



185

CICS

April 2001

In this issue

- 3 Display the DFHVALUE CVDA
- 11 Update on JCHWBOS for CTS 1.3
- 12 Simple tool to manage the data extracted from CICS CSD in a DB2 environment – part 2
- 32 Utility to generate CICS maps
- 48 CICS news

© Xephon plc 2001

update

CICS Update

Published by

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

North American office

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

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; \$270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 (\$23.50) each including postage.

***CICS Update* on-line**

Code from *CICS Update* can be downloaded from our Web site at <http://www.xephon.com/cicsupdate.html>; you will need to supply a word from the printed issue.

Editor

Trevor Eddolls

Disclaimer

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

Contributions

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

© Xephon plc 2001. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

Printed in England.

Display the DFHVALUE CVDA

DESCRIPTION

With this application, you are able to display the DFHVALUE CVDA to a value of XREF. You can use any CICS region, and you must only type the transaction name and the value *or* the CVDA as shown in Figures 1 and 2. That way you always get results like those shown in Figure 3. You must add the transaction DFHV to your PCT and the program CSDFHVAL to your PPT, and then compile CSDFHVAL as a command-level program.

CSDFHVAL

```
*ASM XOPTS(CICS,SP)
*****
* PROGRAMNAME : CSDFHVAL *
* FUNCTION : THIS IS A SIMPLE PROGRAM TO DISPLAY THE DFHVALUE *
* CVDA TO VALUE XREF. *
* THE INPUT CAN BE EITHER *
* TRANSACTION: DFHV VALUE *
* TRANSACTION: DFHV CVDA *
* SUPPORTED CICS VERSIONS: *
* *
* CTS 1.3 *
* *
* THE TERMINAL OUTPUT LOOKS LIKE : *
*
```

```
17/01 01 MI 16:18 FAX +49 911 531 4733 Benutzerservice 001

dfhv cicsdatakey
```

Figure 1: Entering transaction name and value

17/01 01 MI 16:18 FAX +49 911 531 4733 Benutzerservice 001

dfhv 379

Figure 2: Entering CVDA

```
*          DFHVALUE VALUE CVDA X-REFERENCE          *
* -----*
* CICSVER=CICSXXX  SYSID=YYYY  TRANSID=TTTT  PROGRAM=PPPPPPP  *
* -----*
* VALUE          CVDA          *
* 0001          NOTAPPLIC      *
*
* VALUE          = HEX FULLWORD  *
* CVDA           = CHARACTER NAME *
*****
* EXPAND THE DFHEISTG FOR THE REQUIRED USER FIELDS          *
*****
      SPACE
DFHEISTG DSECT
LEN      DS      H
DC       DS      H
HELP     DS      CL4
HELP1    DS      CL4
ALPHAFLD DS      CL12
ADDRESS  DS      D
BINFLD   DS      F
CVBIN    DS      D
PACKIN   DS      F
PACKOUT  DS      F
INPUT    DS      CL20
      SPACE
*****
* THE TERMINAL PRESENTATION OUTPUT AREA BEGINS HERE          *
*****
      SPACE
OUT      DS      0CL312
      SPACE
*****
      HEADER LINE
      SPACE
```

```
DFHVALUE VALUE CVDA X-REFERENCE
-----
CICSVER=CICS530 SYSID=ST01 TRANSID=DFHV PROGRAM=CSDFHVAL
-----
VALUE      CVDA

0379      CICSDATAKEY
```

Figure 3: Example output

```
HDRMSG  DS   XL79
NL       DS   XL1
FILL0   DS   XL62
NL0     DS   XL1
        SPACE
***** ENVIRONMENT LINE
        SPACE
CICSVER DS   CL8
CICS    DS   CL7
SYSIDENT DS  CL8
SYSID   DS   CL4
TRANID  DS   CL10
TRANSID DS   CL4
PROGNA  DS   CL10
PROGRAM DS   CL8
NL1     DS   XL1
        SPACE
***** STARTER STARS
        SPACE
FILL1   DS   XL62
NL2     DS   XL1
        SPACE
***** 1ST LINE OF DATA OUTPUT
        SPACE
VALUE   DS   CL8
FILL2   DS   CL4
CVDA    DS   CL8
NL3     DS   XL1
NL4     DS   XL1
VAL     DS   CL4
```

```

FILL3    DS    CL8
CVDAVAL  DS    CL12
        SPACE
*****
*      THE OUTPUT AREA ENDS HERE                                *
*****
        EJECT
*****
*      HERE BEGINS THE CSECT ITSELF                              *
*****
        SPACE
CSDFHVAL CSECT
CSDFHVAL AMODE 31
CSDFHVAL RMODE ANY
        B      START
        SPACE
PROGNAME DC    CL8'DFHVALUE'          SET
        DC    CL8'&SYSDATE'          EYECATCHER
        DC    CL8'&SYSTIME'          INFORMATION
        SPACE
BEGIN    EQU   *
        SPACE
*****
*      CICS BASIC CODING                                        *
*****
        SPACE
CTS13   EQU   *
        L     12,X'21C'              ADDRESS CURRENT TCB
        L     12,X'D0'(,12)          ADDRESS TCB EXTN
        L     12,X'14'(,12)          ADDRESS AFCX
        L     12,X'08'(,12)          ADDRESS THE CSA
        CLI   X'9F'(12),X'53'        CICS 5.3.0 ?
        BNE   ERROR1                 CICS VERSION/RELEASE NOT SUPPORTED
        MVC   CICS,=C'CICS530'       MOVE VER/REL INTO OUTPUT FIELD
        SPACE
*****
*      PROCESS TERMINAL INPUT                                    *
*****
        SPACE
        LA    5,BINFLD
        MVC   0(16,5),=X'00'         CLEAR AREA
        LA    5,LEN
        MVC   0(2,5),=X'0014'        SET RECEIVE LENGTH
        LA    5,INPUT
        MVC   0(20,5),=X'40'         CLEAR INPUT AREA
        SPACE
        EXEC  CICS RECEIVE INTO(INPUT) LENGTH(LEN)
        SPACE
* -----*
*      INPUT AREA SHOULD BE

```

```

*          XACT VALUE OR CVDA
*          Ø    5 -----> 17
* -----*
          SPACE
          LH    6,LEN                LOAD REAL RECEIVE LENGTH
          LH    7,PREFIX            LOAD PREFIX LENGTH
          SR    6,7                  GET CORRECT DATA LENGTH
          STH   6,LEN                AND STORE IT BACK
          SPACE
* -----*
* CHECK IF 1ST CHARACTER BLANK
* -----*
          SPACE
          CLI   5(5),X'40'          CHECK IF 1ST CHARACTER BLANK
          BE ERROR1                 WRONG INPUT
          LA    7,5(5)              POINT 1ST BYTE OF INPUT DATA
          SPACE
INPUTEST EQU *
          CLC   Ø(1,7),=X'F9'      HIGHER THAN Z
          BH    ERROR1             WRONG INPUT
          CLC   Ø(1,7),=X'C1'      LOWER THAN A
          BL    ERROR1             WRONG INPUT
          SPACE
* -----*
* ALPHAINPUT TEST
* -----*
          SPACE
ALPHA EQU *
          CLC   Ø(1,7),=X'E9'      HIGHER THAN Z
          BH    NUM                 MUST BE NUMERIC
          LH    6,LEN               RELOAD INPUT LENGTH
          LA    5,5(5)              POINT TO INPUT DATA
          LA    7,ALPHAFLD          POINT TO OUTPUT FIELD
          BCTR  6,Ø                 SUBTRACT ONE FROM LENGTH
          EX    6,TEXTMOVE          MOVE CVDA TO FIELD
          B     SEARCH              GOTO SEARCH
          SPACE
NUM EQU *
          LA    5,INPUT             POINT TO INPUT DATA
          LA    7,5(5)              POINT 1ST BYTE OF INPUT DATA
          SPACE
* -----*
* MOVE UNPACK AND CONVERT FOR SEARCH
* -----*
          SPACE
MOUNCO EQU *
          LH    6,LEN                LOAD REAL RECEIVE LENGTH
          LA    8,HELP1             POINT TO HELP1 FIELD
          LA    4,HELP              POINT TO HELP FIELD
          AR    4,6                  ADD LENGTH TO HELP TO GET THE

```

```

*
      BCTR  6,0          CORRECT OFFSET FOR NUMLOOP MOVE
      EX    6,NUMMOVE   SUBTRACT ONE FROM LENGTH
                        MOVE NUMERIC VALUE
      SPACE
      EXEC CICS DELAY FOR SECONDS(SECOND)
      SPACE
      LR    7,4          LOAD CORRECT MOVE REG
      LA    6,X'4'      LOAD MOVE LENGTH
      SPACE
NUMLOOP EQU *
      LA    8,PACKOUT   POINT TO PACKOUT FIELD
      BCTR  6,0          SUBTRACT ONE FROM LENGTH
      EX    6,NUMMOVE   MOVE NUMERIC VALUE
      LA    7,BINFLD    POINT TO BINFLD FIELD
      LA    10,PACKIN   POINT TO PACKIN FIELD
      MVC   0(4,10),=X'00000000' SET FIELD TO ZEROES
      NI    3(8),X'CF'  ZONE IT
      PACK  0(4,8),0(4,8) PACK PACKIN TO PACKIN
      SPACE
      EXEC CICS DELAY FOR SECONDS(SECOND)
      SPACE
      CVB   6,0(0,10)   CONVERT TO BINARY
      SPACE
      EXEC CICS DELAY FOR SECONDS(SECOND)
      SPACE
      ST    6,0(7)      STORE IT INTO BINFLD
      SPACE
*****
*   SEARCH DFHEITAB CVDA TABLE
*   R6 = CVDA  R8 = VALUE
*****
SEARCH EQU *
      SPACE
      EXEC CICS LOAD PROGRAM('DFHEITAB') SET(6)
      SPACE
      L     6,X'58'(6)   POINT TO DFHVALUE IN MODULE
      LA    6,X'15'(6)   POINT TO 1ST CVDA 'NOTAPPLIC'
      LA    8,X'C'(6)    POINT TO 1ST VALUE '00000001'
      SPACE
* -----*
*           TABLE LAYOUT LENGTH X'21'
*           FILLER CVDA VALUE
*           0           1      13-16
* -----*
      SPACE
      LA    5,INPUT
      CLC  5(1,5),=X'E9'  NUMERIC INPUT
      BH   NUMSEAR        YES VALUESEARCH
      LA    5,ALPHAFLD

```



```

SPACE
CVDASEAR EQU *
CLC 0(12,5),0(6) COMPARE CVDA
BE MOVE MOVE INTO OUTPUT
LA 6,X'11'(6) POINT TO NEXT CVDA
LA 8,X'C'(6) POINT TO NEXT VALUE
CLC 0(5,6),=C'T7770' STOPPER SET TO DEVICE OPTION
BE ERRORMSG
B CVDASEAR LOOK FOR NEXT ENTRY
SPACE
NUMSEAR EQU *
LA 5,BINFLD POINT TO BIN VALUE FIELD
SPACE
NUMSEAR1 EQU *
CLC 0(4,5),0(8) COMPARE VALUE
BE MOVE MOVE INTO OUTPUT
LA 6,X'11'(6) POINT TO NEXT CVDA
LA 8,X'11'(8) POINT TO NEXT VALUE
CLC 0(5,6),=C'T7770' STOPPER SET TO DEVICE OPTION
BE ERRORMSG
B NUMSEAR1 CHECK NEXT
SPACE
*****
* PROCESS OUTPUT *
* R6 = CVDATAB R8 = VALUETAB R5 = VALUEOUT R7 = CVDAOUT *
*****
SPACE
MOVE EQU *
LA 5,VAL POINT TO VALUE OUTPUT
LA 7,CVDAVAL POINT TO CVDA OUTPUT
MVC 0(12,7),0(6) MOVE CVDA FROM TABLE TO OUTPUT
ICM 9,15,0(8) LOAD VALUE
LA 4,CVBIN LOAD CONVERTOUT AREA
CVD 9,0(0,4) CONVERT TO DECIMAL
UNPK 0(4,5),0(8,4) UNPACK INTO VAL OUTPUT FIELD
OI 3(5),X'F0' MAKE LAST BYTE DECIMAL
B SENDLIST SEND OUTPUT
SPACE
ERROR EQU *
SPACE
EXEC CICS SEND TEXT FROM(MSGDAT1) ERASE FREEKB
SPACE
B RETURN
SPACE
ERROR1 EQU *
SPACE
EXEC CICS SEND TEXT FROM(MSGDAT2) ERASE FREEKB
SPACE
B RETURN
SPACE

