November 1997

In this issue

3 31-bit I/O in Assembler
6 Year 2000 aid: source scan program
40 DASD space monitoring
43 Using a load library for SCLM – controlled projects
49 Generating structured Assembler programs with ISPF edit macros – part 2
64 Useful Assembler macros – part 2
72 MVS news

© Xephon plc 1997
Contributions
If you have anything original to say about MVS, or any interesting experience to recount, why not spend an hour or two putting it on paper? The article need not be very long -- two or three paragraphs could be sufficient. Not only will you be actively helping the free exchange of information, which benefits all MVS users, but you will also gain professional recognition for your expertise, and the expertise of your colleagues, as well as some material reward in the form of a publication fee -- we pay at the rate of £170 ($250) per 1000 words for all original material published in MVS Update. If you would like to know a bit more before starting on an article, write to us at one of the above addresses, and we'll send you full details, without any obligation on your part.

© Xphon plc 1997. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xphon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (16 pence). To order, contact Xphon at any of the addresses above.

Printed in England.
31-bit I/O in Assembler

INTRODUCTION
With a few exceptions Assembler programs that perform I/O are no longer restricted to operating in 24-bit mode. VSAM, of course, was not subject to that restriction. Although, today, Assembler is rarely used directly for commercial programming, many installations have their own data access methods usually written in Assembler. Such routines are ideal candidates for being made address-mode independent to relieve storage constraints.

GENERAL CONSIDERATIONS
There are a few restrictions that must be taken into consideration if AMODE=31 is used for I/O processing:

• The DCB must be located below the 16 MB line. To do this, either specify RMODE=24 for the program that contains the DCB (normally not a particularly attractive requirement, because one of the aims of using 31-bit addressing is to increase the amount of main-storage available) or dynamically allocate the DCB in the 24-bit area using the STORAGE or GETMAIN system service.

• The MODE=31 keyword must be specified on the OPEN and CLOSE macros.

• The DCBE (DCB-extension) address for BSAM/BPAM must be located below the 16 MB line (ie in 24-bit address space).

• A DCBE must be defined if 31-bit EODAD or SYNAD addresses are used.

• A DCBE must be defined if the data buffers are to be assigned above the 16 MB line (ie in 31-bit address space – RMODE31=BUFF).

Note: Although the DCBE provides further options, these are infrequently used.
The following (infrequently used) I/O macros cannot be used in 31-bit addressing mode:

- ESETL
- PDAB
- PDABD
- PROTOV
- SETL.

The following I/O macros have limited 31-bit addressing capability; the macro must be located below the 16 MB line, although the program may run in AMODE=31:

- READ
- WRITE
- SETPRT.

RDJFCB cannot run in AMODE=31, because the OPEN ...TYPE=J,MODE=31 combination is not permitted.

**SUMMARY**

Programs that use QSAM require little effort to change them to make the program fully 31-bit capable. Programs that use BSAM require more effort.

**QSAM EXAMPLE**

```
QSAM31A CSECT
QSAM31A AMODE 31
QSAM31A RMODE ANY
... set-up base register, etc.
* Dynamically allocate the DCB in 24-bit address space
  STORAGE OBTAIN,LENGTH=DCBL,LOC=BETWEEN
  LR R2,R1   R1: address of the allocated DCB
  MVC Ø(DCBL,R2),FILEDCB    initialize DCB
...  
  OPEN ((R2),(INPUT)),MODE=31
...  
* read loop
```
GET (R2)
* R1: address of the record in the buffer

FILEEND DS $0H end of file

CLOSE ((R2)),MODE=31
STORAGE RELEASE,LENGTH=DCBL,AREA=(R2) release DCB

FILEDCB DCB DDNAME=DD1,DSORG=PS,MACRF=GL,DEVDA=DA,DCBE=XDCBE
DCBL EQU *-FILEDCB DCB length
XDCBE DCBE EODAD=FILEEND,RMODE31=BUFF
END

This sample program shows the maximum gain in 24-bit storage availability.

BSAM EXAMPLE
In comparison with QSAM, BSAM (and BPAM) also require that a DCBE is defined below the 16 MB line.

BSAM31A CSECT
BSAM31A AMODE 31
BSAM31A RMODE ANY
... set-up base register, etc.
* Allocate DCB + DECB (READ macro) below the 16 MB line
STORAGE OBTAIN,LENGTH=INL,LOC=BElOW
LR R2,R1 DCB address
MVC 0(DCBL,R2),FILEDCB initialize the allocated DCB
LA R3,DCBL(R2) address of the READ macro
MVC 0(READL,R3),READ initialize the allocated READ macro

... OPEN ((R2),INPUT),MODE=31

... * read loop
READ (R3),SF,(R2),BUF,MF=E
CHECK (R3)

... FILEEND DS $0H end of file

... CLOSE ((R2)),MODE=31
STORAGE RELEASE,LENGTH=INL,AREA=(R2) release allocated storage

FILEDCB DCB DDNAME=DD2,DEVDA=DA,DSORG=PS,MACRF=R,DCBE=XDCBE
DCB align DCB on word boundary
DCBL EQU *-FILEDCB DCB length
READ READ DECB,DF,0,0,'S',MF=L
READL EQU *-READ DECB length (READ macro)
INL EQU *-FILEDCB total length of DCB + DECB

INTRODUCTION
The program below, YEAR2K, searches partitioned datasets (PDSs) for strings of text. When a specified string is found, the specific string, record, and member are flagged. This is used to:

- Create analysis summaries for both the member and dataset.
- Include the member in any generated JCL that may be used create a maintenance PDS for further conversion consideration.
- Select the record as a sequential dataset so that it may be viewed with ISPF facilities or listed with a subsequent program, YEAR2KL, for analysis of priority and personnel assignment.

SEARCH STRING SPECIFICATIONS
Search strings are defined by the labels WORDLIST through to LASTWORD. The definition is by macro STDEF. This macro is defined within the program source and may contain from one to four operands, as follows:

- The character string. This character string may contain any EBCDIC characters. If embedded blanks, commas, or single quotation marks are included the string must be enclosed in single quote marks. If embedded quote marks or ampersands are desired, each occurrence must be specified as two consecutive specifications of that character (ie " or && to specify ' or &, respectively).
The remaining operands, if present, indicate that the search is qualified to specific segment(s) of the specified string. These operands consist of the single characters W, P, and/or S to denote qualifications of WORD, PREFIX, and/or SUFFIX respectively. These qualifiers have the same meaning as those used in ISPF search and replace commands. For example, if word and prefix are specified for the string ‘DATE’, the strings DATE and DATE2 will be selected, but UPDATE will not be selected. If all three qualifiers are specified for string ‘MM’; MM, MMDDYY, and YYMM will qualify while SUMMARY will not qualify.

Sample character string definitions are shown below:

```
WORDLIST STDEF AGE.W,P
STDEF BIRTH.W,P
STDEF CALENDAR
STDEF CENTURY
STDEF CSADAT
... ...
STDEF YM,P,S,W
STDEF YMD
STDEF YY
LASTWORD DC X'FF'
```

MEMBER SELECTION

Members of the PDS may be limited in two ways:

- FROM=member1 and THRU=member2 PARM fields. These specifications limit member names to those from member1 through to member2, whose respective default values are the first and last member of the PDS. For example, PARM='FROM=C,THRU=M' would restrict analysis to members beginning with characters C through to L and the member M.

- Use of the exclusion dataset (CARDS). Records from this sequential dataset are read and the information from bytes 1-8 is extracted and sorted. Member names that match any of these selections are excluded from analysis. If bytes 2-8 contain an asterisk, all members whose names match the previous characters are excluded. For example, the entries MEMBERX and NAME* would exclude the members MEMBERX and all members whose first four characters are ‘NAME’.
JCL TO COPY SELECTED MEMBERS

Whenever a string is found, an IEBCOPY statement is written to the sequential output dataset OUTJCL to copy the member to a maintenance PDS. This dataset is later edited to remove the COPY statements for members that are to be manually excluded and to customize JOB and target DD statements. A sample of this data can be seen in Figure 1.

```jcl
//COPY2KJR JOB , 'YEAR 2000 ANALYST',.... <--- CUSTOMIZE
//COPYSTEP EXEC PGM=IEBCOPY
//INPUT DD DISP=SHR,DSN=OPER.ONLINE.SOURCLIB
//OUTPUT DD DISP=SHR,DSN=OBJECT.PDS.NAME <--- CUSTOMIZE
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
COPY OUTDD=OUTPUT,INDD=INPUT
SELECT MEMBER=AAG10010
... ... ...
SELECT MEMBER=XMIT9MP 
SELECT MEMBER=XMIT9MP
/*
```

**Figure 1:** Sample IEBCOPY JCL produced by YEAR2K

```
Browse SYST002.YEAR2K.MATCHES
Command --> PAGE
080900 MOVE 'BAD' TO BABY-AGE
081000 WHEN BIRTH-MONTH = T-MONTH (INDX)
081100 COMPUTE NEWBORN-AGE = T-DAYS (INDX) - BIRTH-DAYS 
081200 ADD SYS-DAYS TO NEWBORN-AGE
082100 MOVE DAYS-OF-YEAR (INDX) TO YEAR-DAYS.
082200 ADD H-DAYS TO YEAR-DAYS.
082300 DIVIDE H-YEAR BY 4 GIVING IS-IT-LEAP ROUNDED.
082600 ADD 1 TO YEAR-DAYS.
77 YY1 PIC 99 VALUE ZERO.
03 YY PIC 99.
MOVE CURRENT-DATE TO WSDATE.
PERFORM CK-I-DATE THRU CD-EXIT
PERFORM CK-I-DATE THRU CD-EXIT
PERFORM CK-I-DATE THRU CD-EXIT.
PERFORM CK-I-DATE THRU CD-EXIT
CK-I-DATE.
COMPUTE YY1 = YY OF WKDATE - 1.
(CKYR NOT > YY OF WKDATE)
```

**Figure 2:** Sample browse of OUTPUT file
SELECTED OUTPUT RECORDS

Each record that contains a specified string is written to a sequential file (OUTPUT). To facilitate viewing this data on an 80 character screen line, the first 72 characters of the record are written, followed by the PDS member name, the remainder of the record, and the record count within the PDS member. Also to facilitate viewing, a record of hyphens is placed at the end of each selected PDS member record. A program (YEAR2KL) provides a listing of these individual records for further analysis. An ISPF browse of a sample output file is provided in Figure 2.

SAMPLE OUTPUT LISTINGS

A summary is provided for each PDS member. If the member is excluded, a notation is made. If not excluded, a line is printed that provides the number of records analysed and the number of records where specified strings were found. If available, the ISPF statistics are also printed. These statistics are preceded by the DASD address (TTR) of the PDS member. For each of the specified strings that were found within the member occurrence, counts are provided for the type of occurrence (embedded, word, prefix, or suffix). See Figure 3.

A summary of the entire PDS appears as the final page of the report. This summary includes the total number of members and records that were analysed and the number of selections. A summary of each of the specified strings is provided in Figure 4.

PROGRAM SOURCE

GBLA &N.&IMBED.&OTHER.&WORD.&PREFIX.&SUFFIX
   LCLC &MYNAME
*
   &MYNAME SETC 'YEAR2K'                        CSECT NAME
   RBASE EQU 12                                BASE REGISTER FOR CSECT
   RBAL EQU 10                                 BAL REGISTER
*
   TITLE '&MYNAME'                         LISTING TITLE
*

YEAR2K ANALYSIS REPORT

RECORDS SELECTED TO DSN=SYST002.YEAR2K.MATCHES
IEBCOPY JCL TO DSN=SYST002.YEAR2K.IEBCOPY

MANUALLY EXCLUDED MEMBERS:

L80*

1 AAGI0005 CONTAINS 161 RECORDS OF WHICH 0 CONTAIN OCCURRENCES OF SPECIFIED STRINGS
2 AAGI0010 CONTAINS 2,470 RECORDS OF WHICH 43 CONTAIN OCCURRENCES OF SPECIFIED STRINGS

ISPF STATS: 01612 1.00 07/29/96 07/29/96 15:58 2470 2470 0 APPL001

IMBEDDED WORDS PREFIX SUFFIX STRING
1 0 0 0 CALENDAR
0 2 0 0 DATE
1 0 0 0 GREGJUL
1 0 0 0 GREGORIAN
2 0 0 0 JULGREG
25 0 0 0 JULIAN
7 0 0 0 MMDYY
4 0 0 0 YEAR
0 0 0 1 YM
23 0 0 0 YY
64 2 0 1 * TOTAL *

3 AAGI0015 CONTAINS 888 RECORDS OF WHICH 1 CONTAIN OCCURRENCES OF SPECIFIED STRINGS

IMBEDDED WORDS PREFIX SUFFIX STRING
0 0 0 1 YM
0 0 0 1 * TOTAL *

4 AAGMAPA CONTAINS 40 RECORDS OF WHICH 0 CONTAIN OCCURRENCES OF SPECIFIED STRINGS
5 AAGMAP1 CONTAINS 239 RECORDS OF WHICH 0 CONTAIN OCCURRENCES OF SPECIFIED STRINGS
6 AAGMAP2 CONTAINS 149 RECORDS OF WHICH 0 CONTAIN OCCURRENCES OF SPECIFIED STRINGS
7 AGECALC CONTAINS 221 RECORDS OF WHICH 36 CONTAIN OCCURRENCES OF SPECIFIED STRINGS

ISPF STATS: 01004 1.03 05/22/96 05/22/96 16:10 221 223 768 APPL012
YEAR2K ANALYSIS REPORT
JOB=SYST002I  DSN=OPER.ONLINE.SOURCLIB

643 MEMBERS FOUND
16 MEMBERS EXCLUDED
627 MEMBERS ANALYZED
319 MEMBERS SELECTED

302,369 RECORDS ANALYZED
2,662 RECORDS SELECTED

ANALYSIS SUMMARY:

<table>
<thead>
<tr>
<th>IMBEDDED</th>
<th>WORDS</th>
<th>PREFIX</th>
<th>SUFFIX</th>
<th>STRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>AGE</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>BIRTH</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CALENDAR</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CENTURY</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CSAJYD</td>
</tr>
<tr>
<td>0</td>
<td>203</td>
<td>49</td>
<td>0</td>
<td>DATE</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>DMY</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>GREGJUL</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>GREGORIAN</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>JULGREG</td>
</tr>
<tr>
<td>406</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>JULIAN</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MDY</td>
</tr>
<tr>
<td>164</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MMDDYY</td>
</tr>
<tr>
<td>383</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>SCHEDULE</td>
</tr>
<tr>
<td>969</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YEAR</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YDD</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>YM</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YMD</td>
</tr>
<tr>
<td>1,191</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>YY</td>
</tr>
<tr>
<td>3,229</td>
<td>211</td>
<td>50</td>
<td>6</td>
<td>* TOTAL *</td>
</tr>
</tbody>
</table>

Figure 4: YEAR2K sample final page

******************************************************************************
*** THIS PROGRAM READS ALL THE MEMBERS OF A PDS AND BUILDS A JOB ***
*** STREAM CONTAINING IEBCOPY JCL TO COPY SELECTED MEMBERS OF A ***
*** PDS WHEN THE MEMBER IS FOUND TO CONTAIN CERTAIN IMBEDDED ***
*** CHARACTER STRINGS. ***
*** THE CHARACTER STRINGS ARE FOUND IN THE TABLE DEFINED AT LABEL ***
*** 'STRINGS' BY THE MACRO 'STDEF'. ***
******************************************************************************

MACRO
&NAME STDEF &A,&B,&C,&D
GBLA &N,&IMBED,&OTHER,&WORD,&PREFIX,&SUFFIX
LCLA &K,&F
LCLC &T
&T SETC '&A'
&K SETA K'&A
AIF ('&A'(1,1) NE '').NOTQ
&K SETA &K-2
&T SETC '&A'(2,&K)
.NOTQ AIF (&K GT 0).NOTNULL
 MNOTE 8, 'NULL STRING NOT ALLOWED'
MEXIT
.NOTNULL AIF ('&B' NE 'P' AND '&C' NE 'P' AND '&D' NE 'P').NOTP
&F SETA &F+&PREFIX
.NOTP AIF ('&B' NE 'S' AND '&C' NE 'S' AND '&D' NE 'S').NOTS
&F SETA &F+&SUFFIX
.NOTS AIF ('&B' NE 'W' AND '&C' NE 'W' AND '&D' NE 'W').NOTW
&F SETA &F+&WORD
.NOTW ANOP
&NAME DC ALL(&K-1,&F),CLK&T'
&N SETA &N+1
AIF (N'&SYSLIST EO 1).IMBED
&OTHER SETA 1
MEXIT
.IMBED ANOP
&IMBED SETA 1
MEND MACRO
&LABEL SMUM002 &DSECT-YES,&C-Ø
PUSH PRINT
PRINT GEN

* MACRO TO DESCRIBE PDS BLDL ENTRY WITH ISPF STATISTICS, **
* TO BE USED BY 'BLDL' MACRO. **
* DSECT-YES WILL CAUSE A DSECT TO BE CREATED. **
* DSECT-NO DATA WILL BEGIN ON A DOUBLEWORD BOUNDARY. **
* C_ LABELS WILL BE GU_2XX (_ MAY BE ANY ALPHAMERIC **
* CHARACTER(S), INTENDED FOR GENERATING MULTIPLE **
* COPIES OF THE GENERATED LAYOUT). **
*** THIS MACRO IS A MODIFICATION TO 'GTEUM02' FROM THE ***
*** CONNECTICUT BANK TAPE. THE IMPLEMENTATION OF THIS SOURCE ***
*** MANAGEMENT SYSTEM WAS MUCH EASIER BY UTILIZING THIS EXISTING ***
*** CODE. MUCH GRADITUDE AND APPRECIATION IS GIVEN TO: ***

