report',50)
rje.2 = left(' ',1,' ')
rje.3 =left('Report produced on',18)      ,
      ||left(' ',1,' ')||left(date(),12) ,
      ||left('at',3,' ')||left(time(),10)
rje.4 = left(' ',1,' ')
rje.5 = left('Date',12)||left('time',12),
      ||left('RJE event',16)||left('Line',6),
      ||left('Remote name',12)||left('Subsys',6)
rje.6 = left('-',65,'-')
"EXECIO * DISKW RJERPT (STEM rje.)"
/*-----*/
/* Print NJE report header                               */
/*-----*/
nje.1 = left('JES2 NJE report',50)
nje.2 = left(' ',1,' ')
nje.3 =left('Report produced on',18)      ,
      ||left(' ',1,' ')||left(date(),12) ,
      ||left('at',3,' ')||left(time(),10)
nje.4 = left(' ',1,' ')
nje.5 = left('Date',12)||left('time',12),
      ||left('NJE event',16)||left('Line',8),
      ||left('Node name',10)||left('Subsys',6)
nje.6 = left('-',65,'-')
"EXECIO * DISKW NJERPT (STEM nje.)"
/*-----*/
/* Print JES2 Network SYSOUT Transmission report header */
/*-----*/
xmm.1 = left('JES2 Network SYSOUT Transmission report',50)
xmm.2 = left(' ',1,' ')
xmm.3 =left('Report produced on',18)      ,
      ||left(' ',1,' ')||left(date(),12) ,
      ||left('at',3,' ')||left(time(),10)
xmm.4 = left(' ',1,' ')
xmm.5 = left('Transmission',24)||left('Job',18)||left('Node name',11)
xmm.6 = left('Start / End TOD',24)||left('Original',9),
      ||left('Current',9)||left('Original Execution Next',27),
      ||left('Device',9)||left('Sys',4)
xmm.7 = left('-',85,'-')
"EXECIO * DISKW XMTRPT (STEM xmm.)"

```

```

/*-----*/
/*          Main processing loop          */
/*-----*/
DO FOREVER
  "EXECIO 1 DISKR SMF"
  IF RC = 0 THEN call End_of_file
else do
  PARSE PULL record
  PARSE VAR record header 15 rest
  smf5xrty = c2d(substr(header,2,1))      /* Record type */
  call SMF5x_header
end
End
/*-----*/
/* Part 4: Print out record count at end of input file */
/*-----*/
End_of_file:
/* Close & free all allocated files */
call printrje " "
msg = "Total SMF RJE records read : "left(Totrje,4)
call printrje msg;
call printrje " "
msg ="Line event statistics:"
call printrje msg
msg ="No. of VTAM requests processed:"left(t1,6)
call printrje msg
msg ="No. of exception responses:      "left(t2,5)
call printrje msg
msg ="No. of LUSTATs received:         "left(t3,5)
call printrje msg
msg ="No. of bid rejects:              "left(t4,5)
call printrje msg
msg ="No. of temporary errors:        "left(t5,5)
call printrje msg
call printnje " "
msg = "Total SMF NJE records read : "left(Totnje,4)
call printnje msg
say "JES2 RJE report ....."r5xrje
say "JES2 NJE report ....."r5xnje
say "JES2 SYSOUT Transmission report .."r5xmit
call Printxmt " "
Select
  when TPrec > 0 then  Arate = format((TPrec/Ttime),4,2)
  otherwise           Arate = 0
End
msg = "Total SMF XMIT records read : "left(Totxmt,4)
call Printxmt msg
msg = "Total TP records transmitted: "TPrec
call Printxmt msg
msg = "Total TP transmit time (sec): "Ttime

```

```

call Printxmt msg
msg = "Average transmit rate:          "Arate
call Printxmt msg
"EXECIO 0 DISKR JESMF (FINIS"
"EXECIO 0 DISKW RJERPT (FINIS"
"EXECIO 0 DISKW NJERPT (FINIS"
"EXECIO 0 DISKW XMITRPT (FINIS"
"FREE FILE(JESMF RJERPT NJERPT XMITRPT)"
EXIT 0
SMF5x_header:
/*-----*/
/* Part 2: Basic section of the RJE, NJE, and XMIT          */
/*-----*/
smf5xflg = x2b(c2x(substr(header,01,01)))          /* System indicator */
smf5xrty = c2d(substr(header,02,1))              /* Record type      */
smf5xtme = smf(c2d(substr(header,03,04)))        /* Time record was written*/
smf5xdte = substr(c2x(substr(header,07,04)),3,5) /* Date record was written */
smf5xsid = substr(header,11,04)                  /* System Identification */
SMF_date = left(Date('N',smf5xdte,'J'),11)
Select
  when smf5xrty = "52" then call SMFRJE_rec
  when smf5xrty = "53" then call SMFRJE_rec
  when smf5xrty = "54" then call SMFRJE_rec
  when smf5xrty = "55" then call SMFNJE_rec
  when smf5xrty = "56" then call SMFNJE_rec
  when smf5xrty = "57" then call SMFXMT_rec
  when smf5xrty = "58" then call SMFNJE_rec
  otherwise nop
End /* of select */
return

SMFRJE_rec:
/*-----*/
/* Part 3a: (SMFRJE_rec) get RJE type info for:          */
/* 52: JES2 LOGON/Start Line (SNA only)                  */
/* 53: JES2 LOGOFF/Stop Line (SNA only)                  */
/* 54: JES2 Integrity (SNA only)                        */
/*-----*/
totrje = totrje + 1 /* Count the total records processed */
PARSE var rest
smf5xpof = c2d(substr(rest,1,2)) /* Offset to prod. section */
smf5xprl = c2d(substr(rest,3,2)) /* Length of prod. section */
smf5xprn = c2d(substr(rest,5,2)) /* Number of prod. section */
smf5xido = c2d(substr(rest,7,2)) /* Offset to id section */
smf5xidl = c2d(substr(rest,9,2)) /* Length of id section */
smf5xidn = c2d(substr(rest,11,2)) /* Number of id section */
smf5xsub = c2d(substr(rest,13,2)) /* Subtype id number */
smf5xver = substr(rest,15,2) /* Record version number */
smf5xsys = substr(rest,17,4) /* Subsystem name */

```

```

smf5xrmt =      substr(rest,21,8)          /* Remote name          */
smf5xlin =      substr(rest,29,8)          /* Line name            */
smf5xpsw =      substr(rest,37,8)          /* Line password        */
SMF5xrec_type:
  Select
    when smf5xrty = "52" then call type52
    when smf5xrty = "53" then call type53
    when smf5xrty = "54" then call type54
    otherwise nop
  End
  /* of select */
return

SMFNJE_rec:
  /*-----*/
  /* Part3b: (SMFNJE_rec) get NJE type info for:          */
  /* 55: JES2 Network SIGNON                               */
  /* 56: JES2 Network Integrity                           */
  /* 58: JES2 Network SIGNOFF                             */
  /*-----*/
  Totnje = Totnje + 1          /* Count the total records processed */
  PARSE var rest
smf5ysbs = c2d(substr(rest,1,2))          /* Hasp subsystem id          */
smf5ysub = c2d(substr(rest,3,2))          /* Record subtype            */
smf5ylrr = c2d(substr(rest,5,2))          /* Length of rest of rec.    */
smf5ynnm =      substr(rest,7,8)          /* Node name                  */
smf5ymem = c2d(substr(rest,15,1))          /* Member number              */
  Select
    when smf5ysbs = 2 then Subsys='JES2'
    otherwise          nop
  End
SMF5yrec_type:
  Select
    when smf5xrty = "55" then call type55
    when smf5xrty = "56" then call type56
    when smf5xrty = "58" then call type58
    otherwise nop
  End
  /* of select */
return

SMFXMT_rec:
  /*-----*/
  /* Part 3c: JES2 Network SYSOUT Transmission            */
  /*-----*/
  Totxmt = Totxmt + 1          /* Count the total records processed */
  PARSE var rest
smf57sbs = c2d(substr(rest,1,2))          /* Hasp subsystem id          */
smf57sub = c2d(substr(rest,3,2))          /* Record subtype            */
smf57lrr = c2d(substr(rest,5,2))          /* Length of rest of rec.    */
smf57jid =      substr(rest,7,8)          /* Original job identification*/
smf57cjd =      substr(rest,15,8)          /* Current job identification */

```

```

smf57onn =      substr(rest,23,8)          /* Original node name      */
smf57enn =      substr(rest,31,8)          /* Execution node name     */
smf57nnn =      substr(rest,39,8)          /* Next node name          */
smf57dvn =      substr(rest,47,8)         /* Sysout transmitter device name*/
smf57tss = smf(c2d(substr(rest,55,4)))
                                           /* SYSOUT transmitter start time */
smf57sta =      c2d(substr(rest,55,4))
                                           /* SYSOUT transmitter start time */
smf57dss = , ,                               /* SYSOUT transmitter start date */
                substr(c2x(substr(rest,59,4)),3,5)
smf57tps = smf(c2d(substr(rest,63,04)))
                                           /* SYSOUT transmitter stop time */
smf57end =      c2d(substr(rest,63,04))
                                           /* SYSOUT transmitter stop time */
smf57dps = , ,                               /* SYSOUT transmitter stop date */
                substr(c2x(substr(rest,67,4)),3,5)
smf57acn =      substr(rest,71,8)          /* Network account number  */
smf57tsi =      substr(rest,79,4)          /* Sysout transmitter system id */
smf57cnt = c2d(substr(rest,83,4))          /* Number of logical tp records */
time      = cross(smf57end,smf57sta)
rate      = format((smf57cnt/time),8,3)
Stdate = left(Date('N',smf57dss,'J'),11)
Etdate = left(Date('N',smf57dps,'J'),11)
msg = Stdate smf57tss smf57jid smf57cjd,
smf57onn smf57enn smf57nnn smf57dvn smf57tsi,
smf57acn;call Printxmt msg
msg = Etdate smf57tps" Transmit time (sec): "left(time,6)
call Printxmt msg
msg = left(' ',23,' ')||" TP records: "left(smf57cnt,6)
call Printxmt msg
msg = left(' ',23,' ')||" Transmit rate: "right(rate,7)
call Printxmt msg
Ttime = Ttime + time
TPrec = Tprec + smf57cnt
/*-----*/
/* This is the header for all future extensions to the SMF 57      */
/* record. Sections beyond this point must be accessed by using    */
/* the values stored in the triplets (below) that contain the offset,*/
/* length, and number of sections of the type corresponding to the  */
/* triplet. New sections will be appended to this header and their  */
/* presence can be detected by an increase in the number of triplets */
/* and by a non-zero section offset, length, and number of sections. */
/*-----*/
smf57ntr = c2d(substr(rest,87,2))          /* Number of triplets      */
smf57osw = c2d(substr(rest,91,4))          /* Offset to ess section   */
smf57lsw = c2d(substr(rest,95,2))          /* Length of ess section   */
smf57nsw = c2d(substr(rest,97,2))          /* Number of ess sections  */
/*-----*/
/* Enhanced SYSOUT Support (ESS) section                          */
/* This section contains the OUTPUT descriptor (if any) in SWBTU   */

```



```

/* format (IEFSJPFx plus text units) for the first offloaded data */
/* set included in this SMF record. The SWBTU may be processed */
/* using the SWBTUREC macro or other Scheduler JCL Facility (SJF) */
/* services. */
/*-----*/
IF smf571sw > 0 AND smf57nsw > 0 Then do
    ess = smf57osw - 3
    smf57ln1 = c2d(substr(x.i,ess,2)) /* Length of ESS section */
    smf57sgt = c2d(substr(x.i,ess+2,4)) /* Segment identifier */
    smf57ind = substr(x.i,ess+6,1) /* ESS section indicator */
    smf57sjf = '80000000'x /* Error obtaining swbtu */
                                /* (SWBTU data not present)*/

    if bitand(smf57ind,smf57sjf) = smf57sjf then,
        Esser = "Error obtaining scheduler JCL facility"
    smf57jdt = substr(x.i,ess+8,8) /* JDVT name */
    smf57tul = c2d(substr(x.i,eff+16,2)) /* SWBTU data area length */
    smf57tu = substr(x.i,eff+18,smf57tul) /* SWBTU data area */
end /* of Enhanced SYSOUT Support (ESS) Section */
return

/*-----*/
/* Part 4: Handle individual record types */
/*-----*/
type52:
/*-----*/
/* JES2 LOGON/Start Line (SNA only) */
/* Action that causes the record to be written: */
/* - An SNA line is started /$S LINE(n) command/ */
/* - An SNA line is restarted /$E LINE(n) command/ */
/* - An SNA RJE workstation user signs on */
/*-----*/
smf52sub = c2d(substr(rest,1,2)) /* Subtype id number*/
Select
    when smf52sub= 1 then rjevent='SNA logon '
    otherwise rjevent='SNA start line '
End
msg = Smf_date smf5xtme rjevent smf5xlin smf5xrmt smf5xsys
call printrje msg
return

type53:
/*-----*/
/* JES2 LOGOFF/Stop Line (SNA only) */
/* Action that causes the record to be written: */
/* - An SNA line is stopped /$P LINE(n) command/ */
/* - An SNA line is restarted /$E LINE(n) command/ */
/* - An SNA RJE workstation user signs off */
/*-----*/
smf53sub = c2d(substr(rest,1,2)) /* Subtype id number */
Select

```

```

        when smf53sub= 1 then rjevent='SNA logoff      '
        otherwise           rjevent='SNA stop line  '
    End
smf53ctr = c2d(substr(rest,45,4))    /* No. of VTAM requests processed */
smf53ct4 = c2d(substr(rest,49,4))    /* No. of exception responses */
smf53ct8 = c2d(substr(rest,53,4))    /* No. of LUSTATs received */
smf53ctc = c2d(substr(rest,57,4))    /* No. of bid rejects      */
smf53cta = c2d(substr(rest,61,4))    /* No. of temporary errors */
smf53adp =      substr(rest,65,3)    /* Line identifier, 'SNA' */
msg = Smf_date smf5xtme rjevent smf5xlin smf5xrmt smf5xsys
call printrje msg
inden="Line event counters      :right(smf53ctr,5)|| ,
' - No. of VTAM requests processed '
call printrje inden; t1 =t1 + smf53ctr
indel="                "right(smf53ct4,5)|| ,
' - No. of exception responses'
call printrje inden; t2 = t2 + smf53ct4
indel="                "right(smf53ct8,5)|| ,
' - No. of LUSTATs received'
call printrje inden; t3 = t3 + smf53ct8
indel="                "right(smf53ctc,5)|| ,
' - No. of bid rejects'
call printrje inden; t4 = t4 + smf53ctc
indel="                "right(smf53cta,5)|| ,
' - No. of temporary errors'
call printrje inden; t5 = t5 + smf53cta
return

type54:
/*-----*/
/* JES2 Integrity (SNA only) */
/* Action that causes the record to be written: */
/* An SNA RJE workstation user signs on with an */
/* incorrect password. */
/*-----*/
smf54sub = c2d(substr(rest,1,2))    /* Subtype id number*/
Select
    when smf54sub= 1 then rjevent='SNA invalid LOGON password'
    otherwise           nop
End
msg = Smf_date smf5xtme rjevent smf5xlin smf5xrmt smf5xsys
call printrje msg
return

type55:
/*-----*/
/* JES2 Network SIGNON */
/* Action that causes the record to be written: */
/* A signon record is written at each node $S N(n) commands. */
/*-----*/

```