```

```

ERRORMSG EQU *
SPACE
EXEC CICS SEND TEXT FROM(MSGDAT3) ERASE FREEKB
SPACE
B RETURN
SPACE
SENDLIST EQU *
SPACE
EXEC CICS ASSIGN SYSID(SYSID) PROGRAM(PROGRAM)
SPACE
MVC TRANSID,EIBTRNID
SPACE
EXEC CICS SEND TEXT FROM(OUT) ERASE FREEKB
SPACE
RETURN EQU *
SPACE
EXEC CICS RETURN
SPACE
START DS 0H
SPACE
*****
* LOAD DFHEISTG USER FIELDS *
*****
SPACE
MVC HDRMSG,MSGDAT
MVC CICSVER,=C'CICSVER='
MVC SYSIDENT,=C' SYSID='
MVC TRANID,=C' TRANSID='
MVC PROGNA,=C' PROGRAM='
MVC VALUE,=C'VALUE '
MVC VAL,=C' '
MVC FILL2,=C' '
MVC FILL3,=C' '
MVC CVDA,=C'CVDA '
MVC CVDAVAL,=C' '
MVC ALPHAFLD,=C' '
MVC HELP,=C' '
MVC HELP1,=C' '
MVI NL,X'15'
MVI NL1,X'15'
MVI NL2,X'15'
MVI NL3,X'15'
MVI NL4,X'15'
MVI FILL0,C'-'
MVC FILL0+1,FILL0
MVI NL0,X'15'
MVC FILL1,FILL0
B BEGIN
EJECT
*****

```

```

*          CONSTANTS          *
*****
          SPACE
NUMMOVE  DS      0F
          MVC      0(0,8),0(7)          EXECUTE MOVE NUMERIC VALUE
TEXTMOVE DS      0F
          MVC      0(0,7),0(5)          EXECUTE MOVE CVDA TO FIELD
MSGDAT   DC      CL79'                   DFHVALUE VALUE CVDA X-REFERENCE'
MSGDAT1  DC      CL80'CICS VERSION/RELEASE NOT SUPPORTED !'
MSGDAT2  DC      CL80'INVALID INPUT !!!!!!!!!!!!!!!!!!!!!!!'
MSGDAT3  DC      CL80'SEARCH ARGUMENT NOT FOUND !!!!!!!!!!!!!!!'
BLANKS   DC      CL256' '
ALIGN    DS      0F
SECOND   DC      XL4'00000001'
PREFIX   DC      XL2'0005'
          SPACE
          LTORG
          SPACE
          END      CSDFHVAL

```

Claus Reis

CICS Systems Programmer

Nuernberger Lebensversicherung AG (Germany)

© Xephon 2001

Update on JCHWBOS for CTS 1.3

An article entitled *The CICS Web interface – serving objects* in Issue 164 of *CICS Update*, July 1999, introduced the converter program, JCHWBOS, which made possible the delivery of images through the CICS Web Interface for CICS/ESA Version 4.1, using COMMAREAs, pointers, and the Web template management program DFHWBTL. CICS Transaction Server 1.3 introduced a new, easier-to-use Web API and new resources, including document templates. To use the new API, document templates have to be defined and installed but old-style object delivery is still available, in which case unrecognized document templates are looked for in the DFHHTML DD card and auto-installed.

However, some images are not delivered successfully using the old-style delivery at CTS 1.3. The Web Template Management program

returns 'RESP=008,RESP2=0007 (OBJECTTOOBIG)', even though the templates are in fact considerably smaller than the 32KB allowed.

This is because using the old-style DFHWBTL-based method can result in templates auto-installing, when by default they are defined as 'TYPE=EBCDIC,APPEND=YES'. Defining the image templates as 'TYPE=BINARY,APPEND=NO' and avoiding the auto-install solves this problem.

Alternatively, CICS Support Pack CA8C provides a sample converter, GRAPHIX, which delivers images successfully through the new API, ie DOCUMENT CREATE followed by WEB SEND, provided that the document templates are defined as 'TYPE=BINARY'. Download it from <http://www-4.ibm.com/software/ts/cics/txppacs/ca8c.html>.

J O'Grady
CICS Systems Programmer
Norwich Union (UK)

© Xephon 2001

Simple tool to manage the data extracted from CICS CSD in a DB2 environment – part 2

This month we conclude the code to manage the data extracted from CICS CSD in a DB2 environment.

```
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '     SYSREC00 INTO TABLE PRA10S.TABLSRP'
Queue ' ('
Queue '     TYPERS POSITION(1) CHAR(4),
Queue '     LSRPNAM POSITION(5) CHAR(8),
Queue '     GROUP POSITION(13) CHAR(8),
Queue '     DESCR POSITION(21) CHAR(58),
Queue '     POOLID POSITION(79) CHAR(1),
Queue '     MAXKEYL POSITION(80) CHAR(3),
Queue '     SHRLIM POSITION(83) CHAR(3),
Queue '     STRINGS POSITION(86) CHAR(3),
Queue '     D512 POSITION(89) CHAR(5),
Queue '     D1K POSITION(94) CHAR(5),
Queue '     D2K POSITION(99) CHAR(5),
Queue '     D4K POSITION(104) CHAR(5),
Queue '     D8K POSITION(109) CHAR(5),
```

```

Queue '      D12K POSITION(114) CHAR(5),      '
Queue '      D16K POSITION(119) CHAR(5),      '
Queue '      D20K POSITION(124) CHAR(5),      '
Queue '      D24K POSITION(129) CHAR(5),      '
Queue '      D28K POSITION(134) CHAR(5),      '
Queue '      D32K POSITION(139) CHAR(5),      '
Queue '      I512 POSITION(144) CHAR(5),      '
Queue '      I1K POSITION(149) CHAR(5),        '
Queue '      I2K POSITION(154) CHAR(5),        '
Queue '      I4K POSITION(159) CHAR(5),        '
Queue '      I8K POSITION(164) CHAR(5),        '
Queue '      I12K POSITION(169) CHAR(5),       '
Queue '      I16K POSITION(174) CHAR(5),       '
Queue '      I20K POSITION(179) CHAR(5),       '
Queue '      I24K POSITION(184) CHAR(5),       '
Queue '      I28K POSITION(189) CHAR(5),       '
Queue '      I32K POSITION(194) CHAR(5),       '
Queue '      HSD4K POSITION(199) CHAR(8),      '
Queue '      HSD8K POSITION(207) CHAR(8),      '
Queue '      HSD12K POSITION(215) CHAR(8),     '
Queue '      HSD16K POSITION(223) CHAR(8),     '
Queue '      HSD20K POSITION(231) CHAR(8),     '
Queue '      HSD24K POSITION(239) CHAR(8),     '
Queue '      HSD28K POSITION(247) CHAR(8),     '
Queue '      HSD32K POSITION(255) CHAR(8),     '
Queue '      HSI4K POSITION(263) CHAR(8),      '
Queue '      HSI8K POSITION(271) CHAR(8),      '
Queue '      HSI12K POSITION(279) CHAR(8),     '
Queue '      HSI16K POSITION(287) CHAR(8),     '
Queue '      HSI20K POSITION(295) CHAR(8),     '
Queue '      HSI24K POSITION(303) CHAR(8),     '
Queue '      HSI28K POSITION(311) CHAR(8),     '
Queue '      HSI32K POSITION(319) CHAR(8),     '
Queue '    )'
Queue
Call Write_sysinf
"DELSTACK"
End
lsrp.f3 = recinp.i
Return
Prepare_Sysin_Maps:
if f4 = 1 then do
  tf = 'MAPS'
  Wfile = sfile||tf||')'
  Call Alloc_sysinf
  "NEWSTACK"
  Queue 'CREATE TABLE PRA10S.TABMAPS'
  Queue '      (TYPERES CHAR(4)      NOT NULL,'
  Queue '      MAPSNAM CHAR(8)      NOT NULL,'
  Queue '      GROUP CHAR(8)      NOT NULL,'
Queue '      DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
```

```

Queue '          RESIDENT CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          USAGE   CHAR(9)  NOT NULL WITH DEFAULT ,'
Queue '          USELPA  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          STATUS  CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          RSL     CHAR(6) NOT NULL WITH DEFAULT)'
Queue '      IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABMAPS'
Queue '          ON PRA10S.TABMAPS'
Queue '          (MAPSNAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '      SYSREC00 INTO TABLE PRA10S.TABMAPS'
Queue ' ('
Queue '     TYPERES POSITION(1) CHAR(4),
Queue '     MAPSNAM POSITION(5) CHAR(8),
Queue '     GROUP POSITION(13) CHAR(8),
Queue '     DESCR POSITION(21) CHAR(58),
Queue '     RESIDENT POSITION(79) CHAR(3),
Queue '     USAGE POSITION(82) CHAR(9),
Queue '     USELPA POSITION(91) CHAR(3),
Queue '     STATUS POSITION(94) CHAR(8),
Queue '     RSL POSITION(102) CHAR(6)
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
maps.f4 = recinp.i
Return
Prepare_Sysin_Part:
if f5 = 1 then do
    tf = 'PART'
    Wfile = sfile||tf||')'
    Call Alloc_sysinf
    "NEWSTACK"
    Queue 'CREATE TABLE PRA10S.TABPART'
    Queue '          (TYPERES CHAR(4)          NOT NULL,'
    Queue '          PARTNAM CHAR(8)          NOT NULL,'
    Queue '          GROUP CHAR(8)          NOT NULL,'
Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
Queue '          RESIDENT CHAR(3) NOT NULL WITH DEFAULT ,'

```

```

Queue '          USAGE  CHAR(9)  NOT NULL WITH DEFAULT ,'
Queue '          USELPA CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          STATUS  CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          RSL     CHAR(6)  NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABPART'
Queue '          ON PRA10S.TABPART'
Queue '          (PARTNAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '          SYSREC00 INTO TABLE PRA10S.TABPART'
Queue ' ('
Queue '   TYPERES POSITION(1) CHAR(4),
Queue '   PARTNAM POSITION(5) CHAR(8),
Queue '   GROUP POSITION(13) CHAR(8),
Queue '   DESCR POSITION(21) CHAR(58),
Queue '   RESIDENT POSITION(79) CHAR(3),
Queue '   USAGE POSITION(82) CHAR(9),
Queue '   USELPA POSITION(91) CHAR(3),
Queue '   STATUS POSITION(94) CHAR(8),
Queue '   RSL POSITION(102) CHAR(6)
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
part.f5 = recinp.i
Return
Prepare_Sysin_Prof:
if f6 = 1 then do
tf = 'PROF'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABPROF'
Queue '          (TYPERES CHAR(4)          NOT NULL,'
Queue '          PROFNAM CHAR(8)          NOT NULL,'
Queue '          GROUP CHAR(8)          NOT NULL,'
Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
Queue '          SCRNSIZE CHAR(9) NOT NULL WITH DEFAULT ,'
Queue '          UCTRAN CHAR(3) NOT NULL WITH DEFAULT ,'