* * CHUCK HOFFMAN, SYSTEMS PROGRAMMING, GTEL COMPUTING CENTER ***

* * MODIFICATION OF HIS MACRO ON THE CONNECTICUT BANK TAPE EASED ***
* THE IMPLEMENTATION OF THIS SYSTEM. ***

*********************************************************************

AIF ('&DSECT' EQ 'YES').GUM02A
&LABEL DS 0D , ISPF STATS PDS BLDL ENTRY
AGO .GUM02B

.GUM02A ANOP
&LABEL DSECT , ISPF STATS PDS BLDL ENTRY
.GUM02B ANOP

GU&C.2FF DS XL2 BLDL COUNT OF ENTRIES
GU&C.2LL DS XL2 BLDL LENGTH OF ENTRIES
GU&C.2NAM DS CL8 MEMBER NAME
GU&C.2TTR DS XL3 PDS VALUE 'TTR'
GU&C.2K DS X BLDL VALUE 'K'
GU&C.2Z DS X BLDL VALUE 'Z'
GU&C.2C DS X PDS VALUE 'C'
GU&C.2VER DS X ISPF VERSION NUMBER (BIN)
GU&C.2MOD DS X ISPF MOD NUMBER (BIN)
DS XL2 (UNUSED, X'0000')
GU&C.2DATC DS PL4 ISPF DATE CREATED (PACK)
GU&C.2DATM DS PL4 ISPF DATE MODIFIED (PACK)
GU&C.2TIMM DS XL2 ISPF TIME MODIFIED (PK NOSIGN)
GU&C.2SIZE DS XL2 ISPF SIZE (BIN)
GU&C.2INIT DS XL2 ISPF INITIAL SIZE (BIN)
GU&C.2MODL DS XL2 ISPF COUNT OF MOD LINES (BIN)
GU&C.2ID DS CL7 ISPF USERID
DS CL3 (UNUSED X'404040')

POP PRINT
MEND

&MYNAME CSECT,
STM R14,R12,12(R13) SAVE REGS TO CALLER S.A.
B (BEGIN-&MYNAME)(R15) BRANCH AROUND EYECATCHER
DC A(L'NAME) LENGTH OF CSECT NAME
NAME DC C'&MYNAME' CSECT NAME
DC C' &SYSDATE &SYSTIME ' ASSEMBLY DATE/TIME STAMP
DC C'(C) COPYRIGHT KEITH H. NICAISE 1997'
DC C'ALL RIGHTS RESERVED'
BEGIN LR RBASE,R15 LOAD BASE REGISTER
USING &MYNAME,RBASE ADDRESSABILITY
PRINT NGEN
GETMAIN R,LV-WORKDLEN GET SAVE/WORK AREA
ST R1,8(0,R13) MY S.A. ADDR INTO CALLER S.A.
ST R13,4(0,R1)
CALLER S.A. ADDR INTO MY S.A.
LR R13,R1 R13 POINTS TO MY S.A.
ADDRESSABILITY OF SAVE AREA
R1 POINTS TO CALLER S.A.
R15 R0 AND R1 ARE RESTORED

* EJECT

***********************************************************************
***
*** MAINLINE ROUTINE
***
***********************************************************************

MAIN EQU *
ST R1,R1SAVE
XC COMPCODE,COMPCODE
* L R1,-A(INITIAL) POINT TO INITIALIZATION ROUTINE
BALR RBAL,R1 GO PERFORM INITIALIZATION
*
MAINDIRL BAL RBAL,GETDIR GET MEMBER NAME
LTR R15,R15 END OF DIRECTORY REACHED?
BNZ MAINEND YES
*
L R3,EXCLUDE1 POINT TO CURRENT EXCLUSION
LR R4,R3 POINT TO BEGINNING OF MEMBER NAME
LA R0,7 MAXIMUM LENGTH-1
MAINWC CLI 1(R4),C'**' WILD CARD PATTERN?
BE MAINWXC YES
LA R4,1(R4) POINT TO NEXT CHARACTER
BCT R0,MAINWC CONTINUE
*
MAINWXC SR R4,R3 GET LENGTH-1
*
MAINXL EX R4,MAINXCLC IS MEMBER TO BE EXCLUDED?
BL MAINNX NO
BH MAINXMB MAYBE
*
AP EXCLUDED,-P'1' COUNT EXCLUSION
MVC LINE+9(S),OUTMEM MOVE MEMBER NAME TO OUTPUT LINE
MVC LINE+18(S),-C'EXCLUDED' SET EXCLUSION MESSAGE
MVC LINE+26(6),OCCURPAT SET EDIT PATTERN
ED LINE+26(6),EXCLUDED FORMAT EXCLUSION COUNT
MVI LINE,C'0' SET TO DOUBLE SPACE
BAL RBAL.DOUBLESP ALLOW FOR DOUBLE SPACE
BAL RBAL,PRINT GO PRINT LINE
B MAINDIRL GO GET NEXT MEMBER
*
MAINXCLC CLC OUTMEM(*-*) ,0(R3) IS MEMBER TO BE EXCLUDED?
*
MAINXMB LA R3,L'EXCLUDES(R3) POINT TO NEXT ENTRY
ST R3,EXCLUDE1 SAVE POSITION
B MAINXL GO CHECK
**MAINNX** ST R15, INRECLOC          INITIALIZE FOR GETREC

* **MAINNXTR** BAL RBAL, GETREC       READ RECORD FROM CURRENT MEMBER
  LTR R15, R15               END OF MEMBER REACHED?
  BNZ MAINDIRL             YES
  MVI HIT, 0               CLEAR 'FIND' FLAG

* **CLI** IMDEF, 0             ANY IMBEDDED DEFINITIONA?
  BE MAINNOIM           NO

* **BAL** RBAL, SCAN1       SCAN FOR IMBEDDED ENTRIES

**MAINNOIM** CLI OTDEF, 0     ANY NON-IMBED DEFINITIONA?
  BE MAINNOOT          NO

* **BAL** RBAL, SCAN2        SCAN FOR WORDS, PREFIXES, & SUFFIXES

**MAINNOOT** CLI HIT, 0      ANY FINDS?
  BE MAINNXTR         NO

  AP FINDS, =P'1'    COUNT OCCURRENCE IN RECORD
  L R1, INRECLOC    POINT TO SOURCE IMAGE
  MVC OUTSOURC, 0(R1)     MOVE SOURCE TO OUTPUT AREA
  MVC OUTCOUNT-1(L'OUTCOUNT+1), OCCURPAT SET EDIT PATTERN
  ED OUTCOUNT-1(L'OUTCOUNT+1), RECORDS+1 FORMAT RECORD COUNT
  MVC OUT7380, 72(R1)  MOVE COLUMNS 73-80
  BAL RBAL, PUTOUT WRITE OUTPUT RECORD

  B MAINNXTR       GO CONTINUE

**MAINEND** DS $H

* **LM** R3, R5, TOTREGS  LOAD TOTAL REGISTERS

**MOVETTLS** MVC 0(4*L'TOTALS, R3), 4*L'TOTALS(R3) MOVE GRAND TOTALS
  BXLE R3, R4, MOVETTLS CONTINUE

* **BAL** RBAL, HEADPAGE WRITE OUTPUT RECORD

  MVC LINE+5(6), OCCURPAT SET EDIT PATTERN
  ED LINE+5(6), MEMBERS FORMAT MEMBER NUMBER
  MVC LINE+12(13), =C'MEMBERS FOUND'
  BAL RBAL, PRINT PRINT TOTAL

  MVC LINE+5(6), OCCURPAT SET EDIT PATTERN
  ED LINE+5(6), EXCLUDED FORMAT MEMBER NUMBER
  MVC LINE+12(16), =C'MEMBERS EXCLUDED'
  BAL RBAL, PRINT PRINT TOTAL

  MVC LINE+5(6), OCCURPAT SET EDIT PATTERN
  SP MEMBERS, EXCLUDED COMPUTE REMAINDER
  ED LINE+5(6), MEMBERS FORMAT MEMBER NUMBER

MVC  LINE+12(16),='MEMBERS ANALYZED'
BAL  RBAL,PRINT  PRINT TOTAL

*  
MVC  LINE+5(6),OCCURPAT  SET EDIT PATTERN
ED  LINE+5(6),SELECTED  FORMAT MEMBER NUMBER
MVC  LINE+12(16),='MEMBERS SELECTED'
BAL  RBAL,PRINT  PRINT TOTAL

*  
MVI  LINE,C'0'  SET TO DOUBLE SPACE
BAL  RBAL,DOUBLES  ALLOW FOR DOUBLE SPACE

*  
MVC  LINE+1(10),OCCUR1  SET EDIT PATTERN
ED  LINE+1(10),TRECS  FORMAT TOTAL RECORD COUNT
MVC  LINE+12(16),='RECORDS ANALYZED'
BAL  RBAL,PRINT  PRINT TOTAL

*  
MVC  LINE+1(10),OCCUR1  SET EDIT PATTERN
ED  LINE+1(10),TFINDS  FORMAT TOTAL RECORDS SELECTED
MVC  LINE+12(16),='RECORDS SELECTED'
BAL  RBAL,PRINT  PRINT TOTAL

*  
CP  TFINDS,='P'0'  ANY FINDS?
BZ  MAINNONE  NO

*  
BAL  RBAL,DOUBLES  ALLOW FOR DOUBLE SPACE
MVC  LINE(18),='ANALYSIS SUMMARY:'
BAL  RBAL,PRINT  PRINT TOTAL
MVI  LINE,C'0'  SET TO DOUBLE SPACE
BAL  RBAL,DOUBLES  ALLOW FOR DOUBLE SPACE

*  
BAL  RBAL,DOCOUNTS  PRINT LISTING OF INDIVIDUAL FINDS

*  
MAINNONE  MVC  OUTSEL(3),='/*'  SET END OF DATA
MVC  OUTSEL+3(L'OUTSEL-3),OUTSEL+2  CLEAR REST OF RECORD
LA  R3,OUTSEL  POINT TO OUTPUT LINE
BAL  RBAL,WRITEJCL  WRITE IEBCOPY END OF DATA CONTROL

*  
BEGIN DCB CLOSE

*  
MVC  PRCLOS(PRCLOSLN),CLOSED  INITIALIZE CLOSE LIST
CLOSE (PRINTER),MF=(E,PRCLOSL)  CLOSE IT

*  
MVC  IPCLOS(IPCLOSLN),CLOSED  SET INPUT CLOSE LIST
CLOSE (INPUT),MF=(E,IPCLOSL)  CLOSE INPUT

*  
MVC  PDCLOS(PDCLOSLN),CLOSED  SET PDSDIR CLOSE LIST
CLOSE (PDSDIR),MF=(E,PDCLOSL)  CLOSE PDSDIR

*  
MVC  OPCLOS(OPCLOSLN),CLOSED  SET OUTPUT CLOSE LIST
CLOSE (OUTPUT),MF=(E,OPCLOSL)  CLOSE OUTPUT

*  
MVC  OJCLOS(OJCLOSLN),CLOSED  SET OUTJCL CLOSE LIST

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
CLOSE (OUTJCL),MF=(E,OJCLOSL) CLOSE OUTJCL
*
* END DCB CLOSE
*
END00 LA R15,Ø SET COMPLETION CODE ØØ
ST R15,COMPCODE INTO STORAGE
B ENDING GO TO ENDING
*
EJECT
***********************************************************************
***
*** LINKAGE CONVENTIONS EXITING PROGRAM
***
***********************************************************************
ENDING L R14,COMPCODE R14 SAVES COMP CODE
LR R1,R13 R1 SAVES ADDR OF MY S.A.
L R13,4(Ø,R1) R13 RESTORED, PTR CALLER S.A.
FREEMAIN R,LV=WORKDLEN,A=(R1) FREE MY SAVE/WORK AREA
LR R15,R14 R15 SET TO COMP CODE
LM RØ,R12,20(R13) RØ-R12 RESTORED
L R14,12(Ø,R13) R14 RESTORED
MVI 12(R13),X'FF' SET COMPLETION SIGNAL
BR R14 RETURN TO CALLER
*
*
* BEGIN STUB DEFINE
*
*
EJECT
***********************************************************************
***
*** GET MEMBER NAME FROM DIRECTORY
***
***********************************************************************
*
GETDIR ST RBAL,SAVGDBAL SAVE LINKAGE REGISTER
*
CLI DFLAG,Ø FIRST TIME?
* BNE GDNOT1ST NO
MVI DFLAG,X'FF' SET FLAG
*
GDRD BAL RBAL,READDIR READ DIRECTORY RECORD
LTR R15,R15 NORMAL RETURN?
* BNZ GDRETURN NO
BNZ GDEND NO
*
GDNOT1ST L R2,DIRENTRY LOAD ADDRESS OF MEMBER DATA
*
AP TRECS,RECORDS ACCUMULATE TOTAL RECORDS PROCESSED
ZAP RECORDS,-P'0' CLEAR MEMBER RECORD COUNT
AP MEMBERS,-P'1' COUNT NUMBER OF MEMBERS
*

CLI Ø(R2),X'FF' END OF DIRECTORY BLOCK?
BE GDRD YES
*
MVC OUTMEM,Ø(R2) MOVE MEMBER NAME TO OUTPUT AREA
XR R15,R15 SET NORMAL RETURN
*
GDRETURN L RBAl,SAVGBAl RESTORE LINKAGE REGISTER
BR RBAl RETURN
*
GDEND LA R15,4 SET END-OF-DIRECTORY EXIT
B GDRETURN GO EXIT
*
EJECT
***********************************************************************
*** ***
*** READ DIRECTORY RECORD ***
*** ***
***********************************************************************
*
READDIR ST RBAl,SAVRDBAl SAVE LINKAGE REGISTER
*
L R6,DIRENTRY LOAD ADDRESS OF CURRENT LOCATION
LTR R6,R6 FIRST DIRECTORY BLOCK?
BZ RDNXTDIR YES
*
MVI LINE,C'0' SET TO DOUBLE SPACE
BAL RBAl,DUBLES P ALLOW FOR DOUBLE SPACE
*
MVC LINE+1(6),OCCURP AT SET EDIT PATTERN
ED LINE+1(6),MEMBERS FORMAT MEMBER NUMBER
MVC LINE+9(8),OUTMEM MOVE MEMBER NAME TO OUTPUT LINE
MVC LINE+18(LOCURS).OCCURS
ED LINE+18+OCCUR1-OCURS(L'OCCUR1),RECORDS FORMAT RECORDS
ED LINE+18+OCCUR2-OCURS(L'OCCUR2),FINDS " FIND OCCURRENCES
BAL RBAl,PRINT PRINT MEMBER HEADING LINE
CP FINDS,-P'0' ANY FINDS?
BZ RDNXTMEM NO
*
BAL RBAl,GETSTATS GET MEMBER STATISTICS
LTR R15,R15 STATS OKAY?
BNZ RDNOSTAT NO
OC GU02DATC,GU02DATC CREATION DATE BINARY ZEROS?
BZ RDNOSTAT YES
*
MVC LINE+1(11),=C'ISPF STATS:'
UNPK LINE+13(6),GU02TTR(L'GU02TTR+1) UNPACK TTR NYBLS
NC LINE+13(5),=B'X'F' MASK OUT ZONES
TR LINE+13(5),=C'0123456789ABCDEF' CONVERT TO DIXPLAY
XR Rl,Rl CLEAR REGISTER
IC Rl,GU02MOD GET MODIFICATION
ST R1,DOUBLE SAVE
IC R1,GU02VER GET VERSION
MH R1,=H'100' MOVE 2 DECIMAL DIGITS LEFT
A R1,DOUBLE ADD MODIFICATION
CVD R1,DOUBLE CONVERT TO DECIMAL
MVC LINE+18(7),=X'402021206B2020' SET EDIT PATTERN
ED LINE+18(7),DOUBLE+5 FORMAT VV.MM
ICM R1,B'1111',GU02DATC GET CREATION DATE
ST R1,JGYYDDD SAVE FOR CONVERSIONT
BAL RBAL,JULGREG CONVERT TO MM/DD/YY
MVC LINE+26(8),JGMMDDYY MOVE TO LINE
ICM R1,B'1111',GU02DATM GET CREATION DATE
ST R1,JGYYDDD SAVE FOR CONVERSIONT
BAL RBAL,JULGREG CONVERT TO MM/DD/YY
MVC LINE+35(8),JGMMDDYY MOVE TO LINE
UNPK LINE+46(5),GU02TIMM(3) UNPACK MODIFIED TIME
MVC LINE+45(2),LINE+46 MOVE HH LEFT
MVI LINE+47,C':' SEPARATE HH:MM
LH R1,GU02SIZE LOAD SIZE FROM DIRECTORY
CVD R1,DOUBLE CONVERT TO DECIMAL
MVC LINE+50(7),OCCURPAT SET EDIT PATTERN
ED LINE+50(7),DOUBLE+5 FORMAT SIZE
LH R1,GU02INIT LOAD INITIAL SIZE FROM DIRECTORY
CVD R1,DOUBLE CONVERT TO DECIMAL
MVC LINE+57(7),OCCURPAT SET EDIT PATTERN
ED LINE+57(7),DOUBLE+5 FORMAT SIZE
ICM R1,B'0011',GU02MOD LOAD COUNT OF MOD LINES
CVD R1,DOUBLE CONVERT TO DECIMAL
MVC LINE+64(7),OCCURPAT SET EDIT PATTERN
ED LINE+64(7),DOUBLE+5 FORMAT SIZE
MVC LINE+71(7),GU02ID MOVE USER ID TO LINE
BAL RBAL,PRINT PRINT STATISTICS

RDNOSTAT MVC OUTSEL+L'SELECT-1(8),OUTMEM SET MEMBER NAME
LA R3,OUTSEL POINT TO OUTPUT RECORD
BAL RBAL,WRITEJCL WRITE IEBCOPY SELECT STATEMENT

AP TFINDS,FINDS ACCUMULATE GRAND TOTAL
ZAP FINDS,=P'0' RESET COUNTER
AP SELECTED,=P'1' ACCUMULATE TOTAL SELECTIONS

BAL RBAL,DOCOUNTS PRINT LISTING OF INDIVIDUAL FINDS
MVI OUTAREA,C'-' SET SEED
MVC OUTAREA+1(L'OUTAREA-1),OUTAREA SET INDICATOR LINE
BAL RBAL,PUTOUT WRITE OUTPUT RECORD
B RDNXTMEM GO GET NEXT ENTRY

RDNXTDIR GET PDSDIR,DIRBLOCK READ DIRECTORY RECORD
LA R6,DIRBLOCK+2 POINT TO ENTRY
ST R6,DIRENTRY SAVE ADDRESS (NOT REALLY NEEDED)

