188 CICS update

July 2001

In this issue

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*, and complete issues in Acrobat PDF format, 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 download a copy of our *Notes for Contributors* from www.xephon. com/contnote.html.

Printed in England.

CICS/ESA 4 to CICS TS 1.3 application 'gotchas'

With the IBM announcement that CICS/ESA 4.1. support will end on 31 December 2002, many sites are now upgrading their CICS/ESA 4.1 systems to CICS Transaction Server 1.3. The upgrade is quite simple – the change to the MVS logger is probably the biggest task for the systems programmers. However, there are a few 'gotchas' that might catch the application programmers! Here are the ones I've found during my work upgrading CICS/ESA Version 4 systems to CICS TS Version 1.3. systems.

READQ TS – INVREQ NOT QIDERR

CICS TS will now return an INVREQ for an EXEC CICS READQ TS QUEUE('queue') – where 'queue' is low-values. CICS Version 4 returned QIDERR. Some application programs test for QIDERR to identify that the variable 'queue' has not been set yet. Application logic needs to be changed or the Global User Exit, XTSEREQC, could be used to set EIBRESP to QIDERR and emulate the CICS Version 4 response.

DEFAULT CICS WEB INTERFACE USERID

The CICS Version 4 Web interface facility provided a configuration transaction (CWBC) that allowed a default Web user-id to be specified. The CICS TS Web interface configuration is done using CEDA and a default Web user-id is not part of the new RDO definitions.

To set a default Web user-id with CICS TS 1.3, it must now be set in the Analyser by checking and setting COMMAREA field WBRA_USERID.

EXEC CICS START WITH REQID – IOERR

EXEC CICS START FROM(xxxx...) REQID(xxxx...) no longer operates like it used to under CICS 4. The response of IOERR can now be given when a duplicate REQID is used – this was not checked under CICS/ESA 4 (although the results would be unpredictable if a duplicate REQID already existed).

NEW CONDITIONS: LOCKED AND RECORDBUSY

Even without making use of RLS, the new conditions (locked and recordbusy) can be seen when accessing recoverable resources in CICS TS. Code that accesses recoverable resources needs to be reviewed in the light of these new conditions possibly being received. Check the *CICS Recovery and Restart Guide* for more details.

EIBRCODE AFTER ILLOGIC RESPONSE

Bytes 3 and 4 of EIBRCODE now contain VSAM return codes rather than NULLs. This can be helpful, but code comparing the EIBRCODE with user-defined constants needs to be reviewed.

CVDA NOTAPPLIC

The CVDAs for EXEC CICS INQUIRE TERMINAL requests are no longer made available in certain cases. For example, under CICS/ESA 4, an EXEC CICS INQUIRE TERMINAL for CREATESESS issued against a surrogate terminal would return a 67 (CREATE). Under CICS TS, a CVDA of 1 (NOTAPPLIC) is returned. This is apparently an intentional change to make the code match what the manual states.

THE COUNTDOWN HAS BEGUN!

To help prompt management and your CICS Application Areas into upgrading, here is a small piece of JavaScript to insert into your intranet page to display a countdown to the end of CICS/ESA 4 support:

```
<script language="JavaScript"><!--
today = new Date();
BigDay = new Date("January 1, 2003")
msPerDay = 24 * 60 * 60 * 1000 ;
timeLeft = (BigDay.getTime() - today.getTime());
e_daysLeft = timeLeft / msPerDay;
daysLeft = Math.floor(e_daysLeft);
document.write("CICS/ESA v4.1. support: " + daysLeft + " days
remaining.");
// --></script>
```

J Lemmon
JP – Lemon Tree (UK)

Automatic CICS RDO definition from a PDS library

When we install new application software comprising many programs and maps, we can save time and resources using the $DEFCMD utility.

This utility provides a way to create the RDO definition in DFHCSDUP format starting from a PDS library (source or load module).

There are three steps:

1 IKJEFT01 produces the member list of a library with the LISTDS 'library.name' members command.

2 $CLNLIST cleans the member list, removing useless rows.

3 $DEFCMD builds the RDO macro for the DFHCSDUP utility.