```

```

Queue '          MODENAME CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          PRICOMP  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          JOURNAL  CHAR(2)  NOT NULL WITH DEFAULT ,'
Queue '          MSGJRNL  CHAR(6)  NOT NULL WITH DEFAULT ,'
Queue '          MSGINTEG CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          ONEWTE   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          PROTECT  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          CHAINC   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          DVSUPRT  CHAR(7)  NOT NULL WITH DEFAULT ,'
Queue '          INBFMH   CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          RAQ      CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          LOGREC   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          NEPCCLASS CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          RTIMOUT  CHAR(4)  NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE  INDEX PRA10S.XTABPROF'
Queue '          ON PRA10S.TABPROF'
Queue '          (PROFNAM  ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue '  LOAD DATA RESUME YES LOG YES INDDN  '
Queue '          SYSREC00 INTO TABLE PRA10S.TABPROF'
Queue '  ('
Queue '    TYPERS POSITION(1) CHAR(4),
Queue '    PROFNAM POSITION(5) CHAR(8),
Queue '    GROUP POSITION(13) CHAR(8),
Queue '    DESCR POSITION(21) CHAR(58),
Queue '    SCRNSIZE POSITION(79) CHAR(9),
Queue '    UCTRAN POSITION(88) CHAR(3),
Queue '    MODENAME POSITION(91) CHAR(8),
Queue '    PRICOMP POSITION(99) CHAR(3),
Queue '    JOURNAL POSITION(102) CHAR(2),
Queue '    MSGJRNL POSITION(104) CHAR(6),
Queue '    MSGINTEG POSITION(110) CHAR(3),
Queue '    ONEWTE POSITION(113) CHAR(3),
Queue '    PROTECT POSITION(116) CHAR(3),
Queue '    CHAINC POSITION(119) CHAR(3),
Queue '    DVSUPRT POSITION(122) CHAR(7),
Queue '    INBFMH POSITION(129) CHAR(4),
Queue '    RAQ POSITION(133) CHAR(3),
Queue '    LOGREC POSITION(136) CHAR(3),
Queue '    NEPCCLASS POSITION(139) CHAR(3),

```



```

Queue '      RTIMOUT POSITION(142) CHAR(4)
Queue '    )'
Queue
Call Write_sysinf
"DELSTACK"
End
prof.f6 = recinp.i
Return
Prepare_Sysin_Prog:
if f7 = 1 then do
  tf = 'PROG'
  Wfile = sfile||tf||')'
  Call Alloc_sysinf
  "NEWSTACK"
  Queue 'CREATE TABLE PRA10S.TABPROG'
  Queue '          (TYPERES CHAR(4)          NOT NULL,'
  Queue '          PROGNAM CHAR(8)          NOT NULL,'
  Queue '          GROUP CHAR(8)          NOT NULL,'
  Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
  Queue '          LANGUAGE CHAR(9) NOT NULL WITH DEFAULT ,'
  Queue '          RELOAD CHAR(3) NOT NULL WITH DEFAULT ,'
  Queue '          RESIDENT CHAR(3) NOT NULL WITH DEFAULT ,'
  Queue '          USAGE CHAR(9) NOT NULL WITH DEFAULT ,'
  Queue '          USELPA CHAR(3) NOT NULL WITH DEFAULT ,'
  Queue '          STATUS CHAR(8) NOT NULL WITH DEFAULT ,'
  Queue '          CEDF CHAR(3) NOT NULL WITH DEFAULT ,'
  Queue '          DATALOC CHAR(5) NOT NULL WITH DEFAULT ,'
  Queue '          EXECKEY CHAR(4) NOT NULL WITH DEFAULT ,'
  Queue '          REMSYS CHAR(4) NOT NULL WITH DEFAULT ,'
  Queue '          REMNAM CHAR(8) NOT NULL WITH DEFAULT ,'
  Queue '          TRANSID CHAR(4) NOT NULL WITH DEFAULT ,'
  Queue '          EXECU CHAR(9) NOT NULL WITH DEFAULT)'
  Queue '      IN PRD10S.CICSRDO;'
  Queue 'CREATE INDEX PRA10S.XTABPROG'
  Queue '      ON PRA10S.TABPROG'
  Queue '      (PROGNAM ASC)'
  Queue '      USING STOGROUP PRG10S'
  Queue '      PRIQTY 12'
  Queue '      ERASE NO'
  Queue '      BUFFERPOOL BP1'
  Queue '      CLOSE NO ;'
  Queue
  Call Write_sysinf
  "DELSTACK"
  Wfile = sfile||tf||'#)'
  Call Alloc_sysinf
  "NEWSTACK"
  Queue '  LOAD DATA RESUME YES LOG YES INDDN '
  Queue '  SYSREC00 INTO TABLE PRA10S.TABPROG'
  Queue '  ('
Queue '  TYPERES POSITION(1) CHAR(4),

```

```

Queue '   PROGNAM POSITION(5) CHAR(8),      '
Queue '   GROUP POSITION(13) CHAR(8),      '
Queue '   DESCR POSITION(21) CHAR(58),     '
Queue '   LANGUAGE POSITION(79) CHAR(9),   '
Queue '   RELOAD POSITION(88) CHAR(3),     '
Queue '   RESIDENT POSITION(91) CHAR(3),   '
Queue '   USAGE POSITION(94) CHAR(9),      '
Queue '   USELPA POSITION(103) CHAR(3),    '
Queue '   STATUS POSITION(106) CHAR(8),    '
Queue '   CEDF POSITION(114) CHAR(3),      '
Queue '   DATALOC POSITION(117) CHAR(5),   '
Queue '   EXECKEY POSITION(122) CHAR(4),   '
Queue '   REMSYS POSITION(126) CHAR(4),    '
Queue '   REMNAM POSITION(130) CHAR(8),    '
Queue '   TRANSID POSITION(138) CHAR(4),   '
Queue '   EXECU POSITION(142) CHAR(9)      '
Queue '   )'
Queue
Call Write_sysinf
"DELSTACK"
End
prog.f7 = recinp.i
Return
Prepare_Sysin_Ptnr:
if f8 = 1 then do
tf = 'PTNR'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABPTNR'
Queue '          (TYPERES CHAR(4)          NOT NULL,'
Queue '          PTPNRNAM CHAR(8)          NOT NULL,'
Queue '          GROUP CHAR(8)              NOT NULL,'
Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
Queue '          NETNAME CHAR(8) NOT NULL WITH DEFAULT ,'
Queue '          NETWORK CHAR(8) NOT NULL WITH DEFAULT ,'
Queue '          PROFILE CHAR(8) NOT NULL WITH DEFAULT ,'
Queue '          TPNAME CHAR(64) NOT NULL WITH DEFAULT ,'
Queue '          XTPNAME CHAR(128) NOT NULL WITH DEFAULT)'
Queue ' IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABPTNR'
Queue '          ON PRA10S.TABPTNR'
Queue '          (PTPNRNAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"

```

```

Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue ' SYSREC00 INTO TABLE PRA10S.TABPTNR'
Queue ' ('
Queue ' TYPERS POSITION(1) CHAR(4), '
Queue ' PTNRNAM POSITION(5) CHAR(8), '
Queue ' GROUP POSITION(13) CHAR(8), '
Queue ' DESCR POSITION(21) CHAR(58), '
Queue ' NETNAME POSITION(79) CHAR(8), '
Queue ' NETWORK POSITION(87) CHAR(8), '
Queue ' PROFILE POSITION(95) CHAR(8), '
Queue ' TPNAME POSITION(103) CHAR(64), '
Queue ' XTPNAME POSITION(167) CHAR(128) '
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
ptnr.f8 = recinp.i
Return
Prepare_Sysin_Sess:
if f9 = 1 then do
tf = 'SESS'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABSESS'
Queue ' (TYPERS CHAR(4) NOT NULL,'
Queue ' SESSIONS CHAR(8) NOT NULL,'
Queue ' GROUP CHAR(8) NOT NULL,'
Queue ' DESCR CHAR(58) NOT NULL WITH DEFAULT,'
Queue ' CONN CHAR(4) NOT NULL WITH DEFAULT,'
Queue ' SESSNAM CHAR(4) NOT NULL WITH DEFAULT,'
Queue ' NETNAMEQ CHAR(8) NOT NULL WITH DEFAULT,'
Queue ' MODENAME CHAR(8) NOT NULL WITH DEFAULT,'
Queue ' PROTOCOL CHAR(4) NOT NULL WITH DEFAULT,'
Queue ' MAXIMUM CHAR(7) NOT NULL WITH DEFAULT,'
Queue ' RECPFX CHAR(2) NOT NULL WITH DEFAULT,'
Queue ' RECCOUNT CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' SNDPFX CHAR(2) NOT NULL WITH DEFAULT,'
Queue ' SNDCOUNT CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' SNDSIZE CHAR(5) NOT NULL WITH DEFAULT,'
Queue ' RECSIZE CHAR(5) NOT NULL WITH DEFAULT,'
Queue ' SESSPRI CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' USERID CHAR(8) NOT NULL WITH DEFAULT,'
Queue ' AUTOCONN CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' BUILDCHA CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' USERAREA CHAR(3) NOT NULL WITH DEFAULT,'
Queue ' IOAREA CHAR(11) NOT NULL WITH DEFAULT,'

```

```

Queue '          RELREQ  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          DISCREQ CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          NEPCCLASS CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          RECOVLOPT CHAR(11) NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABSESS'
Queue '          ON PRA10S.TABSESS'
Queue '          (SESSIONS ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '          SYSREC00 INTO TABLE PRA10S.TABSESS'
Queue ' ('
Queue '          TYPERS POSITION(1) CHAR(4),
Queue '          SESSIONS POSITION(5) CHAR(8),
Queue '          GROUP POSITION(13) CHAR(8),
Queue '          DESCR POSITION(21) CHAR(58),
Queue '          CONN POSITION(79) CHAR(4),
Queue '          SESSNAM POSITION(83) CHAR(4),
Queue '          NETNAMEQ POSITION(87) CHAR(8),
Queue '          MODENAME POSITION(95) CHAR(8),
Queue '          PROTOCOL POSITION(103) CHAR(4),
Queue '          MAXIMUM POSITION(107) CHAR(7),
Queue '          RECPFX POSITION(114) CHAR(2),
Queue '          RECCOUNT POSITION(116) CHAR(3),
Queue '          SNDPFX POSITION(119) CHAR(2),
Queue '          SNDCOUNT POSITION(121) CHAR(3),
Queue '          SNDSIZE POSITION(124) CHAR(5),
Queue '          RECSIZE POSITION(129) CHAR(5),
Queue '          SESSPRI POSITION(134) CHAR(3),
Queue '          USERID POSITION(137) CHAR(8),
Queue '          AUTOCONN POSITION(145) CHAR(3),
Queue '          BUILDCHA POSITION(148) CHAR(3),
Queue '          USERAREA POSITION(151) CHAR(3),
Queue '          IOAREA POSITION(154) CHAR(11),
Queue '          RELREQ POSITION(165) CHAR(3),
Queue '          DISCREQ POSITION(168) CHAR(3),
Queue '          NEPCCLASS POSITION(171) CHAR(3),
Queue '          RECOVLOPT POSITION(174) CHAR(11)
Queue '          )'
Queue
Call Write_sysinf

```

```

        "DELSTACK"
    End
sess.f9 = recinp.i
Return
Prepare_Sysin_Term:
if f10 = 1 then do
    tf = 'TERM'
    Wfile = sfile||tf||')'
    Call Alloc_sysinf
    "NEWSTACK"
    Queue 'CREATE TABLE PRA10S.TABTERM'
    Queue '          (TYPERES CHAR(4)          NOT NULL,'
    Queue '          TERMNAM CHAR(8)          NOT NULL,'
    Queue '          GROUP CHAR(8)          NOT NULL,'
    Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
    Queue '          AUTINSTM CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          AUTINSTN CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          TYPETERM CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          NETNAME CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          CONSOLE CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          CONSNAME CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          REMSYS CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          REMNAM CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          REMNET CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          MODENAME CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          PRINTER CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          PRINTERC CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          ALTPRINT CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          ALTPRINC CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          POOL CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          TASKLIM CHAR(5) NOT NULL WITH DEFAULT ,'
    Queue '          USERID CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          NATLANG CHAR(1) NOT NULL WITH DEFAULT ,'
    Queue '          TRANSACT CHAR(4) NOT NULL WITH DEFAULT ,'
    Queue '          TERMPRI CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          INSERVIC CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          SECURNAM CHAR(8) NOT NULL WITH DEFAULT ,'
    Queue '          ATTCHSEC CHAR(10) NOT NULL WITH DEFAULT ,'
    Queue '          BINDSEC CHAR(3) NOT NULL WITH DEFAULT ,'
    Queue '          USEDFTLU CHAR(3) NOT NULL WITH DEFAULT)'
    Queue ' IN PRD10S.CICSRD0;'
    Queue 'CREATE INDEX PRA10S.XTABTERM'
    Queue '          ON PRA10S.TABTERM'
    Queue '          (TERMNAM ASC)'
    Queue '          USING STOGROUP PRG10S'
    Queue '          PRIQTY 12'
    Queue '          ERASE NO'
    Queue '          BUFFERPOOL BP1'
    Queue '          CLOSE NO ;'
    Queue
    Call Write_sysinf