LH R5,DIRBLOCK LOAD NUMBER NUMBER OF BYTES USED
STH R5,DIRSPACE SAVE
SH R5,-H'2' REDUCE BY LENGTH OF FIELD

BNP RDNXTDIR
  IF EMPTY DIRECTORY BLOCK, GO TO NEXT
B  RD1STMEM
  GO PROCESS FIRST ENTRY IN BLOCK
*
RDNXMEM L  R6,DIRENTRY
  LOAD ADDRESS OF CURRENT LOCATION
LH R5,DISRSPACE
  LOAD REMAINING SPACE IN BLOCK
IC R1,11(R6)
  LOAC 'C' FIELD
N  R1,=F'31'
  GET USER AREA HALFWORDS (5 LOW BITS)
LA R1,12(R1,R1)
  BYTES + MEMBER NAME, 'TTR', AND 'C'
SR R5,R1
  DEDUCT CURRENT ENTRY LENGTH
AR R6,R1
  POINT TO NEXT ENTRY
*
RD1STMEM CLI 0(R6),X'FF'
  LAST DIRECTRY ENTRY?
BE RDDIREND
  YES
CH R5,=H'11'
  ROOM FOR ADDITIONAL ENTRIES?
BL RDNXTDIR
  NO
ST R6,DIRENTRY
  SAVE CURRENT POINTER
STH R5,DISRSPACE
  SAVE REMAINING SPACE
MVC TTRN,8(R6)
  SAVE RELATIVE DASD ADDRESS
*
MVI TTRN+3,0
  CLEAR 'N'
CLI TTRN+2,0
  VALID ADDRESS?
BNE RDOKAY
  YES
*
MVC LINE+2(8),0(R6)
  SET MEMBER NAME
MVC LINE+11(9),=C'NOT FOUND' SET ERROR MESSAGE
MVI LINE,C'0'
  SET TO DOUBLE SPACE BEFORE PRINT
BAL RBAL,DOUBLES
  ALLOW FOR DOUBLE SPACE
BAL RBAL,PRINT
  PRINT ERROR LINE
B  RDNXTDIR
  GO PROCESS REMAINDER OF LIST
*
*DOKEY POINT INPUT,TTRN
  POINT TO NOTE LIST RECORD
RDOKAY FIND INPUT,(R6),D
  POINT TO NOTE LIST RECORD
XR R15,R15
  CLEAR RETURN CODE
*
RDRETURN L  RBAL,SAVRDBAL
  RESTORE LINKAGE REGISTER
BR RBAL
  RETURN
*
RDNDIREND LA R15,4
  INDICATE END OF DIRECTORY
B  RDRETURN
  GO RETURN
*
EJECT
***********************************************************************
*** ***
*** LIST NUMBER OF INDIVIDUAL COUNTS FOR EACH FOUND STRING ***
*** ***
***********************************************************************
*
DOCOUNTS ST  RBAL,SAVDCB
  SAVE LINKAGE REGISTER
*
MVC LINE(L'SUBHEAD'),SUBHEAD SET SUBHEADING
BAL RBAL,DOUBLES
  ALLOW FOR DOUBLE SPACE
BAL RBAL,PRINT
  PRINT SUBHEADING
*

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
LA R4,WORDLIST   POINT TO LIST OF STRINGS
LA R2,TOTALS
*
DCLOOP XR R3,R3   CLEAR REGISTER
IC R3,0(R4)   INSERT LENGTH-1 OS STRING
*
CP 0(L'TOTALS,R2),-P'0' ANY OCCURRENCES?
BNE DCFORMAT YES
CLC L'TOTALS(3*L'TOTALS,R2),0(R2) IN OTHER TOTALS?
BE DCLoopX NO
*
DCFORMAT MVC LINE+3(8),-X'20206B2021204022' SET EDIT PATTERN
MVC LINE+12(26),LINE+3 REPPLICATE
ED LINE+2(36),0(R2)   FORMAT IMBEDDED,WORD,PREFIX,SUFFIX
EX R3,DCMOVE   MOVE STRING TO PRINT LINE
BAL RBAL,PRINT   PRINT COUNT FOR STRING
*
AP IMBEDDED,0(L'TOTALS,R2) ACCUMULATE MEMBER TOTALS
AP WORDS,L'TOTALS(L'TOTALS,R2)   "
AP PREFIXS,2*L'TOTALS(L'TOTALS,R2)   "
AP SUFFIXS,3*L'TOTALS(L'TOTALS,R2)   "
*
AP 4*L'TOTALS(L'TOTALS,R2),0(L'TOTALS,R2)   " DATASET TOTALS
AP 5*L'TOTALS(L'TOTALS,R2),L'TOTALS(L'TOTALS,R2)   "
AP 6*L'TOTALS(L'TOTALS,R2),2*L'TOTALS(L'TOTALS,R2)   "
AP 7*L'TOTALS(L'TOTALS,R2),3*L'TOTALS(L'TOTALS,R2)   "
*
ZAP 0(L'TOTALS,R2),-P'0' RESET MEMBER COUNT FOR STRING
MVC L'TOTALS(3*L'TOTALS,R2),0(R2)   " MEMBER TOTALS
*
DCLoopX LA R4,3(R3,R4)   POINT TO NEXT STRING
LA R2,8*L'TOTALS(R2)   POINT TO TOTALS FOR NEXT STRING
CLI 0(R4),X'FF'   LAST STRING?
BNE DCLoop NO
*
MVC LINE+3(8),-X'20206B2021204022' SET EDIT PATTERN
MVC LINE+12(26),LINE+3 REPPLICATE
ED LINE+2(36),IMBEDDED FORMAT IMBEDDED,WORD,PREFIX,SUFFIX
*
ZAP IMBEDDED,-P'0'   RESET IMBEDDED TOTALS FOR STRINGS
MVC WORDS(3*L'TOTALS),IMBEDDED " WORD,PREFIX,SUFFIX
MVC LINE+L'SUBHEAD-6(9),-C'* TOTAL *'
BAL RBAL,PRINT   PRINT TOTAL LINE
*
L RBAL,SAVDCBAl RESTORE LINKAGE REGISTER
BR RBAL RETURN
*
DCMOVE MVC LINE+L'SUBHEAD-6(*-*),2(R4)
*
EJECT

***********************************************************************
***

*** READ RECORD FROM PDS MEMBER

***********************************************************************
*
GETREC  ST  RBAL.SAVGRBAL  SAVE LINKAGE REGISTER
*
          L  R1,INRECLOC  POINT TO RECORD LOCATION
          LTR  R1,R1  FIRST RECORD OF MEMBER?
          BNZ  GRNXTREC  NO
*
GRNXTBLK  LA  R2,DECB  POINT TO DECB
          L  R3,BLOCKLOC  POINT TO AREA ADDRESS
          ST  R3,INRECLOC  SAVE RECORD POINTER
          READ  (R2),SF,INPUT,(R3),MF=E  READ BLOCK FROM MEMBER
          CHECK  (R2)  AWAIT ECB POSTING
*
          LH  R5,INLRECL  LOAD RECORD LENGTH
          LH  R3,INBLKSIZ  LOAD MAXIMUM BLOCK SIZE
          L  R1,DECB+16  LOAD RECORD POINTER WORD (10B)
          SH  R3,14(R1)  SUBTRACT REMAINING COUNT
          L  R1,BLOCKLOC  GET ADDRESS OF BLOCK
          AR  R3,R1  POINT TO END OF BLOCK
          BCTR  R3,0  POINT TO LAST BYTE OF BLOCK
          ST  R3,BLOCKEND  SAVE ENDING ADDRESS
          L  R1,INRECLOC  POINT TO BEGINNING OF BLOCK
          B  GR1STREC  GO PROCESS FIRST RECORD OF BLOCK
*
GR1STREC  L  R1,INRECLOC  GET PREVIOUS RECORD LOCATION
          AH  R1,INLRECL  POINT TO NEXT RECORD
          C  R1,BLOCKEND  PAST END OF BLOCK?
          BNL  GRNXTBLK  YES
*
GRISREC  ST  R1,INRECLOC  SAVE ADDRESS OF RECORD
          XR  R15,R15  SET 'RECORD FOUND' CODE
          AP  RECORDS,-P'1'  COUNT RECORD
*
GRRETURN  L  RBAL,SAVGRBAL  RESTORE LINKAGE REGISTER
          BR  RBAL  RETURN
*
GREOF  LA  R15,4  SET 'RECORD NOT FOUND' CODE (EOF)
          B  GRRETURN  GO RETURN
*
EJECT

***********************************************************************
*
*** SCAN FOR IMBEDDED STRINGS

***********************************************************************
*
SCAN1  ST  RBAL,SAVS1BAL  SAVE LINKAGE REGISTER
*
          XR  R3,R3  CLEAR REGISTER

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
L R6,INRECLOC  LOAD ADDRESS OF INPUT RECORD
LA R8,72  NUMBER OF BYTES
LR R5,R8  FOR LENGTH-1 COMPARISON

* S1LOOP2 LA R15,WORDLIST  POINT TO LIST OF STRINGS
LA R14,TOTALS  POINT TO ACCUMULATORS FOR 1ST STRING
BCTR R5,0  REMAINING LENGTH - 1

* S1LOOP1 IC R3,0(R15)  INSERT LENGTH-1 OF WORDLIST STRING

* CLI 1(R15),0  IMBEDDED?
BNE S1LOOP1X  NO

* CR R5,R3  PAST END OF INPUT?
BL S1LOOP1X  YES

* EX R3,S1CLC  MATCH FOUND?
BNE S1LOOP1X  NO

* AP 0(L'TOTALS,R14),-P'1' COUNT OCCURRENCE
MVI HIT,X'FF'  FLAG RECORD

* S1LOOP1X LA R15,3(R3,R15)  POINT TO NEXT WORDLIST ENTRY
LA R14,8+L'TOTALS(R14)  POINT TO CORRESPONDING TOTALS

* CLI 0(R15),X'FF'  END OF LIST?
BNE S1LOOP1  NO

* LA R6,1(R6)  POINT TO NEXT CHARACTER
BCT R8,S1LOOP2  CONTINUE

* L RBAL,SAVS1BAL  RESTORE LINKAGE REGISTER
BR RBAL  RETURN

* S1CLC CLC 2(*-* ,R15),0(R6)

* EJECT

**********************************************************************
***  ***
*** SCAN FOR WORDS, PREFIXES, & SUFFIXES  ***
***  ***
**********************************************************************

* SCAN2 ST RBAL,SAVS2BAL  SAVE LINKAGE REGISTER

* XR R3,R3  CLEAR REGISTER
L R6,INRECLOC  LOAD ADDRESS OF INPUT RECORD
BCTR R6,0  DECREMENT TO PREVIOUS BYTE
LA R8,72  NUMBER OF BYTES
XR R7,R7  INITIALIZE LENGTH

* S2LOOP2 LA R15,WORDLIST  POINT TO LIST OF STRINGS
LA R14, TOTALS  POINT TO ACCUMULATORS FOR 1ST STRING
*  BAL RBAL, GETWORD  SCAN FOR VALID STRING
*  LTR R8, R8  RECORD DEPLETED?
BNP S2RETURN  YES
*  CLC =C'DATE-WRITTEN.',0(R6) COBOL COMMENT DATE?
BE S2LOOP2  YES
CLC =C'DATE-COMPILED.',0(R6)
BE S2LOOP2  YES
*  S2LOOP1 IC R3,0(R15)  INSERT LENGTH-1 OF WORDLIST STRING
*  CR R7, R3  PAST END OF INPUT?
BL S2LOOP1X  YES
*  BNE S2NOTW  CAN'T BE WORD MATCH UNLESS SAME SIZE
*  TM 1(R15), WORDBIT  WORD COMPARISON?
BZ S2NOTW  NO
*  EX R3, S1CLC  MATCH FOUND?
BNE S2RETURN  NO
*  BNE S2NOTW  NO (TO ALLOW PREFIX/SUFFIX TO INCLUDE FULL WORD MATCH)
*  AP L'TOTALS(L'TOTALS, R14), =P'1' COUNT OCCURRENCE
B S2FOUND  GO FLAG RECORD
*  S2NOTW TM 1(R15), PREFBIT  PREFIX COMPARISON?
BZ S2NOTWP  NO
*  EX R3, S1CLC  MATCH FOUND?
BNE S2NOTWP  NO
*  AP 2*L'TOTALS(L'TOTALS, R14), =P'1' COUNT OCCURRENCE
B S2FOUND  GO FLAG RECORD
*  S2NOTP TM 1(R15), SUFXBIT  SUFFIX COMPARISON?
BZ S2LOOP1X  NO
*  LA R1, 0(R6, R7)  POINT TO END OF INPUT STRING
SR R1, R3  LESS LENGTH OF WORDLIST SUFFIX
*  EX R3, S2CLC  MATCH FOUND?
BNE S2LOOP1X  NO
*  AP 3*L'TOTALS(L'TOTALS, R14), =P'1' COUNT OCCURRENCE
*  S2FOUND MVI HIT, X'FF'  FLAG RECORD
S2LOOP1X LA R15,3(R3,R15)  POINT TO NEXT WORDLIST ENTRY
      LA R14,8*L'TOTALS(R14) POINT TO CORRESPONDING TOTALS
 *
      CLI  0(R15),X'FF' END OF LIST?
      BNE S2LOOP1 NO
 *
      B S2LOOP2 CONTINUE
 *
S2RETURN L RBAL,SAVS2BAL RESTORE LINKAGE REGISTER
      BR RBAL RETURN
 *
S2CLC  CLC 2(*-*.R15),0(R1)
 *
EJECT
**********************************************************************
***     ***
*** WRITE JCL FOR IEBCOPY ***  ***
***     ***
**********************************************************************
*
BUILDJCL ST RBAL,SAVBJBAL SAVE LINKAGE REGISTER
 *
      LM R3,R5,-A(FIRSTJCL,L'FIRSTJCL,LASTJCL) LOAD REGISTERS
 *
BJLOOP BAL RBAL,WRITEJCL WRITE JCL RECORD
      BXLE R3,R4,BJLOOP CONTINUE
 *
      L RBAL,SAVBJBAL RESTORE LINKAGE REGISTER
      BR RBAL RETURN
 *
EJECT
**********************************************************************
***     ***
*** WRITE IEBCOPY RECORD ***  ***
***     ***
**********************************************************************
*
WRITEJCL ST RBAL,SAVWJBAL SAVE LINKAGE REGISTER
 *
      MVC JCLOUT,0(R3) MOVE IMAGE
      CLC INPUTDD,0(R3) IS THIS INPUT DD STATEMENT?
      BNE WJNOTIDDD NO
      MVC JCLOUT+L'INPUTDD(44),HEADDSN INSERT DSN
 *
WJNOTIDDD PUT OUTJCL,JCLOUT WRITE RECORD
 *
      L RBAL,SAVWJBAL RESTORE LINKAGE REGISTER
      BR RBAL RETURN
 *
EJECT
**********************************************************************
***     ***
**WRITE COPY OF SOURCE**

* PUTOUT ST RBAL,SAVPBAL SAVE LINKAGE REGISTER
* PUT OUTPUT,OUTAREA WRITE RECORD
* L RBAL,SAVPBAL RESTORE LINKAGE REGISTER
BR RBAL RETURN
*
EJECT

**SCAN FOR ALPHAMERIC STRING**

* GETWORD ST RBAL,SAVGWBAL SAVE LINKAGE REGISTER
* LA R6,1(R6,R7) POINT PAST CURRENT STRING
SR R8,R7 SUBTRACT LENGTH-1 OF PREVIOUS STRING
BCTR R8,Ø " OTHER BYTE
LTR R8,R8 ANY REMAINING DATA?
BNP GWRETURN NO
*
LA R1,0(R6,R8) POINT TO END OF TEXT
EX R8,GWTRT1 FIND FIRST NON-BLANK/SPECIAL
BZ GWNULL EXIT IF NONE FOUND
*
LR R7,R1 GET STARTING ADDRESS OF STRING
SR R7,R6 COMPUTE LENGTH-1 OF EMPTY SPACE
SR R8,R7 REDUCE TOTAL LENGTH
BNP GWRETURN NO (SHOULDN'T HAPPEN)
*
LR R6,R1 POINT TO BEGINNING OF STRING
AR R1,R8 POINT TO DEFAULT END (SHOULDN'T BE)
EX R8,GWTRT2 FIND FIRST BLANK/SPECIAL
LR R7,R1 SET CURRENT POSITION
SR R7,R6 COMPUTE LENGTH OF STRING
BCTR R7,Ø LENGTH - 1
*
GWRETURN L RBAL,SAVGWBAL RESTORE LINKAGE REGISTER
BR RBAL RETURN
*
GWNULL XR R8,R8 FORCE NULL LENGTH
B GWRETURN EXIT
*
GWTRT1 TRT 0(*-*,R6),TRTTAB1
GWTRT2 TRT 0(*-*,R6),TRTTAB2
*
EJECT
**CONVERT JULIAN DATE TO GREGORIAN DATE**

* JULGREG ST RBAL,SAVJGBAL SAVE LINKAGE REGISTER
* ZAP JGDAYS,JGYYDDD+2(2) SAVE DAYS FROM BEGINNING OF YEAR
  ZAP JGMONTHS,='P'1' INITIALIZE MONTH
* LA R15,JANUARY POINT TO FIRST MONTH OF YEAR
  LA R0,L'JANUARY SIZE OF DAYS/MONTH FIELD
  LA R1,DECEMBER POINT TO LAST MONTH OF YEAR
* ZAP FEBRUARY,'P'28' SET NON-LEAP YEAR DAYS
* CLC =X'2000',JGYYDDD YEAR 20XX?
  BE JGYR2000
* JG20THCN TM JGYYDDD+1.1 LEAP YEAR?
  BO JGLOOP NO
  TM JGYYDDD+1,X'12'
  BNM JGLOOP NO
  JGYR2000 AP FEBRUARY,'P'1' ADJUST
* JGLOOP CP JGDAYS,0(L'JANUARY,R15) CURRENT MONTH?
  BNH JGFOUN D YES
  AP JGMONTHS,='P'1' INCREMENT MONTH
  SP JGDAYS,0(L'JANUARY,R15) DECREMENT DAYS PER CURRENT MONTH
  BXLE R15,R0,JGLOOP CONTINUE
* JGFOUND UNPK JGMMDYY(2),JGMONTHS UNPACK MONTH
  UNPK JGMMDYY+3(2),JGDAYS UNPACK DAY
  UNPK JGMMDYY+6(3),JGYYDDD+1(2) UNPACK YEAR
  MVI JGMMDYY+2,','/" SEPARATE MONTH AND DAY
  MVI JGMMDYY+5,','/" SEPARATE DAY AND YEAR
  OI JGMMDYY+1,','0'' FORCE MONTH NUMERIC
  OI JGMMDYY+4,','0'' FORCE DAY NUMERIC
  OI JGMMDYY+7,','0'' FORCE YEAR NUMERIC
* JGRETURN L RBAL,SAVJGBAL LOAD LINKAGE REGISTER
  BR RBAL RETURN
* EJECT