```

smf55fg1 = x2b(c2x(substr(rest,16,1))) /* Signon status flags*/
smf55lpw =      substr(rest,17,8)      /* Line password */
smf55npw =      substr(rest,25,8)      /* Node password */
smf55lnm =      substr(rest,33,8)      /* Line name */
if smf55fg1 = '10000000'b then njevent='Signon response'
else
                        njevent='Signon      '
msg = Smf_date smf5xtme njevent smf55lnm smf5ynnm smf5ymem Subsys
call printnje msg
return

```

type56:

```

/*-----*/
/* JES2 Network Integrity */
/* Action that causes the record to be written: */
/* Whenever a sign-on attempt contains an incorrect line */
/* or node password. */
/*-----*/
smf56fg1 = x2b(c2x(substr(rest,16,1))) /*signon status flags*/
smf56lpw =      substr(rest,17,8)      /*line password */
smf56npw =      substr(rest,25,8)      /*node password */
smf56lnm =      substr(rest,33,8)      /*line name */
if smf56fg1 = '10000000'b then njevent='Invalid response '
else
                        njevent='Invalid signon      '
msg = Smf_date smf5xtme njevent smf56lnm smf5ynnm smf5ymem Subsys
call printnje msg
return

```

type58:

```

/*-----*/
/* JES2 Network SIGNOFF */
/* Action that causes the record to be written: */
/* The records written at each node whenever a networking */
/* session is ended. */
/*-----*/
njevent='Signoff      '
smf58rv1 = substr(rest,16,1) /* reserved */
smf58lnm = substr(rest,17,8) /* line name */
msg = Smf_date smf5xtme njevent smf58lnm smf5ynnm smf5ymem Subsys
call printnje msg
return

```

Printrje:

```

/*-----*/
/*      Print each RJE report line */
/*-----*/
PARSE arg lineout1
"EXECIO 1 DISKW RJERPT (STEM lineout)"
if rc ^= 0 then
do
say "printrje RC =" RC

```

```

        exit rc
    end
Return
/* end of Printrje */

Printnje:
/*-----*/
/* Print each NJE report line */
/*-----*/
    PARSE arg lineout1
    "EXECIO 1 DISKW NJERPT (STEM lineout)"
    if rc ^= 0 then
        do
            say "Printnje RC =" RC
            exit rc
        end
    end
Return
/* end of Printnje */

Printxmt:
/*-----*/
/* Print each XMIT report line */
/*-----*/
    PARSE arg lineout1
    "EXECIO 1 DISKW XMITRPT (STEM lineout)"
    if rc ^= 0 then
        do
            say "Printxmt RC =" RC
            exit rc
        end
    end
Return
/* end of Printxmt */

SMF: procedure
/*-----*/
/* REXX - convert SMF time to hh:mm:ss:hd format */
/*-----*/
arg time
time1 = time % 100
hh = time1 % 3600;          hh = right("0"||hh,2)
mm = (time1 % 60) - (hh * 60); mm = right("0"||mm,2)
ss = time1 - (hh * 3600) - (mm * 60); ss = right("0"||ss,2)
fr = time // 1000;          fr = right("0"||fr,2)
rtime = hh||":"||mm||":"||ss||":"||fr
return rtime

CROSS: procedure
/*-----*/
/* Cover the midnight crossover */
/*-----*/
arg endtime,startime
Select
    when endtime >= startime then nop

```

```
        otherwise endtime = endtime + 8640000  
    End  
    diftm = (endtime - starttime)*0.01  
    return diftm
```

Mile Pekic
Systems Programmer (Serbia and Montenegro)

© Xephon 2005

Catalog activity reporting with ICETOOL

We are learning to use ICETOOL as our tool of choice for simple reporting. ICETOOL provides us with the speed and performance of DFSORT as well as a fair amount of flexibility through the use of symbols to map record layouts. This short article provides an example of how to use ICETOOL to create a couple of simple reports detailing ICF catalog activity.

We have included two sample jobs for your review and use. The first job produces two unique reports. One of the features that we find very useful is the OUTFIL statement. This ICETOOL keyword allows you to create multiple output files with a single pass over the input data. In this job we are producing three output files. One of these simply copies the type 65 and 66 records to a collector file so that we can use them in the future. The other two output files are simple reports by record type. Another feature we like in ICETOOL is the CHANGE keyword. This allows you to match the values in an input field, and change them to a different value. In our examples, we examine the type field and change the value to a more detailed text display.

The second sample job uses all the same features as sample job 1, but produces a single output file. The output file is a simple report combining the type 65 and 66 records. For this example we used SORT rather than COPY to sequence the records in chronological order. We have also provided an excerpt of the report output in Figure 1.

ICETOOL works very well with records that have a somewhat fixed layout. Self-defining records may also be processed with ICETOOL, but require different techniques for processing. We hope that these examples provide some insight into this useful utility. Future articles will provide additional examples of how to use ICETOOL.

SAMPLE JOB 1

```
//jobname JOB your job information
//*
/*JOBPARM      LINES=9999
//*
//STEP0010 EXEC PGM=ICETOOL
//STEPLIB DD DISP=SHR,DSN=only.if.needed
//SYMNAMES DD DISP=SHR,DSN=local.symbol.library(SMFHEADR)
//          DD DISP=SHR,DSN=local.symbol.library(SMF65)
//          DD DISP=SHR,DSN=local.symbol.library(SMF66)
//INPUT DD DISP=SHR,DSN=local.smf.data.set
//SAVEFILE DD DISP=(NEW,DELETE),DSN=&&SAVEFILE,
//           UNIT=SYSALLDA,SPACE=(CYL,(50,10)),
//           DSORG=PS,RECFM=VBS,BLKSIZE=0
//TOOLMSG DD SYSOUT=* ICETOOL MESSAGES
//DFSMSG DD SYSOUT=* DFSORT MESSAGES
//SYMNOUT DD SYSOUT=*
//REPORT65 DD SYSOUT=*
//REPORT66 DD SYSOUT=*
//TOOLIN DD *
COPY FROM(INPUT) USING(CPY1)
/*
//CPY1CNTL DD *
OPTION SPANINC=RC4,VLSHRT
OUTFIL FNAMES=SAVEFILE,
INCLUDE=(SMFXRTY,EQ,65,OR,SMFXRTY,EQ,SMF066)
OUTFIL FNAMES=REPORT65,CONVERT,
INCLUDE=(SMFXRTY,EQ,SMF065),
OUTREC=(1:SMFXDTE,EDIT=(TTTT/TT/TT),12:SMFXTME,TM1,
EDIT=(TT:TT:TT),22:SMF65JNM,
34:SMF65SUB,CHANGE=(6,C'IN',C'INSERT',
C'DE',C'DELETE',
C'UP',C'UPDATE'),
NOMATCH=(C'??????'),
43:SMF65ENM,89:SMF65TYP,
CHANGE=(37,C'A',C'NON-VSAM DATA SET',
C'B',C'GENERATION DATA GROUP BASE',
C'C',C'CLUSTER',
C'D',C'DATA SET')
```

```

C'E',C'VSAM EXTENSION RECORD      ',
C'F',C'FREE SPACE                  ',
C'G',C'ALTERNATE INDEX             ',
C'H',C'ACTIVE GENERATION DATA SET (GDS) ',
C'I',C'INDEX                        ',
C'J',C'GDG EXTENSION RECORD        ',
C'K',C'VSAM VOLUME RECORD (VVR)    ',
C'L',C'LIBRARY CONTROL SYSTEM LIBRARY RECORD',
C'M',C'MASTER CATALOG              ',
C'N',C'NON-VSAM HEADER RECORD      ',
C'O',C'OBJECT ACCESS METHOD (OAM) NON-VSAM ',
C'P',C'PAGE SPACE                  ',
C'Q',C'VVR HEADER                  ',
C'R',C'PATH                         ',
C'T',C'TRUE NAME RECORD            ',
C'U',C'USER CATALOG                ',
C'V',C'VOLUME                      ',
C'W',C'LIBRARY CONTROL SYSTEM VOLUME ',
C'X',C'ALIAS                       ',
C'Y',C'UPGRADE                     ',
C'Z',C'VVR HEADER PRIMARY          ',
X'ØØ',C'NORMAL NON-VSAM RECORD     ',
X'Ø1',C'JES3 RECORD                '),
NOMATCH=(C'?????? UNKNOWN RECORD TYPE ??????'),
HEADER2=(1:DATE,35:'OUR COMPANY NAME GOES HERE',79:TIME,/,
39:'ICF CATALOG ACTIVITY RPT',/,/,
4:'DATE',14:'TIME',22:'JOBNAME',34:'ACTION',
49:'DATA SET NAME')
OUTFIL FNames=REPORT66,CONVERT,
INCLUDE=(SMFXRTY,EQ,SMFØ66),
OUTREC=(1:SMFXDTE,EDIT=(TTTT/TT/TT),12:SMFXTME,TM1,
EDIT=(TT:TT:TT),22:SMF66JNM,
34:SMF65SUB,CHANGE=(6,C'IN',C'INSERT',
C'DE',C'DELETE',
C'UP',C'UPDATE'),
NOMATCH=(C'??????'),
43:SMF66ENM,89:SMF66TYP,
CHANGE=(37,C'A',C'NON-VSAM DATA SET      ',
C'B',C'GENERATION DATA GROUP BASE      ',
C'C',C'CLUSTER                          ',
C'D',C'DATA SET                          ',
C'E',C'VSAM EXTENSION RECORD            ',
C'F',C'FREE SPACE                        ',
C'G',C'ALTERNATE INDEX                  ',
C'H',C'ACTIVE GENERATION DATA SET (GDS) ',
C'I',C'INDEX                            ',
C'J',C'GDG EXTENSION RECORD              ',
C'K',C'VSAM VOLUME RECORD (VVR)         ',
C'L',C'LIBRARY CONTROL SYSTEM LIBRARY RECORD',
C'M',C'MASTER CATALOG                    '),

```

```

        C'N',C'NON-VSAM HEADER RECORD          ',
        C'O',C'OBJECT ACCESS METHOD (OAM) NON-VSAM ',
        C'P',C'PAGE SPACE                      ',
        C'Q',C'VVR HEADER                     ',
        C'R',C'PATH                           ',
        C'T',C'TRUE NAME RECORD                ',
        C'U',C'USER CATALOG                    ',
        C'V',C'VOLUME                          ',
        C'W',C'LIBRARY CONTROL SYSTEM VOLUME  ',
        C'X',C'ALIAS                          ',
        C'Y',C'UPGRADE                         ',
        C'Z',C'VVR HEADER PRIMARY             ',
        X'00',C'NORMAL NON-VSAM RECORD        ',
        X'01',C'JES3 RECORD                    '),
        NOMATCH=(C'?????? UNKNOWN RECORD TYPE ??????'),
        HEADER2=(1:DATE,35:'OUR COMPANY NAME GOES HERE',79:TIME,/,
        39:'ICF CATALOG ACTIVITY RPT',/,/,
        4:'DATE',14:'TIME',22:'JOBNAME',34:'ACTION',
        49:'DATA SET NAME')
/*
//

```

SAMPLE JOB 2

```

//jobname JOB your job information
/*
/*JOBPARM      LINES=9999
/*
//STEP0010 EXEC PGM=ICETOOL
//STEPLIB DD DISP=SHR,DSN=only.if.needed
//SYMNAMES DD DISP=SHR,DSN=local.symbol.library(SMFHEADR)
//          DD DISP=SHR,DSN=local.symbol.library(SMF65)
//          DD DISP=SHR,DSN=local.symbol.library(SMF66)
//INPUT DD DISP=SHR,DSN=local.smf.data.set
//TOOLMSG DD SYSOUT=* ICETOOL MESSAGES
//DFSMSG DD SYSOUT=* DFSORT MESSAGES
//SYMNOUT DD SYSOUT=*
//REPORT65 DD SYSOUT=*
//REPORT66 DD SYSOUT=*
//TOOLIN DD *
        SORT FROM(INPUT) USING(SRT1)
/*
//SRT1CNTL DD *
        OPTION SPANINC=RC4,VLSHRT
        SORT FIELDS=(SMFXDTE,A,SMFXTME,A)
        OUTFIL FNAMES=REPORT65,CONVERT,
        INCLUDE=(SMFXRTY,EQ,SMF065,OR,SMFXRTY,EQ,SMF066),
        OUTREC=(1:SMFXRTY,EDIT=(TT),
        5:SMFXDTE_DT1,EDIT=(TTTT/TT/TT),16:SMFXTME,TM1,

```



```

EDIT=(TT:TT:TT),26:SMF65JNM,
38:SMF65SUB,CHANGE=(6,C'IN',C'INSERT',
                    C'DE',C'DELETE',
                    C'UP',C'UPDATE'),
                    NOMATCH=(C'??????'),
47:SMF65ENM,93:SMF65TYP,
CHANGE=(37,C'A',C'NON-VSAM DATA SET',
        C'B',C'GENERATION DATA GROUP BASE',
        C'C',C'CLUSTER',
        C'D',C'DATA SET',
        C'E',C'VSAM EXTENSION RECORD',
        C'F',C'FREE SPACE',
        C'G',C'ALTERNATE INDEX',
        C'H',C'ACTIVE GENERATION DATA SET (GDS)',
        C'I',C'INDEX',
        C'J',C'GDG EXTENSION RECORD',
        C'K',C'VSAM VOLUME RECORD (VVR)',
        C'L',C'LIBRARY CONTROL SYSTEM LIBRARY RECORD',
        C'M',C'MASTER CATALOG',
        C'N',C'NON-VSAM HEADER RECORD',
        C'O',C'OBJECT ACCESS METHOD (OAM) NON-VSAM',
        C'P',C'PAGE SPACE',
        C'Q',C'VVR HEADER',
        C'R',C'PATH',
        C'T',C'TRUE NAME RECORD',
        C'U',C'USER CATALOG',
        C'V',C'VOLUME',
        C'W',C'LIBRARY CONTROL SYSTEM VOLUME',
        C'X',C'ALIAS',
        C'Y',C'UPGRADE',
        C'Z',C'VVR HEADER PRIMARY',
        X'00',C'NORMAL NON-VSAM RECORD',
        X'01',C'JES3 RECORD'),
        NOMATCH=(C'?????? UNKNOWN RECORD TYPE ??????')),
HEADER2=(1:DATE,45:'OUR COMPANY NAME GOES HERE',90:TIME,/,
        39:'ICF CATALOG ACTIVITY RPT',/,/,
        1:'RT',9:'DATE',18:'TIME',26:'JOBNAME',38:'ACTION',
        52:'DATA SET NAME',93:'TYPE OF DATA ENTITY')
/*
//

```

SMFHEADR ICETOOL SYMBOL FILE

```

*-----*
*
* ICETOOL SYMBOL MAP FOR SMF RECORD HEADER
*
* This layout will accommodate both types of SMF records, those with
* subtypes and those without. If the individual record layout is for*

```

```

* a record without subtypes, the symbol definition for the actual SMF*
* record should begin with a POSITION,19 directive.
*
* Source documentation: SA22-7630-7
* Local symbols added where appropriate
*---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---*
SMF_RDW,1,4,BI Record descriptor word
SMFXLEN,=,2,BI Record length
SMFXSEG,*,2,BI Segment length
SMFXFLG,*,1,BI Flag byte
SMFXRTY,*,1,BI SMF record type
SMF017,17 *** Local symbol ***
SMF018,18 *** Local symbol ***
SMF021,21 *** Local symbol ***
SMF065,65 *** Local symbol ***
SMF066,66 *** Local symbol ***
SMFXTME,*,4,BI Time since midnight in
* hundredths of a second, since
* the record was moved into the
* SMF buffer
SMFXDTE,*,4,PD Date when the record was moved
SMFXDTE_DT1,=,4,DT1 Format the date for reporting
* into the SMF buffer, in the form
* 0ccydddf, where c is 0 for 19xx
* and c is 1 for 20xx.
SMFXSID,*,4,CH System identifier
SMFXSSI,*,4,CH Subsystem identifier
SMFXSTY,*,2,BI Subtype indicator

```

SMF65 ICETOOL SYMBOL FILE