```

```

"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue ' SYSREC00 INTO TABLE PRA10S.TABTERM'
Queue ' ('
Queue ' TYPERES POSITION(1) CHAR(4), '
Queue ' TERMNAM POSITION(5) CHAR(8), '
Queue ' GROUP POSITION(13) CHAR(8), '
Queue ' DESCR POSITION(21) CHAR(58), '
Queue ' AUTINSTM POSITION(79) CHAR(4), '
Queue ' AUTINSTN POSITION(83) CHAR(8), '
Queue ' TYPETERM POSITION(91) CHAR(8), '
Queue ' NETNAME POSITION(99) CHAR(8), '
Queue ' CONSOLE POSITION(107) CHAR(3), '
Queue ' CONSNAME POSITION(110) CHAR(8), '
Queue ' REMSYS POSITION(118) CHAR(4), '
Queue ' REMNAM POSITION(122) CHAR(4), '
Queue ' REMNET POSITION(126) CHAR(8), '
Queue ' MODENAME POSITION(134) CHAR(8), '
Queue ' PRINTER POSITION(142) CHAR(4), '
Queue ' PRINTERC POSITION(146) CHAR(3), '
Queue ' ALTPRINT POSITION(149) CHAR(4), '
Queue ' ALTPRINC POSITION(153) CHAR(3), '
Queue ' POOL POSITION(156) CHAR(8), '
Queue ' TASKLIM POSITION(164) CHAR(5), '
Queue ' USERID POSITION(169) CHAR(8), '
Queue ' NATLANG POSITION(177) CHAR(1), '
Queue ' TRANSACT POSITION(178) CHAR(4), '
Queue ' TERMPRI POSITION(182) CHAR(3), '
Queue ' INSERVIC POSITION(185) CHAR(3), '
Queue ' SECURNAM POSITION(188) CHAR(8), '
Queue ' ATTCHSEC POSITION(196) CHAR(10), '
Queue ' BINDSEC POSITION(206) CHAR(3), '
Queue ' USEDFTLU POSITION(209) CHAR(3) '
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
term.f10 = recinp.i
Return
Prepare_Sysin_Tran:
if f11 = 1 then do
tf = 'TRAN'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABTRAN'
Queue ' (TYPERES CHAR(4) NOT NULL,'

```

```

Queue '          TRANNAM CHAR(8)          NOT NULL,'
Queue '          GROUP   CHAR(8)          NOT NULL,'
Queue '          DESCR   CHAR(58) NOT NULL WITH DEFAULT ,'
Queue '          PROGRAM CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          TWASIZE  CHAR(5)  NOT NULL WITH DEFAULT ,'
Queue '          PROFILE  CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          PARTSET  CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          STATUS   CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          TASKDLOC  CHAR(5)  NOT NULL WITH DEFAULT ,'
Queue '          TASKDKEY  CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          STOCLEAR  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          RUNAWAY   CHAR(7)  NOT NULL WITH DEFAULT ,'
Queue '          SHUTD     CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          ISOLATE   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          DYNAMIC   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          REMSYS    CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          REMNAM    CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          TRPROF    CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          LOCALQ    CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          PRIORITY  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          TRANCLAS  CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          ALIAS     CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          TASKREQ   CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          XTRANID   CHAR(8)  NOT NULL WITH DEFAULT ,'
Queue '          TPNAME    CHAR(64) NOT NULL WITH DEFAULT ,'
Queue '          XTPNAME   CHAR(128) NOT NULL WITH DEFAULT,'
Queue '          DTIMOUT  CHAR(4)  NOT NULL WITH DEFAULT ,'
Queue '          INDOUBT   CHAR(7)  NOT NULL WITH DEFAULT ,'
Queue '          RESTART   CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          SPURGE    CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          TPURGE    CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          DUMP      CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          TRACE     CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          CONFDATA  CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          RESSEC    CHAR(3)  NOT NULL WITH DEFAULT ,'
Queue '          CMDSEC    CHAR(3)  NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABTRAN'
Queue '          ON PRA10S.TABTRAN'
Queue '          (TRANNAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"

```

```

Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue ' SYSREC00 INTO TABLE PRA10S.TABTRAN'
Queue ' ('
Queue ' TYPERS POSITION(1) CHAR(4),
Queue ' TRANNAM POSITION(5) CHAR(8),
Queue ' GROUP POSITION(13) CHAR(8),
Queue ' DESCR POSITION(21) CHAR(58),
Queue ' PROGRAM POSITION(79) CHAR(8),
Queue ' TWASIZE POSITION(87) CHAR(5),
Queue ' PROFILE POSITION(92) CHAR(8),
Queue ' PARTSET POSITION(100) CHAR(8),
Queue ' STATUS POSITION(108) CHAR(8),
Queue ' TASKDLOC POSITION(116) CHAR(5),
Queue ' TASKDKEY POSITION(121) CHAR(4),
Queue ' STOCLEAR POSITION(125) CHAR(3),
Queue ' RUNAWAY POSITION(128) CHAR(7),
Queue ' SHUTD POSITION(135) CHAR(8),
Queue ' ISOLATE POSITION(143) CHAR(3),
Queue ' DYNAMIC POSITION(146) CHAR(3),
Queue ' REMSYS POSITION(149) CHAR(4),
Queue ' REMNAM POSITION(153) CHAR(8),
Queue ' TRPROF POSITION(161) CHAR(8),
Queue ' LOCALQ POSITION(169) CHAR(3),
Queue ' PRIORITY POSITION(172) CHAR(3),
Queue ' TRANCLAS POSITION(175) CHAR(8),
Queue ' ALIAS POSITION(183) CHAR(4),
Queue ' TASKREQ POSITION(187) CHAR(4),
Queue ' XTRANID POSITION(191) CHAR(8),
Queue ' TPNAME POSITION(199) CHAR(64),
Queue ' XTPNAME POSITION(263) CHAR(128),
Queue ' DTIMOUT POSITION(391) CHAR(4),
Queue ' INDOUBT POSITION(395) CHAR(7),
Queue ' RESTART POSITION(402) CHAR(3),
Queue ' SPURGE POSITION(405) CHAR(3),
Queue ' TPURGE POSITION(408) CHAR(3),
Queue ' DUMP POSITION(411) CHAR(3),
Queue ' TRACE POSITION(414) CHAR(3),
Queue ' CONFDATA POSITION(417) CHAR(3),
Queue ' RESSEC POSITION(420) CHAR(3),
Queue ' CMDSEC POSITION(423) CHAR(3)
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
tran.f11 = recinp.i
Return
Prepare_Sysin_Trcl:
if f12 = 1 then do
tf = 'TRCL'
Wfile = sfile||tf||')'

```



```

Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABTRCL'
Queue '          (TYPERES CHAR(4)          NOT NULL,'
Queue '          TRCLNAM CHAR(8)          NOT NULL,'
Queue '          GROUP CHAR(8)          NOT NULL,'
Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'
Queue '          MAXACT CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          PURGET CHAR(7) NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABTRCL'
Queue '          ON PRA10S.TABTRCL'
Queue '          (TRCLNAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '          SYSREC00 INTO TABLE PRA10S.TABTRCL'
Queue ' ('
Queue ' TYPERES POSITION(1) CHAR(4),
Queue ' TRCLNAM POSITION(5) CHAR(8),
Queue ' GROUP POSITION(13) CHAR(8),
Queue ' DESCR POSITION(21) CHAR(58),
Queue ' MAXACT POSITION(79) CHAR(3),
Queue ' PURGET POSITION(82) CHAR(7)
Queue ' )'
Queue
Call Write_sysinf
"DELSTACK"
End
trcl.f12 = recinp.i
Return
Prepare_Sysin_Type:
if f13 = 1 then do
tf = 'TYPE'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'CREATE TABLE PRA10S.TABTYPE'
Queue '          (TYPERES CHAR(4)          NOT NULL,'
Queue '          TYPENAM CHAR(8)          NOT NULL,'
Queue '          GROUP CHAR(8)          NOT NULL,'
Queue '          DESCR CHAR(58) NOT NULL WITH DEFAULT ,'

```

Queue	'	DEVICE	CHAR(8)	NOT NULL WITH DEFAULT	'
Queue	'	TERMMOD	CHAR(1)	NOT NULL WITH DEFAULT	'
Queue	'	SESSTYPE	CHAR(8)	NOT NULL WITH DEFAULT	'
Queue	'	LDC	CHAR(8)	NOT NULL WITH DEFAULT	'
Queue	'	SHIP	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	PAGSIZE	CHAR(7)	NOT NULL WITH DEFAULT	'
Queue	'	ALTPAGE	CHAR(7)	NOT NULL WITH DEFAULT	'
Queue	'	ALTSUFF	CHAR(1)	NOT NULL WITH DEFAULT	'
Queue	'	FMHPARM	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	OBOPERID	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	AUTOPAGE	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	DEFSCREEN	CHAR(7)	NOT NULL WITH DEFAULT	'
Queue	'	ALTSCREEN	CHAR(7)	NOT NULL WITH DEFAULT	'
Queue	'	APLKYBD	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	APLTEXT	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	AUDIBLEA	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	COLOR	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	COPY	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	DUALCASE	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	EXTENDED	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	HILIGHT	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	KATAKANA	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	LIGHTPEN	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	MSRCONTR	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	OBFORMAT	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	PARTITIO	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	PRINTADA	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	PROGSYMB	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	VALIDATI	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	FORMFEED	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	HORIZFOR	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	VERTICAL	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	TEXTKYBD	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	TEXTPRIN	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	QUERY	CHAR(4)	NOT NULL WITH DEFAULT	'
Queue	'	OUTLINE	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	SOSI	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	BACKTRAN	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	CGCSGID	CHAR(11)	NOT NULL WITH DEFAULT	'
Queue	'	ASCII	CHAR(2)	NOT NULL WITH DEFAULT	'
Queue	'	SNDSIZE	CHAR(5)	NOT NULL WITH DEFAULT	'
Queue	'	RECSIZE	CHAR(5)	NOT NULL WITH DEFAULT	'
Queue	'	BRACKET	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	LOGMODE	CHAR(8)	NOT NULL WITH DEFAULT	'
Queue	'	LOGMODEC	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	ERRLASTL	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	ERRINTES	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	ERRCOLOR	CHAR(9)	NOT NULL WITH DEFAULT	'
Queue	'	ERRHILIG	CHAR(9)	NOT NULL WITH DEFAULT	'
Queue	'	AUTOCONN	CHAR(3)	NOT NULL WITH DEFAULT	'
Queue	'	ATI	CHAR(3)	NOT NULL WITH DEFAULT	'

```

Queue '          TTI          CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          CREATES CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          RELREQ    CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          DISCREQ  CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          NEPCCLASS CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          SIGNOFF  CHAR(6) NOT NULL WITH DEFAULT ,'
Queue '          XRFSSOFF CHAR(7) NOT NULL WITH DEFAULT ,'
Queue '          ROUTEMSG CHAR(8) NOT NULL WITH DEFAULT ,'
Queue '          LOGONMSG  CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          BUILDCHA  CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          USERAREA  CHAR(3) NOT NULL WITH DEFAULT ,'
Queue '          IOAREA   CHAR(11) NOT NULL WITH DEFAULT ,'
Queue '          UCTRAN   CHAR(6) NOT NULL WITH DEFAULT ,'
Queue '          RECOVOPT  CHAR(11) NOT NULL WITH DEFAULT ,'
Queue '          RECOVNOT  CHAR(11) NOT NULL WITH DEFAULT)'
Queue '          IN PRD10S.CICSRD0;'
Queue 'CREATE INDEX PRA10S.XTABTYPE'
Queue '          ON PRA10S.TABTYPE'
Queue '          (TYPENAM ASC)'
Queue '          USING STOGROUP PRG10S'
Queue '          PRIQTY 12'
Queue '          ERASE NO'
Queue '          BUFFERPOOL BP1'
Queue '          CLOSE NO ;'
Queue
Call Write_sysinf
"DELSTACK"
Wfile = sfile||tf||'#)'
Call Alloc_sysinf
"NEWSTACK"
Queue ' LOAD DATA RESUME YES LOG YES INDDN '
Queue '          SYSREC00 INTO TABLE PRA10S.TABTYPE'
Queue ' ('
Queue '          TYPERS POSITION(1) CHAR(4),
Queue '          TYPENAM POSITION(5) CHAR(8),
Queue '          GROUP POSITION(13) CHAR(8),
Queue '          DESCR POSITION(21) CHAR(58),
Queue '          DEVICE POSITION(79) CHAR(8),
Queue '          TERMMOD POSITION(87) CHAR(1),
Queue '          SESSTYPE POSITION(88) CHAR(8),
Queue '          LDC POSITION(96) CHAR(8),
Queue '          SHIP POSITION(104) CHAR(3),
Queue '          PAGSIZE POSITION(107) CHAR(7),
Queue '          ALTPAGE POSITION(114) CHAR(7),
Queue '          ALTSUFF POSITION(121) CHAR(1),
Queue '          FMHPARM POSITION(122) CHAR(3),
Queue '          OBOPERID POSITION(125) CHAR(3),
Queue '          AUTOPAGE POSITION(128) CHAR(3),
Queue '          DEFSCREEN POSITION(131) CHAR(7),
Queue '          ALTSCREEN POSITION(138) CHAR(7),
Queue '          APLKYBD POSITION(145) CHAR(3),