**GET PDS ISPF STATISTICS**

* GETSTATS ST RBAL,SAVGSBAL SAVE LINKAGE REGISTER

XC  BLDLNTRY(BDLLEN),BLDLNTRY  CLEAR ENTRY WORK AREA
MVI  GU02FF+1,X'01'  SET ENTRY COUNT TO 1
MVI  GU02LL+1,X'50'  SET ENTRY LENGTH TO 80
MVC  GU02Nam,OUTMEM  MOVE MEMBER NAME INTO BDL AREA
LA  R1,INPUT  R1 POINTS TO OPEN DCB
LA  R0,BDLNLTRY  R0 POINTS TO BDL ENTRY AREA
BLDL (R1),(R0)  EXECUTE BDL
LTR  R15,R15  TEST RETURN CODE
*  
*  *  
*  BNZ  GSRETURN  EXIT IF NOT NORMAL RETURN  
*  
TM  GU02C,X'80'  IF AN ALIAS
BNO  GSRETURN  THEN
LA  R15,12  TURN ON ALIAS FLAG
* 
GSRETURN  L  RBAL,SAVGSBAL  RESTORE LINKAGE REGISTER
BR  RBAL  RETURN
* 
*  END STUB DEFINE  
*  
***********************************************************************
***  ***  ***
***  PRINT ROUTINE  ***  ***
***  ***  
***********************************************************************
*
PRINT  PUT  PRINTER,LINE  PRINT LINE
MVI  LINE,C' '  SET SEED
MVC  LINE+1(L'LINE),LINE  CLEAR LINE
DOUBLES  BCTR  R9,RBAL  RETURN IF PAGE NOT FULL
*
HEADPAGE  MVC  PAGENO,-X'40202120'  SET EDIT PATTERN
ED  PAGENO,PAGES  FORMAT PAGE NUMBER
AP  PAGES,=P'1'  INCREMENT PAGE COUNT
PUT  PRINTER,HEADER  PRINT PAGE HEADING
LA  R9,56  SET LINES/PAGE
MVI  LINE,C'0'  SET TO DOUBLE SPACE AFTER HEADER
BR  RBAL  RETURN
*
***********************************************************************
***  ***  ***
***  FIXED DATA AREA  ***  ***
***  ***  
***********************************************************************
*
SUBHEAD  DC  C'OIMBEDDED  WORDS  PREFIX  SUFFIX  STRING'
*
OCCURS  DC  C'CONTAINS'

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
OCCUR1 DC X'40206B2020206B202120'
DC C' RECORDS OF WHICH'
OCCUR2 DC X'40206B2020206B202120'
DC C' CONTAIN OCCURRENCES OF SPECIFIED STRINGS'
LOCCURS EQU "-OCCURS"
OCCURPAT DC X'402020202120'
*
SELECT DC C' SELECT MEMBER= '
*
FIRSTJCL DC CL80'//COPY2KYR JOB ,"YEAR 2000 ANALYST","...
<--- CUSTOMIZE'
DC CL80'//COPYSTEP EXEC PGM=IEBCOPY'
INPUTDD DC C'//INPUT DD DISP-SHR,DSN='
DC CL(80-L'INPUTDD)' '
DC CL80'//OUTPUT DD DISP-SHR,DSN=OBJECT.PDS.NAME
<--- CUSTOMIZE'
DC CL80'//SYSPRINT DD SYSOUT="'
DC CL80'//SYSIN DD *'
LASTJCL DC CL80' COPY OUTDD=OUTPUT,INDD=INPUT'
*
&WWORD SETA 4 FULL WORD MATCH VALUE
&PREFIX SETA 2 PREFIX MATCH VALUE
&SUFFIX SETA 1 SUFFIX MATCH VALUE
WORDBIT EQU &WWORD FULL WORD MATCH INDICATOR
PREFIXBIT EQU &PREFIX PREFIX MATCH INDICATOR
SUFIXBIT EQU &SUFFIX SUFFIX MATCH INDICATOR
WORDLIST DS 0C
PUSH PRINT
PRINT GEN
STDEF AGE,W,P
STDEF BIRTH,W,P
STDEF CALENDAR
STDEF CENTURY
STDEF CSADAT
STDEF CSAFEID
STDEF CSAJYD
STDEF DATE,W,P
STDEF DMY
STDEF GREGJUL
STDEF GREGORIAN
STDEF JULGREG
STDEF JULIAN
STDEF MDY
STDEF MMDDYY
STDEF SCHEDULE
STDEF TODAY,W
STDEF YEAR
*
STDEF YD,P,S,W
STDEF YDD
STDEF YM,P,S,W
STDEF YMD
STDEF YY
LASTWORD DC X'FF'  
NOTE THAT THIS MUST IMMEDIATELY FOLLOW LIST OF CHARACTER STRINGS

POP PRINT
IMDEF DC AL1(&IMBED)
OTDEF DC AL1(&OTHER)

TRTTAB1 DC 256X'0'
ORG TRTTAB1+X'81' LOWER CASE 'A'
DC X'8182838486878889'
ORG TRTTAB1+X'91' LOWER CASE 'J'
DC X'919293949596979899'
ORG TRTTAB1+X'A2' LOWER CASE 'S'
DC X'A2A3A4A5A6A7A8A9'
ORG TRTTAB1+C'@'
DC C'@'
ORG TRTTAB1+C'#'
DC C'#'
ORG TRTTAB1+C'$'
DC C'$'
ORG TRTTAB1+C'A'
DC C'ABCDEFGHI'
ORG TRTTAB1+C'J'
DC C'JKLMNOPQR'
ORG TRTTAB1+C'S'
DC C'STUVWXYZ'
ORG TRTTAB1+C'0'
DC C'0123456789'
ORG

TRTTAB2 DC 256X'FF'
ORG TRTTAB2+X'81' LOWER CASE 'A'
DC 9X'0'
ORG TRTTAB2+X'91' LOWER CASE 'J'
DC 9X'0'
ORG TRTTAB2+X'A2' LOWER CASE 'S'
DC 8X'0'
ORG TRTTAB2+C'@'
DC X'0'
ORG TRTTAB2+C'#'
DC X'0'
ORG TRTTAB2+C'$'
DC X'0'
ORG TRTTAB2+C'A'
DC 9X'0'
ORG TRTTAB2+C'J'
DC 9X'0'
ORG TRTTAB2+C'S'
DC 9X'0'
ORG TRTTAB2+C'0'
DC 10X'0'
ORG

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
* OPEND OPEN (,),MF=L
CLOSED CLOSE (,),MF=L
LTORG
*
    PUSH PRINT SAVE CURRENT PRINT OPTIONS
PRINT GEN PRINT EXPANDED MACRO
READ DCB,D,SF,MF=L
POP PRINT REINSTATE PREVIOUS PRINT OPTIONS
*
** EJECT
*******************************************************************************
*** ***
*** PERFORM INITIALIZATION TO SAVE BASE ADDRESSING SPACE ***
*** ***
*******************************************************************************
*
INITIAL ST RBAL,SAVILBAL SAVE LINKAGE REGISTER
*
    LA R8,2048(RBASE) LOAD RBASE + HALF PAGE
    LA R8,2048(R8) LOAD RBASE + FULL PAGE
    USING &MYNAME,RBASE,R8 ADDRESSABILITY
*
    MVC JGMOTBL(13*G'MGJTOB),JGMOTBLD COPY JULGREG DAYS/MONTH
*
* BEGIN DCB INITIALIZATION
*
    MVC PRINTER(PRINTERL),PRINTERD INITIALIZE DCB
*
    MVC INPUT(INPUTL),INPUTD INITIALIZE INPUT DCB
*
    MVC PDSDIRPDSDIRL),PDSDIRD INITIALIZE PDSDIR DCB
*
    MVC OUTPUT(OUTPUTL),OUTPUTD INITIALIZE OUTPUT DCB
*
    MVC OUTJCL(OUTJCLL),OUTJCLD INITIALIZE OUTJCL DCB
*
    MVC CARDS(CARDSL),CARSD INITIALIZE CARDS DCB
*
* END DCB INITIALIZATION
*
*
* BEGIN DCB OPENS
*
    MVC PROPENL(PROPENLN),OPEND INITIALIZE SET PRINTER OPEN LIST
OPEN (PRINTER, (OUTPUT)),MF=(E,PROPENL) OPEN PRINTER
*
    MVC IPOPENL(IPOPENLN),OPEND SET INPUT OPEN LIST
OPEN (INPUT, (INPUT)),MF=(E,IPOPENLN) OPEN INPUT
*
    MVC PDOPENL(PDOPENLN),OPEND SET PDSDIR OPEN LIST
OPEN (PDSDIR, (INPUT)),MF=(E,PDOPENL) OPEN PDSDIR

MVC OPOPENL(OPOPENLN),OPEND SET OUTPUT OPEN LIST
OPEN (OUTPUT,(OUTPUT)),MF=(E,OPOPENL) OPEN OUTPUT

MVC OJOPENL(OJOPENLN),OPEND SET OUTJCL OPEN LIST
OPEN (OUTJCL,(OUTPUT)),MF=(E,OJOPENL) OPEN OUTJCL

MVC DECBA(DECBA LN),DECBD INITIALIZE DECB

LA R3, INPUT
GET ADDRESS OF PDS DCB
 USING IHADCB, R3
ESTABLISH ADDRESSABILITY
LH R5, DCBRECL
LOAD RECORD LENGTH
STH R5, INLRECL
SAVE
LH R3, DCBBSIZE
LOAD MAXIMUM BLOCK SIZE
STH R3, INBLKSIZ
SAVE
LA R3, 100(R3)
ADD PAD
DROP R3
DROP ADDRESSABILITY
GETMAIN R, LV-(R3)
GET WORK AREA FOR INPUT BLOCKS
ST R1, BLOCKLOC
SAVE ADDRESS

MVC CDOPENL(CDOPENLN),OPEND SET CARDS OPEN LIST
OPEN (CARDS,(INPUT)),MF=(E,CDOPENL) OPEN CARDS

* END DCB OPENS

XC INRECLOC, INRECLOC
ASSURE INITIALLY ZERO (SHOULD BE)
MVI DFLAG, 0
"
ZAP FINDS, =P'0'
INITIALIZE STRING FOUND COUNT
ZAP MEMBERS, =P'0'
INITIALIZE MEMBERS IN PDS
ZAP SELECTED, =P'0'
INITIALIZE SELECTED MEMBERS
ZAP EXCLUDED, =P'0'
INITIALIZE EXCLUDED MEMBERS
ZAP RECORDS, =P'0'
INITIALIZE RECORDS IN 1ST MEMBER
ZAP TREC, =P'0'
INITIALIZE RECORDS IN ALL MEMBER
ZAP TFINDS, =P'0'
INITIALIZE SELECTIONS IN ALL MEMBERS
ZAP IMBEDDED, =P'0'
INITIALIZE 1ST MEMBER IMBEDDED COUNT
MVC WORDS(11*L'TOTALS), IMBEDDED " WORD, PREFIX, SUFFIX, 1ST STR

LA R15, TOTALS
POINT TO FIRST TOTAL
LA R0, 8*L'TOTALS
SIZE OF ENTRY
LA R1, GRANDS
POINT TO GRAND TOTALS
STM R15, R1, TOTREGS
SAVE FOR OTHER LOOPS

ILT MC V 8*L'TOTALS (8*L'TOTALS, R15), 0(R15) " NEXT LINE
BXLE R15, R0, ILT MC V CONTINUE

MVC OUTSEL(L'SELECT), SELECT MOVE IEBCOPY SELECT STATEMENT
MVC OUTSEL + L'SELECT (L'OUTSEL + L'SELECT), OUTSEL + L'SELECT - 1 CLR
TIME
ST R1,JGYYDDD SAVE JULIAN DATE
BAL RBAL,JULGREG CONVERT TO MM/YY/DD
MVC HEADER(L'HEAD),HEAD INITIALIZE HEADER
MVC HEADER+L'HEAD(L'HEADER-L'HEAD),HEADER+L'HEAD-1 CLEAR
MVC PAGENO-4(4),=C'PAGE' SET PAGE NUMBER ID
ZAP PAGES,-P'1' INITIALIZE PAGE COUNT
MVC DDNAME,JCLDDN MOVE IEBCOPY JCL FILE NAME
BAL RBAL,GETNAMES GET SELECTION DSN
MVC JCLOUT(44),HEADDRS MOVE OUTJCL DSN TO SAVE AREA
MVC DDNAME,OUTDDN MOVE SELECTION FILE NAME
BAL RBAL,GETNAMES GET SELECTION DSN
MVC LINE+1(24),=C'RECORDS SELECTED TO DSN=' SET JCL DS NAME
MVC LINE+25(L'HEADDRS),HEADDRS MOVE FILE DSN TO PRINT LINE
MVC DDNAME,PSDDN MOVE SELECTION FILE NAMES
BAL RBAL,GETNAMES PUT JOB/DSN NAMES IN HEADER
MVC HEADDATE,JGMMDDYY MOVE MM/YY/DD TO HEADING
BAL RBAL,HEADPAGE PRINT PAGE HEADER
BAL RBAL,PRINT PRINT SELECTION DSN
BAL RBAL.DOUBLESP ALLOW FOR DOUBLE SPACE
MVC LINE(29),=C'IEBCOPY JCL TO DSN=' SET ID
MVC LINE+20(L'HEADDRS),JCLOUT MOVE FILE DSN TO PRINT LINE
BAL RBAL,PRINT PRINT SELECTION DSN
BAL RBAL.DOUBLESP ALLOW FOR DOUBLE SPACE
BAL RBAL,BUILDJCL WRITE FIRST PART OF IEBCOPY JCL
*
LA R3,EXCLUDES POINT TO FIRST ELEMENT
LA R4,EXCLUDEX-EXCLUDES(R3) POINT TO LAST EXCLUDE
ST R3,EXCLUDE1 SAVE BEGINNING ADDRESS
MVC LINE(27),=C'MANUALLY EXCLUDED MEMBERS:'
BAL RBAL.DOUBLESP ALLOW FOR DOUBLE SPACE
BAL RBAL,PRINT PRINT EXCLUSION SUBHEADER
MVI LINE,C'0' SET TO DOUBLE SPACE
BAL RBAL.DOUBLESP ALLOW FOR DOUBLE SPACE
*
ILCDLOOP GET CARDS,CARDAREA READ EXCLUSION CARD
MVC 0(L'EXCLUDES,R3),CARDAREA MOVE MEMBER NAME TO EXCL TABLE
LA R3,L'EXCLUDES(R3) POINT TO NEXT ENTRY
CR R3,R4 PAST END OF SAVE AREA?
BL 1LCDLOOP NO
*
CARDEOF MVC CDCCLOSL(CDCCLOSLN),CLOSED SET CARDS CLOSE LIST
CLOSE (CARDS),MF=(E,CDCCLOSL) CLOSE CARDS
*
MVC 0(L'EXCLUDES,R3),=8X'FF' SET HIGH VALUES
ST R3,EXCLUDE2 SAVE LAST CARD IMAGE
C R3,EXCLUDE1 ANY EXCLUSIONS?
BNE 1LSORT NO
MVC LINE+5(8),=C'* NONE *' INDICATE NO EXCLUSIONS
BAL RBAL,PRINT PRINT INDICATION
B 1EXIT GO EXIT

* ILSORT L R3, EXCLUDE1 LOAD START OF LIST
* ILSORTL2 LA R4, L'EXCLUDES(R3) POINT TO NEXT ELEMENT OF VECTOR
  C R4, EXCLUDE2 AT END OF VECTOR?
  BE ILSORTX2 YES (BUT PRINT LAST ENTRY)
  BH ILEXIT YES
*
ILSORTL1 CLC 0(L'EXCLUDES, R4), 0(R3) CURRENT ENTRY LOWER?
  BH ILSORTX1 NO
*
  XC 0(L'EXCLUDES, R3), 0(R4) SWAP
  XC 0(L'EXCLUDES, R4), 0(R3) VECTOR
  XC 0(L'EXCLUDES, R3), 0(R4) ELEMENTS
*
ILSORTX1 LA R4, L'EXCLUDES(R4) POINT TO NEXT ENTRY
  C R4, EXCLUDE2 AT END OF LIST?
  BL ILSORTL1 NO
*
ILSORTX2 MVC LINE+5(L'EXCLUDES), 0(R3) MOVED SORTED ENTRY
  BAL RBAL, PRINT PRINT ENTRY
*
  LA R3, L'EXCLUDES(R3) POINT TO NEXT ENTRY
  B ILSORTL2 CONTINUE
*
ILEXIT MVI LINE, C'0' SET TO DOUBLE SPACE
  BAL RBAL, DOUBLES P ALLOW FOR DOUBLE SPACE
*
  L RBAL, SAVILBAL RESTORE LINKAGE REGISTER
  BR RBAL RETURN
*
EJECT

*******************************************************************************
*** ***
*** GET JOB AND PDS DSN NAMES ***
*** ***
*** ** Thanks to Mr. Mark Hoffman for this logic **
*** ***
*******************************************************************************
*
GETNAMES ST RBAL, SAVGNBAL SAVE LINKAGE REGISTER
*
XR R15, R15 ADDRESS OF PSA
USING PSA, R15 ESTABLISH ADDRESSABILITY
L R14, FLCCVT ADDRESS OF CVT
DROP R15 DROP ADDRESSABILITY TO PSA
USING CVTMAP, R14 ESTABLISH ADDRESSABILITY TO CVT
L R15, CVTTCBP ADDRESS OF NEXT TCB POINTE BP
L R15, 4(0, R15) ADDRESS OF CURRENT TCB
DROP R15 DROP ADDRESSABILITY TO CVT