Before executing the $JOBCSD utility you have to:

- Allocate a sequential dataset (Fixed and LRECL 80) where you must write the command to obtain the member LIST – LISTDS 'LIBRARY NAME' MEMBERS. It may be a member of a partitioned dataset.
- Specify a partitioned dataset where the program DEFCMD will write the RDO macro – in the source code the name is USER.TSO.CNTL(RDOCMD).

It's important to check the parameter values for:

- CX – the name of the CICS region (first dsname qualifier).
- ENV – the CICS environment (production or test).
- VERI – the CICS version (V410, V530, etc).
- GRP – the RDO group.
- TIPO – BMS (MAPSET) PGM (PROGRAM) (default is PGM).

A second important advantage is that you can isolate a single package with its modules residing in one library from another package in a particular DFHCSD. It's easy to move these objects from a CICS region to another CICS region in a test or production environment.

The $JOBCSD JCL may be executed as a PROC. The $CLNLIST and $DEFCMD utilities are written in REXX.

$CLNLIST

```
/***********************REXX**************************/
/* THIS PROGRAM READS A LIST OF MODULES PRODUCED BY   */
/* IKJEFT01 AND CLEANS USELESS ROWS                   */
/*TRACE R*/
ADDRESS TSO
CONT = 0
FLAG = 0
J = 0
'EXECIO * DISKR INPLIST ( STEM VARJ.'
LIMITE = VARJ.0
SAY '*   NUMBER OF RECORDS IN THE INPUT FILE ...:' VARJ.0
DO J=J+1 WHILE FLAG = 0
   RECO = SUBSTR(VARJ.J,1,50)
   F = FIND(RECO,'--MEMBERS--')
   IF F > 0 THEN  FLAG = 1
      ELSE NOP
  'EXECIO * DISKR INPLIST ( STEM VARJ.'
END
J=J-1
DO J=J+1 TO LIMITE
   RECKO = SUBSTR(VARJ.J,1,10)
   SELECT
       WHEN RECKO = 'END       ' THEN
            'EXECIO * DISKR INPLIST ( STEM VARJ.'
       WHEN RECKO = 'READY     ' THEN
            'EXECIO * DISKR INPLIST ( STEM VARJ.'
       OTHERWISE
             CALL  WRITEREC
   END
END
SAY  '*** NUMBER OF RESOURCES ===> ' CONT
EXIT
WRITEREC:
RECO = SUBSTR(VARJ.J,3,8)
R = SPACE(RECO,0)
R = '  ' R
CONT = CONT + 1
QUEUE R
'EXECIO * DISKW OUT'
'EXECIO * DISKR INPLIST ( STEM VARJ.'
RETURN
```