```

Queue ' APLTEXT POSITION(148) CHAR(3), '
 Queue ' AUDIBLEA POSITION(151) CHAR(3), '
 Queue ' COLOR POSITION(154) CHAR(3), '
 Queue ' COPY POSITION(157) CHAR(3), '
 Queue ' DUALCASE POSITION(160) CHAR(3), '
 Queue ' EXTENDED POSITION(163) CHAR(3), '
 Queue ' HIGHLIGHT POSITION(166) CHAR(3), '
 Queue ' KATAKANA POSITION(169) CHAR(3), '
 Queue ' LIGHTPEN POSITION(172) CHAR(3), '
 Queue ' MSRCONTR POSITION(175) CHAR(3), '
 Queue ' OBFORMAT POSITION(178) CHAR(3), '
 Queue ' PARTITIO POSITION(181) CHAR(3), '
 Queue ' PRINTADA POSITION(184) CHAR(3), '
 Queue ' PROGSYMB POSITION(187) CHAR(3), '
 Queue ' VALIDATI POSITION(190) CHAR(3), '
 Queue ' FORMFEED POSITION(193) CHAR(3), '
 Queue ' HORIZFOR POSITION(196) CHAR(3), '
 Queue ' VERTICAL POSITION(199) CHAR(3), '
 Queue ' TEXTKYBD POSITION(202) CHAR(3), '
 Queue ' TEXTPRIN POSITION(205) CHAR(3), '
 Queue ' QUERY POSITION(208) CHAR(4), '
 Queue ' OUTLINE POSITION(212) CHAR(3), '
 Queue ' SOSI POSITION(215) CHAR(3), '
 Queue ' BACKTRAN POSITION(218) CHAR(3), '
 Queue ' CGCSGID POSITION(221) CHAR(11), '
 Queue ' ASCII POSITION(232) CHAR(2), '
 Queue ' SNDSIZE POSITION(234) CHAR(5), '
 Queue ' RECSIZE POSITION(239) CHAR(5), '
 Queue ' BRACKET POSITION(244) CHAR(3), '
 Queue ' LOGMODE POSITION(247) CHAR(8), '
 Queue ' LOGMODEC POSITION(255) CHAR(3), '
 Queue ' ERRLASTL POSITION(258) CHAR(3), '
 Queue ' ERRINTES POSITION(261) CHAR(3), '
 Queue ' ERRCOLOR POSITION(264) CHAR(9), '
 Queue ' ERRHILIG POSITION(273) CHAR(9), '
 Queue ' AUTOCONN POSITION(282) CHAR(3), '
 Queue ' ATI POSITION(285) CHAR(3), '
 Queue ' TTI POSITION(288) CHAR(3), '
 Queue ' CREATES POSITION(291) CHAR(3), '
 Queue ' RELREQ POSITION(294) CHAR(3), '
 Queue ' DISCREQ POSITION(297) CHAR(3), '
 Queue ' NEPCCLASS POSITION(300) CHAR(3), '
 Queue ' SIGNOFF POSITION(303) CHAR(6), '
 Queue ' XRFSOFF POSITION(309) CHAR(7), '
 Queue ' ROUTEMSG POSITION(316) CHAR(8), '
 Queue ' LOGONMSG POSITION(324) CHAR(3), '
 Queue ' BUILDCHA POSITION(327) CHAR(3), '
 Queue ' USERAREA POSITION(330) CHAR(3), '
 Queue ' IOAREA POSITION(333) CHAR(11), '
 Queue ' UCTRAN POSITION(344) CHAR(6), '
 Queue ' RECOVOPT POSITION(350) CHAR(11), '

```

        Queue '      RECOVNOT POSITION(361) CHAR(11)
            Queue '    )'
            Queue
            Call Write_sysinf
            "DELSTACK"
        End
type.f13 = recinp.i
Return
Prepare_Sysin_Tables:
tf = 'TABLES'
Wfile = sfile||tf||')'
Call Alloc_sysinf
"NEWSTACK"
Queue 'DROP TABLESPACE PRD10S.CICSRDO ;'
Queue 'COMMIT ;'
Queue 'CREATE TABLESPACE CICSRDO'
Queue '  IN PRD10S'
Queue '  USING STOGROUP PRG10S'
Queue '                PRIQTY 20'
Queue '                SECQTY 20'
Queue '                ERASE NO'
Queue '  BUFFERPOOL BP1'
Queue '  CLOSE NO ;'
Queue 'COMMIT ;'
Queue
Call Write_sysinf
"DELSTACK"
Return
Alloc_sysinf:
ADDRESS TSO
dd=OUTTRAP(dd.)
"ALLOC DA('"Wfile"') F(FILEOUT) SHR REUSE"
dd=OUTTRAP('OFF')
if rc = 0 then do
            typfunc = 'AllocSysin'
            Call CX_Error_func
            say mess
            "Free F(FILEIN)"
            "Free F(FILEOUT)"
            Exit
        End
Return
Write_sysinf:
"EXECIO * DISKW FILEOUT (FINIS)"
if rc = 0 then do
            typfunc = 'WriteSysin'
            Call CX_Error_func
            say mess
            "Free F(FILEIN)"
            "Free F(FILEOUT)"
            Exit

```

```

                                End
"FREE F(FILEOUT)"
Return
Write_type_file:
dd=OUTTRAP(dd.)
ADDRESS TSO "DELETE '"Wfile'"
"ALLOC DA('"Wfile"') DIR(0) SPACE(1,0) DSORG(PS)" ,
"RECFM(F,B) LRECL(460) BLKSIZE(27600) TRACKS" ,
"UNIT(WORKA) NEW CATALOG F(FILEOUT)"
dd=OUTTRAP('OFF')
if rc = 0 then do
                                typfunc = 'Allocfile'
                                Call CX_Error_func
                                say mess
                                Exit
                                End
dd=OUTTRAP(dd.)
"EXECIO" nrec "DISKW FILEOUT (STEM REC. FINIS"
"FREE F(FILEOUT)"
dd=OUTTRAP('OFF')
if rc = 0 then do
                                typfunc = 'Writetfile'
                                Call CX_Error_func
                                say mess
                                Exit
                                End
Return
CX_Error_func:
Select
when typfunc = 'Allocfilein' then
    mess = 'Allocation INPUT file in error.'
when typfunc = 'Readfilein' then
    mess = 'Read INPUT file in error.'
when typfunc = 'AllocSysin' then
    mess = 'Allocation SYSIN DB2 file in error for RDO resource 'tf
when typfunc = 'WriteSysin' then
    mess = 'Write SYSIN DB2 file in error for RDO resource 'tf
when typfunc = 'Allocfile' then
    mess = 'Allocation 'Wfile' file in error.'
when typfunc = 'Writetfile' then
    mess = 'Write 'Wfile' file in error.'
Otherwise nop
End
Return

```

SAMPLE JOB TO EXECUTE CICSDB2 TOOL

```

//TOOLCXDB JOB (LTYZ1100),
//          CLASS=S,REGION=0M,
//          MSGCLASS=X,

```

```

//          MSGLEVEL=(1,1),
//          NOTIFY=&SYSUID
//*
/*JOBPARM BYTES=999999,LINES=9999
//*
//*****
/* DELETE/DEFINE SYSIN FILE (DB2 Statements) FOR DB2 UTILITY      *
//*****
//DELETE EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN      DD  *
DELETE USR0110.CICSSVIL.DFHCS0.DB2.SYSIN NONVSAM
/*
//*
//DEFINE EXEC PGM=IEFBR14
//SYSPRINT DD  SYSOUT=*
//SYSOUT DD   SYSOUT=*
//DB2SYSIN DD  DISP=(NEW,CATLG,DELETE),
//            DSN=USR0110.CICSSVIL.DFHCS0.DB2.SYSIN,
//            VOL=SER=CIXH02,UNIT=3390,
//            DCB=(DSORG=PS,BLKSIZE=27920,RECFM=FB,LRECL=80),
//            SPACE=(CYL,(1,1,10))
//*
//*****
/* EXEC CICSDB2 CLIST TO: - CREATE TABLESPACE                      *
/*                          - CREATE TABLES                       *
/*                          - LOAD TABLES                         *
//*****
//DB2FUNC EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSEXEC DD DSN=CICS.USERLIB.CLIST,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
CICSDB2 CICSSVIL.DFHCS0.DB2.SORT
/*      |
/*      |-----> Input file (created by CICS EXTRACT function)
//*

```

SAMPLE COMMANDS TO A DB2 ENVIRONMENT

After execution of the CICSDB2 utility, it is possible to approach a DB2 environment using the QMF interface and/or other products in order to carry out query, select, update, or any other required operation on the database environment.

Some examples of commands for the QMF interface are:

```
SELECT * FROM PRA10S.TABPROG
```

```
SELECT COUNT(*) FROM PRA10S.TABTRAN
```

```

WHERE GROUP LIKE 'GR%'
AND TRANNAM LIKE 'TM%'

SELECT TRANNAM FROM PRA10S.TABTRAN
WHERE TRANNAM LIKE 'TM%'
UNION
SELECT PROGNAME FROM PRA10S.TABPROG
WHERE PROGNAME LIKE 'TM%'

SELECT * FROM PRA10S.TABTRAN
ORDER BY TWASIZE,TRANNAM DESC

SELECT * FROM PRA10S.TABTERM
ORDER BY TYPETERM

SELECT * FROM PRA10S.TABFILE
ORDER BY GROUP

UPDATE PRA10S.TABPROG
SET DESCR = 'Definition of remote program for EARE project.'
WHERE REMSYS = 'CICSTST3'

```

Espedito Morvillo
Systems Programmer (Italy)

© Xephon 2001

Utility to generate CICS maps

MAPDEF generates Assembler map definitions by processing input map images. Practice proves that it is far easier to draw a map than to specify macros with lots of parameters. If you don't have SDF II or some similar product, try to make a map with MAPDEF.