```

*---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---*
*
* ICETOOL SYMBOL MAP FOR SMF RECORD TYPE 65
*
* Source documentation: SA22-7630-7
* Local symbols added where appropriate
*---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---*
POSITION,19
SMF65SBS,*,4,BI Reserved
SMF65SUB,*,2,CH The action taken on the catalog
* entry, valid values are:
* IN ( Insert )
* DE ( Delete )
* UP ( Update )
SMF65SUB_IN,C'IN' Insert
SMF65SUB_DE,C'IN' Delete
SMF65SUB_UP,C'IN' Update
SMF65POF,*,4,BI Offset of product section from

```

*		start of the record, including
*		the RDW.
SMF65PLN,* ,2,BI		Length of the product section
SMF65PNO,* ,2,BI		Number of product sections
SMF65DOF,* ,4,BI		Offset of data section from
*		start of the record, including
*		the RDW.
SMF65DLN,* ,2,BI		Length of the data section
SMF65DNO,* ,2,BI		Number of data sections
SMF65VER,* ,2,CH		Version of the type 65 record
SMF65PNM,* ,8,CH		Catalog management product
*		identifier.
SMF65JNM_1,* ,1,CH	*** Local symbol ***	
SMF65JNM_2,=,2,CH	*** Local symbol ***	
SMF65JNM_3,=,3,CH	*** Local symbol ***	
SMF65JNM_4,=,4,CH	*** Local symbol ***	
SMF65JNM_5,=,5,CH	*** Local symbol ***	
SMF65JNM_6,=,6,CH	*** Local symbol ***	
SMF65JNM_7,=,7,CH	*** Local symbol ***	
SMF65JNM,=,8,CH	Job name. The job log identifi-	
*	cation consists of the the job	
*	name, time and date that the	
*	reader recognized the JOB card	
*	(for this job). If a system task	
*	caused the record to be written,	
*	the job name and user identifi-	
*	cation fields contain blanks and	
*	the time and date fields contain	
*	zeros.	
SMF65RST,* ,4,BI	Time in hundredths of a second,	
*	that the reader recognized the	
*	JOB card (for this job).	
SMF65RDT,* ,4,DT1	Date when the reader recognized	
*	the JOB card (for this job), in	
*	the form 00cydddF.	
SMF65UID,* ,8,CH	User-defined identification	
*	field (taken from common exit	
*	parameter area, not from USER=	
*	parameter on job statement).	
SMF65FNC,* ,1,CH	Contains 'S' if a dataset was	
*	scratched; 'U' if only catalog	
*	entries were modified.	
SMF65FNC_S,C'S'	Scratched	
SMF65FNC_U,C'U'	Catalog entries modified.	
SMF65CNM,* ,44,CH	Name of the catalog in which	
*	record was updated or deleted.	
SMF65TYP,* ,1,CH	Entry type identifier.	
SMF65TYP_A,C'A'	Non-VSAM dataset	
SMF65TYP_B,C'B'	Generation data group base	
SMF65TYP_C,C'C'	Cluster	

SMF65TYP_D,C'D'	Dataset
SMF65TYP_E,C'E'	VSAM extension record
SMF65TYP_F,C'F'	Free space
SMF65TYP_G,C'G'	Alternate index
SMF65TYP_H,C'H'	Active generation dataset (GDS)
*	entry in GDG base
SMF65TYP_I,C'I'	Index
SMF65TYP_J,C'J'	GDG extension record
SMF65TYP_K,C'K'	VSAM volume record (VVR)
SMF65TYP_L,C'L'	Library control system library
*	record
SMF65TYP_M,C'M'	Master catalog
SMF65TYP_N,C'N'	Non-VSAM header record
SMF65TYP_O,C'O'	Object Access Method (OAM) non-
*	VSAM record
SMF65TYP_P,C'P'	Page space
SMF65TYP_Q,C'Q'	VVR header
SMF65TYP_R,C'R'	Path
SMF65TYP_T,C'T'	True name record
SMF65TYP_U,C'U'	User catalog
SMF65TYP_V,C'V'	Volume
SMF65TYP_W,C'W'	Library control system volume
SMF65TYP_X,C'X'	Alias
SMF65TYP_Y,C'Y'	Upgrade
SMF65TYP_Z,C'Z'	VVR header primary
SMF65TYP_Ø,X'ØØ'	Normal non-VSAM record
SMF65TYP_1,C'Ø1'	JES3 record
SMF65ENM_Ø1,*,Ø1,CH	Local symbol
SMF65ENM_Ø2,=,Ø2,CH	Local symbol
SMF65ENM_Ø3,=,Ø3,CH	Local symbol
SMF65ENM_Ø4,=,Ø4,CH	Local symbol
SMF65ENM_Ø5,=,Ø5,CH	Local symbol
SMF65ENM_Ø6,=,Ø6,CH	Local symbol
SMF65ENM_Ø7,=,Ø7,CH	Local symbol
SMF65ENM_Ø8,=,Ø8,CH	Local symbol
SMF65ENM_Ø9,=,Ø9,CH	Local symbol
SMF65ENM_1Ø,=,1Ø,CH	Local symbol
SMF65ENM_11,=,11,CH	Local symbol
SMF65ENM_12,=,12,CH	Local symbol
SMF65ENM_13,=,13,CH	Local symbol
SMF65ENM_14,=,14,CH	Local symbol
SMF65ENM_15,=,15,CH	Local symbol
SMF65ENM_16,=,16,CH	Local symbol
SMF65ENM_17,=,17,CH	Local symbol
SMF65ENM_18,=,18,CH	Local symbol
SMF65ENM_19,=,19,CH	Local symbol
SMF65ENM_2Ø,=,2Ø,CH	Local symbol
SMF65ENM,=,44,CH	Entry name.
SMF65NNM,*,44,CH	Reserved.
SMF65CRC,*,2,BI	Catalog record for updated or

* deleted entry (the length of
 * this record is contained in the
 * first two bytes of this field).

SMF66 ICETOOL SYMBOL FILE

```

*-----*
*
* ICETOOL SYMBOL MAP FOR SMF RECORD TYPE 66
*
* Source documentation: SA22-7630-7
* Local symbols added where appropriate
*-----*
POSITION,19
SMF66SBS,*,4,BI          Reserved
SMF66SUB,*,2,CH          The action taken on the catalog
*                          entry, valid values are:
*                          IN ( Insert )
*                          DE ( Delete )
*                          UP ( Update )
SMF66SUB_IN,C'IN'        Insert
SMF66SUB_DE,C'IN'        Delete
SMF66SUB_UP,C'IN'        Update
SMF66POF,*,4,BI          Offset of product section from
*                          start of the record, including
*                          the RDW.
SMF66PLN,*,2,BI          Length of the product section
SMF66PNO,*,2,BI          Number of product sections
SMF66DOF,*,4,BI          Offset of data section from
*                          start of the record, including
*                          the RDW.
SMF66DLN,*,2,BI          Length of the data section
SMF66DNO,*,2,BI          Number of data sections
SMF66VER,*,2,CH          Version of the type 65 record
SMF66PNM,*,8,CH          Catalog management product
*                          identifier.
SMF66JNM_1,*,1,CH        *** Local symbol ***
SMF66JNM_2,=,2,CH        *** Local symbol ***
SMF66JNM_3,=,3,CH        *** Local symbol ***
SMF66JNM_4,=,4,CH        *** Local symbol ***
SMF66JNM_5,=,5,CH        *** Local symbol ***
SMF66JNM_6,=,6,CH        *** Local symbol ***
SMF66JNM_7,=,7,CH        *** Local symbol ***
SMF66JNM,=,8,CH          Job name. The job log identifi-
*                          cation consists of the the job
*                          name, time, and date that the
*                          reader recognized the JOB card
*                          (for this job). If a system task
*                          caused the record to be written,

```

*	the job name and user identification fields contain blanks and
*	the time and date fields contain
*	zeros.
SMF66RST,*,4,BI	Time in hundredths of a second,
*	that the reader recognized the
*	JOB cad (for this job).
SMF66RDT,*,4,DT1	Date when the reader recognized
*	the JOB card (for this job), in
*	the form ØcyydddF.
SMF66UID,*,8,CH	User-defined identification
*	field (taken from common exit
*	parameter area, not from USER=
*	parameter on job statement).
SMF66FNC,*,1,CH	Contains 'R' if catalog entry is
*	renamed.
SMF66FNC_R,C'R'	Renamed
SMF66FNC_U,C'U'	Catalog entries modified.
SMF66CNM,*,44,CH	Name of the catalog in which
*	record was updated or deleted.
SMF66TYP,*,1,CH	Entry type identifier.
SMF66TYP_A,C'A'	Non-VSAM dataset
SMF66TYP_B,C'B'	Generation data group base
SMF66TYP_C,C'C'	Cluster
SMF66TYP_D,C'D'	Dataset
SMF66TYP_E,C'E'	VSAM extension record
SMF66TYP_F,C'F'	Free space
SMF66TYP_G,C'G'	Alternate index
SMF66TYP_H,C'H'	Active generation dataset (GDS)
*	entry in GDG base
SMF66TYP_I,C'I'	Index
SMF66TYP_J,C'J'	GDG extension record
SMF66TYP_K,C'K'	VSAM volume record (VVR)
SMF66TYP_L,C'L'	Library control system library
*	record
SMF66TYP_M,C'M'	Master catalog
SMF66TYP_N,C'N'	Non-VSAM header record
SMF66TYP_O,C'O'	Object Access Method (OAM) non-
*	VSAM record
SMF66TYP_P,C'P'	Page space
SMF66TYP_Q,C'Q'	VVR header
SMF66TYP_R,C'R'	Path
SMF66TYP_T,C'T'	True name record
SMF66TYP_U,C'U'	User catalog
SMF66TYP_V,C'V'	Volume
SMF66TYP_W,C'W'	Library control system volume
SMF66TYP_X,C'X'	Alias
SMF66TYP_Y,C'Y'	Upgrade
SMF66TYP_Z,C'Z'	VVR header primary
SMF66TYP_Ø,X'ØØ'	Normal non-VSAM record

RT	DATE	TIME	JOBNAME	ACTION	DATA SET NAME	TYPE OF DATA ENTITY
66	2005/09/03	18:00:01	DFHSM	UPDATE	SYS9.DSIT.DSQDBDEF.DSQTSEDEF.D0509003.T0735129	PATH
65	2005/09/03	18:00:03	AQRT09PJ	DELETE	AQRT01T.EDL070B.RUN070B.ERROR	NON-VSAM DATA SET
65	2005/09/03	18:00:03	AQRT09PJ	DELETE	AQRT01T.EDL070B.RUN070B.REPORT	NON-VSAM DATA SET
65	2005/09/03	18:00:04	AQRT09PJ	DELETE	AQRT01T.EDL070B.RUN070B.OUTPUT	NON-VSAM DATA SET
65	2005/09/03	18:00:04	AQRT09PJ	DELETE	AQRT01T.EZ.TEST.REPORT1	NON-VSAM DATA SET
65	2005/09/03	18:00:04	LACP6HP6	DELETE	QLAN55Q.RD5MEMD.CP6H10A	NON-VSAM DATA SET
66	2005/09/03	18:00:04	SMFDUMP	UPDATE	V235010	LIBRARY CONTROL
66	2005/09/03	18:00:04	SMFDUMP	UPDATE	RMVTS05	SYSTEM VOLUME
65	2005/09/03	18:00:05	SFCP6HP6	DELETE	SFPAPAT.CPRDTMP.CP6H10A	LIBRARY CONTROL
65	2005/09/03	18:00:05	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN064B.ERROR	NON-VSAM DATA SET
65	2005/09/03	18:00:05	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN064B.REPORT	NON-VSAM DATA SET
65	2005/09/03	18:00:05	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN064B.OUTPUT	NON-VSAM DATA SET
65	2005/09/03	18:00:05	AQRT01PL	DELETE	AQRT02T.LANSI347.SORT001.OUTPUT	NON-VSAM DATA SET
65	2005/09/03	18:00:05	AQRT01PL	DELETE	AQRT02T.LANSI347.SORT002.OUTPUT	NON-VSAM DATA SET
65	2005/09/03	18:00:06	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN043B.ERROR	NON-VSAM DATA SET
65	2005/09/03	18:00:06	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN043B.REPORT	NON-VSAM DATA SET
65	2005/09/03	18:00:06	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN043B.OUTPUT	NON-VSAM DATA SET
65	2005/09/03	18:00:06	AQRT01PL	DELETE	AQRT02T.LANSI347.RUN043B.V	NON-VSAM DATA SET
65	2005/09/03	18:00:06	AQRT01PL	DELETE	AQRT02T.LCLDATA.LOWLVL.OUTPUT.SORTED	NON-VSAM DATA SET
65	2005/09/03	18:00:07	AQRT01PL	DELETE	AQRT02T.LCLDATA.LOWLVL.LOAD.SUCCESS	NON-VSAM DATA SET
65	2005/09/03	18:00:07	DFHSM	UPDATE	SYS3.Q5ALTEC7.D050831.T234551.Q5ALTEC7	CLUSTER
65	2005/09/03	18:00:07	DFHSM	UPDATE	SYS3.Q5ALTEC7.D050831.T234551.Q5ALTEC7	CLUSTER
65	2005/09/03	18:00:07	DFHSM	DELETE	SYS3.Q5ALTEC7.D050831.T234551.Q5ALTEC7	CLUSTER
65	2005/09/03	18:00:07	DFHSM	DELETE	SYS3.Q5ALTEC7.D050831.T234551.Q5ALTEC7	CLUSTER
65	2005/09/03	18:00:07	DFHSM	DELETE	SYS3.Q5ALTEC7.D050831.T234551.Q5ALTEC7	CLUSTER

Figure 1: Sample report for example job 2

SMF66TYP_1,C'01'	JES3 record
SMF66ENM_01,*,01,CH	Local symbol
SMF66ENM_02,=,02,CH	Local symbol
SMF66ENM_03,=,03,CH	Local symbol
SMF66ENM_04,=,04,CH	Local symbol
SMF66ENM_05,=,05,CH	Local symbol
SMF66ENM_06,=,06,CH	Local symbol
SMF66ENM_07,=,07,CH	Local symbol
SMF66ENM_08,=,08,CH	Local symbol
SMF66ENM_09,=,09,CH	Local symbol
SMF66ENM_10,=,10,CH	Local symbol
SMF66ENM_11,=,11,CH	Local symbol
SMF66ENM_12,=,12,CH	Local symbol
SMF66ENM_13,=,13,CH	Local symbol
SMF66ENM_14,=,14,CH	Local symbol
SMF66ENM_15,=,15,CH	Local symbol
SMF66ENM_16,=,16,CH	Local symbol
SMF66ENM_17,=,17,CH	Local symbol
SMF66ENM_18,=,18,CH	Local symbol
SMF66ENM_19,=,19,CH	Local symbol
SMF66ENM_20,=,20,CH	Local symbol
SMF66ENM,=,44,CH	Entry name.
SMF66NNM,*,44,CH	Reserved.
SMF66CRC,*,2,BI	Catalog record for updated or
*	deleted entry (the length of
*	this record is contained in the
*	first two bytes of this field).

The *Update* family

In addition to *MVS Update*, the Xephon family of *Update* publications now includes *AIX Update*, *CICS Update*, *DB2 Update*, *MQ Update*, *RACF Update*, and *TCP/SNA Update*. Details of all of these can be found on the Xephon Web site at www.xephon.com.

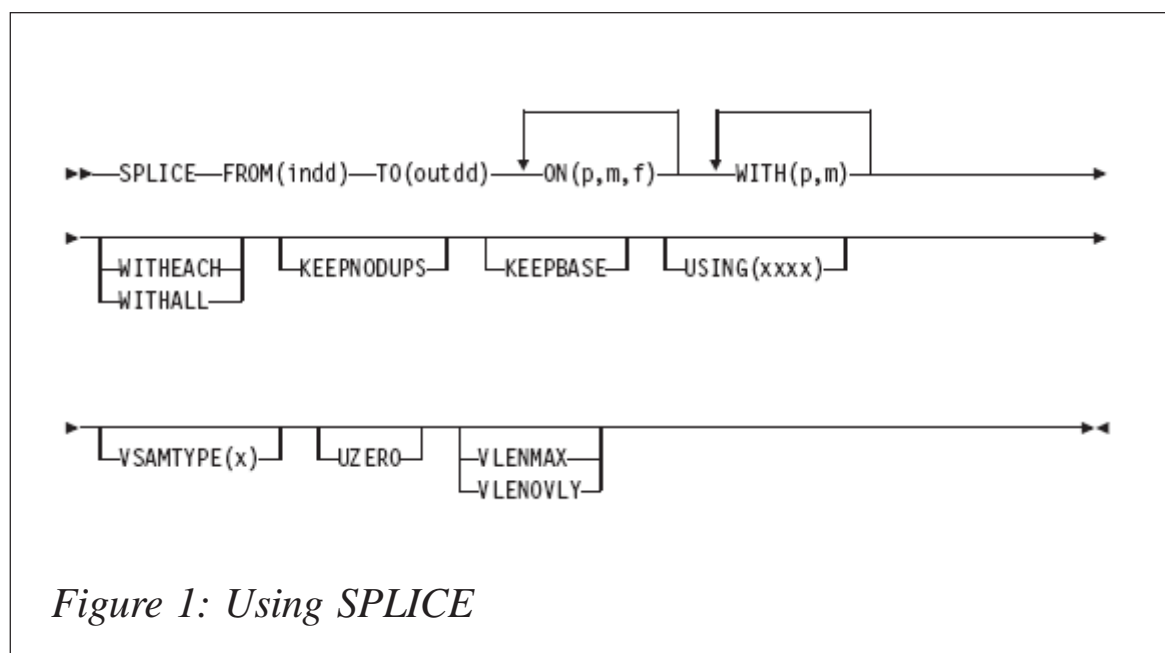
Using SPLICE, a DFSORT feature, to group information together using a large group of key values

INTRODUCTION

I recently submitted an article describing the use of SPLICE in a RACF-specific example. This was somewhat complicated and dealt with our complete RACF dump information. For those of you who don't have anything to do with RACF reporting, I thought I'd write it up again using a simple example. The original example had to cater for over 1700 key values, and this then shut out other methods of selecting and processing the data. This example will deal with a lot less data to enable a better overview of how SPLICE works, but the concept remains the same.

SPLICE

SPLICE is a relatively new feature of DFSORT and if you've not been keeping your eyes open is easy to overlook.



If you've got z/OS 1.6+ then you've got SPLICE; if you've got z/OS 1.4 or z/OS 1.5, you will need one of the following PTFs:

- DFSORT V1R5 PTF UQ95214.
- DFSORT V1R4 PTF UQ95213.

They are free and just need to be downloaded and applied.

SPLICE is a standard feature of DFSORT and is described in the IBM manual *DFSORT Application Programming Guide*, <http://publibz.boulder.ibm.com/epubs/pdf/ice1ca10.pdf>.

SPLICE splices together specified fields from records with matching (key) field values (ON), ie duplicates, but with differing information (WITH) to enable the joining of fields from different types of input record to create output records with information from two or more records. This is illustrated in Figure 1.

We will be using the WITHALL option, which produces, for each duplicate, a new record combining the original record with the information from each duplicate to deliver $n-1$ records (where n is the number of records found per key value).

DESCRIPTION

First, I put together a simple input file, which can be described as follows:

Type 0001 records:

User descriptor records (key records):

Pos 1-4 Type

Pos 20-26 Userid

Pos 40-44 Department

Type 0002, 0003 and 0004 records:

Money transfer records: Type 0002 Returns, Type 0003 Sales, Type 0004 Vouchers

Pos 1-4 Type

Pos 6-13 Date

Pos 16-23 Reference number

Pos 28-36 Value

Pos 60-66 Userid

Example input file:

```

ISREDDE2      USERID.SPLICE.EG.DATA                      Columns 00001 00072
Command ===>                                       Scroll ===> CSR
***** ***** Top of Data *****
=COLS>  -----1-----2-----3-----4-----5-----6-----+
000100 0001                                uid4711                                10001
000200 0002 21092005 00120341                2203.20                                uid4732
000300 0003 22092005 00133456                490.43                                 uid5311
000400 0003 20092005 00112343                4700.44                                uid5412
000500 0001                                uid4732                                10001
000600 0002 20092005 00096785                224.98                                 uidshop
000700 0002 21092005 00087655                429.33                                 uidshop
000800 0003 21092005 00082341                1801.11                                uidshop
000810 0003 20092005 00012343                2200.44                                uid5412
000900 0004 22092005 00000124                100.00                                 uidshop
001000 0001                                uid4811                                10001
001100 0002 20092005 00096781                2324.38                                uid5311
001200 0002 21092005 00087652                3929.53                                uid4711
001300 0002 20092005 00096783                4524.68                                uid5412
001400 0003 20092005 00096784                1324.78                                uid4811
001500 0003 21092005 00087656                1129.83                                uid5311
001600 0003 21092005 00082347                1001.91                                uid4811
001610 0003 21092005 00212343                500.54                                 uid5412
001700 0004 22092005 00000128                200.00                                 uid5311
001800 0001                                uid4912                                10001
001900 0001                                uid5111                                10001
002000 0002 20092005 00296784                1124.78                                uid5232
002100 0003 21092005 00287656                1429.83                                uid5311
002200 0003 21092005 00282347                1701.91                                uid5232
002300 0001                                uid5232                                10002
002400 0002 21092005 00387652                3129.53                                uid4711
002500 0002 20092005 00396783                4824.68                                uid4711
002600 0003 20092005 00396784                1224.78                                uid4811
002610 0003 22092005 00182347                1501.99                                uid5232
002700 0004 20092005 00300714                100.00                                 uid5311
002800 0001                                uid5311                                10001
002900 0002 20092005 00196175                1121.38                                uid5311
003000 0002 21092005 00187265                121.53                                 uid4711
003100 0002 20092005 00196355                1629.68                                uid4711
003200 0003 20092005 00196445                4149.78                                uid4811
003300 0003 21092005 00187535                4128.83                                uid5311
003400 0003 21092005 00182621                491.91                                 uid4811
003500 0004 22092005 00100714                100.00                                 uid5311
003600 0001                                uid5412                                10002
003700 0001                                uidshop                                20000
***** ***** Bottom of Data *****

```

EXAMPLE REPORT

To produce a list of the userids belonging to department 10001 and the sales attributed to these users:

- Select the userids belonging to department 10001.
- Write the results to a temporary dataset with the userids in the same position as those in the type 0002, 0003, and 0004 records. Retain record type in positions 1–4.
- Select the type 0003 records and write 'as is' to a second temporary dataset.
- Perform a SPLICE ON the userid WITH value and date on a concatenation of the two previously produced temporary datasets.
- Discard the records not belonging to the selected userids.
- Sort on userid and date and report the sales per selected userid.

JCL

```
//USERIDX JOB (SR00,SR016882,SR016882),'SPLICE...',
//          CLASS=0,MSGCLASS=H,REGION=0M,
//          NOTIFY=USERID,MSGLEVEL=(1,1)
//* *****
//* DATE: 22.09.2005   TIME: 10:25:50   USER: USERID
//* LIB : USERID.SYS.CNTL(SPLICE01)
//* *****
//* DOC : .....
//*          .....
//* *****
//DEL01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE USERID.TEMP3
DELETE USERID.COPY.TEMP1
DELETE USERID.COPY.TEMP2
DELETE USERID.COPY.TEMP3
DELETE USERID.OUTDD.TEMP
SET MAXCC = 0
/*
//SPLICE EXEC PGM=ICETOOL
//DFSMSG DD SYSOUT=*
```

```

//TOOLMSG DD SYSOUT=*
//PRINT00 DD SYSOUT=*
//INDD1 DD DISP=SHR,DSN=USERID.SPLICE.EG.DATA
//TEMP0001 DD DISP=(NEW,DELETE,DELETE),SPACE=(CYL,(500,50,0))
//TEMP1 DD DSN=&&TEMP1,UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
//TEMP2 DD DSN=&&TEMP2,UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
//TEMP3 DD DISP=(,CATLG),DSN=USERID.TEMP3,SPACE=(CYL,(1,1))
//TEMP4 DD DISP=(,CATLG),DSN=USERID.COPY.TEMP1,SPACE=(CYL,(1,1))
//TEMP5 DD DISP=(,CATLG),DSN=USERID.COPY.TEMP2,SPACE=(CYL,(1,1))
//TEMP6 DD DISP=(,CATLG),DSN=USERID.COPY.TEMP3,SPACE=(CYL,(1,1))
//OUTDD DD DISP=(,CATLG),DSN=USERID.OUTDD.TEMP,SPACE=(CYL,(1,1))
//CON DD DSN=*.TEMP1,VOL=REF=*.TEMP1,DISP=(OLD,CATLG)
// DD DSN=*.TEMP2,VOL=REF=*.TEMP2,DISP=(OLD,CATLG)
//PRINT00 DD SYSOUT=*
//TOOLIN DD *
*****
COPY FROM(INDD1) TO(TEMP1) USING(CTL1)
COPY FROM(TEMP1) TO(TEMP4)
COPY FROM(INDD1) TO(TEMP2) USING(CTL2)
COPY FROM(TEMP2) TO(TEMP5)
SPLICE FROM(CON) TO(TEMP3) ON(60,7,CH) -
WITHALL WITH(5,55)
COPY FROM(TEMP3) TO(TEMP6)
SORT FROM(TEMP3) TO(OUTDD) USING(CTL3)
DISPLAY FROM(OUTDD) LIST(PRINT00) BLANK -
TITLE('DEPARTMENT 10001 SALES PER USER AND DATE') -
ON(60,7,CH) HEADER('USERID') -
ON(6,8,CH) HEADER('DATE') -
ON(28,9,CH) HEADER('VALUE') -
BETWEEN(1) LINES(32)
/*
//CTL1CNTL DD *
INCLUDE COND=(40,5,CH,EQ,C'10001')
OUTREC FIELDS=(1:1,4,5:55X'40',60:20,7)

/*
//CTL2CNTL DD *
INCLUDE COND=(1,4,CH,EQ,C'0003')
OUTREC FIELDS=(1,15,16:12X'40',28:28,9,37:23X'40',
60:60,7)

/*
//CTL3CNTL DD *
OMIT COND=(1,4,CH,EQ,C'0003')
SORT FIELDS=(60,7,CH,A,6,8,CH,A)
OUTREC FIELDS=(1,66)
/*

```

RESULTS AND STAGE RESULTS

First copy:

```
ISREDDE2    USERID.COPY.TEMP1                Columns 00001 00066
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
=COLS>  ----+----1----+----2----+----3----+----4----+----5----+----6----+
000001 0001                                     uid4711
000002 0001                                     uid4732
000003 0001                                     uid4811
000004 0001                                     uid4912
000005 0001                                     uid5111
000006 0001                                     uid5311
***** ***** Bottom of Data *****
```

Second copy:

```
ISREDDE2    USERID.COPY.TEMP2                Columns 00001 00066
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
=COLS>  ----+----1----+----2----+----3----+----4----+----5----+----6----+
000001 0003 22092005                            490.43          uid5311
000002 0003 20092005                            4700.44         uid5412
000003 0003 21092005                            1801.11         uidshop
000004 0003 20092005                            2200.44         uid5412
000005 0003 20092005                            1324.78         uid4811
000006 0003 21092005                            1129.83         uid5311
000007 0003 21092005                            1001.91         uid4811
000008 0003 21092005                             500.54         uid5412
000009 0003 21092005                            1429.83         uid5311
000010 0003 21092005                            1701.91         uid5232
000011 0003 20092005                            1224.78         uid4811
000012 0003 22092005                            1501.99         uid5232
000013 0003 20092005                            4149.78         uid4811
000014 0003 21092005                            4128.83         uid5311
000015 0003 21092005                             491.91         uid4811
***** ***** Bottom of Data *****
```

SPLICE

```
ISREDDE2    USERID.COPY.TEMP3                Columns 00001 00066
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
=COLS>  ----+----1----+----2----+----3----+----4----+----5----+----6----+
000001 0001 20092005                            1324.78         uid4811
000002 0001 21092005                            1001.91         uid4811
000003 0001 20092005                            1224.78         uid4811
000004 0001 20092005                            4149.78         uid4811
000005 0001 21092005                             491.91         uid4811
000006 0003 22092005                            1501.99         uid5232
```

```

000007 0001 22092005          490.43          uid5311
000008 0001 21092005          1129.83         uid5311
000009 0001 21092005          1429.83         uid5311
000010 0001 21092005          4128.83         uid5311
000011 0003 20092005          2200.44         uid5412
000012 0003 21092005           500.54         uid5412
***** ***** Bottom of Data *****

```

SORT

```

ISREDDE2  USERID.OUTDD.TEMP          Columns 00001 00066
Command ==>          Scroll ==> CSR
***** ***** Top of Data *****
=COLS>  -+--+1--+2--+3--+4--+5--+6--+
000001 0001 20092005          1324.78         uid4811
000002 0001 20092005          1224.78         uid4811
000003 0001 20092005          4149.78         uid4811
000004 0001 21092005          1001.91         uid4811
000005 0001 21092005           491.91         uid4811
000006 0001 21092005          1129.83         uid5311
000007 0001 21092005          1429.83         uid5311
000008 0001 21092005          4128.83         uid5311
000009 0001 22092005           490.43         uid5311
***** ***** Bottom of Data *****

```

DISPLAY

```

COMMAND INPUT ==>          SCROLL ==> CSR
***** ***** TOP OF DATA *****
DEPARTMENT 10001 SALES PER USER AND DATE

USERID  DATE      VALUE
-----  -+--+
uid4811 20092005    1324.78
uid4811 20092005    1224.78
uid4811 20092005    4149.78
uid4811 21092005    1001.91
uid4811 21092005     491.91
uid5311 21092005    1129.83
uid5311 21092005    1429.83
uid5311 21092005    4128.83
uid5311 22092005     490.43
***** ***** BOTTOM OF DATA *****

```

COMMENTS

The first COPY produces a list of all userids belonging to

department 10001.

The second COPY produces a list of all type 0003 records.

The SPLICE selects duplicates; it produces a base record from the first duplicate. This base record is then subsequently overlaid with the 'WITH' fields from the following duplicates and written out to produce $n-1$ entries (where n is the number of similar key field records).

The record type in this example is not overwritten with the duplicate values.

When the base record originates from a type 0001 record the resulting output still has the type 0001 in positions 1 to 4.

The duplicates that have a type 0003 in positions 1 to 4 are duplicates, but do not originate from 0001 type records from department 10001. These records are then OMITted in the SORT.

CONCLUSION

SPLICE is a powerful feature of DFSORT that enables the selection of records based on a large list of key values. In our example these come from the same input file; this doesn't have to be the case. What is important is that the key values of the base records and the following duplicates must be in the same position. Furthermore, if there are no type identifiers (0001 and 0003 in this example) it is necessary to add one in a unique location in the initial COPY phases (eg position 68 = 1 for the first dataset and position 68 = 2 in the second dataset, then OMIT the position 68 = 2 records in the SORT phase). What is also important is that the two or more input files are concatenated with the base records' dataset first.

This example is a working example and runs as described when using the input dataset and JCL above. Try it and see.

Rolf Parker
Systems Programmer (Germany)

© Xephon 2005

A glimpse at the WLM's predictions – part 2

This month we conclude the code for a sample report writer that provides a starting point from which one can begin to ponder WLM's prediction process.