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
USING TCB,R15 ESTABLISH ADDRESSABILITY CURRENT TCB
L R14,TCBTIO ADDRESS OF TCB
USING TCB,R14 ESTABLISH ADDRESSABILITY TO TCB
MVC HEADJOBN,TIOCNJOB MOVE JOB NAME TO HEADER
MVC HEADJOBN-4(4),=C'JOB=' SET JOBNAME ID

* DROP R15 DROP ADDRESSABILITY TO TCB
LA R15,TIOELNGH ADDRESS OF FIRST TIOT ENTRY
DROP R14 DROP ADDRESSABILITY (HLASM OBJECTS)
USING TIOENTRY,R15 ESTABLISH ADDRESSABILITY TO TIOT

* GNTIOTLP CLI TIOELNGH,X'00' END OF TIOT CHAIN?
BE GNSRETURN YES (SHOULDN'T HAPPEN)
CLC TIOEDDNM(8),DDNAME PDS NAME FOUND?
BE GNDSN YES
XR R0,R0 CLEAR REGISTER
IC R0,TIOELNGH INSERT ENTRY LENGTH
AR R15,R0 POINT TO NEXT ENTRY
B GNTIOTLP CONTINUE

* GNDSN XR R1,R1 CLEAR REGISTER
ICM R1,7,TIOEJFCB ADDRESS OF JFCB
USING JFCB,R1 ESTABLISH ADDRESSABILITY TO JFCB
MVC HEADDNM,JFCBDSNM MOVE DSN TO HEADER
MVC HEADDNM-4(4),=C'DSN=' SET DSN ID IN HEADER
DROP R1,R15 DROP ADDRESSING TO JFCB,TIOT,ENTRY

* * *
GNSRETURN RBAL.SAVGNBAL RESTORE LINKAGE REGISTER
BR RBAL RETURN

* EJECT

***********************************************************************
*** FIXED DATA AREA
***
***********************************************************************

HEAD DC C'1YEAR2K ANALYSIS REPORT'
*
* BEGIN DCB CONSTANTS
*

PRINTERD DCB DDNAME=PRINTER,DEVD=DA,DSORG=PS,LRECL=133,
   BLKSIZE=133,MACRF=(PM),RECFM=FBA
*
INPUTD DCB DDNAME=INPUT,DSORG=PO,MACRF=R,EODAD=GM
*
PDSDIRD DCB DDNAME=INPUT,DSORG=PS,MACRF=GM,EODAD=GDEND,LBLKSIZE=256,
   RECFM=F,LRECL=256
PDSDDN EQU PDSDIRD+DCBDDNAM-DCBRELAD
*
OUTPUTD DCB DDNAME=OUTPUT,DSORG=PS,MACRF=PM

OUTDDN EQU OUTPUTD+DCBDDNAM-DCBRELAD
*
OUTJCLD DCB DDNAME=OUTJCL,DSORG=PS,MACRF=PM
JCLDDN EQU OUTJCLD+DCBDDNAM-DCBRELAD
*
CARDSD DCB DDNAME=CARDS,DSORG=PS,MACRF=GM,EODAD=CARDEOF,
RECFM=FB,LRECL=80
*
* END DCB CONSTANTS
*
*
* END CONSTANTS
*

LTORG
*
EJECT
***********************************************************************
***
*** DSECT FOR MY SAVE AREA AND VARIABLES.
***
***
***********************************************************************
WORKD DSECT
MYSAVE DS 18F MY REGISTER SAVE AREA
COMPCODE DS F PROGRAM COMPLETION CODE
RETCDE DS F INTERNAL RETURN CODE
RISAVE DS F INITIAL VALUE IN R1
TOTREGS DS 3F
BLOCKLOC DS F
BLOCKEND DS F
INLRECL DS H
INBLKSIZ DS H
INRECLOC DS F
TRTN DS F
PAGES DS PL2
HIT DS C
DFLAG DS C
MEMBERS DS PL3
SELECTED DS PL3
EXCLUDED DS PL3
RECORDS DS PL4
TRECS DS PL4
TFINDS DS PL4
DOUBLE DS D
DDNAME DS CL8
*
* BEGIN STUB LINK SAVE
*
SAVBJBAL DS A BAL REGISTER SAVE AREA FOR BUILDJCL
SAVDCBAL DS A BAL REGISTER SAVE AREA FOR DOCOUNTS
SAVGDBAL DS A BAL REGISTER SAVE AREA FOR GETDIR
SAVGNBAL DS A             BAL REGISTER SAVE AREA FOR GETNAMES
SAVGRBAL DS A             BAL REGISTER SAVE AREA FOR GETREC
SAVGSBAL DS A             BAL REGISTER SAVE AREA FOR GETSTATS
SAVGWBAL DS A             BAL REGISTER SAVE AREA FOR GETWORD
SAVI L BAL DS A           BAL REGISTER SAVE AREA FOR INITIAL
SAVJGBAL DS A             BAL REGISTER SAVE AREA FOR JULGREG
SAVPOBAL DS A             BAL REGISTER SAVE AREA FOR PUTOUT
SAVRDBAL DS A             BAL REGISTER SAVE AREA FOR READDIR
SAVSLBAL DS A             BAL REGISTER SAVE AREA FOR SCAN1
SAVS2BAL DS A             BAL REGISTER SAVE AREA FOR SCAN2
SAVWJBAL DS A             BAL REGISTER SAVE AREA FOR WRITEJCL
*
* END STUB LINK SAVE
*
SPACE
*
* BEGIN OPEN/CLOSE LIST
*
*
DS OD
*
PROPENL OPEN ( , ), MF=L
PROPENLN EQU *-PROPENL
PRCLOS L CLOSE ( ), MF=L
PRCLOSL EQU *-PRCLOS L
*
IOPENL OPEN ( , ), MF=L
IOPENLN EQU *-IOPENL
IPCLOS L CLOSE ( ), MF=L
IPCLOSL EQU *-IPCLOSL
*
PDOPENL OPEN ( , ), MF=L
PDOPENLN EQU *-PDOPENL
PDCLOS L CLOSE ( ), MF=L
PDCLOSL EQU *-PDCLOSL
*
OOPENL OPEN ( , ), MF=L
OOPENLN EQU *-OOPENL
OPCLOS L CLOSE ( ), MF=L
OPCLOSL EQU *-OPCLOSL
*
OJOPENL OPEN ( , ), MF=L
OJOPENLN EQU *-OJOPENL
OJCLOS L CLOSE ( ), MF=L
OJCLOSL EQU *-OJCLOSL
*
COPENL OPEN ( , ), MF=L
COPENLN EQU *-COPENL
CDCLOS L CLOSE ( ), MF=L
CDCLOSL EQU *-CDCLOSL
*
* END OPEN/CLOSE LIST
*
* BDLNTRY SMUM002 DSECT=NO BDL FORMAT ENTRY
* BDLLEN EQU *-BLDLNTRY LENGTH OF BDL ENTRY
READ DECBA, SF, MF=L DECBA FOR PDS
DECBALN EQU *-DECB
*
* BEGIN DCB DSECTS
*
PRINTER DCB DDNAME=PRINTER,DEVD=DA,DSORG=PS,LRECL=133,
BLKSIZE=133,MACRF=(PM),RECFM=FBA
PRINTERL EQU *-PRINTER
*
INPUT DCB DDNAME=INPUT,DSORG=PO,MACRF=R,EODAD=GREOF
INPUTL EQU *-INPUT
*
PDSDIR DCB DDNAME=INPUT,DSORG=PS,MACRF=GM,EODAD=GDEND,BLKSIZE=256,
RECFM=F,LRECL=256
PDSDIRL EQU *-PDSDIR
*
OUTPUT DCB DDNAME=OUTPUT,DSORG=PS,MACRF=PM
OUTPUTL EQU *-OUTPUT
*
OUTJCL DCB DDNAME=OUTJCL,DSORG=PS,MACRF=PM
OUTJCLL EQU *-OUTJCL
*
CARDS DCB DDNAME=C ARDS,DSORG=PS,MACRF=GM,EODAD=CARDEOF,
RECFM=FB,LRECL=80
CARDSL EQU *-CARDS
*
* END DCB DSECTS
*
JGMOTBL DS PL2'0'
JANUARY DS P'31'
*M A M J J A S O N  N
FEBRUARY DS P'28,31,30,31,30,31,30,31,30,31,30,31,30,31,30'
DECEMBER DS P'31'
JGDAYS DS PL2
JGMONTHS DS PL2
JGMDDYY DS C'MM/DD/YY'
JGYDDD DS F
* END DSECT INSERT
*
HEADER DS CL133
ORG HEADER+L'HEAD+10
HEADJOBN DS CL8,C' ' DSN='
HEADDSN DS CL44,5C
HEADDATE DS CL8
ORG HEADER+L'HEADER-5
PAGENO DS CL4
ORG
*
JCLOUT DS CL80
OUTAREA  DS  CL93
          ORG  OUTAREA
OUTSOURC DS  CL72
OUTMEM  DS  CL8
OUT7380  DS  CL8
OUTCOUNT DS  CL5
          ORG

OUTSEL  DS  CL80

LINE  DS  CL133

DIRENTRY  DS  F  POINTER TO DIRECTORY ENTRY
DIRSPACE  DS  H  SPACE IN DIRECTORY BLOCK

DIRBLOCK DS  CL256

FINDS  DS  CL4
IMBEDDED DS  PL3
WORDS  DS  PL(L'IMBEDDED)
PREFIXS  DS  PL(L'IMBEDDED)
SUFFIXS  DS  PL(L'IMBEDDED)

TOTALS  DS  ØPL(L'IMBEDDED)
.TOTALS  ANOP
&N  SETA  &N-1
   DS  ØPL(L'TOTALS)
   AIF  (&N  GT  Ø).TOTALS
GRANDS  DS  ØPL(L'TOTALS)

EXCLUDE1 DS  F
EXCLUDE2 DS  F
CARDAREA DS  CL80
EXCLUDES DS  3Ø0CL8
EXCLUDEX DS  CL8
   DS  ØD
WORKDLEN EQU  *-WORKD

PRINT GEN

IHAPSA  MAP OF PSA  DSECT=PSA
IKJTCB  MAP OF TCB  DSECT=TCB
TIOT  DSECT
IEFTIOT1  MAP OF TIOT
CVT  DSECT=YES  MAP OF CVT  DSECT=CVTMAP
JFCB  DSECT  MAP OF JFCB
                MAP OF JFCBPREF
JFCBPREF DS  CL16  PREFIX
   IEFJFCBN LIST=NO  JFCB PROPER

DCBD  DSORG=PO,DEVD=DA  A.T.
INTRODUCTION

A frequent problem in performance reporting and monitoring is the manipulation and management of the vast amounts of data produced by SMF, RMF, and third-party product reporters. Various data reduction and reporting tools have evolved over the years to address this problem, perhaps one of the most widely installed being Barry Merrill’s SAS/MXG product. The software provides a basic set of SAS routines that re-format raw SMF data into SAS files (databases).
Sets of reports and trending macros are also provided.

The following example demonstrates the power and efficiency of SAS in data manipulation and presentation. First we used the IBM utility DCOLLECT (see JOB SASJDIV). After the job volspaz read data from SAS databases created in the first step (SASJDIV) and create a report.

The following code was developed in an MVS/ESA 5.2, SAS6.096, and SAS/MXG 13.13 environment. Although levels of MXG and MVS are probably irrelevant, some features of SAS Version 6 are used that do not appear in SAS Version 5 (a competent SAS programmer should be able to remove or re-create these features as required). Specific SAS Version 6 attributes are noted in the example.

SASJDIV

```sas
//SASJDIV JOB COM,'SASDIV',CLASS=W,MSGCLASS=Ø
//*/
/*/ TRAITEMENT : COLLECTE DANS CPE POUR CFT ET RACF
//*/
//DIV EXEC SAS,REGION=ØM,
// WORK='150.20',
// OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL'
//CFT DD DSN=SAS.SMF.CFT,DISP=SHR
//SMF DD DSN=SAS.SMF.RAC,DISP=SHR
//REPORT DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//SASLIST DD SYSOUT=Ø
//SYSIN DD *

OPTIONS PAGESIZE=60 LINESIZE=132 ;

%CPSTART(MODE=BATCH,
 SYSTEM=MVS,
 ROOT=SAS.SAS608.CPE.,
 PDB=SAS.BERCY.DIVPDB.,
 DISP=OLD,
 ROOTSERV=,
 SHARE=N/A,
 MXGSR=( 'SAS.BERCY.SOURCLIB' 'SAS.MXG.SOURCLIB'),
 MXGLIB=SAS.MXG.FORMATS
);

%INCLUDE SOURCLIB(TYPECFT);
RUN;
%INCLUDE SOURCLIB(TYPE80A);
RUN;
```
%CMPROCES(
  COLLECTR=GENERIC,
  TOOLNM=SASDS,
  UNIT=DISK,
  GENLIB=WORK
);

%CPREDUCE();

/**** REPORT OUTPUT ****/

%INCLUDE REPORT(OPTIONS);
%INCLUDE REPORT(HIER);
%INCLUDE REPORT(RJCFT);
%INCLUDE REPORT(RJRAACF1);
%INCLUDE REPORT(RJRAACF2);
/*
/*
/*
/* DELETE THE FILES AFTER PROCESSING
/*
/*
//DELETE EXEC PGM=IDCAMS,COND=(0,NE,DIV.SAS)
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
DELETE SAS.SMF.CFT
DELETE SAS.SMF.RAC
/*
/*

VOLSPAZJCL

//VOLSPAZ JOB EXP,'VOLSPAZ',CLASS=W,MSGCLASS=O,MSGLEVEL=(1,1),
// NOTIFY=DUNAND,USER=SYSOP8,PASSWORD=MANNXX
//DELOUT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
DELETE EXPL69.DISQUE.LIST
IF MAXCC <= 8 THEN SET MAXCC=0
/*
/* VOLSPACE EXEC SAS,REGION=8M,
// WORK='200,50',
// OPTIONS='MEMSIZE=16M DMSBATCH BATCH TERMINAL'
//SOURCLIB DD DSN=SAS.BERCY.REPORTS,DISP=SHR
//LIBRARY DD DSN=SAS.MXG.FORMATS,DISP=SHR,
//SASLIST DD DSN=EXPL69.DISQUE.LIST,DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(TRK.(1,1),RLSE),
// DCB=(RECFM=F,LRECL=133,BLKSIZE=0),MGMTCLAS=DEL32
//SYSIN DD *

OPTIONS PAGESIZE=60 LINESIZE=132;

%LET RETCODE=.:
Using a load library for SCLM–controlled projects

INTRODUCTION

SCLM uses a default naming convention for the partitioned datasets (Project.Group.Type). By default all datasets of a project have the same High Level Qualifier (Project). The second qualifier indicates the group in the hierarchy (eg DEVT,TEST,PROD). The low-level qualifier indicates the dataset type (eg SOURCE,OBJ,LOAD).

One problem, which I have faced with the above naming convention for libraries, is the large number of load libraries. Each project may
have as many load libraries as the number of groups in the hierarchy. In our installation we have one CICS region for each group in the hierarchy. SCLM promotion purges the load module from the original group. If a program has to be accessed from a CICS region after promotion, all the load libraries at the top of the group in the hierarchy should be concatenated to the DFHRP. As an example consider the hierarchy of three groups:

DEVT-----TEST-----PROD

The concatenation of DFHRPL for different CICS regions should be as follows:

<table>
<thead>
<tr>
<th>Region</th>
<th>DFHRPL</th>
<th>DD</th>
<th>Project</th>
<th>Load Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROD</td>
<td>//DFHRPL</td>
<td>DD</td>
<td>DSN=project.PROD.LOAD</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>//DFHRPL</td>
<td>DD</td>
<td>DSN=project.TEST.LOAD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DD</td>
<td>DSN=project.PROD.LOAD</td>
<td></td>
</tr>
<tr>
<td>DEVT</td>
<td>//DFHRPL</td>
<td>DD</td>
<td>DSN=project.DEVT.LOAD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DD</td>
<td>DSN=project.TEST.LOAD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>DD</td>
<td>DSN=project.PROD.LOAD</td>
<td></td>
</tr>
</tbody>
</table>

If an installation has a large number of projects and a number of levels in the project hierarchy, the number of datasets in DFHRPL may exceed the limits. We have solved this problem by allocating one common load library for each group in the hierarchy (COMMON.group.LOAD). The CICS start-up procedure concatenates only the load library corresponding to the group for the CICS region. The following changes are required for the SCLM project definition:

• The language definition for Linkage Editor (FLMLE370). The definition includes an additional translator for copying the load module to ‘COMMON.DEVT.LOAD’.

• REXX program COPYBLD. This program is invoked by the above translator to perform the copy.

• Language definition for promote (FLMPROCP). This translator is invoked during the promote process.