The input map image is stored in a dataset with a fixed or variable-record format. The maximum record length is 133 bytes. The output dataset must have a record length of 80 bytes.

MAP IMAGE DEFINITION

The definition consists of four parts, each beginning with a right bracket in the first position, followed by the keyword. Schematically, it looks like:

```
)name_of_mapset,parameters
```



```

)Attributes
marker1:parameters
marker2:parameters
marker3:parameters
...
)Variable
variable1:long_name_variable
variable2:long_name_variable,attributes
... or
short_prefix*:long_prefix_variable
...
* comment
)Map:map_name1,parameters
-----
|   MAP IMAGE 1   |
-----

)Map:map_name2,parameters
-----
|   MAP IMAGE 2   |
-----
...

```

PARAMETER DESCRIPTION

The following parameters are available:

-))mapset, parameters

At the beginning you assign the name of the MAPSET that is used in the macro definition DFHMSD. You specify the name after the two left brackets, followed by the parameters of the DFHMSD macro, delimited by commas. MODE is the only mandatory parameter.

-)Attributes
marker1:parameters
marker2:parameters
marker3:parameters

This part defines the attributes that are assigned to the markers. A marker specifies field attributes in the map image. A marker can be a special character (except *), any letter or digit, or combinations of these, with the maximum length being two characters.

The delimiter between the marker name and the parameters is a colon(:). Parameters have the same syntax as in the DFHMDF

macro. Parameter ATTRB is mandatory; don't specify POS and LENGTH parameters.

- ```
)variable
variable1:long_variable_name,attributes
variable2:long_variable_name
short_prefix*:long_variable_prefix
```

This part is optional. In cases where the full field name is longer than the field length, use any short name instead of it. In this part you define the relationship between the short name and the full name in the DFHMDF macro.

The second reason for using this part is the partial assignment of attributes to some specific variables. Attributes are specified after the variable name, and separated by commas. Any attribute specified in the marker is ignored.

If there are a lot of short fields that are filled by similar elements, we give them names beginning with the same prefix. In that case, you can set the short prefix into the field and then the suffix. The program will replace the short prefix with the full prefix specified in the variable definition.

- ```
* - comment
```

In any part of definition you can insert a comment line, marked with an asterisk in the first column.

- ```
)Map:map_name,parameters

MAP IMAGE
```

This part contains the map image. The part that begins with ' )Map:map\_name'. Map\_name is followed by the parameters of the DFHMDFI macro; only SIZE is mandatory.

You can draw a map image from the next row until the end of the dataset or until the next row that starts with a right bracket. The map image looks exactly the same as on the CICS screen.

Input and output variables are marked with some of the previously defined markers in the attribute part. The marker is placed after the starting position of the variable. Field length is the space

between this marker and the next marker. The next marker will be some other field marker or end marker. The end marker can be any character that is defined in the attribute part with the parameters 'ATTR=(ASKIP)'.

A variable name is written into the marked space. If there is not enough space, you can write a short name, and assign it to a long name in the variable part.

If you want to mark part of the text in a specific way, put a marker at the beginning of the part, and an end marker at the end of the text.

Programs use the following criterion to separate marked text from marked variables: if a parameter ATTRB contains subparameter ASKIP, it is a text marker, otherwise it is a variable marker.

If you don't specify variable names, the program defines them according to the following rule: RxxPyy where xx is a row number and yy is a column number of the field.

## EXAMPLE JOB TO SUBMIT MAPDEF

The following JCL can be used to submit a job:

```
//useridA JOB MSGCLASS=X,MSGLEVEL=(2,1),NOTIFY=userid,CLASS=A
//MAPDEF EXEC PGM=MAPDEF
//STEPLIB DD DSN=userid.USER.LOAD,DISP=SHR
//IN DD DSN=input_dataset,DISP=SHR
//OUT DD DSN=output_dataset,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

Note: in the SYSPRINT dataset you have a detailed report. In the process of generating macros, the program prints the input line and the corresponding macros. If the return code is not zero, a warning appears in the report.

## EXAMPLES

A map image looks like:

```
)MAPSAMP,MODE=INOUT,LANG=PLI
)Attributes
```

```

+:ATTRB=(ASKIP)
@:ATTRB=(BRT,UNPROT,IC)
#:ATTRB=(UNPROT,BRT)
$:ATTRB=(ASKIP,BRT)
&:ATTRB=(BRT,PROT)
)Variable
N:NUMBER
T:TYPETIME
A:ADDITARG
J:JOBNOTES
E:JOBERROR
M:JOBMESSC
S:JOBSYSCL
)Map:MSRDD1,SIZE=(24,80)
----- Display Job Information -----
Command$====>#COMMAND +

&message +

Select from last $====>@N + #T+ 1-99, (M)inutes,(H)ours, or (D)ays

Job Name $====>#JOBNAME + Job name or mask
Job ID $====>#JOBID + Job ID or mask
Job Net ID $====>#NETID + Net ID or mask
Additional arguments $====>#A+ (Y)es or (N)o

Optional Selection Criteria
Queue Type $====>#QUEUETY + queue type
Job Group ID $====>#JOBGID + job group ID or mask
Job Error Text $====>#JOBERRT + error text
Jobs with Notes only $====>#J+ (Y)es or (N)o
Initial Errors only $====>#E+ (Y)es or (N)o
Job Message Class $====>#M+ Jobs with Sysout Class$====>#S+

Press$ENTER+key to display.
Press$DOWN+key to select between time range
Press$END+key to return to the previous menu.

```

## MAPDEF will give the following result:

```

MAPSAMP DFHMSD TYPE=&SYSPARM,TIOAPFX=YES, C
 MODE=INOUT,LANG=PLI,STORAGE=AUTO,CTRL=(FREEKB,FRSET)
MSRDD1 DFHMDI SIZE=(24,80)
 DFHMDF POS=(01,01),LENGTH=77,ATTRB=(ASKIP), C
 INITIAL='----- Display Job Information ---C
 -----'
 DFHMDF POS=(02,01),LENGTH=07,ATTRB=(ASKIP), C
 INITIAL='Command'
 DFHMDF POS=(02,09),LENGTH=04,ATTRB=(ASKIP,BRT), C
 INITIAL='====>'
COMMAND DFHMDF POS=(02,14),LENGTH=64, C

```

```

ATTRB=(UNPROT,BRT)
DFHMDF POS=(02,79),LENGTH=01,ATTRB=(ASKIP)
message DFHMDF POS=(04,03),LENGTH=75, C
ATTRB=(BRT,PROT)
DFHMDF POS=(06,03),LENGTH=16,ATTRB=(ASKIP), C
INITIAL='Select from last'
DFHMDF POS=(06,26),LENGTH=04,ATTRB=(ASKIP,BRT), C
INITIAL='====>'
NUMBER DFHMDF POS=(06,31),LENGTH=02, C
ATTRB=(BRT,UNPROT,IC)
DFHMDF POS=(06,34),LENGTH=01,ATTRB=(ASKIP)
TYPETIM DFHMDF POS=(06,36),LENGTH=01, C
ATTRB=(UNPROT,BRT)
DFHMDF POS=(06,38),LENGTH=01,ATTRB=(ASKIP)
DFHMDF POS=(06,44),LENGTH=34,ATTRB=(ASKIP), C
INITIAL='1-99,(M)inutes,(H)ours,or(D)ays'
DFHMDF POS=(08,03),LENGTH=08,ATTRB=(ASKIP), C
INITIAL='Job Name'
DFHMDF POS=(08,26),LENGTH=04,ATTRB=(ASKIP,BRT), C
INITIAL='====>'
JOBNAME DFHMDF POS=(08,31),LENGTH=08, C
ATTRB=(UNPROT,BRT)
DFHMDF POS=(08,40),LENGTH=01,ATTRB=(ASKIP)
DFHMDF POS=(08,44),LENGTH=16,ATTRB=(ASKIP), C
INITIAL='Job name or mask'
DFHMDF POS=(09,03),LENGTH=06,ATTRB=(ASKIP), C
INITIAL='Job ID'
DFHMDF POS=(09,26),LENGTH=04,ATTRB=(ASKIP,BRT), C
INITIAL='====>'
JOBID DFHMDF POS=(09,31),LENGTH=08, C
ATTRB=(UNPROT,BRT)
DFHMDF POS=(09,40),LENGTH=01,ATTRB=(ASKIP)
DFHMDF POS=(09,44),LENGTH=16,ATTRB=(ASKIP), C
INITIAL='Job ID or mask'
DFHMDF POS=(10,03),LENGTH=10,ATTRB=(ASKIP), C
INITIAL='Job Net ID'
DFHMDF POS=(10,26),LENGTH=04,ATTRB=(ASKIP,BRT), C
INITIAL='====>'
NETID DFHMDF POS=(10,31),LENGTH=08, C
ATTRB=(UNPROT,BRT)
DFHMDF POS=(10,40),LENGTH=01,ATTRB=(ASKIP)
DFHMDF POS=(10,44),LENGTH=16,ATTRB=(ASKIP), C
INITIAL='Net ID or mask'
DFHMDF POS=(11,03),LENGTH=20,ATTRB=(ASKIP), C
INITIAL='Additional arguments'
DFHMDF POS=(11,26),LENGTH=04,ATTRB=(ASKIP,BRT), C
INITIAL='====>'
ADDITAR DFHMDF POS=(11,31),LENGTH=01, C
ATTRB=(UNPROT,BRT)
DFHMDF POS=(11,33),LENGTH=01,ATTRB=(ASKIP)
DFHMDF POS=(11,44),LENGTH=13,ATTRB=(ASKIP), C

```

|         |        |                                                                               |   |
|---------|--------|-------------------------------------------------------------------------------|---|
|         |        | INITIAL='(Y)es or (N)o'                                                       |   |
|         | DFHMDF | POS=(13,02),LENGTH=27,ATTRB=(ASKIP),<br>INITIAL='Optional Selection Criteria' | C |
|         | DFHMDF | POS=(14,03),LENGTH=10,ATTRB=(ASKIP),<br>INITIAL='Queue Type'                  | C |
|         | DFHMDF | POS=(14,26),LENGTH=04,ATTRB=(ASKIP,BRT),<br>INITIAL='====>'                   | C |
| QUEUETY | DFHMDF | POS=(14,31),LENGTH=08,<br>ATTRB=(UNPROT,BRT)                                  | C |
|         | DFHMDF | POS=(14,40),LENGTH=01,ATTRB=(ASKIP)                                           |   |
|         | DFHMDF | POS=(14,44),LENGTH=10,ATTRB=(ASKIP),<br>INITIAL='queue type'                  | C |
|         | DFHMDF | POS=(15,03),LENGTH=12,ATTRB=(ASKIP),<br>INITIAL='Job Group ID'                | C |
|         | DFHMDF | POS=(15,26),LENGTH=04,ATTRB=(ASKIP,BRT),<br>INITIAL='====>'                   | C |
| JOBGID  | DFHMDF | POS=(15,31),LENGTH=08,<br>ATTRB=(UNPROT,BRT)                                  | C |
|         | DFHMDF | POS=(15,40),LENGTH=01,ATTRB=(ASKIP)                                           |   |
|         | DFHMDF | POS=(15,44),LENGTH=20,ATTRB=(ASKIP),<br>INITIAL='job group ID or mask'        | C |
|         | DFHMDF | POS=(16,03),LENGTH=14,ATTRB=(ASKIP),<br>INITIAL='Job Error Text'              | C |
|         | DFHMDF | POS=(16,26),LENGTH=04,ATTRB=(ASKIP,BRT),<br>INITIAL='====>'                   | C |
| JOBERRT | DFHMDF | POS=(16,31),LENGTH=08,<br>ATTRB=(UNPROT,BRT)                                  | C |
|         | DFHMDF | POS=(16,40),LENGTH=01,ATTRB=(ASKIP)                                           |   |
|         | DFHMDF | POS=(16,44),LENGTH=10,ATTRB=(ASKIP),<br>INITIAL='error text'                  | C |
|         | DFHMDF | POS=(17,03),LENGTH=20,ATTRB=(ASKIP),<br>INITIAL='Jobs with Notes only'        | C |
|         | DFHMDF | POS=(17,26),LENGTH=04,ATTRB=(ASKIP,BRT),<br>INITIAL='====>'                   | C |
| JOBNOTE | DFHMDF | POS=(17,31),LENGTH=01,<br>ATTRB=(UNPROT,BRT)                                  | C |
|         | DFHMDF | POS=(17,33),LENGTH=01,ATTRB=(ASKIP)                                           |   |
|         | DFHMDF | POS=(17,44),LENGTH=13,ATTRB=(ASKIP),<br>INITIAL='(Y)es or (N)o'               | C |
|         | DFHMDF | POS=(18,03),LENGTH=19,ATTRB=(ASKIP),<br>INITIAL='Initial Errors only'         | C |
|         | DFHMDF | POS=(18,26),LENGTH=04,ATTRB=(ASKIP,BRT),<br>INITIAL='====>'                   | C |
| JOBERR0 | DFHMDF | POS=(18,31),LENGTH=01,<br>ATTRB=(UNPROT,BRT)                                  | C |
|         | DFHMDF | POS=(18,33),LENGTH=01,ATTRB=(ASKIP)                                           |   |
|         | DFHMDF | POS=(18,44),LENGTH=13,ATTRB=(ASKIP),<br>INITIAL='(Y)es or (N)o'               | C |
|         | DFHMDF | POS=(19,03),LENGTH=17,ATTRB=(ASKIP),<br>INITIAL='Job Message Class'           | C |
|         | DFHMDF | POS=(19,26),LENGTH=04,ATTRB=(ASKIP,BRT),                                      | C |