```
PTPMDP=c2d(substr(x.i,ptff+8,4))
/* Projected max. % of processor demanded */
/* at priority */
PTCPUU=c2d(substr(x.i,ptff+12,4)) /* Cpu using samples at priority */
PTCPUD=c2d(substr(x.i,ptff+16,4)) /* Cpu delay samples at priority */
PTW2UR=c2d(substr(x.i,ptff+20,4))
/* Wait-to-using ratio at priority *16 */
W2UR = format(PTW2UR/16,3,2)
PTAPU =c2d(substr(x.i,ptff+24,4))
/* Actual measured proc.used at priority */
PTPPU =c2d(substr(x.i,ptff+28,4))
/* Projected proc.time to be used at prty */
PTACMD=c2d(substr(x.i,ptff+32,4)) /* Achievable cumulative max demand */
/* for priorities affected by a move */
PTACMD= format(PTACMD/10,3,1)
PTIMAXD=c2d(substr(x.i,ptff+40,4)) /* Initial cumulative max demand */
PTIMAXD=format(PTIMAXD/10,3,1)
PTWMAXD=c2d(substr(x.i,ptff+44,4)) /* Projected cumulative max demand*/
PTWMAXD=format(PTWMAXD/10,3,1)
PTIAMTW=c2d(substr(x.i,ptff+48,4)) /* Initial avg. mean time to wait */
Select
  when PTIAMTW >'0' then P1IAMTW=format(PTIAMTW/1000,5,2)
  otherwise P1IAMTW = PTIAMTW
End
PTWAMTW=c2d(substr(x.i,ptff+52,4)) /*Projected avg. mean time to wait*/
Select
  when PTWAMTW >'0' then P1WAMTW=format(PTWAMTW/1000,5,2)
  otherwise P1WAMTW = PTWAMTW
End
PTSCPUU=c2d(substr(x.i,ptff+56,4)) /* Sample based cpu using samples */
/* at priority */
PTSCPUD=c2d(substr(x.i,ptff+60,4)) /* Sample based cpu delay samples */
/* at priority */
/* Printed variables: */
pty.ext = right(PTPRTY,4), /* Dispatch priority */
right(PTNP,4), /* New dispatch priority */
right(PTIMDP,4), /* % of CPU demanded - init */
right(PTPMDP,4), /* - projected*/
right(W2UR,5), /* Wait-to-using ratio */
right(PTAPU,5), /* CPU used - actual measured*/
right(PTPPU,5), /* - projected */
```

```

        right(PTACMD,5),          /* Cum. max demand - achievable */
        right(PTIMAXD,5),        /*           - initial          */
        right(PTWMAXD,5),        /*           - projected        */
        right(PIIAMTW,5),        /* MTTW - initial avg          */
        right(PIWAMTW,5),        /*           - projected avg.    */
        right(PTCPUU,5),         /* Cpu samples - using         */
        right(PTCPUD,5),         /*           - delay            */
        right(PTSCPUU,5),        /* Sample based cpu samples - using */
        right(PTSCPUD,5)         /*           - delay            */
end                               /* of extent*/
"EXECIO * DISKW s99p(STEM pty. )"
END                               /*           of Priority table*/
syspgc.1 =left(' ',1)
syspgc.2 =left('Part 3: WLM Plotting',25)
syspgc.3 =left('-',92,'-')
"EXECIO * DISKW s99p (STEM syspgc.)"
Select
when PAGP_LSTX > 0 then do
syspgg.1 =left(' ',14) left('Page fault rate:',18) left(xar,400)
syspgg.2 =left(' ',14) left('AUX. storage rate:',18) left(yar,400)
"EXECIO * DISKW s99p (STEM syspgg.)"
end
otherwise do
syspgg.1 =left('No SYSTEM wide paging delays found',60)
syspgg.2 =left('Class Per. Dly',14)
syspgg.3 =left('-',92,'-')
"EXECIO * DISKW s99p (STEM syspgg.)"
end
END
/*-----*/
/* Subtype 1: TRACE table entry - The trace */
/* table contains information about the policy and resource */
/* adjustment decisions made during the last policy interval */
/*-----*/
if (TOF <> 0) & (TLN <> 0) Then
do
trcc.1 = left(' ',1)
trcc.2 = left(i,3),
left('Part 3: TRACE entry at interval for period > 0',60)
trcc.3 = left('Class',6) left('Period',7) left('LPI',4),
left('SPI',5) left('WLM action code',26)
left('PAIid',6),
left('RAIid',5)
trcc.4 = left('-',65,'-')
/* EXECIO * DISKW s99p (STEM trcc. )" */
do j = 0 to TON -1
tff = (TOF + (j*TLN))- 3
EXTnum = j+1
TPID =c2d(substr(x.i,tff,1)) /* Policy adjustment interval id. */
TRID =c2d(substr(x.i,tff+1,1)) /* Resource adjustment interval id.*/

```

```

TCOD =c2d(substr(x.i,tff+2,2))      /* Policy adjustment action code */
TJOB =  substr(x.i,tff+4,8)         /* Job name of address space */
                                     /* effected by action being traced */
                                     /* (blank if not an address space */
                                     /* level action) */
TLPI =c2d(substr(x.i,tff+12,4))     /* Projected local performance */
                                     /* index to be acheived as a result*/
                                     /* of the traced action */
TSPI =c2d(substr(x.i,tff+16,4))     /* Projected sysplex performance */
                                     /* index to be acheived as a result*/
                                     /* of the traced action */
TGSR =c2d(substr(x.i,tff+20,4))     /* Projected resource group */
                                     /* service rate be acheived as a */
                                     /* result of the traced action */
TDT1 =c2d(substr(x.i,tff+24,4))     /* System use */
TDT2 =c2d(substr(x.i,tff+28,4))     /* System use */
TDT3 =c2d(substr(x.i,tff+32,4))     /* System use */
TRGN =  substr(x.i,tff+36,8)        /* Resource group Name */
                                     /* (blank if no resource group) */
TCNM =  substr(x.i,tff+44,8)        /* Service Class Name */
TPER =c2d(substr(x.i,tff+52,2))     /* Service period number */
TASID=c2d(substr(x.i,tff+54,2))     /* Address Space ID number */
LPI  = TLPI/100
SPI  = TSPI/100
c =1
SELECT
  when TPER > '0' then do
    trac.c = left(TCNM,8),
              right(tper,2),
              right(lpi,5),
              right(spi,5),
              right(tcod,5),
              left(TCOD,22),
              right(tpid,4),
              right(trid,5)
    /* "EXECIO * DISKW s99p (STEM trac. )" */
    c = c +1
  end
  otherwise nop
END /* select */
pa = 0
end
end
END /* of main loop */
/* Close & free all allocated files */
"EXECIO 0 DISKW S99P(FINIS "
"free FILE(S99P)"
exit
/*-----*/

```

```

/* Error exit routine */
/*-----*/
ERROR: say 'The following command produced non-zero RC =' RC
      say SOURCELINE(SIGL)
      exit
SUBTYPE3:
parse arg dof
/*-----*/
/* SMF99 subtype 3: Self-Defining section */
/*-----*/
DOX = DOF - 3
S993COF = c2d(substr(x.i,dof,4)) /* Offset to class information*/
S993CLN = c2d(substr(x.i,dof+4,2)) /* Length of class information*/
S993CON = c2d(substr(x.i,dof+6,2)) /* Number of class information*/
/*-----*/
/* SMF99 subtype 3 Class Data section - Contains information */
/* identifying the Service Class described in this record */
/*-----*/
COF = S993COF - 3
S99_PNAM.i= substr(x.i,cof,8) /* Service class name */
S993CPOF= c2d(substr(x.i,dof+8,4)) /* Offset to class period sec.*/
S993CPLN= c2d(substr(x.i,dof+12,2)) /* Length of class period sec.*/
S993CPON= c2d(substr(x.i,dof+14,2)) /* Number of class period sec.*/
/*-----*/
/* SMF99 subtype 3 period self definition section */
/*-----*/
IF (S993CPLN > 0) Then do
  do pp = 0 to S993CPON-1
    CPOF = (S993CPOF + (pp*S993CPLN))- 3
    pa = 1
    S993_PRPOF = c2d(substr(x.i,cpof,4))
                                /* Offset to paging rate plot for
                                this period */
    S993_PRPLN = c2d(substr(x.i,cpof+4,2)) /*Length of paging rate plot */
    S993_PRPON = c2d(substr(x.i,cpof+6,2)) /*Number of paging rate plots*/
  Select
    when S993_PRPON > 0 then call PPR S993_PRPOF S993_PRPLN S993_PRPON
    otherwise nop
  End
  pb = 1
  S993_MPLOF = c2d(substr(x.i,cpof+8,4)) /* Offset to MPL delay plots
                                for this period */
  S993_MPLLN = c2d(substr(x.i,cpof+12,2)) /* Length of MPL delay plots */
  S993_MPLON = c2d(substr(x.i,cpof+14,2)) /* Number of MPL delay plots */
  Select
    when S993_MPLOF > 0 then call MPL S993_MPLOF S993_MPLLN S993_MPLON
    otherwise nop
  End
  pc = 1
  S993_RUAOF = c2d(substr(x.i,cpof+16,4)) /* Offset to ready user average

```

```

                                plots for period          */
S993_RUALN = c2d(substr(x.i,cpof+20,2)) /* Length of ready user average
                                plot                          */
S993_RUAON = c2d(substr(x.i,cpof+22,2)) /* Number of ready user average
                                plots                          */
Select
  when S993_RUAOF > 0 then call RUA  S993_RUAOF S993_RUALN  S993_RUAON
  otherwise nop
End
  pd = 1
  S993_SWPOF = c2d(substr(x.i,cpof+24,4)) /* Offset to swap delay plots
                                for this period          */
  S993_SWPLN = c2d(substr(x.i,cpof+28,2)) /* Length of swap delay plot */
  S993_SWPON = c2d(substr(x.i,cpof+30,2)) /* Number of swap delay plots*/
Select
  when S993_SWPOF > 0 then call SDP  S993_SWPOF S993_SWPLN  S993_SWPON
  otherwise nop
End
  pe = 1
  S993_PASOF = c2d(substr(x.i,cpof+32,4)) /* Offset to proportional
                                aggregate speed plots for this period */
  S993_PASLN = c2d(substr(x.i,cpof+36,2)) /* Length of proportional
                                aggregate speed plot          */
  S993_PASON = c2d(substr(x.i,cpof+38,2)) /* Number of proportional
                                aggregate speed plots          */
Select
  when S993_PASOF > 0 then call PAS  S993_PASOF S993_PASLN  S993_PASON
  otherwise nop
End
  pf = 1
  S993_QMPLOF= c2d(substr(x.i,cpof+40,4)) /* Offset to queue delay plots
                                for this period          */
  S993_QMPLLN= c2d(substr(x.i,cpof+44,2)) /*Length of queue delay plot */
  S993_QMPLON= c2d(substr(x.i,cpof+46,2)) /*Number of queue delay plots*/
Select
  when S993_QMPLOF > 0 then call QDP  S993_QMPLOF S993_QMPLLN
  S993_QMPLON
  otherwise nop
End
  S993_QRUAOF= c2d(substr(x.i,cpof+48,4)) /* Offset to queue ready user
                                average for this period    */
  S993_QRUALN= c2d(substr(x.i,cpof+52,2)) /* Length of queue ready user
                                average plot                */
  S993_QRUAON= c2d(substr(x.i,cpof+54,2)) /* Number of queue ready user
                                average plot                */
  pg = 1
Select
  when S993_QRUAOF > 0 then call QRU  S993_QRUAOF S993_QRUALN
  S993_QRUAON
  otherwise nop

```

```

End
S993_INT_CLASS_NAME=substr(x.i,cpof+56,8) /*Internal class name of the
                                         period. For non-discretionary
                                         periods this will be the same
                                         as the external class name.
                                         For discretionary periods this
                                         will be of the form $SRMDIxx.
                                         For dynamic periods this will
                                         be of the form $SRMSxxx.   */

S993_QSTPOF= c2d(substr(x.i,cpof+64,4)) /* Offset to queue service time
                                         for this period           */
S993_QSTPLN= c2d(substr(x.i,cpof+68,2)) /* Length of queue service time
                                         plot                       */
S993_QSTPON= c2d(substr(x.i,cpof+70,2)) /* Number of queue service time
                                         plot                       */

  ph = 1
Select
  when S993_QSTPOF > 0 then call QST  S993_QSTPOF S993_QSTPLN
S993_QSTPON
  otherwise nop
End
S993_AINSOF= c2d(substr(x.i,cpof+72,4)) /* Offset to active server
                                         instance plot for this period */
S993_AINSLN= c2d(substr(x.i,cpof+76,2)) /* Length of active server
                                         instance plot           */
S993_AINSON= c2d(substr(x.i,cpof+78,2)) /* Number of active server
                                         instance plot           */

  pj = 1
Select
  when S993_AINSOF > 0 then call ASI S993_AINSOF S993_AINSLN S993_AINSON
  otherwise nop
End
S993_ASTROF= c2d(substr(x.i,cpof+80,4)) /*Offset to virtual storage
                                         plot for active server instances for
                                         this period           */
S993_ASTRLN= c2d(substr(x.i,cpof+84,2))
                                         /* Length of virtual storage plot*/
S993_ASTRON= c2d(substr(x.i,cpof+86,2))
                                         /* Number of virtual storage plot*/

  pk = 1
Select
  when S993_ASTROF > 0 then call VS1 S993_ASTROF S993_ASTRLN S993_ASTRON
  otherwise nop
End
S993_TSTROF= c2d(substr(x.i,cpof+88,4)) /* Offset to virtual storage
                                         plot for total server instances for
                                         this period           */
S993_TSTRLN= c2d(substr(x.i,cpof+92,2))
                                         /* Length of virtual storage plot*/

```