• REXX program COPYPRO.COPYPRO copies the load module to ‘COMMON.targroup.LOAD’, where ‘targroup’ is the target group for promotion.
**PROGRAM SOURCE**

***************************************************************
* FLMLE370 -- 370/LINKAGE EDITOR LANGUAGE DEFINITION FOR SCLM *
* *
* MODIFIED TO ADD A STEP TO COPY THE LOAD MODULE CREATED INTO *
* A COMMON LOAD LIBRARY *
* *
***************************************************************

FLMLANGL LANG=LE370, CANEDIT=N, VERSION=L370V1.0

FLMTRNSL CALLNAME='LINKED/370',
FUNCTION=BUILD,
COMPILE=IEWL,
VERSION=F64,
GOODRC=0,
OPTIONS=(DCBS,MAP)

1 (* SYSLIN *)
FLMALLOC IOTYPE=S, KEYREF=INCL, RECFM=FB, LRECL=80,
RECNUM=20000, DDNAME=SYSLIN

2 (* LOAD MODULE NAME *)
FLMALLOC IOTYPE=L, KEYREF=REF

3 (* SYSLMOD *)
FLMALLOC IOTYPE=P, KEYREF=LOAD, RECFM=U, LRECL=0,
RECNUM=500, DIRBLKS=20, DDNAME=SYSLMOD

4 (* SYSLIB *)
FLMALLOC IOTYPE=A, DDNAME=SYSLIB

ADD THE LIBRARIES TO BE CONCATENATED TO SYSLIB HERE

5 (* N/A *)
FLMALLOC IOTYPE=N

6 (* SYSPRINT *)
FLMALLOC IOTYPE=O, KEYREF=LMAP, RECFM=FB, LRECL=121,
RECNUM=2500, PRINT=Y, DDNAME=SYSPRINT

7 (* N/A *)
FLMALLOC IOTYPE=N

8 (* SYSPUT *)
FLMALLOC IOTYPE=W, RECFM=U, LRECL=0, RECNUM=5000,
DDNAME=SYSPUT

9 (* N/A *)
FLMALLOC IOTYPE=N

10 (* N/A *)

FLMALLOC IOTYPE=N
*
* 11  (* N/A *)
FLMALLOC IOTYPE=N
*
* 12  (* SYSTERM *)
FLMALLOC IOTYPE=A,DDNAME=SYSTERM
FLMCOPYLB NULLFILE
*
******************************************************************************
* - COPY LOAD MODULE TO COMMON LIBRARY for BUILD PROCESS *
******************************************************************************
FLMTRNSL CALLNAM='COPY LOAD MODULE',
FUNCTN=BUILD,
COMPILE=COPYBLD,
DSNAME=library,
CALLMETH=TSOLNK,
VERSION=2.1,
OPTIONS:@@FLMMBR,
GOODRC=0,
PORDER=1
*
* DDNAME ALLOCATIONS
*
FLMALLOC IOTYPE=W,DDNAME=SYSIN
FLMALLOC IOTYPE=W,DDNAME=SYSUT1
FLMALLOC IOTYPE=U,DDNAME=SYSPRINT
FLMALLOC IOTYPE=U,DDNAME=SYSLMOD
*
/* REXX */
/*****************************************************************************/
/* Program : COPYBLD */
/* Used to copy the load module created during SCLM build */
/* process into a common load library */
/* */
/* Name of the member is passed as a parameter */
/* */
/* Name of the common load library is */
/* FLMPRJ.FLMGRP.FLMTYP */
/* */
/* where FLMPRJ : Common project for all load libraries */
/* FLMGRP : lowest level group in the hierarchy */
/* used by SCLM */
/* FLMTYP : type used for LOAD modules */
/* */
/* Replace the constant definition for the above variables*/
/* with the installations local values */
/****************************************************************************/
arg arg
flmprj = 'COMMON'
flmgrp = 'yyyyyyy'
flmtyp = 'LOAD'
msg_status = msg("off")
parse VALUE arg with mem ','
mem = strip(mem,T)
DSTDSN = flmprj || '.' || flmgrp || '.' flmtyp
"FREE FI(DST1)"
"ALLOC FI(DST1) DA('"DSTDSN"') SHR"
if rc <> 0 then
do
  say ' error in allocating ' dstdsn
  return 8
end
TEXT.0 = 2
TEXT.1 = " COPY INDD-SYSLMOD,OUTDD-DST1"
TEXT.2 = " SELECT M-(("mem"..R)"
"EXECIO * DISKW SYSIN (STEM TEXT. FINIS"
"CALL 'SYS1.LINKLIB(IEBCOPY)"
ret_code = RC
"FREE FI(DST1)"
return ret_code

***********************************************************************
* FLMPRCOP - Language Definition for Copy during Promote

***********************************************************************

FLMTRNSL CALLNAM='COPY FOR PROMOTE'.
   FUNCTN= COPY,
   COMPILE= COPYPRO,
   DNAME=library,
   CALLMETH=TSOLNK,
   VERSION=2.1,
   OPTIONS= @@FLMTOG, @@FLMTYP, @@FLMMBR, PDSDATA=Y,
   GOODRC=0,
   PORDER+1

* DDNAME ALLOCATIONS

* FLMALLOC IOTYPE=W,DDNAME=SYSIN
* FLMALLOC IOTYPE=W,DDNAME=SYSUT1
* FLMALLOC IOTYPE=W,DDNAME=SYSUT2
* FLMALLOC IOTYPE=A,DDNAME=SCRI
* FLMCPYLB @@FLMDSN

/* REXX*/

Program : COPYPRO
* Used to copy a load module during SCLM promote
* process into a Common load library

Common load library name is
@@FLMPRJ.@@FLMTOG.@@FLMTYP
@@FLMPRJ - High level qualifier for common load
@@FLMTOG - Target group - passed as parameter
@@FLMTYP - Type - passed as parameter
@@FLMMBR - member name - passed as parameter
Copies only load modules (only for @@FLMTYP='LOAD')
If the installation uses another qualifier for load library type, change in the program
if @@FLMTYP <> 'LOAD' then return 0
DSTDSN = @@FLMPRJ."@@FLMTOG"."@@FLMTYP"
"FREE FI(DSTI)"
"ALLOC FI(DSTI) DA(’”DSTDSN”’) SHR"
if rc <> 0 then do
  say ’error in allocating ’ dstdsn
  return 8
end
TEXT.Ø = 2
TEXT.1 = ”COPY INDD~SRC1.OUTDD=DST1”
TEXT.2 = ”SELECT M=((”@@FLMMBR”,,R))”
"EXECIO * DISKW SYSIN (STEM TEXT. FINIS"
"CALL ’SYS1.LINKLIB(IEBCOPY)’"
ret_code = RC
"FREE FI(DSTI)"
return ret_code
Generating structured Assembler programs with ISPF edit macros – part 2

This month we round off our look at ISPF edit macros.

AJULGREG EDIT MACRO

PROC Ø DEBUG
ISREDIT MACRO (INIT DEBUG) NOPROCESS
IF &SUBNAME = ? THEN DO
ISPEXEC DISPLAY PANEL(AINDCB)
EXIT
END
IF &INIT = DEBUG OR &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
ISREDIT PROCESS DEST
IF &LASTCC ← 0 THEN +
  DO
    ISREDIT FIND FIRST "* BEGIN DCB INIT" 1
    IF &LASTCC ← 0 THEN +
      DO
        SET ZEDSMG = &STR(POSITIONING ERROR)
        SET ZEDLMSG = &STR(NO '* END STUB DEF' CONSTANT)
        ISPEXEC SETMSG MSG(ISRZ001)
        EXIT CODE(12)
      END
    ELSE +
      DO
        ISREDIT (DEST) ← CURSOR
        SET DEST ← &EVAL(&DEST-2)
      END
  END
ELSE +
  ISREDIT (DEST) ← LINENUM .ZDEST
ISREDIT LINE_AFTER &DEST ← DATALINE "*
ISREDIT LINE_AFTER &EVAL(&DEST+1) ← DATALINE " +
MVC JGMOTBL(13*'JGMOTBL'),JGMOTBLD COPY JULGREG DAYS/MONTH"
IF INIT ← INIT THEN +
  DO
    ISREDIT FIND FIRST "HEADPAGE" 2 35
    IF &LASTCC ← 0 THEN +
      DO
        SET ZEDSMG = &STR(POSITIONING ERROR)
        SET ZEDLMSG = &STR(NO CALL TO 'HEADPAGE')
        ISPEXEC SETMSG MSG(ISRZ001)
        EXIT CODE(12)
      END
    ELSE +
      DO

ISREDIT (DEST) = CURSOR
END
SET &DEST = &EVAL(&DEST-1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
TIME "
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
ST R1,JGYYDDD SAVE JULIAN DATE "
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
BAL RBAL,JULGREG CONVERT TO JULIAN DATE TO GREGDATE "
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
MVC HEADDATE,JGMMDDYY MOVE MM/DD/YY TO HEADER "
END
ISREDIT FIND "* END STUB DEF" 1
IF &LASTCC = 0 THEN +
   DO.
      ISREDIT (DEST) = CURSOR
      SET DEST = &EVAL(&DEST-2)
   END
ELSE SET DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "*
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
EJECT "
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
***********************************************************************
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
***
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
*** CONVERT JULEAN DATE TO GREGORIAN DATE ***
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
*** ***
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
***********************************************************************
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "*
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
JULGREG ST RBAL,SAVJGBAL SAVE LINKAGE REGISTER "
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE " +
CLI JGYYDDD,1 IS ACTUAL CENTURY PRESENT? "
SET &DEST = &EVAL(&DEST+1)

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
ISREDIT Line AFTER &DEST = DATALINE " +
   BH JGACTUAL YES
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE " +
   TR JGYYDDD(1),=X'1920' CENTURY=0 --> 19XX, 1-->20XX
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE " +
   JGACTUAL ZAP JGDAYS,JGYYDDD+2(2) SAVE DAYS FROM BEGINNING OF YEAR
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE " +
   ZAP JGMONTHS,-P'1' INITIALIZE MONTH
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   LA R15,JANUARY LOAD ADDRESS OF DAYS/MONTH TABLE
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   LA R0,L'JANUARY ... WIDTH OF TABLE
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   LA R1,DECEMBER ... END OF TABLE
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   ZAP FEBRUARY,-P'28' SET NON LEAP YEAR DAYS
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   CLC =X'2000',JGYYDDD YEAR 2000?
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   BE JGYR2000 YES
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   JG20THCN TM JGYYDDD+1,1 LEAP YEAR?
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   BO JGLOOP NO
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   TM JGYYDDD+1,X'12'
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   BM JGLOOP NO
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
   JGYR2000 AP FEBRUARY,-P'1' ADJUST
SET &DEST = &EVAL(&DEST+1)
ISREDIT Line AFTER &DEST = DATALINE ++
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
JGLOOP CP JGDAYS.Ø(L‘JANUARY,R15) CURRENT MONTH?
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
BNH JGFOUND YES
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
AP JGMONTHS.=P‘1’ INCREMENT MONTH
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
SP JGDAYS.Ø(L‘JANUARY,R15) DECREMENT DAYS PER CURRENT MONTH"
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
BXLE R15.RØ, JGLOOP CONTINUE
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE "*
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE "+
JGFOUND UNPK JGMDDYY(2), JGMONTHS UNPACK MONTH
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
UNPK JGMDDYY+3(2), JGDAYS UNPACK DAY
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
MVI JGMDDYY+2,C‘/’ SEPARATE MONTH AND DAY
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
MVI JGMDDYY+5,C‘/’ SEPARATE DAY AND YEAR
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
OI JGMDDYY+1,C‘Ø’ FORCE MONTH NUMERIC
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
OI JGMDDYY+4,C‘Ø’ FORCE DAY NUMERIC
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
OI JGMDDYY+7,C‘Ø’ FORCE YEAR NUMERIC
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE "*
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE "+
JGRETURN L RBAL, SAVJGBAL LOAD LINKAGE REGISTER
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE~AFTER &DEST = DATALINE " +
BR RBAL RETURN
ISREDIT FIND "* END CONSTANT" 1
IF &LASTCC = Ø THEN + DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "*"
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGMOTBLD DC PL2'0,31,28,31,30,31,30,31,31,30,31,30,31,30,31'"
ISREDIT FIND "* END DSECT IN" 1
IF &LASTCC = 0 THEN +
  DO
    ISREDIT (DEST) = CURSOR
    SET DEST = &EVAL(&DEST-1)
  END
ELSE SET DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "*"
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGMOTBL DS PL2'0'
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JANUARY DS P'31'
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
* M A M J J A S O N
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
FEBRUARY DS P'28,31,30,31,30,31,30,31,30,31,30,31,30,31,30'"
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
DECEMBER DS P'31'
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGDAYS DS PL2
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGMONTHS DS PL2
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGMMDDYY DC C'MM/DD/YY'
SET &DEST = &EVAL(&DEST+1)
ISREDIT LINE_AFTER &DEST = DATALINE "+
JGYYDDD DS F
ISREDIT FIND FIRST "* END STUB LINK SAVE" 1
IF &LASTCC = 0 THEN +
  DO
    SET ZEDSMSE = &STR(POSSITIONING ERROR)
    SET ZEDLMSE = &STR(NO "* END STUB DEF' CONSTANT)
    ISPEXEC SETMSG MSG(ISRZ001)
    EXIT CODE(12)
  END
ELSE +
DO
  ISREDIT (DEST) = CURSOR
  SET DEST = &EVAL(&DEST-2)
END

ISREDIT LINE_AFTER &DEST = DATALINE "+

SAVGJGBAL DS A BAL REGISTER SAVE AREA FOR JULGREG
EXIT CODE(0)

ASTUB EDIT MACRO
PROC Ø DEBUG
ISREDIT MACRO (SUBNAME PREFIX DEBUG) NOPROCESS
  IF &SUBNAME = ? THEN DO
    ISPEXEC DISPLAY PANEL(ASTUB)
    EXIT
  END
  DO WHILE &LENGTH(&STR(&STARS)) LT 65
    SET &STARS = &STR(&STR(&STARS)&STR(*))
  END
  SET &SPACES = &STR(&STR(&SPACES)&STR( ))
END

ISREDIT (RETX) = CURSOR
  IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
  ISREDIT PROCESS DEST
  IF &LASTCC = 0 THEN +
    DO
      SET ZEDSMG = &STR(COMMENT COMMAND PENDING)
      SET ZEDLMSG = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
        NO '* END STUB DEF' CONSTANT)
      ISPEXEC SETMSG MSG(ISRZ001)
      EXIT CODE(12)
    END
ELSE +
  DO
    ISREDIT (DEST) = CURSOR
    SET DEST = &EVAL(&DEST-2)
  END
ELSE +
  ISREDIT (DEST) = LINENUM .ZDEST
  SET &NAME = &STR(SAV&PREFIX.BAL)
  SET &SAVE = &STR(&SUBSTR(1:9.&SUBNAME.&SPACES))
  SET &SAVE = &STR(&SAVE.ST RBAL,&NAME.&SPACES)
  SET &SAVE = &STR(&SUBSTR(1:35,&SAVE)&STR(SAVE LINKAGE REGISTER))
  SET &LOAD = &STR( L RBAL,&NAME.&SPACES)
  SET &LOAD = &STR(&SUBSTR(1:35,&LOAD)&STR(RESTORE LINKAGE REGISTER))
  SET &RETURN = &STR( BR RBAL&SPACES)
  SET &DC = &STR(&NAME DS A&SPACES)
  SET &DC = &STR(&SUBSTR(1:35,&DC) +
    &STR(BAL REGISTER SAVE AREA FOR &SUBNAME))
  ISREDIT LINE_AFTER &DEST = DATALINE "+

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
ISREDIT LINE AFTER &EVAL(&DEST+1) = DATALINE " EJECT"
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "****STARS.***"
ISREDIT LINE_AFTER &EVAL(&DEST+3) = DATALINE "****SPACES.***"
ISREDIT LINE_AFTER &EVAL(&DEST+4) = DATALINE "****SPACES.***"
ISREDIT LINE_AFTER &EVAL(&DEST+5) = DATALINE "****SPACES.***"
ISREDIT LINE_AFTER &EVAL(&DEST+6) = DATALINE "****STARS.***"
ISREDIT LINE_AFTER &EVAL(&DEST+7) = DATALINE "*
ISREDIT LINE_AFTER &EVAL(&DEST+8) = DATALINE "&SAVE"
ISREDIT LINE_AFTER &EVAL(&DEST+9) = DATALINE "*
ISREDIT LINE_AFTER &EVAL(&DEST+10) = DATALINE "&LOAD"
ISREDIT LINE_AFTER &EVAL(&DEST+11) = DATALINE "&RETURN"
ISREDIT FIND "* END STUB LINK" 1
ISREDIT (LINEX) = CURSOR
ISREDIT LINE_AFTER &EVAL(&LINEX-2) = DATALINE "&DC"
ISREDIT LOCATE &DEST
SET &BAL = &STR( BAL RBAL,&SUBNAME&SPACES)
SET &BAL = &STR(&SUBSTR(1:35,&BAL)LINK TO &SUBNAME&SPACES)
ISREDIT LOCATE &RETX
ISREDIT LINE_AFTER &RETX = DATALINE "*
ISREDIT LINE_AFTER &RETX+1 = DATALINE "&BAL"
EXIT CODE(0)

ABAT EDIT MACRO
ISREDIT MACRO (MEMBER)
IF &MEMBER = ? THEN DO
ISPEXEC DISPLAY PANEL(ABAT)
EXIT
END
SET &DEFAULT = &STR(ABATSkel)
IF &MEMBER ! = &STR() THEN SET &DEFAULT = &STR(&MEMBER)
ISREDIT COPY &DEFAULT AFTER .ZFIRST
IF &LASTCC = 0 THEN DO
SET ZEDSMGS = &STR(&DEFAULT NOT FOUND)
SET ZEDILMSG = &STR(MEMBER &DEFAULT CANNOT BE FOUND IN PDS)
ISPEXEC SETMSG MSG(ISRZ001)
EXIT
END
ISREDIT (PROG) = MEMBER
ISREDIT CHANGE @@@@@@@@ &PROG

AINDCB EDIT MACRO
PROC Ø DEBUG
ISREDIT MACRO (DCBNAME PREFIX DEBUG) NOPROCESS
IF &SUBNAME = ? THEN DO
ISPEXEC DISPLAY PANEL(AINDCB)
EXIT
END
IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
ISREDIT PROCESS DEST
IF &LASTCC == 0 THEN +

DO

ISREDIT FIND FIRST "* END DCB INITIAL" 1

IF &LASTCC == 0 THEN +

DO

SET ZEDMSMG = &STR(COMMENT COMMAND PENDING)
SET ZEDLMSG = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
NO '* END STUB DEF' CONSTANT)
ISPEXEC SETMSG MSG(ISRZ001)
EXIT CODE(12)
END
ELSE +

DO

ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END

END
ELSE +

ISREDIT (DEST) = LINENUM .ZDEST
SET DF = &STR(&SUBSTR(1:2,&DCBNAME))
IF &PREFIX == &STR() THEN SET &DF = &PREFIX
SET &LINE = &STR( MVC &DCBNAME(&DCBNAME.L),&DCBNAME.D +
INITIALIZE &DCBNAME DCB)
ISREDIT LINE_AFTER &DEST = DATALINE "**"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
ISREDIT FIND "* END DCB OPEN" 1

IF &LASTCC == 0 THEN +

DO

ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)

SET &LINE = &STR( MVC &DF.OPENL(&DF.OPENLN),OPEND +
SET &DCBNAME OPEN LIST)
ISREDIT LINE_AFTER &DEST = DATALINE "**"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
SET &LINE = &STR( OPEN (&DCBNAME,(INPUT)),MF=(E,&DF.OPENL) +
OPEN &DCBNAME)
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "&LINE"
ISREDIT FIND "* END DCB CLOSE" 1

IF &LASTCC == 0 THEN +

DO

ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)