```

 INITIAL='==>'
JOBMESS DFHMDF POS=(19,31),LENGTH=01, C
 ATTRB=(UNPROT,BRT)
 DFHMDF POS=(19,33),LENGTH=01,ATTRB=(ASKIP)
 DFHMDF POS=(19,44),LENGTH=22,ATTRB=(ASKIP), C
 INITIAL='Jobs with Sysout Class'
 DFHMDF POS=(19,67),LENGTH=04,ATTRB=(ASKIP,BRT), C
 INITIAL='==>'
JOBSYSC DFHMDF POS=(19,72),LENGTH=01, C
 ATTRB=(UNPROT,BRT)
 DFHMDF POS=(19,74),LENGTH=01,ATTRB=(ASKIP)
 DFHMDF POS=(21,02),LENGTH=05,ATTRB=(ASKIP), C
 INITIAL='Press'
 DFHMDF POS=(21,08),LENGTH=05,ATTRB=(ASKIP,BRT), C
 INITIAL='ENTER'
 DFHMDF POS=(21,14),LENGTH=21,ATTRB=(ASKIP), C
 INITIAL='key to display. Press'
 DFHMDF POS=(21,36),LENGTH=04,ATTRB=(ASKIP,BRT), C
 INITIAL='DOWN'
 DFHMDF POS=(21,41),LENGTH=32,ATTRB=(ASKIP), C
 INITIAL='key to select between time range'
 DFHMDF POS=(22,02),LENGTH=05,ATTRB=(ASKIP), C
 INITIAL='Press'
 DFHMDF POS=(22,08),LENGTH=03,ATTRB=(ASKIP,BRT), C
 INITIAL='END'
 DFHMDF POS=(22,12),LENGTH=35,ATTRB=(ASKIP), C
 INITIAL='key to return to the previous menu.'
DFHMDF TYPE=FINAL
END

```

## PROGRAM MAPDEF

```

MAPDEF: PROC OPTIONS(MAIN);
/*****/
/* PROGRAM GENERATES ASSEMBLER MAP DEFINITION BASED ON MAP IMAGE */
/*****/
/*****/
/* DATASETS */
/*****/
DCL IN FILE RECORD SEQL INPUT;
DCL OUT FILE RECORD SEQL OUTPUT;
/*****/
/* WORKING VARIABLES */
/*****/
DCL 1 VAR BASED(PTR_VAR),
 2 NEXT PTR INIT(NULL),
 2 TYPE CHAR(1),
 2 FIELD CHAR(7) VAR,
 2 NAME_VAR CHAR(7) VAR,
 2 DESCR_VAR CHAR(70) VAR;

```

```

DCL 1 ATR BASED(PTR_ATR),
 2 NEXT PTR INIT(NULL),
 2 TYPE CHAR(1) INIT('K'),
 2 NAME_ATR CHAR(2) VAR,
 2 DESCR_ATR CHAR(70) VAR;
DCL (PTR_VAR,PP(3),PTR_VARG(2) INIT(NULL,NULL)) PTR;
DCL (PTR_ATR,PTR_ATRG INIT(NULL)) PTR;
DCL PTR_CURR PTR INIT(NULL);

DCL RECORD CHAR(133) VAR INIT('');
DCL 1 RECORD_DEF BASED(ADDR(RECORD)),
 2 LEN_REC BIN FIXED,
 2 CH(133) CHAR(1);
DCL 1 RECORD_OUT,
 2 LABEL CHAR(7),
 2 X CHAR(1) INIT(' '),
 2 MACRO CHAR(7),
 2 COMMENT CHAR(56),
 2 CONTINUE CHAR(9);
DCL (I,J,ROW,POS,LEN) BIN FIXED;
DCL RET_CODE BIN FIXED(31);
DCL WORK CHAR(80) VAR;
DCL NOT_EOF BIT INIT('1'B);
/***** BUILTIN FUNCTIONS *****/
DCL (NULL,SUBSTR,INDEX,ADDR,LENGTH) BUILTIN;
/***** ON CONDITIONS *****/
ON ERROR SNAP SYSTEM;
ON ENDFILE(IN) NOT_EOF='0'B;
/***** MAIN PROCEDURE *****/
I,J=1;
RET_CODE=0;
CALL READ_MAP_DEF(' ',' ');
DO WHILE(NOT_EOF);
 IF SUBSTR(RECORD,1,2) = '))'
 THEN CALL GEN_MAP_SET;
 ELSE
 IF SUBSTR(RECORD,1,2) = ')V'
 THEN CALL GEN_MAP_FIELDS;
 ELSE
 IF SUBSTR(RECORD,1,2) = ')A'
 THEN CALL GEN_MAP_ATTRIBUTES;
 ELSE
 IF SUBSTR(RECORD,1,2) = ')M'
 THEN CALL GEN_MAP;
 ELSE CALL MESSAGE('1'B,1,'');
END;
CALL OUT_ASM_MAP(' ','DFHMSD','TYPE=FINAL',' ');
CALL OUT_ASM_MAP(' ','END',' ',' ');
CALL PLIRETC(RET_CODE); /* SET EXTERNAL RETURN CODE */

/*****

```



```

/* PROCEDURE READS MAP IMAGE */
/*****
READ_MAP_DEF: PROC(M1,M2);
 DCL (M1,M2) CHAR(1);
 DO UNTIL(¬NOT_EOF | (SUBSTR(RECORD,1,1) ¬='*' & RECORD ¬=' '));
 READ FILE(IN) INTO(RECORD);
 PUT SKIP EDIT('SOURCE->>>',RECORD) (A);
 IF M1=' ' & (SUBSTR(RECORD,1,1) ¬='*') THEN ROW=ROW+1;
 END;
 DO LEN_REC=LENGTH(RECORD) TO 1 BY -1 WHILE(CH(LEN_REC)=' '); END;
 IF SUBSTR(RECORD,1,1) = ')'
 THEN DO;
 IF SUBSTR(RECORD,2,1) = ')'
 THEN M1=')';
 ELSE M1=': ';
 M2=', ';
 END;
 IF M1 ¬= ' '
 THEN DO;
 I=INDEX(RECORD,M1);
 J=INDEX(SUBSTR(RECORD,I),M2);
 IF J=Ø THEN J=LENGTH(RECORD)-I+1;
 END;
 RETURN;
END;

/*****
/* PROCEDURE PRINTS ASSEMBLER MAP DEFINITION */
/*****
OUT_ASM_MAP: PROC(L,M,O,N);
 DCL L CHAR(*) VAR;
 DCL M CHAR(*) VAR;
 DCL O CHAR(*) VAR;
 DCL N CHAR(*) VAR;
 LABEL=L;
 MACRO=M;
 COMMENT=O;
 CONTINUE=N;
 WRITE FILE(OUT) FROM(RECORD_OUT);
 PUT SKIP EDIT(RECORD_OUT) (X(7),A,A,A,A,A);
 RETURN;
END;

/*****
/* PROCEDURE GENERATE MAPSET (DFHMSD MACRO) */
/*****
GEN_MAP_SET: PROC;
 WORK=SUBSTR(RECORD,I+2,J-3);
 CALL OUT_ASM_MAP(WORK, 'DFHMSD',
 'TYPE=&SYSPARM,TIOAPFX=YES,', 'C');
 WORK=SUBSTR(RECORD,I+J);

```

```

IF INDEX(WORK,'STORAGE') = 0 &
 INDEX(WORK,'BASE') = 0
THEN WORK=WORK||',STORAGE=AUTO';
IF INDEX(WORK,'LANG') = 0
THEN WORK=WORK||',LANG=PLI';
IF INDEX(WORK,'CTRL') = 0
THEN WORK=WORK||',CTRL=(FREEKB,FRSET)';
CALL OUT_ASM_MAP(' ',' ',WORK,' ');
CALL READ_MAP_DEF(' ',' ');
END;

/*****
/* PROCEDURE GENERATES FIELDS (DFHMDF MACRO) */
*****/
GEN_MAP_FIELDS: PROC;
 DCL V CHAR(1) INIT('V');
 DCL (K,L) BIN FIXED;
 DCL PPR PTR;
 CALL READ_MAP_DEF(':',',',' ');
 IF SUBSTR(RECORD,1,1) = ')'
 THEN RETURN;
 DO WHILE(NOT_EOF & SUBSTR(RECORD,1,1) = ')');
 K=1;
 ALLOC VAR;
 VAR.TYPE=V;
 VAR.DESCR_VAR='';
 IF SUBSTR(RECORD,I-1,1) = '*'
 THEN DO;
 VAR.TYPE = 'G';
 FIELD=SUBSTR(RECORD,1,I-2);
 K=2;
 END;
 ELSE FIELD=SUBSTR(RECORD,1,I-1);
 IF SUBSTR(RECORD,I+1,6) = 'ATTRB='
 THEN CALL MESSAGE('1'B,3,FIELD);
 L=INDEX(RECORD,' ,ATTRB=');
 IF L = 0
 THEN NAME_VAR=SUBSTR(RECORD,I+1);
 ELSE DO;
 DESCR_VAR=SUBSTR(RECORD,L+1);
 L=L-I-1;
 NAME_VAR=SUBSTR(RECORD,I+1,L);
 END;
 IF PTR_VARG(K)=NULL
 THEN PTR_VARG(K)=PTR_VAR;
 ELSE PP(K)->VAR.NEXT=PTR_VAR;
 PP(K)=PTR_VAR;
 CALL READ_MAP_DEF(':',',',' ');
 END;
 /* PUTTING GENERIC VARIABLES, IF THEY EXIST, TO END */
 PP(1)->VAR.NEXT=PTR_VARG(2);

```

```

END;
/*****
/* PROCEDURE GENERATES FIELD ATTRIBUTES */
*****/
GEN_MAP_ATTRIBUTES: PROC;
 DCL (K,L) BIN FIXED;
 DCL POMA CHAR(2) VAR;
 CALL READ_MAP_DEF(':',',','');
 IF SUBSTR(RECORD,1,1) = ')'
 THEN RETURN;
 DO WHILE(NOT_EOF & SUBSTR(RECORD,1,1)≠')');
 POMA=SUBSTR(RECORD,1,I-1);
 WORK=SUBSTR(RECORD,I+1);
 DO PTR_ATR=PTR_ATRG REPEAT(ATR.NEXT) WHILE(PTR_ATR ≠NULL)
 UNTIL(ATR.NAME_ATR=POMA | ATR.DESCR_ATR = WORK);
 END;
 IF PTR_ATR=NULL
 THEN CALL ATTRIBUTE(POMA,WORK);
 ELSE DO;
 IF ATR.NAME_ATR = POMA
 THEN CALL MESSAGE('1'B,5,ATR.NAME_ATR);
 ELSE DO;
 CALL MESSAGE('Ø'B,6,ATR.NAME_ATR);
 CALL ATTRIBUTE(POMA,WORK);
 END;
 END;
 CALL READ_MAP_DEF(':',',','');
END;
END;