$DEFCMD

```
/********************** REXX ********************************/
```

```
/* THIS PROGRAM BUILDS THE MACRO IN DFHCSDUP FORMAT         */
/* STARTING FROM A PDS DATASET CONTAINING PROGRAMS AND MAPS */
/* TRACE R */
ADDRESS TSO
PARSE ARG PARAM
SAY ' PARAMETERS: ' PARAM
CXNAME = SUBSTR(PARAM,1,8)
ENV    = SUBSTR(PARAM,10,5)
VERI   = SUBSTR(PARAM,16,4)
GRPN   = SUBSTR(PARAM,21,8)
TIPO   = SUBSTR(PARAM,30,3)
GRPN = SPACE(GRPN,0)
CONT = 0
'EXECIO * DISKR INP ( STEM VAR.'
CALL SCRIVJ
IF TIPO = 'PGM'  THEN CALL CSDPGM
   ELSE  CALL  CSDMAP
SAY ' - ' CONT 'RESOURCES ARE BEEN BUILT'
EXIT
CSDPGM:
DO J=1 TO VAR.0
/*DELSTACK*/
/*NEWSTACK*/
RIS = SUBSTR(VAR.J,3,10)
RISO = SPACE(RIS,0)
R.1 = ' DEFINE PROGRAM('RISO') GROUP('GRPN')'
R.2 = '  LANGUAGE(COBOL) RELOAD(NO) RESIDENT(NO) USAGE(NORMAL)'
R.3 = '  USELPACOPY(NO) STATUS(ENABLED) CEDF(YES) DATALOCATION(BELOW)'
R.4 = '  EXECKEY(USER) EXECUTIONSET(FULLAPI)'
  CONT = CONT + 1
  DO X=1 TO 4
    R.X=SUBSTR(R.X,1,72)
      QUEUE R.X
      'EXECIO * DISKW OUT'
  END
 'EXECIO * DISKR INP ( STEM VAR.'
END
RETURN
CSDMAP:
DO J=1 TO VAR.0
RIS = SUBSTR(VAR.J,3,10)
RISO = SPACE(RIS,0)
R.1 = ' DEFINE MAPSET('RISO') GROUP('GRPN')'
R.2 = '  RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED)'
  CONT = CONT + 1
   DO X=1 TO 2
      R.X=SUBSTR(R.X,1,72)
      QUEUE R.X
      'EXECIO * DISKW OUT'
   END
'EXECIO * DISKR INP ( STEM VAR.'
END
```

```
RETURN
SCRIVJ:
USERI = USERID()
USERII = USERI'LI'
/*-- BUILD THE JCL FOR DFHCSDUP ------------------*/
J.1 = '//'USERII 'JOB CLASS=A,MSGCLASS=A,NOTIFY='USERI
J.2 = '//*-------------------------------------------------*'
J.3 = '//* ATTENTION: CHECK THE DSNAME OF DFHCSD   !!!     *'
J.4 = '//* ATTENTION: CHECK THE VERSION OF CICS    !!!     *'
J.5 = '//*-------------------------------------------------*'
J.6 = '//STEPØ1   EXEC PGM=DFHCSDUP'
J.7 = '//STEPLIB  DD  DSN='ENV'.'VERI'.SDFHLOAD,DISP=SHR'
J.8 = '//DFHCSD   DD  DSN='CXNAME'.'ENV'.'VERI'.DFHCSD,DISP=SHR'
J.9 = '//SYSPRINT DD  SYSOUT=*'
J.1Ø = '//SYSIN    DD *'
 DO X=1 TO 1Ø
 J.X=SUBSTR(J.X,1,72)
 QUEUE J.X
 'EXECIO * DISKW OUT'
 END
 RECO = 'ADD GROUP('GRPN')  LIST(LSTNAME)'
 QUEUE RECO
'EXECIO * DISKW OUT'
RETURN
```

$JOBCSD

```
//JOBNAME  JOB CLASS=A,MSGCLASS=A,REGION=2M,NOTIFY=&SYSUID
//*  THIS JOB BUILDS RDO STATEMENTS TO DEFINE PROGRAM AND MAPSET    *
//*  STARTING FROM PDS LIBRARY CONTAINING LOAD OR SOURCE MODULES ---*
//*- ATTENTION:                                                    --*
//*- BEFORE EXECUTING THIS JOB WE MUST MODIFY THE PARAMETERS BELOW  *
//*- CX = THE NAME OF REGION CICS (FIRST DSNAME QUALIFIER)     -----*
//*- ENV = CICS ENVIRNMENT (PRODUCTION OR TEST)--------------------*
//*- VERI = CICS VERSION -------------------------------------------*
//*- GRP  = RDO GROUP ----------------------------------------------*
//*- TIPO = BMS (MAPSET) PGM (PROGRAM) -----------------------------*
//ELABCSD PROC CX='',ENV='CICST',VERI='V41Ø',GRP='GRPNAME',TIPO='BMS'
//*--- CREATE RESOURCES TO BE DELETED FROM DFHCSD ---------*
//LISTPDS  EXEC PGM=IKJEFTØ1,DYNAMNBR=5ØØ
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD DSN=&&EXTRL,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(15,2)),
//    DCB=(LRECL=8Ø,RECFM=FB,BLKSIZE=24ØØØ)
//SYSUDUMP DD DUMMY
//SYSTSIN  DD DSN=USER.TSO.CNTL($LIST),DISP=SHR
//*- THIS STEP PRODUCES THE SEQUENTIAL DATASET CONTAINING THE ------*
//*- MEMBERS THAT WILL BE DELETED IN DFHCSD ------------------------*
//*                JCL TO EXECUTE A CLIST                          *
//DELCXPDS EXEC PGM=IKJEFTØ1,DYNAMNBR=1ØØ,
//         PARM=($CLNLIST)
//***-- THE SYST.CMDPROC.REXX CONTAIN THE $CLNLIST PROC -------------***
```

```
//SYSPROC  DD DISP=SHR,DSN=USER.CMDPROC.REXX
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//INPLIST  DD DSN=&&EXTRL,DISP=(OLD,DELETE)
//OUT    DD DSN=&&EXTRL2,DISP=(,PASS),UNIT=SYSDA,SPACE=(TRK,(15,2)),
//    DCB=(LRECL=80,RECFM=FB,BLKSIZE=24000)
//*OUT      DD  SYSOUT=B
//*- BUILD RDO MACRO ---------------------------------------------*
//*               JCL TO EXECUTE A CLIST                         *
//DELRICSD EXEC PGM=IKJEFT01,DYNAMNBR=100,
//         PARM=($DEFCMD,&CX,&ENV,&VERI,&GRP,&TIPO)
//*---- THE SYST.CMDPROC.REXX CONTAIN THE $DEFCMDS PROC ----------***
//SYSPROC  DD DISP=SHR,DSN=USER.CMDPROC.REXX
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN  DD DUMMY
//INP DD  DSN=&&EXTRL2,DISP=(OLD,DELETE)
//OUT DD  DSN=USER.TSO.CNTL(RDOCMD),DISP=SHR
//*
//   PEND
//*
//STEPTLTR EXEC ELABCSD,CX=CICSNAME
```