SET &LINE = &STR( MVC &DF.CLOS(&DF.CLOSIN),CLOSED +
SET &DCBNAME CLOSE LIST)
ISREDIT LINE_AFTER &DEST = DATALINE "**"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
SET &LINE = &STR( CLOSE (&DCBNAME),MF=(E,&DF.CLOS) +

CLOSE &DCBNAME
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "&LINE"
ISREDIT FIND "* END DCB CONST" 1
IF &LASTCC = 0 THEN +
  DO
    ISREDIT (DEST) = CURSOR
    SET DEST = &EVAL(&DEST-2)
  END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DCBNAME.D)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)+
  &STR(DCB DDNAME=&DCBNAME,DSORG=PS,MACRF=GM,EODAD=&DF.EOF))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST - DATALINE "&LINE"
ISREDIT FIND "* END OPEN/CLOS" 1
IF &LASTCC = 0 THEN +
  DO
    ISREDIT (DEST) = CURSOR
    SET DEST = &EVAL(&DEST-2)
  END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DF.OPENL)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)&STR(open (.),MF=L))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
SET &LINE = &STR(&DF.OPENLN)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)&STR(EQU *-&DF.OPENL))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
SET &LINE = &STR(&DF.CLOS L)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)&STR(close (.),MF=L))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
SET &LINE = &STR(&DF.CLOS LN)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)&STR(EQU *-&DF.CLOS L))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
ISREDIT FIND "* END DCB DSECT" 1
IF &LASTCC = 0 THEN +
  DO
    ISREDIT (DEST) = CURSOR
    SET DEST = &EVAL(&DEST-2)
  END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DCBNAME)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)+
  &STR(DCB DDNAME=&DCBNAME,DSORG=PS,MACRF=GM,EODAD=&DF.EOF))
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
SET &LINE = &STR(&DCBNAME.L)
SET &LINE = &STR(&STRSUBSTR(1:9,&LINE)&STR(EQU *-&DCBNAME))
ISREDIT LINE_AFTER &DEST = DATALINE "&LINE"
EXIT CODE(0)
AOUTDCB EDIT MACRO

PROC 0 DEBUG
ISREDIT MACRO (DCBNAME PREFIX DEBUG) NOPROCESS
IF &SUBNAME = ? THEN DO
ISPEXEC DISPLAY PANEL(AOUTDCB)
EXIT
END
IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
ISREDIT PROCESS DEST
IF &LASTCC = 0 THEN +
ISREDIT FIND FIRST "* END DCB INITIAL" 1
IF &LASTCC = 0 THEN +
DO
SET ZEDMSG = &STR(COMMENT COMMAND PENDING)
SET ZEDLMSG = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
NO '* END STUB DEF' CONSTANT)
ISPEXEC SETMSG MSG(ISRZ001)
EXIT CODE(12)
END
ELSE +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
END
ELSE +
ISREDIT (DEST) = LINENUM .ZDEST
SET DF = &STR(&SUBSTR1:2,&DCBNAME))
IF &PREFIX = &STR() THEN SET &DF = &PREFIX
SET &LINE = &STR( MVC &DCBNAME(&DCBNAME.L),&DCBNAME.D +
INITIALIZE &DCBNAME DCB)
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
ISREDIT FIND "* END DCB OPEN" 1
IF &LASTCC = 0 THEN +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR( MVC &DF.OPENL(&DF.OPENLN),OPEND +
SET &DCBNAME OPEN LIST
ISREDIT LINE_AFTER &DEST = DATALINE "*
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
SET &LINE = &STR( OPEN (&DCBNAME.(OUTPUT)),MF=(E,&DF.OPENL) +
OPEN &DCBNAME)
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "&LINE"
ISREDIT FIND "* END DCB CLOSE" 1
IF &LASTCC = 0 THEN +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(MVC &DF.CLOSCL(&DF.CLOSCLN),CLOSED +
SET &DCBNAME CLOSE LIST)
ISREDIT LINE_AFTER &DEST = DATALINE "*"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
SET &LINE = &STR(CLOSE (&DCBNAME).MF=(E, &DF.CLOSCL) +
CLOSE &DCBNAME)
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "&LINE"
ISREDIT FIND "* END DCB CONST" 1
IF &LASTCC = 0 THEN +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DCBNAME.D)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)+
&STR(DCB DDNAME-&DCBNAME.DSORG-PS.MACRF=PM)
ISREDIT LINE_AFTER &DEST = DATALINE "*"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
ISREDIT FIND "* END OPEN/CLOS" 1
IF &LASTCC = 0 THEN +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DF.OPENL)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)&STR(OPEN (,),MF=L)
ISREDIT LINE_AFTER &DEST = DATALINE "*"
ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE "&LINE"
SET &LINE = &STR(&DF.OPENLN)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)&STR(EQU *-&DF.OPENLN)
ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "&LINE"
SET &LINE = &STR(&DF.CLOSCL)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)&STR(CLOSE (),MF=L)
ISREDIT LINE_AFTER &EVAL(&DEST+3) = DATALINE "&LINE"
SET &LINE = &STR(&DF.CLOSCLN)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)&STR(EQU *-&DF.CLOSCL)
ISREDIT LINE_AFTER &EVAL(&DEST+4) = DATALINE "&LINE"
ISREDIT FIND "* END DCB DSECT" 1
IF &LASTCC = 0 THEN +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
ELSE SET DEST = &EVAL(&DEST+1)
SET &LINE = &STR(&DCBNAME)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)+

&STR(DCB DDNAME-&DCBNAME,DSORG=PS,MACRF=PM)
ISREDIT LINE_AFTER &DEST - DATALINE "+"
ISREDIT LINE_AFTER &EVAL(&DEST+1) - DATALINE "&LINE"
SET &LINE - &STR(&DCBNAME,L)
SET &LINE = &STR(&SUBSTR(1:9,&LINE)&STR(EQU *-&DCBNAME))
ISREDIT LINE_AFTER &EVAL(&DEST+2) - DATALINE "&LINE"
EXIT CODE(0)

ACMD EDIT MACRO
ISREDIT MACRO (MEMBER)
IF &MEMBER = ? THEN DO
  ISPEXEC DISPLAY PANEL(ACMD)
  EXIT
END
SET &DEFAULT = &STR(ACMDSKEL)
IF &MEMBER = &STR() THEN SET &DEFAULT = &STR(&MEMBER)
ISREDIT COPY &DEFAULT AFTER .ZFIRST
IF &LASTCC = 0 THEN DO
  SET ZEDSMSG = &STR(&DEFAULT NOT FOUND)
  SET ZEDLMSG = &STR(MEMBER &DEFAULT CANNOT BE FOUND IN PDS)
  ISPEXEC SETMSG MSG(ISRZ001)
  EXIT
END
ISREDIT (PROG) - MEMBER
ISREDIT CHANGE @@@@@@@@ &PROG

ACSA EDIT MACRO

PROC Ø DEBUG
ISREDIT MACRO (SUBNAME PREFIX DEBUG) NOPROCESS
  IF &SUBNAME = ? THEN DO
    ISPEXEC DISPLAY PANEL(ACSA)
  EXIT
END
DO WHILE &LENGTH(&STR(&STARS)) LT 65
  SET &STARS = &STR(&STR(&STARS)&STR(*))
  SET &SPACES = &STR(&STR(&SPACES)&STR( ))
END
SET &CSACOM = &STR( CSA&SUBSTR(1:58,&SPACES))
IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
ISREDIT PROCESS DEST
IF &LASTCC = 0 THEN +
  DO
    ISREDIT FIND FIRST "* END DSECT" 1
  IF &LASTCC = 0 THEN +
    DO
      SET ZEDSMSG = &STR(COMMENT COMMAND PENDING)
      SET ZEDLMSG = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
                       NO '* END DSECT' CONSTANT)
      ISPEXEC SETMSG MSG(ISRZ001)
      EXIT CODE(12)
  END
© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
ELSE +
    DO
        ISREDIT (DEST) = CURSOR
        SET DEST = &EVAL(&DEST-2)
        END
    END
    ELSE +
    ISREDIT (DEST) = LINENUM .ZDEST
    SET &EXEC = &STR( EXEC CICS ADDRESS CSA(CSAREG)&SPACES)
    SET &REG = &STR(CSAREG EQU R9&SPACES)
    SET &USNG = &STR( USING DFHCSADS.CSAREG&SPACES)
    ISREDIT LINE_AFTER &DEST = DATALINE "**"
    ISREDIT LINE_AFTER &EVAL(&DEST+1) = DATALINE " EJECT"
    ISREDIT LINE_AFTER &EVAL(&DEST+2) = DATALINE "***&STARS.***"
    ISREDIT LINE_AFTER &EVAL(&DEST+3) = DATALINE "***&SPACES.***"
    ISREDIT LINE_AFTER &EVAL(&DEST+4) = DATALINE "***&CSACOM.***"
    ISREDIT LINE_AFTER &EVAL(&DEST+5) = DATALINE "***&SPACES.***"
    ISREDIT LINE_AFTER &EVAL(&DEST+6) = DATALINE "***&STARS.***"
    ISREDIT LINE_AFTER &EVAL(&DEST+7) = DATALINE "*"
    ISREDIT LINE_AFTER &EVAL(&DEST+8) = DATALINE " COPY DFHCSADS"
    ISREDIT LINE_AFTER &EVAL(&DEST+9) = DATALINE "*"
    ISREDIT FIND FIRST " * END ADDRESS" 1
    ISREDIT (LINEX) = CURSOR
    ISREDIT LINE_AFTER &EVAL(&LINEX-2) = DATALINE "&EXEC"
    ISREDIT LINE_AFTER &EVAL(&LINEX-1) = DATALINE "&REG"
    ISREDIT LINE_AFTER &EVAL(&LINEX) = DATALINE "&USNG"
    ISREDIT LINE_AFTER &EVAL(&LINEX+1) = DATALINE "*
    EXIT CODE(0)

ATCA EDIT MACRO

PROC Ø DEBUG
    ISREDIT MACRO (SUBNAME PREFIX DEBUG) NOPROCESS
    IF &SUBNAME = ? THEN DO
        ISPSEXEC DISPLAY PANEL(ATCA)
        EXIT
    END
    DO WHILE &LENGTH(&STR(&STARS)) LT 65
        SET &STARS = &STR(&STR(&STARS)&STR(*))
    SET &SPACES = &STR(&STR(&SPACES)&STR( ))
    END
    SET &TCACOM = &STR( T C A&SUBSTR(1:58,&SPACES))
    IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
    ISREDIT (RETX) = CURSOR
    ISREDIT PROCESS DEST
    IF &LASTCC ≠ Ø THEN +
        DO
            ISREDIT FIND FIRST " * END DSECTS" 1
            IF &LASTCC ≠ Ø THEN +
                DO
                    SET ZEDSMMSG = &STR(COMMENT COMMAND PENDING)
SET ZEDLM = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
NO ' ' END DSECT' CONSTANT)
ISPEXEC SETMSG MSG(ISRZ001)
EXIT CODE (12)
ELSE +
DO
ISREDIT (DEST) = CURSOR
SET DEST = &EVAL(&DEST-2)
END
END
ELSE +
ISREDIT (DEST) = LINENUM.ZDEST
SET &MAC = &STR(
   DFHTCA CICSsTYC-CONFIG&SPACES)
ISREDIT LINE_AFTER &DEST = DATALINE 
ISREDIT LINE_AFTER &DEST-1 = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+1) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+1) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+2) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+3) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+4) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+5) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+6) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+7) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+8) = DATALINE
ISREDIT LINE AFTER &EVAL(&DEST+9) = DATALINE
ISREDIT LOCATE &RETX
SET &EXEC = &STR(
   L TCACBAR,CSACDTA-DFHCSADS(CSAREG))
SET &USNG = &STR(USING DFHCSADS,CAREG&SPACES)
ISREDIT (LINEX) = CURSOR
ISREDIT LINE_AFTER &EVAL(&LINEX) = DATALINE
ISREDIT LINE_AFTER &EVAL(&LINEX+1) = DATALINE
ISREDIT LINE_AFTER &EVAL(&LINEX+2) = DATALINE
EXIT CODE (0)

ATWA EDIT MACRO

PROC Ø DEBUG
ISREDIT MACRO (SUBNAME PREFIX DEBUG) NOPROCESS
IF &SUBNAME = ? THEN DO
ISPEXEC DISPLAY PANEL(ATWA)
EXIT
END
DO WHILE &LENGTH(&STR(&STARS)) LT 65
SET &STARS = &STR(&STR(&STARS)&STR(*))
SET &SPACES = &STR(&STR(&SPACES)&STR( ))
END
SET &TACOM = &STR( T W A&SUBSTR(1:58,&SPACES))
IF &DEBUG = DEBUG THEN CONTROL LIST SYMLIST CONLIST
ISREDIT PROCESS DEST
IF &LASTCC ^= Ø THEN +

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
DO
ISREDIT FIND FIRST "* END DSECTS" 1
IF &LASTCC → Ø THEN +
   DO
      SET ZEDSMG = &STR(COMMENT COMMAND PENDING)
      SET ZEDLMSG = &STR(ENTER AN 'A' OR 'B' LINE COMMAND +
                             NO '* END DSECT' CONSTANT)
      ISPEXEC SETMSG MSG(ISRZ001)
      EXIT CODE(12)
   END
ELSE +
   ISREDIT (DEST) = CURSOR
   SET DEST = &EVAL(&DEST-2)
   END
ELSE +
   ISREDIT (DEST) = LINENUM .ZDEST
   SET &EXEC = &STR(
               EXEC CICS ADDRESS TWA(TWAREG)&SPACES)
   SET &REG = &STR(TWAREG EQU R13&SPACES)
   SET &USNG = &STR( USING TWADS,TWAREG&SPACES)
   ISREDIT
   LINE AFT &DEST = DATALINE "**"
   ISREDIT LINE AFT &EVAL(&DEST+1) = DATALINE "  EJECT"
   ISREDIT LINE AFT &EVAL(&DEST+2) = DATALINE "***&STARS.***"
   ISREDIT LINE AFT &EVAL(&DEST+3) = DATALINE "***&SPACES.***"
   ISREDIT LINE AFT &EVAL(&DEST+4) = DATALINE "***&TWACOM.***"
   ISREDIT LINE AFT &EVAL(&DEST+5) = DATALINE "***&SPACES.***"
   ISREDIT LINE AFT &EVAL(&DEST+6) = DATALINE "***&STARS.***"
   ISREDIT LINE AFT &EVAL(&DEST+7) = DATALINE "*
   ISREDIT LINE AFT &EVAL(&DEST+8) = DATALINE "TWADS DSECT"
   ISREDIT LINE AFT &EVAL(&DEST+9) = DATALINE "TWA DS ØC"
   ISREDIT LINE AFT &EVAL(&DEST+10) = DATALINE "*
   ISREDIT FIND FIRST "* END ADDRESS" 1
   ISREDIT (LINEX) = CURSOR
   ISREDIT LINE AFT &EVAL(&LINEX-2) = DATALINE "&EXEC"
   ISREDIT LINE AFT &EVAL(&LINEX-1) = DATALINE "&REG"
   ISREDIT LINE AFT &EVAL(&LINEX) = DATALINE "&USNG"
   ISREDIT LINE AFT &EVAL(&LINEX+1) = DATALINE "*
   EXIT CODE(Ø)

Keith H Nicaise
Technical Services Manager
Touro Infirmary (USA)

© Xephon 1997
Useful Assembler macros – part 2

We continue our look at the following Assembler macros: EXTR, BSM24, BSM31.

EXITR MACRO
* EXTR RETURNS TO CALLER (IT MUST BE USED TOGETHER WITH MACRO INITR):
* RESTORES REGISTERS;
* FREEMAINS GETMAINED SAVEAREA;
* SETS RETURNCODE, DEFAULTS TO ZERO; USE ABSOLUTE OR REGISTER NOTATION
* WHEN EXITR IS CALLED, R13 MUST POINT TO THE SAVE AREA GETMAINED FROM
* MACRO INITR.
* *
* UNDER MVS/370 RETURN WILL BE DONE VIA BRANCH ON R14
* CODE FOR SUPPORT OF NON-XA (MVS/370) WILL ONLY BE GENERATED IF
* GLOBAL VARIABLE &MVS370-SUP OR &SYSSPLV=1.
* CODE FOR SUPPORT OF XA/ESA WILL ONLY BE GENERATED IF &SYSSPLV > 1
* UNDER MVS/XA/ESA RETURN WILL BE DONE USING BSM 0,R14 TO RESTORE
* CALLERS ADDRESSING MODE; WHATEVER THE SUBROUTINE WAS CALLED BY
* BALR R14,R15 OR BY BASSM R14,R15, R14 WILL CONTAIN CALLER'S
* ADDRESSING MODE. NOTE THAT UNDER MVS/XA/ESA THE EXITR REQUIRES THIS
* KIND OF CALL TECHNIQUE; IF THE SUBROUTINE IS LINKED BY EG
* LA R14,RETURN , L R15,ADDRSUBR AND BR R15 WHERE ADDRSUBR IS A 31-
* BIT ADDRESS MISSING MODE BIT INDICATION IN BIT 0, THE EXITR WILL
* RETURN IN 24 BIT MODE ALTHOUGH IT SHOULD RETURN IN 31-BIT MODE.
* TO FORCE A BRANCH RETURN AVOIDING BSM MODE CHANGE UNDER XA/ESA USE
* PARAMETER BRANCH-YES.
* STANDARD RETURN REGISTER IS 14, BUT A DIFFERENT RETURN REGISTER
* CAN BE REQUESTED VIA THE PARAMETER RETREG.
* Generates additional support code as explained in INITR MACRO IF
* THE INITR MACRO IS INVOKED WITH PARAMETER GENCODE=YES.
* IF PARAMETER EXIT IS SET TO A VALUE, A TERMINATION CLEAN UP ROUTINE
* WITH ENTRY-LABEL EQUAL TO THE VALUE INDICATED IN THIS PARAMETER
* WILL BE INVOKED WITH A BAL RI5,EXITNAME.
* THE FOLLOWING BRANCH LABELS WILL BE EXTERNALLY AVAILABLE WHEN
* GENCODE-YES IS USED:
* EXIT: NORMAL EXIT WITH RC-VALUE IN FIELD RETCODE
* QUICKOUT: EXIT WITH RC=0 BUT WITHOUT INACTIVATING ESTAE
* EXITRC4: EXIT WITH RC=4
* EXITRCB: EXIT WITH RC=B
* EXITRC12: EXIT WITH RC=12
* EXITRC16: EXIT WITH RC=16
* *
&NAME EXITR &RC=0, &EXIT=.