/*****
/* PROCEDURE GENERATES FIELD ATTRIBUTE */
*****/
ATTRIBUTE: PROC(A,0);
 DCL A CHAR(*) VAR, O CHAR(*) VAR;
 ALLOC ATR;
 IF PTR_ATRG=NULL
 THEN PTR_ATRG=PTR_ATR;
 ELSE PP(3)->ATR.NEXT=PTR_ATR;
 PP(3)=PTR_ATR;
 ATR.NAME_ATR=A;
 ATR.DESCR_ATR=0;
 IF INDEX(0,'PROT') > Ø | INDEX(0,'ASKIP') = Ø
 THEN ATR.TYPE='V';
END;

/*****
/* PROCEDURE GENERATE MAP */
*****/
GEN_MAP: PROC;
 DCL INDK CHAR(1);

```

```

DCL LENN BIN FIXED INIT(0);
WORK=SUBSTR(RECORD,I+1,J-2);
CALL OUT_ASM_MAP(WORK,'DFHMDI',SUBSTR(RECORD,I+J),' ');
ROW=0;
CALL READ_MAP_DEF(' ',' ');
DO WHILE(NOT_EOF & SUBSTR(RECORD,1,1)≠' ');
 POS=1;
 CALL FIND_MARKER(POS,I,LEN,LENN);
 IF LENN ≠ 0 /* IF MARKER IS NOT AT THE BEGGINING */
 THEN PTR_CURR=NULL; /* IT IS CONSTANNT */
 ELSE POS=I; /* OTHERWISE VARIABLE */
 DO WHILE(POS < LEN_REC);
 IF PTR_CURR ≠ NULL
 THEN INDK=PTR_CURR->ATR.TYPE;
 ELSE INDK='K';
 IF INDK = 'K'
 THEN CALL GEN_CONSTANT;
 ELSE CALL GEN_VARIABLE;
 END;
 CALL READ_MAP_DEF(' ',' ');
END;
END;

/*****
/* PROCEDURE GENERATES CONSTANT */
/*****
GEN_CONSTANT: PROC;
DCL (R,P,D) PIC'99';
DCL (M INIT(0),
 LENN INIT(0)) BIN FIXED;
IF PTR_CURR ≠ NULL
THEN WORK=PTR_CURR->ATR.DESCR_ATR;
ELSE WORK='ATTRB=(ASKIP)';
CALL FIND_MARKER(POS,M,LEN,LENN);
LEN=LENN;
IF PTR_CURR = NULL
THEN DO;
 I=INDEX(SUBSTR(RECORD,POS,LEN),' ');
 IF I > 0
 THEN DO;
 LEN=I-1;
 M=POS+I+6;
 PTR_CURR=NULL;
 END;
END;
IF LEN > 0
THEN DO;
 R=ROW;
 P=POS;
 D=LEN;
 CALL OUT_ASM_MAP(' ','DFHMDF',

```

```

 'POS=('||R||','||P||'),LENGTH='||D||','||WORK||','||C');
 IF LEN < 48
 THEN CALL OUT_ASM_MAP(' ',' ',
 'INITIAL='''||SUBSTR(RECORD,POS,LEN)||''',' ');
 ELSE DO;
 CALL OUT_ASM_MAP(' ',' ',
 'INITIAL='''||SUBSTR(RECORD,POS,47),'C');
 CALL OUT_ASM_MAP(' ',' ',
 SUBSTR(RECORD,POS+47,LEN-47)||''',' ');
 END;
 END;
 POS=M;
 RETURN;
END;

/*****
/* PROCEDURE GENERATES VARIABLE */
*****/
GEN_VARIABLE: PROC;
 DCL (R,P,D) PIC'99';
 DCL (M INIT(Ø),
 LENN INIT(Ø)) BIN FIXED;
 DCL WORK_NAME_VAR CHAR(7) VAR;

 WORK=PTR_CURR->ATR.DESCR_ATR;
 P=POS;
 CALL FIND_MARKER(POS,M,LEN,LENN);
 WORK_NAME_VAR=SUBSTR(RECORD,POS,LENN);
 DO PTR_VAR=PTR_VARG(1) REPEAT(VAR.NEXT) UNTIL(PTR_VAR=NULL |
 VAR.FIELD=WORK_NAME_VAR |
 (VAR.TYPE='G' &
 VAR.FIELD=SUBSTR(WORK_NAME_VAR,1,LENGTH(VAR.FIELD))));
 END;
 IF PTR_VAR ≠ NULL
 THEN DO;
 IF VAR.TYPE = 'G'
 THEN WORK_NAME_VAR=VAR.NAME_VAR||
 SUBSTR(WORK_NAME_VAR,LENGTH(VAR.FIELD)+1);
 ELSE WORK_NAME_VAR=VAR.NAME_VAR;
 IF VAR.DESCR_VAR ≠ ''
 THEN WORK=VAR.DESCR_VAR;
 END;
 R=ROW;
 D=LEN;
 IF LENN > 7
 THEN CALL MESSAGE('Ø'B,2,WORK_NAME_VAR);
 IF LENN = Ø
 THEN DO;
 D=M-P-1;
 WORK_NAME_VAR='R' || R || 'P' || P;
 CALL MESSAGE('Ø'B,4,WORK_NAME_VAR);

```

```

 END;
 ELSE P=POS;
 CALL OUT_ASM_MAP(WORK_NAME_VAR,'DFHMDF',
 'POS=('||R||','||P||'),LENGTH='||D||',' ','C');
 CALL OUT_ASM_MAP(' ',' ',WORK,' ');
 P=P+D;
 IF P < 80 & INDEX(WORK,'UNPROT') > 0 & CH(P+1) = ' '
 THEN DO;
 DO PTR_ATR=PTR_ATRG REPEAT(ATR.NEXT)
 WHILE(PTR_ATR = NULL & ATR.NAME_ATR = CH(P));
 END;
 P=P+1;
 IF PTR_ATR = NULL
 THEN WORK=ATR.DESCR_ATR;
 ELSE WORK='ATTRB=(ASKIP)';
 CALL OUT_ASM_MAP(' ','DFHMDF',
 'POS=('||R||','||P||'),LENGTH=01,'||WORK,' ');
 END;
 POS=M;
 RETURN;
END;

/*****
/* PROCEDURE RECOGNIZES MARKER */
*****/
FIND_MARKER: PROC(TP,NP,DP,DN);
 DCL (TP, /* CURRENT START OF STRING */
 NP, /* NEXT START OF STRING */
 DP, /* LENGTH OF GEN_MAP_FIELDS */
 DN, /* LENGTH OF NONBLANK STRING */
 N,M,K INIT(1)) BIN FIXED;
 DO TP=TP TO LENGTH(RECORD) WHILE(CH(TP)=' '); END;
 M=LENGTH(RECORD)-TP+1;
 DO PTR_ATR=PTR_ATRG REPEAT(ATR.NEXT) WHILE(PTR_ATR = NULL);
 N=INDEX(SUBSTR(RECORD,TP,M),ATR.NAME_ATR);
 IF N > 0
 THEN DO;
 M=N-1;
 PTR_CURR=PTR_ATR;
 K=LENGTH(ATR.NAME_ATR);
 END;
 END;
 NP=TP+M+K;
 DP=M;
 DO DN=TP+M-1 TO TP BY -1 WHILE(CH(DN)=' ');END;
 DN=DN-TP+1;
 RETURN;
END;

/*****
/* PROCEDURE PRINTS MESSAGE */
*****/

```

```

/*****/
MESSAGE: PROC(BREAK, CODE, TEXT);
 DCL BREAK BIT;
 DCL CODE BIN FIXED;
 DCL TEXT CHAR(*) VAR;
 DCL MESS(6) CHAR(45) VAR INIT(
 'MDØ1 THERE IS NO DEFINITION)),)V,)A OR)M',
 'MDØ2 VARIABLE NAME IS TO LONG ->',
 'MDØ3 ATTRIBUTE IS DESCR_ATRN IN SEGMENT)V ->',
 'MDØ4 VARIABLE IS NOT DEFINED AND IT WILL BE: ',
 'MDØ5 MARKER IS ALREADY DEFINED: ',
 'MDØ6 DESCRIPTION IS THE SAME AS MARKER: '
);
 PUT SKIP EDIT('WARNING *** ', MESS(CODE)) (A);
 IF TEXT = ''
 THEN PUT EDIT(TEXT) (A);
 IF BREAK
 THEN STOP;
 RET_CODE=4;
END;
END MAPDEF;

```

---

*Emina Spasic and Dragan Nikolic*  
*Systems Programmers*  
*Postal Saving Bank (Yugoslavia)*

© Xephon 2001

## **Need help with a CICS problem or project?**

Maybe we can help:

- If it's on a topic of interest to other subscribers, we'll commission an article on the subject, which we'll publish in *CICS Update*, and which we'll pay for – it won't cost you anything.
- If it's a more specialized, or more complex, problem, you can advertise your requirements (including one-off projects, freelance contracts, permanent jobs, etc) to the thousands of CICS professionals who visit *CICS Update*'s home page every week. This service is also free of charge.

Visit the *CICS Update* Web site, <http://www.xephon.com/cicsupdate.html>, and follow the link to *Suggest a topic* or *Opportunities for DB2 specialists*.

# CICS news

---

IBM has added to or updated the following SupportPacs. In the CICS freeware category there is CICS Web Interface 3270 Bridge – support for dynamic modification of attribute bytes, and CICS terminal definitions with autoinstall considerations.

In the third-party contributions category there's one called CICS to SMTP samples.

For further information contact your local IBM representative.

URL: <http://www-4.ibm.com/software/ts/cics/txppacs>.

\* \* \*

It has been estimated that IBM customers are using CICS to process over 30 billion transactions per day with a commercial value of several trillion dollars/week. CICS case studies describing some of the ways organizations are meeting their business needs with CICS technology are available from the IBM Web site.

For further information contact your local IBM representative.

URL: [www.ibm.com/cics/library](http://www.ibm.com/cics/library).

\* \* \*

HostBridge Technology has announced HostBridge, which allows organizations to integrate CICS transactions into scalable e-business applications.

HostBridge facilitates this integration by providing a platform-neutral, XML pathway from CICS transactions to middle-tier application servers. In doing so, HostBridge

completely circumvents the use of 3270 datastreams and the many layered components associated with screen scraping-based solutions. It provides an alternative to terminal emulation and screen scraping-based technologies to 'e-enable' production environments. HostBridge provides a direct XML pathway to middle-tier application servers.

For further information contact:

HostBridge Technology, 100 East 7th Avenue, Stillwater, OK 74074, USA.

Tel: (866) 965 2427.

URL: [http://www.hostbridge.com/press/20010122\\_hbAnnounce.html](http://www.hostbridge.com/press/20010122_hbAnnounce.html).

\* \* \*

IBM has modified its CICS TS for OS/390 Version 1.3, removing availability in Canada and scrapping the 90-day telephone support as part of the Migration Assistance Package. Two products have been removed from the qualifying products list.

Between now and 4 May, if a qualifying CICS product is installed and an upgrade to CICS Transaction Server for OS/390 Version 1.3 is made, a Migration Assistance Package worth \$2,500 is provided at no additional charge.

Qualifying products include CICS/ESA Version 3 Release 3, CICS/ESA Version 4, and CICS TS for OS/390 Version 1.3.

For further information contact your local IBM representative.

URL: <http://www.software.ibm.com>.



**xephon**