```

S993_TSTRON= c2d(substr(x.i,cpof+94,2))
/* Number of virtual storage plot*/
Select
  when S993_TSTROF > 0 then call VS2 S993_TSTROF S993_TSTRLN S993_TSTRON
  otherwise nop
End
  End
  End
Return
PPR:
/*-----*/
/* SMF99 subtype 3 Period Paging Rate Plot section */
/* - Contains paging rate information for the service period. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do zp = 0 to num -1
    ofp = (off + (zp*len))- 3
PPRP_PNUM.i =c2d(substr(x.i,ofp,4)) /* Period number */
PPRP_BW =c2d(substr(x.i,ofp+4,4)) /* Bucket width */
PPRP_LSTX.i =c2d(substr(x.i,ofp+8,4)) /*Last plotted x bucket*/
PPRP_POINTS_OF.i =c2d(substr(x.i,ofp+16,4))
/* Offset of point entries */
PPRP_POINTS_ON.i =c2d(substr(x.i,ofp+20,2))
/* Number of point entries */
PPRP_POINTS_LN.i =c2d(substr(x.i,ofp+22,2))
/* Length of a point entry */
Select
  when PPRP_POINTS_ON.i >0 then call XYPROC ,
    PPRP_POINTS_OF.i PPRP_POINTS_LN.i PPRP_POINTS_ON.i PPRP_LSTX.i
  otherwise nop
End
pprrec.pa=left(S99_PNAM.i,7) left(PPRP_PNUM.i,2) left('PPR',3),
left('Avg.ASID size:',18) left(xar,400); pa = pa + 1
pprrec.pa=left(' ',14,' ') left('Page fault rate:',18) ,
left(yar,400); pa = pa + 1
"EXECIO * DISKW s99P (STEM pprrec.)"
end /* end of Period Paging Rate Plotloop */
Return
MPL:
/*-----*/
/* SMF99 subtype 3 MPL Delay Plot section - */
/* Contains MPL information for the service period. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do zm = 0 to num -1
    ofm = (off + (zm*len))- 3
S99_MPLP_PNUM.i =c2d(substr(x.i,ofm,4)) /* Period number */
S99_MPLP_BW =c2d(substr(x.i,ofm+4,4)) /* Bucket width */

```

```

S99_MPLP_LSTX      =c2d(substr(x.i,ofm+8,4)) /*Last plotted x bucket*/
S99_MPLP_POINTS_OF =c2d(substr(x.i,ofm+16,4))
                                     /* Offset of point entries */
S99_MPLP_POINTS_ON =c2d(substr(x.i,ofm+20,2))
                                     /* Number of point entries */
S99_MPLP_POINTS_LN =c2d(substr(x.i,ofm+22,2))
                                     /* Length of a point entry */

Select
  when S99_MPLP_POINTS_ON >0 then call XYPROC ,
  S99_MPLP_POINTS_OF  S99_MPLP_POINTS_LN  S99_MPLP_POINTS_ON
S99_MPLP_LSTX
  otherwise nop
End
mpprec.pb= left(S99_PNAM.i,7) left(S99_MPLP_PNUM.i,2) left('MPL',3),
           left('Pct of ready users',18) left(xar,400); pb = pb + 1
mpprec.pb =left(' ',14,' ')           left('MPL delay - msec:',18) ,
           left(yar,400); pb = pb + 1
"EXECIO * DISKW s99P (STEM mpprec.)"
end /* end of MPL Delay Plotloop */

Return
RUA:
parse arg off len num
if (len <> 0) Then do
do zr = 0 to num -1
  ofr = (off + (zr*len))- 3
  /*-----*/
  /* SMF99 subtype 3 Ready User Average Plot section - */
  /* Contains ready user average information for the service */
  /* period. */
  /*-----*/
S99_RUAP_PNUM.i =c2d(substr(x.i,ofr,4)) /* Period number */
S99_RUAP_BW =c2d(substr(x.i,ofr+4,4)) /* Bucket width */
S99_RUAP_LSTX =c2d(substr(x.i,ofr+8,4)) /*Last plotted x bucket*/
S99_RUAP_POINTS_OF =c2d(substr(x.i,ofr+16,4))
                                     /* Offset of point entries */
S99_RUAP_POINTS_ON =c2d(substr(x.i,ofr+20,2))
                                     /* Number of point entries */
S99_RUAP_POINTS_LN =c2d(substr(x.i,ofr+22,2))
                                     /* Length of a point entry */

Select
  when S99_RUAP_POINTS_ON >0 then call XYPROC ,
  S99_RUAP_POINTS_OF  S99_RUAP_POINTS_LN  S99_RUAP_POINTS_LN
S99_RUAP_LSTX
  otherwise nop
End
ruarec.pc= left(S99_PNAM.i,7) left(S99_RUAP_PNUM.i,2) left('RUA',3),
           left('MPL slots free:',18) left(xar,400); pc = pc + 1
ruarec.pc =left(' ',14,' ')           left('Max Ready users:',18) ,
           left(yar,400); pc = pc + 1
"EXECIO * DISKW s99p (STEM ruarec.)"

```



```

        end    /* end of Ready User Average Plot  loop */
Return
SDP:
/*-----*/
/* SMF99 subtype 3 Swap Delay Plot section - */
/* Contains information about delays caused by address spaces */
/* in the service period being swapped out. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
do zs = 0 to num -1
ofs = (off + (zs*len))- 3
S99_SWPP_PNUM.i =c2d(substr(x.i,ofs,4)) /* Period number */
S99_SWPP_BW =c2d(substr(x.i,ofs+4,4)) /* Bucket width */
S99_SWPP_LSTX =c2d(substr(x.i,ofs+8,4)) /* Last plotted x bucket*/
S99_SWPP_POINTS_OF=c2d(substr(x.i,ofs+16,4))
/* Offset of point entries */
S99_SWPP_POINTS_ON=c2d(substr(x.i,ofs+20,2))
/* Number of point entries */
S99_SWPP_POINTS_LN=c2d(substr(x.i,ofs+22,2))
/* Length of a point entry */
Select
when S99_SWPP_POINTS_ON >0 then call XYPROC ,
S99_SWPP_POINTS_OF S99_SWPP_POINTS_LN S99_SWPP_POINTS_ON
S99_SWPP_LSTX
otherwise nop
End
sdprec.pd= left(S99_PNAM.i,7) left(S99_SWPP_PNUM.i,2) left('SWP',3),
left('Avg.swap time-msec',18) left(xar,400); pd = pd + 1
sdprec.pd =left(' ',14,' ') left('Swap delay - msec:',18) ,
left(yar,400); pd = pd + 1
"EXECIO * DISKW s99p (STEM sdprec. )"
end /* end of Swap Delay Plot loop */
Return
PAS:
/*-----*/
/* SMF 99 subtype 3 Proportional Aggregate Speed Plot section. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
do za = 0 to num -1
ofa = (off + (za*len))- 3
S99_PASP_PNUM =c2d(substr(x.i,ofa,4)) /* Period number */
S99_PASP_BW =c2d(substr(x.i,ofa+4,4)) /* Bucket width */
S99_PASP_LSTX =c2d(substr(x.i,ofa+8,4)) /*Last plotted x bucket */
S99_PASP_POINTS_OF=c2d(substr(x.i,ofa+16,4))
/* Offset of point entries */
S99_PASP_POINTS_ON=c2d(substr(x.i,ofa+20,2))
/* Number of point entries */
S99_PASP_POINTS_LN=c2d(substr(x.i,ofa+22,2))

```

```

                                                    /* Length of a point entry */
Select
  when S99_PASP_POINTS_ON          >0 then call XYPROC ,
    S99_PASP_POINTS_OF    S99_PASP_POINTS_LN  S99_PASP_POINTS_ON
S99_PASP_LSTX
  otherwise nop
End
pasrec.pe =left(S99_PNAM.i,7) left(S99_PASP_PNUM,2) left('PAS',3),
           left('Proportional speed:',18) left(xar,300); pe = pe + 1
pasrec.pe =left(' ',14,' ') left('Speed PI:',18) ,
           left(yar,300) ; pe = pe + 1
"EXECIO * DISKW s99p (STEM pasrec.)"
end /* end of Proportional Aggregate Speed loop */
Return
QDP:
/*-----*/
/* SMF99 subtype 3: Queue Delay Plot section. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do zq = 0 to num -1
    ofq = (off + (zq*len))- 3
S99_QMPLP_PNUM      =c2d(substr(x.i,ofq,4)) /* Period number */
S99_QMPLP_CLASS_NAME= substr(x.i,ofq+4,8) /* Class name of DISP
                                           where server address spaces
                                           are running */
S99_QMPLP_BW       =c2d(substr(x.i,ofq+12,4)) /* Bucket width */
S99_QMPLP_LSTX     =c2d(substr(x.i,ofq+16,4)) /*Last plotted x bucket*/
S99_QMPLP_POINTS_OF=c2d(substr(x.i,ofq+24,4))
                                           /* Offset of point entries */
S99_QMPLP_POINTS_ON=c2d(substr(x.i,ofq+28,2))
                                           /* Number of point entries */
S99_QMPLP_POINTS_LN=c2d(substr(x.i,ofq+30,2))
                                           /* Length of a point entry */
S99_QMPLP_Q_TYPE=x2b(c2x(substr(x.i,ofq+32,1))) /* Work queue type */
S993WQT = substr(S99_QMPLP_Q_TYPE,1,1)
Select
  when S993WQT = '1' then type ='Queue manager'
                                           /* Queue manager type work*/
  otherwise type ='Batch'
                                           /* Batch type work queue */
End
S99_QMPLP_SUBSYS_TYPE= substr(x.i,ofq+36,4) /* Subsystem type of
                                           owner of queue. Only applies to batch
                                           queue servers. */
S99_QMPLP_SUBSYS_NAME= substr(x.i,ofq+40,8) /* Subsystem name of
                                           owner of queue. Only applies to batch
                                           queue servers. */
Select
  when S99_QMPLP_POINTS_ON          >0 then call XYPROC ,

```

```

S99_QMPLP_POINTS_OF  S99_QMPLP_POINTS_LN  S99_QMPLP_POINTS_ON
S99_QMPLP_LSTX
  otherwise nop
End
qdprec.pf= left(S99_PNAM.i,7) left(S99_QMPLP_PNUM,2) left('QDP',3),
           left('Work req. ratio:',18) left(xar,400); pf = pf + 1
qdprec.pf= left(' ',14,' ') left('Delay per work:',18) ,
           left(yar,400);   pf = pf + 1
"EXECIO * DISKW s99p (STEM qdprec.)"
end /* end Queue Delay Plot loop */
Return
QRU:
/*-----*/
/* SMF99 subtype 3 Queue Ready User Avg Plot section.          */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do zg = 0 to num -1
    ofg = (off + (zg*len))- 3
S99_QRUAP_PNUM      =c2d(substr(x.i,ofg,4))      /* Period number      */
S99_QRUAP_CLASS_NAME= substr(x.i,ofg+4,8)      /* Class name of DISP
                                           where server address spaces
                                           are running          */
S99_QRUAP_BW       =c2d(substr(x.i,ofg+12,4))   /* Bucket width       */
S99_QRUAP_LSTX     =c2d(substr(x.i,ofg+16,4))
                                           /* Last plotted x bucket */
S99_QRUAP_POINTS_OF =c2d(substr(x.i,ofg+24,4))
                                           /* Offset of point entries */
S99_QRUAP_POINTS_ON =c2d(substr(x.i,ofg+28,2))
                                           /* Number of point entries */
S99_QRUAP_POINTS_LN =c2d(substr(x.i,ofg+30,2))
                                           /* Length of a point entry */

Select
  when S99_QRUAP_POINTS_ON >0 then call XYPROC ,
    S99_QRUAP_POINTS_OF  S99_QRUAP_POINTS_LN  S99_QRUAP_POINTS_ON
S99_QRUAP_LSTX
  otherwise nop
End
qrurec.pj= left(S99_PNAM.i,7) left(S99_QRUAP_PNUM,2) left('QRU',3),
           left('X axis - desc.?:',18) left(xar,400); pj = pj + 1
qrurec.pj= left(' ',14,' ') left('Y axis - desc.?:',18) ,
           left(yar,400);   pj = pj + 1
"EXECIO * DISKW s99p (STEM qrurec.)"
end /* end of Queue Ready User Avg */
Return
ASI:
/*-----*/
/* SMF99 subtype 3 Active Server Instances Plot section.          */
/*-----*/
parse arg off len num

```

```

if (len <> 0) Then do
  do as = 0 to num -1
    ofj = (off + (as*len))- 3
S99_AINS_PNUM      =c2d(substr(x.i,ofj,4))          /* Period number */
S99_AINS_BW       =c2d(substr(x.i,ofj+4,4))        /* Bucket width */
S99_AINS_LSTX     =c2d(substr(x.i,ofj+8,4))        /*Last plotted x bucket*/
S99_AINS_POINTS_OF=c2d(substr(x.i,ofj+16,4))
                                                         /*Offset of point entries*/
S99_AINS_POINTS_ON=c2d(substr(x.i,ofj+20,2))
                                                         /* Number of point entries*/
S99_AINS_POINTS_LN=c2d(substr(x.i,ofj+22,2))
                                                         /* Length of a point entry*/

Select
  when S99_AINS_POINTS_ON      >0 then call XYPROC ,
    S99_AINS_POINTS_OF  S99_AINS_POINTS_LN  S99_AINS_POINTS_ON
S99_AINS_LSTX
  otherwise nop
End
asirec.pg= left(S99_PNAM.i,7) left(S99_AINS_PNUM,2) left('ASI',3),
           left('X axis - desc.?:',18) left(xar,400); pg = pg + 1
asirec.pg= left(' ',14,' ') left('Y axis - desc.?:',18) ,
           left(yar,400); pg = pg + 1
"EXECIO * DISKW s99p (STEM asirec.)"
end
  end
Return
QST:
/*-----*/
/* SMF99 subtype 3 Queue service time Plot section. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do aw = 0 to num -1
    afg = (off + (aw*len))- 3
S99_QSTP_PNUM      =c2d(substr(x.i,afg,4))          /* Period number */
S99_QSTP_CLASS_NAME= substr(x.i,afg+4,8)          /* Class name of where
server address spaces are
running */
S99_QSTP_BW       =c2d(substr(x.i,afg+12,4)) /* Bucket width */
S99_QSTP_LSTX     =c2d(substr(x.i,afg+16,4))
                                                         /* Last plotted x bucket */
S99_QSTP_POINTS_OF=c2d(substr(x.i,afg+24,4))
                                                         /* Offset of point entries */
S99_QSTP_POINTS_ON=c2d(substr(x.i,afg+28,2))
                                                         /* Number of point entries */
S99_QSTP_POINTS_LN=c2d(substr(x.i,afg+30,2))
                                                         /* Length of a point entry */

Select
  when S99_QSTP_POINTS_ON > 0 then call XYPROC ,
    S99_QSTP_POINTS_OF  S99_QSTP_POINTS_LN  S99_QSTP_POINTS_LN

```

```

S99_QSTP_LSTX
  otherwise nop
End
qstrec.ph= left(S99_PNAM.i,7) left(S99_QSTP_PNUM,2) left('QST',3),
           left('X axis - desc.?:',18) left(xar,400); ph = ph + 1
qstrec.ph= left(' ',14,' ') left('Y axis - desc.?:',18) ,
           left(yar,400);   ph = ph + 1
  "EXECIO * DISKW s99p (STEM qstrec.)"
  end
end   /* end of Queue service time Plot   */
Return
VS1:
  /*-----*/
  /* SMF99 subtype 3 VS Plot for active server instances.      */
  /*-----*/
parse arg off len num
if (len <> 0) Then do
  do cw = 0 to num -1
    cfg = (off + (cw*len))- 3
S99_ASTR_PNUM      =c2d(substr(x.i,cfg,4))          /* Period number   */
S99_ASTR_BW       =c2d(substr(x.i,cfg+4,4))        /* Bucket width    */
S99_ASTR_LSTX     =c2d(substr(x.i,cfg+8,4))        /*Last plotted x bucket */
S99_ASTR_POINTS_OF=c2d(substr(x.i,cfg+16,4))
                                                         /*offset of point entries*/
S99_ASTR_POINTS_ON=c2d(substr(x.i,cfg+20,2))
                                                         /* Number of point entries*/
S99_ASTR_POINTS_LN=c2d(substr(x.i,cfg+22,2))
                                                         /* Length of a point entry*/
S99_ASTR_C_USED  =x2b(c2x(substr(x.i,cfg+24,1)))    /* Plot Curve used */
VSLINE= substr(S99_ASTR_C_USED,1,1)
Select
  when VSLINE = '1' then Curve ='Below line'   /* VS curve below 16MB */
                                                         /* was used last time server */
                                                         /* instances have been */
                                                         /* adjusted */
  otherwise          Curve ='Above line'   /* VS curve above 16MB */
End
                                                         /* was used last time server */
                                                         /* instances have been */
                                                         /* adjusted */