Marco Busichella
Systems Programmer (Italy)

Documenting CICS programs

We have more than 2,500 CICS programs in our production environment. Many of them are very old, without proper documentation, and their authors have left our company. That's why we write REXX procedures to generate simplified schemas from PL/I, COBOL, or Assembler source. The CICSDOC REXX procedure scans specified PDS or PDSE datasets and analyses each member in the following way:

- Recognizes CICS programs that are written in one of the specified languages; other members are skipped.
- Identifies program name and programming language.
- Searches for all EXEC CICS LINK and EXEC CICS XCTL statements and picks up the names of the called programs and calling methods.

The syntax of EXEC CICS LINK or XCTL statements enables the usage of either constants or variables to specify program names. When you use constants for program names, the report looks like:

```
Type|PgmName  |Method|SubPgm| Variable |CICS DSN
--------------------------------------------------------------------------
ASM |DFH$WBSA |LINK  |      |'DFH$WBST'|CICSTS12.CICS.SDFHSAMP(DFH$WBSA)
--------------------------------------------------------------------------
ASM |DFH$WBSC |LINK  |      |'DFH$WBST'|CICSTS12.CICS.SDFHSAMP(DFH$WBSC)
ASM |DFH$WBSC |LINK  |      |'DFHWBENV'|CICSTS12.CICS.SDFHSAMP(DFH$WBSC)
--------------------------------------------------------------------------
COB |DFHØBAT1.|LINK  |      |'DFHØBAT4'|CICSTS12.CICS.SDFHSAMP(DFHØBAT1)
--------------------------------------------------------------------------
```

Where variables are used, the procedure finds their values in declarations or in assignment statements. In that situation, the report has the following format:

```
-------------------------------------------------------------------------------
Type| PgmName |Method|SubPgm    |Variable      |CICS DSN
-------------------------------------------------------------------------------
COB |DFHØBAT2.| LINK |          |DPL-PROG-NAME |CICSTS12.CICS.SDFHSAMP(DFHØBAT2)
COB |DFHØBAT2.|      |'DFHØBAT7'|DPL-PROG-NAME |CICSTS12.CICS.SDFHSAMP(DFHØBAT2)
COB |DFHØBAT2.|      |'DFHØBAT5'|DPL-PROG-NAME |CICSTS12.CICS.SDFHSAMP(DFHØBAT2)
-------------------------------------------------------------------------------
PLI |DFHPXCC: |LINK  |          |TARGET_PROGRAM|CICSTS12.CICS.SDFHSAMP(DFH$PXCC)
PLI |DFHPXCC:        |'DFH$AXCS'|TARGET_PROGRAM|CICSTS12.CICS.SDFHSAMP(DFH$PXCC)
-------------------------------------------------------------------------------
COB |DFHØVZTR.|XCTL  |          |WØ1-SENDNAME  |CICSTS12.CICS.SDFHSAMP(DFHØVZTR)
COB |DFHØVZTR.|      |'DFHØVZTS'|WØ1-SENDNAME  |CICSTS12.CICS.SDFHSAMP(DFHØVZTR)
-------------------------------------------------------------------------------
```

We can imagine that a program uses one variable and assigns different program names to it. In cases where some of these programs are called by XCTL and others by LINK, the procedure cannot resolve the calling method for each program. In this situation, users have to assign an appropriate method manually.