© 1997. Xophon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
&BRANCH-NO.
&RETREG-14

&NAME
* DS ØH

MHELP 2

GBLC &MSIZE FROM INITR
GBLC &GETPOOL FROM INITR
GBLC &MVS370S FROM INITR
GBLC &SYSSPLV MACRO LEVEL
GBLC &EXITR FROM EXITR
GBLC &GENCO FROM INITR
GBLC &XLATEF FROM INITR
GBLC &ID FROM INITR
GBLA &IDLEN FROM INITR
GBLC &ESTALST FROM EXITR
GBLC &ESTAEND FROM EXITR
GBLC &STAXLST FROM EXITR
GBLC &STAXEND FROM EXITR
GBLC &RETRYR1 FROM EXITR
GBLC &RETRYR2 FROM EXITR
GBLC &SECBS FROM INITR
GBLC &TERBS FROM INITR
GBLC &QARBS FROM INITR
GBLC &TRLATE FROM EXITR
GBLC &TRTAB FROM EXITR
GBLC &ABRET FROM EXITR
GBLC &ESTAER FROM INITR
GBLC &STAXR FROM INITR
GBLC &TSTAUT FROM INITR
LCLC &NONXA

SLEVEL TEST SET SYSSPLV

&NONXA SETC 'EX1'.&SYSNDX'
&XABR SETC 'EX2'.&SYSNDX'
&HASDWA SETC 'EX3'.&SYSNDX'
&STAXBS SETC 'EX4'.&SYSNDX'
&STAXOS SETC 'EX5'.&SYSNDX'
&NTSTAT SETC 'EX6'.&SYSNDX'
&NOAPFON SETC 'EX7'.&SYSNDX'
&BYAPFON SETC 'EX8'.&SYSNDX'
&NOSUPON SETC 'EX9'.&SYSNDX'
&BYSUPON SETC 'EX6'.&SYSNDX'
&BYSUNAF SETC 'EX7'.&SYSNDX'
&RECOVRR SETC 'EX6'.&SYSNDX'
&STAXIT SETC 'EX6'.&SYSNDX'
AIF ('&GENCO' EQ 'NO').NOGENCO
AIF ('&EXITR' EQ 'ONEXITRGENCO').NOGENCO

&EXITR SETC 'ONEXITRGENCO'
B EXIT NORMAL EXIT

EXITRC4 DS ØH
LA R15.4 GET RC 4
ST R15,RETCODE SET RETCODE
EXITRC8 DS @H.
LA R15,8
ST R15.RETCODE
B EXIT
GO EXIT
EXITRC12 DS @H.
LA R15,12
ST R15.RETCODE
B EXIT
GO EXIT
EXITRC16 DS @H.
LA R15,16
ST R15.RETCODE
B EXIT
GO EXIT
AIF ('&ESTAE'R EQ 'NO').NOESTA1

* ESTAE EXIT ROUTINE
&RECOVRR DS @H.
PUSH USING
USING *,R15
USING SDWA,R1
LA R4,12
CR R0,R4
BNE &HASOWA
R0.0(R2)

SAVE PREVIOUS BASE REGS
SET UP BASE REGISTER
SET UP ADDRESSABILITY TO SDWA
PUT 12 IN REGISTER FOR COMPARE
IS SDWA PRESENT?
LOAD RETRY ADDR FROM PARM LIST
SET RC TO RETRY ADDR IN R0
RETURN WITH RETRY ADDR

&HASDWA DS @H.
Enter here if SDWA present
ST R14,12(R13)
SAVE RETURN ADDRESS
L R2,SDWAPARM
LOAD PARAM LIST ADDR FROM SDWA
ST R2,SDWASR01
SAVE POINTER TO ESTAE PARM LIST
L R2,4(R2)
LOAD RETRY ADDRESS

SETRP RC=4,.RETTADDR=(2).RETREGS= YES.FRESDWA= YES.REGS=(14)
DROP R15,R1
DROP LOCAL ADDRESSABILITY
POP USING
RESTORE PREVIOUS BASE REGS

* &RETRYR1 DS @H.
RETRY ROUTINE WITH NO SDWA
&RETRYR2 DS @H.
ESTAE RETRY ROUTINE WITH SDWA
LM R12,R13,8(R1)
LOAD REGS FOR ESTAE PARM LIST
AIF ('&SECBS' EQ '0').ESTNSEC
L &SECBS,8+8(R1)
LOAD SECONDARY BASE IN PARM

ESTNSEC ANOP
AIF ('&TERBS' EQ '0').ESTNTER
L &TERBS,8+12(R1)
LOAD TERTIARY BASE IN PARM

ESTNTER ANOP
AIF ('&QARBS' EQ '0').ESTNQAR
L &QARBs,8+16(R1)
LOAD QUARTERNARY BASE IN PARM

ESTNQAR ANOP
LA R15,6ABRET
SET SEVERE ERROR
ST R15.RETCODE
INDICATE SEVERE ERROR
B QUICKOUT
AND EXIT

NOESTA1 ANOP

© 1997. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
AIF ('&STAXR' EQ 'NO').NOSTAX1

* STAX ATTENTION EXIT

&STAXIT DS 0H.
PUSH USING          SAVE PREVIOUS BASE REGS
USING * ,R15       ADDRESS TEMPORARILY
SAVE (14,12)       SAVE REGS
BALR R12,0          SET UP BASE

&STAXBS DS 0H.
L R15,&STAXOS      SET UP BASE OFFSET
SR R12,R15         SET UP REAL BASE
DROP R15            LEAVE TEMPORARY ADDRESSING

* CLEAN UP WHAT NEED TO

DROP R13            LEAVE ADDRESSING WORKAREA
USING WORKAREA,R9   ADDRESS WORKAREA
L R9,B(R1)          GET USER DATA
OI OPTIONS,ATTN    SET ATTN FLAG
DROP R9             LEAVE LOCAL ADDR TO WORKAREA
USING WORKAREA,R13  ADDRESS WORKAREA NORMALLY AGAIN
RETURN (14,12),RC=8  RETURN
POP USING          RESTORE PREVIOUS BASE REGS

&STAXOS DC A(&STAXBS-&ID)  STAX BASE OFFSET

NOSTAX1 ANOP

AIF ('&ESTAER' EQ 'NO').NOESTA2

&ESTALST ESTAE &RECOVRR,MF=L CREATE MODEL ESTAE PARM LIST
&ESTAEND EQU * NAME ITS END

NOESTA2 ANOP

AIF ('&STAXR' EQ 'NO').NOSTAX2

&STAXLIST STAX &STAXIT,MF=L  STAX LIST FORM
&STAXEND EQU * NAME ITS END

NOSTAX2 ANOP

AIF ('&XLATEF' EQ 'NO').NOXLATE

&TRLATE TR 0(0,R14),&TRTAB EXECUTED TRANSLATE INSTRUCTION

&TRTAB DC 256AL1(*-&TRTAB) UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C::'  UPPERCASE TRANSLATE TABLE NATIONAL CHAR
DC C'@'         UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C:'#'  UPPERCASE TRANSLATE TABLE NATIONAL CHAR
DC C'#'         UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C:')'  UPPERCASE TRANSLATE TABLE NATIONAL CHAR
DC C'.'         UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C'a'  UPPERCASE TRANSLATE TABLE
DC C'ABCDEFHGI'  UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C'j'  UPPERCASE TRANSLATE TABLE
DC C'JKLMNOPQR'  UPPERCASE TRANSLATE TABLE
ORG &TRTAB+C's'  UPPERCASE TRANSLATE TABLE
DC C'STUVWXYZ'  UPPERCASE TRANSLATE TABLE

NOXLATE ANOP

*

EXIT DS 0H.

AIF ('&ESTAER' EQ 'NO').NOESTA3

NOESTA3 ANOP
QUICKOUT DS "0H.

NOGENCO ANOP
AIF ('&GENCO' NE 'YES').GENCOR
AIF ('&EXIT' EQ '').NOEXIT
BAL R15,&EXIT

NOEXIT ANOP
AIF ('&TSTAUT' EQ 'NO').NOTSTAT
TESTAUTH KEY=NOSTATE=YES,RLLEVEL=1,BRANCH=YES TEST FOR STATE
TM OPTIONR,SUPVSTAT IN SUPERVISOR STATE ON AT ENTRY
BZ &NOSUPON NOT SUPERVISOR STATE AT ENTRY
LTR R15,R15 TEST FOR SUPERVISOR STATE
BZ &BYSUPON IS ALREADY IN SUPERVISOR STATE
TESTAUTH FCTN=1,KEY=NO,RLLEVEL=1,BRANCH=YES TEST FOR AUTH
LTR R15,R15 TEST FOR SUPERVISOR STATE
BZ &BYSUNAF IS ALREADY AUTH FOR MODESET
AUTHON BRANCH=YES TURN ON APF FOR MODESET

&BYSUNAF DS "0H.
MODESET MODE=SUP RETURN TO SUPERVISOR STATE
B &BYSUPON PROCEED

&NOSUPON DS "0H.
LTR R15,R15 TEST FOR SUPERVISOR STATE
BNZ &BYSUPON IS NOT IN SUPERVISOR STATE
MODESET MODE=PROB RETURN TO PROBLEM STATE

&BYSUPON DS "0H.
TM OPTIONR,APFON WAS APF ON AT ENTRY
BZ &NOAPFON NOT APF AT ENTRY
AUTHON BRANCH=YES ENSURE APF IS ON AT EXIT
B &BYAPFON PROCEED

&NOAPFON DS "0H.
AUTHOFF BRANCH=YES ENSURE APF IS OFF AT EXIT

&BYAPFON DS "0H.

NOTSTAT ANOP
L 14,RETCODE GET RETURN CODE
MNOTE 0, 'RETURN CODE WILL BE TAKEN FROM FIELD RETCODE ONLY'

GENCOR ANOP
LR 1,13 . SET UP FOR FREEMAIN
L 13,4(13) . R13 -> PREV SAVEAREA
AIF ('&RETREG' EQ '15').RETR15
AIF ('&RETREG' EQ '15').RETR15
AIF ('&RETREG' EQ '14').RETR14
AIF ('&RETREG' EQ '14').RETR14
AIF ('&RETREG' EQ '13').RETR13
AIF ('&RETREG' EQ '13').RETR13
ST &RETREG.20+4*&RETREG.(13) SET RETURN REG IN PREV SAVE

RETR14 ANOP
AIF ('&GENCO' EQ 'YES').EXITRE
AIF ('&RC'(1,1) EQ '(').RCRET
LA 14,&RC . SET RETURN CODE
AGO .EXITRE
RCRET ANOP
LR 14,&RC(1) . SAVE RETURN CODE
AGO .EXITRE
RETR15 ANOP
LR 14,15 . SAVE RETURN REGISTER
AIF ('&GENCO' EQ 'YES').EXITRE
AIF ('&RC' EQ '0').EXITRE
MNOTE B,'RETURN CODE WHILE RETURNING ON R15 CANNOT BE SET'
RETR13 ANOP
AIF ('&GENCO' EQ 'YES').EXITRE
MNOTE B,'RETURN REGISTER CANNOT BE &RETREG'
EXITRE ANOP
L 0,&MSIZE . SET UP FOR FREEMAIN
AIF ('&MVS370S' EQ 'NOTSUP').BYPNON1
AIF ('&SYSSPLV' LT '2').NONXA BYPASS IF NOT XA/ESA MACLEVEL
TESTXA (15) . FIND OUT WHICH MODE
LTR 15,15 . TEST MODE
BP &NONXA . THEN NON XA MODE
BYPNON1 ANOP
FREEMAIN RU,LV=(0),A=(1),SP=&GETPOOL . FREEMAIN SAVEAREA
AIF ('&BRANCH' EQ 'YES').USEBR1
AIF ('&BRANCH' EQ 'NO').BSMBR
MNOTE B,'BRANCH MUST BE EITHER YES OR NO'
USEBR1 ANOP
AIF ('&MVS370S' EQ 'NOTSUP').USEBR2
B &XABR . RETURN VIA BRANCH
AGO .NONXA
BSMBR ANOP
LR 15,14 . SET RETURN CODE
L 14,12(13) . RESTORE R14
LM 0,12,20(13) . RESTORE R0 TO R12
BSM 0,&RETREG . BRANCH BACK TO CALLER
NONXA ANOP
AIF ('&MVS370S' EQ 'NOTSUP').BYPNON2
&NONXA DS 0H .
LA 15,&GETPOOL . INDICATE SUBPOOL NO
SLL 15,24 . INDICATE SUBPOOL NO
OR 0,15 . SET UP FOR FREEMAIN
FREEMAIN R,LV=(0),A=(1) . FREEMAIN SAVEAREA
BYPNON2 ANOP
AIF ('&MVS370S' EQ 'SUP').USEBR2
AIF ('&BRANCH' EQ 'YES').USEBR2
MEXIT
USEBR2 ANOP
&XABR DS 0H .
LR 15,14 . SET RETURN CODE
L 14,12(13) . RESTORE R14
LM 0,12,20(13) . RESTORE R0 TO R12
BSM24 MACRO

* 
* SET ADDRESSING MODE TO 24 BIT IF RUNNING UNDER XA/ESA 
* NEUTRAL UNDER MVS/370 
* 
* USES WORK REGISTER, DEFAULT TO R15 
* WORKREGISTER CAN BE OVERWRITTEN BY BSM (RX) 
* WORK REG CONTAINS ADDR OF NEXT INSTR AND ADDR MODE (24) 
* 
* CODE FOR SUPPORT OF NON-XA (MVS/370) WILL ONLY BE GENERATED IF 
* GLOBAL VARIABLE FROM INITR &MVS370S-SUP IS SPECIFIED OR &SLEVEL=1: 
* IF MACRO INITR IS NOT USED AND &SLEVEL > 1, IT IS STILL POSSIBLE 
* TO FORCE GENERATION OF MVS/370 VIA THE PARAMETER MVS370-SUP. 
* CODE FOR SUPPORT OF XA/ESA WILL ONLY BE GENERATED IF &SLEVEL > 1. 
* 
* MACRO
&NAME BSM24 &REG,&MVS370-NOTSUP 
GBLC &MVS370S COMES FROM INITR IF THIS MACRO IS USED 
GBLC &SYSSPLV MACRO LEVEL 
SPLEVEL TEST SET SYSSPLV 
LCLC &NONXA 
&NONXA SETC 'B24'. '&SYSNDX'. 
AIF ('&MVS370S' NE '').XASUPP 
&MVS370S SETC '&MVS370' . SET ONLY FROM PARAMETER IF INITR IS NOT USED 
INTSUPP ANOP 
AIF ('&MVS370S' EQ 'NOTSUP').SUPP 
AIF ('&MVS370S' EQ 'SUP').SUPP 
MNOTE 8. 'MVS370 MUST BE INDICATED AS NOTSUP OR SUP' 
MEXIT 
SUPP ANOP 
AIF ('&SYSSPLV' GT 'I').XASUPP XA-MACRO LEVEL 
&MVS370S SETC 'SUP' FORCE MVS370 SUPPORT 
XASUPP ANOP 
AIF ('&REG' EQ '').RNULL 
AIF ('&REG'(1,1) EQ '') .AREG 
AGO .RNULL 
AREG ANOP 
&REGR SETC '&REG(1)' 
AGO .REG 
RNULL ANOP 
&REGR SETC '15' 
REG ANOP 
&NAME DS 0H . 
AIF ('&MVS370S' EQ 'NOTSUP').XA 
AIF ('&SYSSPLV' LT '2').NONXA BYPASS IF NOT XA/ESA MACLEVEL
Suggested articles for *MVS Update*

From time to time, subscribers contact us suggesting subjects for articles they would like to see covered in future issues of *MVS Update*. Therefore, partly to inspire prospective authors and partly to see if anyone already has some existing material that might be appropriate, here is a list of subjects that readers have shown an interest in:

- User experiences with Workload Manager
- Open Edition/MVS
- Parallel Processing
- The year 2000 and MVS
- MVS internals
- MVS security
- Performance
- Tuning.

If you are interested in contributing an article, and would like further details, or if there is an area that you would like to see covered in a future issue of *MVS Update*, please contact the editor, Jaime Kaminski, on +44 1635 33598 (telephone), or 106006.1540@compuserve.com (e-mail).
MVS news

Advanced Software Technologies Company (ASTCO) has released version 3.3 of ASTUTE, a dataset and catalog management system for MVS or OS/390. Features in version 3.3 include year 2000 compliance, support for four-digit UCB device numbers and improvements to ASTUTE’s DASD Management Language.

For further information contact:
Advanced Software Technologies Company Ltd, 113 N. Washington Street, Suite 202, PO Box 10826, Rockville, MD 20850, USA
Tel: (301) 424 9455
Fax: (301) 294 8584.

***

Compute (Bridgend) Ltd has announced release 9.8 of SELCOPY, its year 2000 compliant file manipulation utility. A major enhancement is support for DB2 processing using Dynamic SQL, allowing the user to define run-time SQL statements. Compute has also announced release 9.8 of CBL-VCAT, its ICF catalog tuning/display utility, which in addition to being year 2000 compliant, supports CSA storage requests from above the 16 MB line, variable-length RRDS (VRRDS) and local timestamp reporting.

For further information contact:
Compute (Bridgend) Ltd, 8 Merthyr Mawr Road, Bridgend, Mid-Glamorgan, CF31 3NH, UK
Tel: (01656) 652222
Fax: (01656) 652227 or
Compute (Bridgend) Ltd, 38 Guided Court, Rexdale, Ontario, Canada, M9V 4K6
Tel: (416) 746 4447
Fax: (416) 746 5870.

***

IBM has announced version three of its ADSTAR Distributed Storage Manager (ADSM) for MVS. Major enhancements include: optimized back-up and restore performance through intelligent and adaptive processing; a new server-to-server feature which allows data to be shared among multiple ADSM servers. The user interfaces for ADSM Version 3 are redesigned to allow quick navigation through large file systems. In addition, a new Web-based administrative interface allows ADSM control and operation from an intranet.

Contact your local IBM marketing representative for further information.

***