Select
  when S99_ASTR_POINTS_ON ^= 0 then call TWOC ,
    S99_ASTR_POINTS_OF S99_ASTR_POINTS_LN S99_ASTR_POINTS_ON
  otherwise nop
End
vs1rec.pk= left(S99_PNAM.i,7) left(S99_ASTR_PNUM,2) left('VS1',3),
           left('X axis - desc.?: ',19) left(x1arw,400); pk = pk + 1
vs1rec.pk= left(' ',14,' ') left('Y1 axis - desc.?: ',19) ,
           left(y1arw,400); pk = pk + 1
vs1rec.pk= left(' ',14,' ') left('Y2 axis - desc.?: ',19) ,
           left(y2arw,400); pk = pk + 1

```

```

"EXECIO * DISKW s99p (STEM vs1rec.)"
end
  end
Return
VS2:
/*-----*/
/* SMF99 subtype 3 VS Plot for Total server instances. */
/*-----*/
parse arg off len num
if (len <> 0) Then do
  do cw = 0 to num -1
    cfg = (off + (cw*len))- 3
S99_TSTR_PNUM      =c2d(SUBSTR(x.i,cfg,4))      /* Period number */
S99_TSTR_BW       =c2d(SUBSTR(x.i,cfg+4,4))    /* Bucket width */
S99_TSTR_LSTX     =c2d(SUBSTR(x.i,cfg+8,4))    /*Last plotted x bucket*/
S99_TSTR_POINTS_OF=c2d(SUBSTR(x.i,cfg+16,4))
/*Offset of point entries*/
S99_TSTR_POINTS_ON=c2d(SUBSTR(x.i,cfg+20,2))
/* Number of point entries*/
S99_TSTR_POINTS_LN=c2d(SUBSTR(x.i,cfg+22,2))
/* Length of a point entry*/
S99_TSTR_C_USED  =x2b(c2x(SUBSTR(x.i,cfg+24,1))) /* Plot Curve used*/
VSLINE= SUBSTR(S99_TSTR_C_USED,1,1)
Select
  when VSLINE = '1' then Curve ='Below line' /* VS curve below 16MB */
/* was used last time server */
/* instances have been */
/* adjusted */
  otherwise          Curve ='Above line' /* VS curve above 16MB */
End /* was used last time server */
/* instances have been */
/* adjusted */

Select
  when S99_TSTR_POINTS_ON ^= 0 then call TWOC ,
  S99_TSTR_POINTS_OF S99_TSTR_POINTS_LN S99_TSTR_POINTS_ON
  otherwise nop
End
  end
Return
SUBTYPE4:
/*-----*/
/* SMF99 subtype 4: self defining section / main loop - start */
/*-----*/
parse arg dof
DOZ = dof -3
S994DEVCLID = c2d(SUBSTR(x.i,dof,4)) /*Identifier of device cluster */
S994IOPTOF  = c2d(SUBSTR(x.i,dof+4,4)) /*Offset to I/O priority table*/
S994IOPTLN  = c2d(SUBSTR(x.i,dof+8,2)) /*Length of I/O priority info*/
S994IOPTON  = c2d(SUBSTR(x.i,dof+10,2)) /*Num. of priority table sec.*/
S994IOPLTOF = c2d(SUBSTR(x.i,dof+12,4)) /*Offset to the I/O plot sec.*/

```

```

S994IOPLTLN = c2d(SUBSTR(x.i,doz+16,2)) /* Length of I/O plot sec. */
S994IOPLTON = c2d(SUBSTR(x.i,doz+18,2)) /* Number of I/O plot sec. */
/*-----*/
/* SMF99 subtype 4 device cluster priority table section - */
/* Contains information about priorities being used in */
/* the device cluster */
/*-----*/
IF (S994IOPTON > 0) Then do
  do po = 0 to S994IOPTON -1
    poof = (S994IOPTOF + (po*S994IOPTLN))- 3
  pz = 1
  S99_IPTPRTY =c2d(SUBSTR(x.i,poof,2)) /* I/O priority */
  S99_IPTNP =c2d(SUBSTR(x.i,poof+2,2))
                /* New I/O priority;0=not changed) */
  S99_IPTIMDP =c2d(SUBSTR(x.i,poof+4,4))
                /* Initial max. % of time work at */
                /* priority demand I/O, initial */
                /* value before any priority moves */
                /* Percentage scaled by 10 */
  S99_IPTPMDP =c2d(SUBSTR(x.i,poof+8,4)) /* Projected max. % of I/O */
                /* time demanded at priority */
  S99_IPTW2UR =c2d(SUBSTR(x.i,poof+12,4)) /* Wait-to-using ratio at */
                /* priority (*16) */
  S99_IPTIMAXD =c2d(SUBSTR(x.i,poof+16,4)) /*Init cumulative max.demand */
                /* Percentage scaled by 10. */
  S99_IPTWMAXD =c2d(SUBSTR(x.i,poof+20,4))
                /* Projected cumulative max.demand */
                /* Percentage scaled by 10. */
  end
IF (S994IOPLTON > 0) Then do
  do pw = 0 to S994IOPLTON -1
    pwo = (S994IOPLTOF + (pw*S994IOPLTLN))- 3
    /*-----*/
    /* SMF99 subtype 4 I/O plot information - Contains */
    /* the information plotted on the device clusters I/O plot */
    /* Note: The description for I/O curve is missing in SMF book -*/
    /* assumed to be same as description for subtypes 1,3, and 5 */
    /*-----*/
    IO_PLOT_BW =c2d(SUBSTR(x.i,pwo,4)) /* Bucket width */
    IO_PLOT_LSTX =c2d(SUBSTR(x.i,pwo+4,4))
                /* Last plotted x bucket */
    IO_PLOT_POINTS_OF=c2d(SUBSTR(x.i,pwo+12,4))
                /* Offset of point entries */
    IO_PLOT_POINTS_ON=c2d(SUBSTR(x.i,pwo+16,2))
                /* Number of point entries */
    IO_PLOT_POINTS_LN=c2d(SUBSTR(x.i,pwo+18,2))
                /* Length of a point entry */
  Select
  when IO_PLOT_POINTS_ON > 0 then call XYPROC ,
  IO_PLOT_POINTS_OF IO_PLOT_POINTS_LN IO_PLOT_POINTS_ON IO_PLOT_LSTX

```

```

otherwise nop
End
iodrec.pz= left('Clust. ',6) left(S994DEVCLID,3) left('I/O',3),
           left('Max I/O demand:',18) left(xar,400); pz = pz + 1
iodrec.pz= left('Prty: ',6) left(S99_IPTPTY,7) ,
           left('Delay/Using ratio:',18) left(yar,400); pz = pz + 1
"EXECIO * DISKW s99p (STEM iodrec.)"
end /* end of I/O plot loop */
end
Return
SUBTYPE5:
/*-----*/
/* SMF99 subtype 5: Monitored AS information/main loop - start */
/*-----*/
parse arg dof
DOF = dof -3
S995ANAM = SUBSTR(x.i,dof,8) /* Jobname */
S995ACNM = SUBSTR(x.i,dof+8,8) /* Class that space is in */
S995APNUM = c2d(SUBSTR(x.i,dof+16,4)) /* Period number space is in */
S995APCS = c2d(SUBSTR(x.i,dof+20,4)) /*Protective central stor tar*/
S995ARCS = c2d(SUBSTR(x.i,dof+24,4)) /* Restrictive central stor tar*/
S995APPS = c2d(SUBSTR(x.i,dof+28,4)) /* Protective proc stor tar */
S995ARPS = c2d(SUBSTR(x.i,dof+32,4)) /* Restrictive proc stor tar */
/*-----*/
/* SMF99 subtype 5 CPLT plot data - Contains information */
/* plotted on the Central storage paging plot for */
/* the monitored address space in this record */
/*-----*/
S995_CPLT_BW = c2d(SUBSTR(x.i,dof+36,4)) /* Bucket width */
S995_CPLT_LSTX = c2d(SUBSTR(x.i,dof+40,4)) /*Last plotted x bucket */
S995_CPLT_POINTS_OF=c2d(SUBSTR(x.i,dof+48,4))
/* Offset of point entries*/
S995_CPLT_POINTS_ON=c2d(SUBSTR(x.i,dof+52,2))
/* Number of point entries*/
S995_CPLT_POINTS_LN=c2d(SUBSTR(x.i,dof+54,2))
/* Length of a point entry*/
/*-----*/
/* SMF99 subtype 5 PPLT plot data - Contains information */
/* plotted on the Processor storage paging plot for */
/* the monitored address space in this record */
/*-----*/
S995_PPLT_BW = c2d(SUBSTR(x.i,dof+56,4)) /* Bucket width */
S995_PPLT_LSTX = c2d(SUBSTR(x.i,dof+60,4))
/* Last plotted x bucket */
S995_PPLT_POINTS_OF=c2d(SUBSTR(x.i,dof+68,4))
/* Offset of point entries*/
S995_PPLT_POINTS_ON=c2d(SUBSTR(x.i,dof+72,2))
/* Number of point entries*/
S995_PPLT_POINTS_LN=c2d(SUBSTR(x.i,dof+74,2))
/* Length of a point entry*/

```



```

S995ASID = c2d(SUBSTR(x.i,dof+76,2))
S995EXT_CLASS_NAME=SUBSTR(x.i,dof+78,8)
S995_FLAGS=x2b(c2x(SUBSTR(x.i,dof+86,1)))
  z1 = substr(S995_FLAGS,1,1)
  z2 = substr(S995_FLAGS,2,1)
  z3 = substr(S995_FLAGS,3,1)
  z4 = substr(S995_FLAGS,4,1)
  z5 = substr(S995_FLAGS,5,1)
  z6 = substr(S995_FLAGS,6,1)
  Select
  when z1 =1 then s1='Stg protected now'
  otherwise      s1=' '
  End
  Select
  when z2 =1 then s2='STGcrit_explicit'
  otherwise      s2=' '
  End
  Select
  when z3 =1 then s3='Ctar set by stcr_hsk'
  otherwise      s3=' '
  End
  Select
  when z4 =1 then s4='Ptar set by stcr_hsk'
  otherwise      s4=' '
  End
  Select
  when z5 =1 then s5='Ctar set by_PA '
  otherwise      s5=' '
  End
  Select
  when z6 =1 then s6='Ptar set by Pa '
  otherwise      s6=' '
  End
flagST=s1||s2||s3||s4||s5||s6
  Select
  when S995_CPLT_POINTS_ON > 0 then call CURVT ,
  S995_CPLT_POINTS_OF S995_CPLT_POINTS_LN S995_CPLT_POINTS_ON ,
  S995_CPLT_LSTX
  otherwise nop
  End
  cstrec.1 = left(S995ACNM,8) left(S995APNUM,1) left('CSP',3),
             left('ASID size / frames:',18) left(x1wrw,400)
  cstrec.2 = left(S995ANAM,14),
             left('Page in rate:',18) left(y1wrw,400)

```

```

cstrec.3 = left('Asid #:',7)                left(S995ASID,6),
          left('Page in cost:',18)          left(y2wrw,400)
cstrec.4 = left(' ',14,' ') left('Captured ms/sec:',18) left(y3wrw,400)
          "EXECIO * DISKW s99p (STEM cstrec.)"
/* Select
  when S995_PPLT_POINTS_ON > 0 then call CURVT ,
S995_PPLT_POINTS_OF S995_PPLT_POINTS_LN S995_PPLT_POINTS_ON ,
S995_PPLT_LSTX
  otherwise nop
  End */
Return
SUBTYPE9:
parse arg dof
KOF = dof -3
/*-----*/
/* SMF99 subtype 9 self defining section */
/*-----*/
S9_IOSUBS_OFF= c2d(SUBSTR(x.i,kof,4)) /*Offset to IO Subsystem sec.*/
S9_IOSUBS_LEN= c2d(SUBSTR(x.i,kof+4,2))
/*Length of IO Subsystem sec.*/
S9_IOSUBS_NUM= c2d(SUBSTR(x.i,kof+6,2)) /*Number of IO Subsystem */
S9_IOSUB_PLOF= c2d(SUBSTR(x.i,kof+8,4)) /*Offset to IO Subsys.plots */
S9_IOSUB_PLLN= c2d(SUBSTR(x.i,kof+12,2)) /* Length of IO Subsys.plots*/
S9_IOSUB_PLON= c2d(SUBSTR(x.i,kof+14,2)) /* Number of IO Subsys.plots*/
S9_CHANNEL_OF= c2d(SUBSTR(x.i,kof+16,4)) /* Offset to Channel section*/
S9_CHANNEL_LN= c2d(SUBSTR(x.i,kof+20,2)) /* Length of Channel section*/
S9_CHANNEL_ON= c2d(SUBSTR(x.i,kof+22,2)) /* Number of Channel section*/
/*-----*/
/* SMF99 subtype 9 Chpid data entry - Represents I/O subsystem */
/* information. */
/*-----*/
IF (S9_IOSUBS_NUM > 0) Then do
  do io = 0 to S9_IOSUBS_NUM-1
    ioof = (S9_IOSUBS_OFF + (io*S9_IOSUBS_LEN))- 3
S9_FLAG1=x2b(c2x(SUBSTR(x.i,ioof,1))) /* IO subsystem flags */
iosubs1 = substr(S9_FLAG1,1,1) /* IO subsystem is */
Select /* eligible for */
  when iosubs1 = 1 then dyn='Dynamic' /* dynamic mngmnt. */
  otherwise dyn='Not Dyn'
End
iosubs2 = substr(S9_FLAG1,2,1)
Select
  when iosubs2 = 1 then Flo='Float capable' /* IO subsys.is float */
  otherwise Flo='No Float capable' /* capable */
End
iosubs3 = substr(S9_FLAG1,3,1)
Select
  when iosubs3 = 1 then Cnf='Config change' /* CONFIG change has */
  otherwise Cnf='No Config chg.' /* been made. */
End

```

```

        iosubs4 = substr(S9_FLAG1,4,1)          /*No point was plotted */
Select                                          /*this interval. If = 1*/
        when iosubs4 = 1 then Plt='No plot point' /
*S99IOSUB_INTERVAL_VELOC*/
        otherwise          Plt='Point plotted.' /*is not relevant */
End
        iosubs5 = substr(S9_FLAG1,5,1)          /*No channel data was */
Select                                          /*written this interval.*/
        when iosubs5 = 1 then Chd='No ch. data' /*Set on when the structu*/
        otherwise          Chd='Ch. data  '    /*for the channel data */
End                                              /*does not exit      */
        iosubs6 = substr(S9_FLAG1,6,1)
Select                                          /*While calculating the */
        when iosubs6 = 1 then Rtc='Ratio converted' /*busy to connect ratio*/
        otherwise          Rtc='Ratio no conv. ' /*an entry ratio was */
End                                              /*converted from >1.6 to */
                                                /*1.3                    */
S9_CONTROL_FLAGS =x2b(c2x(SUBSTR(x.i,i0of+1,1))) /* Control flags */

        contflg1= substr(S9_CONTROL_FLAGS,1,1) /* Indicates that this SMF */
Select                                          /* 99/9 record represents */
        when contflg1= 1 then Cor='Not correct 99.9 '
                                                /* situation that caused */
        otherwise          Cor=' '             /* IBM not to create valid */
End                                              /* SMF 99/9 data. Identify */
                                                /* condition by checking */
                                                /* the following indicators.*/
                                                /* Data in this SMF 99/9 */
                                                /* is invalid other than */
                                                /* possibly the S9_IOSUB_INX*/
                                                /* (valid when empty slot */
                                                /* or old slot entry) */

        contflg2= substr(S9_CONTROL_FLAGS,2,1)
Select                                          /* Indicates that registry */
        when contflg2= 1 then Reg='No registry data '
        otherwise          Reg=' '             /* data does not exist */
End

        contflg3= substr(S9_CONTROL_FLAGS,3,1)
Select                                          /* WLM LPAR Cluster struct. */
        when contflg3= 1 then Con='ClustStr connc.' /* is not connected */
        otherwise          Con=' '
End

        contflg4= substr(S9_CONTROL_FLAGS,4,1) /* SMF 99 buffer for the */
Select                                          /* subtype 9 data was not*/
        when contflg4= 1 then Buf='No buff.process ' /* processed from the */
        otherwise          Buf=' '             /* previous interval */
End

        contflg5= substr(S9_CONTROL_FLAGS,5,1) /* Indicates empty */
Select                                          /* slot entry */
        when contflg5= 1 then Emp='Empty slot '

```

```

        otherwise          Emp=' '
End
    contflg6= substr(S9_CONTROL_FLAGS,6,1)      /* Indicates old      */
Select                                          /* slot entry        */
    when contflg6= 1 then Old='Old slot '
    otherwise          Old=' '
End
    contflg7= substr(S9_CONTROL_FLAGS,7,1)      /* Unknown reason    */
Select
    when contflg7= 1 then Unk='Unknown reason '
    otherwise          Unk=' '
End
err= Cor||Reg||Buf||Emp||Old||Unk
/*-----*/
/* Identify the reason that caused the error while creating 99.9 */
/*-----*/
Select
when S9_IOSUB_PLON = 0 then do
perror =err
ploter.1 = left('I/O Velocity Plot error occured',40)
ploter.2 = left('Plot error reason:',19) left(perror,60)
ploter.3 = left(' ',1)
"EXECIO * DISKW s99p (STEM ploter.)"
end
otherwise          nop
End
S9_ROW_INDEX = c2d(SUBSTR(x.i,iopf+2,2)) /* Ch.path data row index */
S9_DIAG_TOKEN = c2d(SUBSTR(x.i,iopf+5,3)) /* Internal diagnose data */
S9_TOKEN_NED = c2d(SUBSTR(x.i,iopf+8,32)) /* Token NED of I/O Subsys.*/
S9_IOSUB_TARGET_VELOCITY = , /* I/O subsystem target*/
                c2d(SUBSTR(x.i,iopf+40,4)) /* velocity set by WLM */
S9_IOSUB_VELOCITY = , /* The actual I/O velocity */
                c2d(SUBSTR(x.i,iopf+44,4)) /* of the subsystem */
S9_IOSUB_AVG_SVC_TIME=c2d(SUBSTR(x.i,iopf+48,4))
                                                /* Average service time */
S9_IOSUB_CHECK_POINT_TIME = , /* Average service time*/
                c2d(SUBSTR(x.i,iopf+52,4)) /*of the subsystem when*/
                                                /* SRM set a target */
S9_IOSUB_INDEX= c2d(SUBSTR(x.i,iopf+56,2)) /* IO Subsystem Index */
S9_LCU_SQNUMBER=c2d(SUBSTR(x.i,iopf+58,2)) /* LCU sequence number */
S9_CONTROL_UNITS =c2d(SUBSTR(x.i,iopf+60,16))
                                                /* Control unit numbers(1:8)*/
                                                /* associated with LCU */
S9_IOSUB_PROJECTED_VELOCITY = , /* Projected velocity */
                c2d(SUBSTR(x.i,iopf+76,4)) /* to be expected if a */
                                                /* change is to be made */
S9_IOSUB_INTERVAL_VELOCITY = , /* The I/O velocity that is */
                c2d(SUBSTR(x.i,iopf+80,4)) /* used to plot a point. */
S9_AVG_BUSY_TO_CONNECT_RATIO = , /* The average busy to */
                c2d(SUBSTR(x.i,iopf+84,4)) /* connect time ratio */

```

```

/* calculated by WLM */
/* during the copy interval */
S9_CLEAR_INTERVAL=c2d(SUBSTR(x.i,iroof+88,2)) /* Clear interval index*/
S9_TIMESTAMP_SYSTEM= SUBSTR(x.i,iroof+96,8) /* Identifies the system*/
/* that made a CONFIG change*/

End
End
/*-----*/
/* SMF99 subtype 9 I/O subsystem Plot section. */
/*-----*/
IF (S9_IOSUB_PLON > 0) Then do
do vo = 0 to S9_IOSUB_PLON-1
vof = (S9_IOSUB_PLOF + (vo*S9_IOSUB_PLLN))- 3
SMF99_IOSUB_BW =c2d(SUBSTR(x.i,vof,4)) /* Bucket width */
SMF99_IOSUB_LSTX =c2d(SUBSTR(x.i,vof+4,4))
/* Last plotted x bucket */
SMF99_IOSUB_POINTS_OF=c2d(SUBSTR(x.i,vof+12,4))
/* Offset of point entries */
SMF99_IOSUB_POINTS_ON=c2d(SUBSTR(x.i,vof+16,2))
/* Number of point entries */
SMF99_IOSUB_POINTS_LN=c2d(SUBSTR(x.i,vof+18,2))
/* Length of a point entry */

Select
when SMF99_IOSUB_POINTS_ON 0 then call XYPROC ,
SMF99_IOSUB_POINTS_OF SMF99_IOSUB_POINTS_LN SMF99_IOSUB_POINTS_LN ,
SMF99_IOSUB_LSTX
otherwise nop
End
iovrec.1 = left('I/O Velocity Plot',29)
iovrec.2 = left('Contention factor:',18) left(xar,400)
iovrec.3 = left('I/O Velocity:',18) left(yar,400)
iovrec.4 = left(' ',1)
"EXECIO * DISKW s99p (STEM iovrec.)"
End
/*-----*/
/* SMF99 subtype 9 Chpid data entry - Represents */
/* Channel Data information. */
/*-----*/
IF (S9_CHANNEL_ON > 0) Then do
do cv = 0 to S9_CHANNEL_ON-1
cvf = (S9_CHANNEL_OF + (cv*S9_CHANNEL_LN))- 3
S9_CHANNEL_ID =c2d(SUBSTR(x.i,cvf,1)) /*IO Subsystem ch. ID */
S9_CHANNEL_TYPE =c2d(SUBSTR(x.i,cvf+1,1)) /*IO Subsystem ch.type*/
S9_CHANNEL_FLAG =x2b(c2x(SUBSTR(x.i,cvf+2,1)))
/*IO Subsystem ch.flags */

Chwlm = substr(S9_CHANNEL_FLAG,1,1)
Select
when Chwlm = 1 then CWM='Managed by WLM ' /*Indicates - this ch. */
otherwise CWM='Not WLM managed' /*can be managed by WLM*/
End

```

```

S9_CHANNEL_UTILIZATION = , /* IO Subsystem Ch. */
      c2d(SUBSTR(x.i,cvf+4,4)) /* utilization for curr.*/
S9_CHANNEL_PROJECTED_UTILIZATION = , /* data collect interval*/
      c2d(SUBSTR(x.i,cvf+8,1)) /* Projected IO Subsys. */
S9_CHANNEL_PROJ_CURRENT_PATH_LOAD = , /* Ch.util. for the */
      c2d(SUBSTR(x.i,cvf+9,1)) /* curr. data collection*/
S9_CHANNEL_#SYSTEMS_CONTRIBUTED = , /* interval (percentage)*/
      c2d(SUBSTR(x.i,cvf+10,2)) /* Percent use for path */
S9_CHANNEL_PORT_BUSY_COUNT = , /* projected for the */
      c2d(SUBSTR(x.i,cvf+12,4)) /* current connfig */
End /* during the */
End /* calibration pass. */
Return /* Count of systems */
XYPROC: /* contributed to this */
/*-----*/ /* channel data */
/* SMF99 subtype 1, 3, and 5 - map to format points on a */ /* Ch. port busy count */
/* plot with one curve. */ /*-----*/
parse arg off1 len1 num1 last
xar = left(' ',2)
yar = left(' ',2)
g =1
if (len1 <> 0) Then do
do aa = 0 to num1 -1
ofx = (off1 + (aa*len1))- 3
Select
when (aa+1) = last then xarr =left(' ',2)
otherwise xarr =left(' ',2)
End
yarr =left(' ',2)
XVAL.g =right(c2d(SUBSTR(x.i,ofx,4)),5) /*X value of point plotted */
YVAL.g =right(c2d(SUBSTR(x.i,ofx+4,4)),5) /*Y value of point plotted */
xar =xar||xarr||xval.g
yar =yar||yarr||yval.g
g = g + 1
end
Return xar yar
TWOC:
/*-----*/
/* SMF99 subtype 3 - map to format points on a plot with 2 curves*/
/*-----*/
parse arg off2 len2 num2

```

```

x1arw = left(' ',2)
y1arw = left(' ',2)
y2arw = left(' ',2)
h =1
if (len2 <> 0) Then do
  do bb = 0 to num2 -1
    bff = (off2 + (bb*len2))- 3
  x1ark =left(' ',2)
  y1ark =left(' ',2)
  y2ark =left(' ',2)
  XVAL2.h= right(c2d(SUBSTR(x.i,ofx,4)),4) /*X value of point plotted*/
  Y1VAL.h= right(c2d(SUBSTR(x.i,ofx+4,4)),4) /*Y value of point plotted
                                         on first curve */
  Y2VAL.h= right(c2d(SUBSTR(x.i,ofx+8,4)),4) /* Y value of point plotted
                                         on second curve */

  x1arw =x1arw||x1ark||xval2.h
  y1arw =y1arw||y1ark||y1val.h
  y2arw =y2arw||y2ark||y2val.h
  h = h + 1
end
Return x1arw y1arw y2arw
CURVT:
/*-----*/
/* SMF99 subtype 5 - map to format points on a plot with 3 curves*/
/*-----*/
parse arg off3 len3 num3 la3
x1wrw = left(' ',2)
y1wrw = left(' ',2)
y2wrw = left(' ',2)
y3wrw = left(' ',2)
q =1
if (len3 <> 0) Then do
  do qq = 0 to num3 -1
    qqf = (off3 + (qq*len3))- 3
  Select
  when (qq+1) = la3 then x1wrk =left(' ',2)
  otherwise           x1wrk =left(' ',2)
End
y1wrk =left(' ',2)
y2wrk =left(' ',2)
y3wrk =left(' ',2)
XVA.q = right(c2d(SUBSTR(x.i,qqf,4)),5) /*X value of point plotted */
Y1VA.q = right(c2d(SUBSTR(x.i,qqf+4,4)),5) /*Y value of point plotted
                                         on 1st curve */
Y2VA.q = right(c2d(SUBSTR(x.i,qqf+8,4)),5) /*Y value of point plotted
                                         on 2nd curve */
Y3VA.q = right(c2d(SUBSTR(x.i,qqf+12,4)),5) /*Y value of point
                                         plotted on 3rd curve */

x1wrw =x1wrw||x1wrk||xva.q
y1wrw =y1wrw||y1wrk||y1va.q

```

```

y2wrw =y2wrw||y2wrk||y2va.q
y3wrw =y3wrw||y3wrk||y3va.q
q = q + 1
end
Return x1wrw y1wrw y2wrw y3wrw
SMF:
/*-----*/
/* REXX - convert SMF time to hh:mm:ss:hd format */
/*-----*/
arg time
time1    = time % 100
hh       = time1 % 3600
hh       = RIGHT("0"||hh,2)
mm       = (time1 % 60) - (hh * 60)
mm       = RIGHT("0"||mm,2)
ss       = time1 - (hh * 3600) - (mm * 60)
ss       = RIGHT("0"||ss,2)
fr       = time // 1000
fr       = RIGHT("0"||fr,2)
rtime   = hh||": "||mm||": "||ss||": "||fr

```

Mile Pekic

Systems Programmer (Serbia and Montenegro)

© Xephon 2005

Why not share your expertise and earn money at the same time? *MVS Update* is looking for program code, REXX EXECs, JavaScript, etc, that experienced users of z/OS and OS/390 have written to make their life, or the lives of their users, easier. We are also looking for explanatory articles, and hints and tips, from experienced users. We would also like suggestions on how to improve MVS performance.

We will publish your article (after vetting by our expert panel) and send you a cheque, as payment, and two copies of the issue containing the article once it has been published. Articles can be of any length and should be e-mailed to the editor, Trevor Eddolls, at trevore@xephon.com.

A free copy of our Notes for Contributors, which includes information about payment rates, is available from our Web site at www.xephon.com/nfc.

The product, a platform and application-UC4 has announced a new version of its enterprise-wide batch processing and job scheduling solution, UC4:global 05.

independent job-scheduling tool, provides migration through external job plan dependencies. The new version includes an open interface to output management systems. UC4:global 05 also provides an audit trail and reporting to address compliance with Sarbanes-Oxley and other regulations. In terms of version control, the new release extends that functionality to track exactly what was changed by whom and when.

UC4:global 05 includes UC4.Executor for JMX, an agent which monitors and actively controls Java environments.

For further information contact:

URL: www.uc4.com/press_release_051005.htm.

* * *

Cressida Technology has announced that ReQuest for WebSphere MQ now runs on z/OS. ReQuest currently also supports Windows, AIX, Solaris, HP-UX, and SUSE Linux platforms.

Request for WebSphere MQ is a message tracking, message reporting, message replay, point-in-time message recovery, charge-back, accounting, and auditing solution. ReQuest uses filtering technology to analyse critical message activity information already contained in WMQ logs.

For further information contact:

URL: www.cressida.info/products_cressida_ReQuest.htm.

* * *

Innovation Data Processing has announced FDRCRYPT, an encryption back-up utility specifically designed for z/OS. It will transparently employ the new CP Assist for Cryptographic Function (CPACF) Advanced Encryption Standard (AES) hardware feature for both data encryption and decryption on all z9-109 models.

FDRCRYPT is an optional add-on to the FDR back-up and recovery suite. It can be used to encrypt back-up data being sent off-site (such as disaster recovery tape back-ups) against unauthorized access. It can also be used to encrypt all back-up data.

For further information contact:

URL: www.innovationdp.fdr.com/products/fdrcrypt/index.cfm.

* * *

OpenTech Systems has announced CopyCrypt, a new tape data encryption utility. The product protects tape data, in compliance with government mandates such as HIPPA, Sarbanes-Oxley, and the GLBA, by encrypting data during the tape copy operation to ensure that the data cannot be read by an unauthorized source.

CopyCrypt exploits the cryptographic hardware standard on IBM mainframes by interfacing with IBM's Integrated Cryptographic Services Facility to request encryption services and manage encryption keys. The product supports all standard encryption algorithms, such as DES, 3DES, and AES.

For further information contact:

URL: www.opentechsystems.com/pr/OT-CC-PR101005.pdf.