The report about all members of the input library is placed in the dataset with the symbolic name CICSDOC. In the SYSPRINT dataset, you get the list of all scanned members classified as CICS if it contains any EXEC CICS statements, or non-CICS if there are no CICS statements.

EXAMPLE JOB TO SUBMIT CICSDOC

The following JCL can be used to submit a job:

```
//useridC   JOB CLASS=A,MSGCLASS=X,MSGLEVEL=(Ø,Ø),NOTIFY=&SYSUID
//CICSDOC   EXEC PGM=IKJEFTØ1,DYNAMNBR=5Ø,REGION=4M
//SYSPROC   DD  DSN=userid.USER.CLIST,DISP=SHR
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//CICSDOC   DD  DSN=userid.#CICDOC.LIST,DISP=(NEW,CATLG),
//          UNIT=SYSDA,DCB=(RECFM=FB,LRECL=1Ø5,BLKSIZE=Ø),
//          SPACE=(TRK,(1Ø,5),RLSE)
//SYSTSIN   DD  *
     %CICSDOC  CICSTS12.CICS.SDFHSAMP
/*
```

PROCEDURE CICSDOC

```
/****************************** REXX *********************************/
/* Procedure forms documentation for CICS programs                   */
/* Input: library with CICS programs                                 */
/* Output: Documentation in CICSDOC dataset and list of scanned      */
/*         members in SYSPRINT                                       */
/* For each program report is formed with the following information:  */
/*    * type (PL/I, COBOL, or Assembler)                             */
/*    * program name                                                 */
/*    * calling method  (xctl, link)                                 */
/*    * SubPgm                                                       */
/*    * Variable                                                     */
/*    * Dsname of member that contains program                       */
/* Trace ?R */
 ARG Arg1 Arg2
 userid=SYSVAR(SYSUID)
 prefix=SYSVAR(SYSPREF)
 "PROFILE NOPREFIX"
 rrc=Ø
 signal on error
 rrc = Make_CICS_Doc(Arg1,Arg2)
 error:
 If prefix <> ''
 Then "PROFILE PREFIX("prefix")"
Return rrc
/* Make CICS Doc                                                  */
Make_CICS_Doc: Procedure
Arg DsName, Volume
 rrc=Ø
 If SYSDSN(DsName) <> 'OK'
 Then Do
      Say 'Missing dataset name'
      rrc=12
      End
 Else Do
      findm=Ø
      Records.Ø=Ø
      t=OUTTRAP('dsnc.',,NOCONCAT)
```

```
      "LISTDS "DsName
      t=OUTTRAP('OFF')
      PARSE UPPER VAR dsnc.3 recfm lrecl blksize dsorg
      If dsorg = 'PS' OR (dsorg = 'PO' AND Index(DsName,'(') > Ø)
      Then Do
           rcu=Analyze_program(DsName, Volume)
           rrc=MAX(rrc,rcu)
           End
      Else
      If dsorg = 'PO'
      Then Do;
           t=OUTTRAP('dsnc.',,NOCONCAT)
           "LISTDS "DsName" members "
           t=OUTTRAP('OFF')
           Do i=1 To dsnc.Ø
              If INDEX(dsnc.i,'MEMBERS') > Ø
              Then Leave
           End
           "EXECIO Ø DISKW  CICSdoc  (OPEN)"
           call print_header 'cicsdoc'
           Do i=i+1 To dsnc.Ø
              Parse Var Dsnc.i Member Rest
              Ds_Name=DsName||'('||Member||')'
              rcu=Analyze_program(Ds_Name, Volume)
              rrc=MAX(rrc,rcu)
           End
           "EXECIO Ø DISKW  CICSdoc  (FINIS)"
           "EXECIO Ø DISKW  sysprint (OPEN)"
           LibName.1=DsName
           "EXECIO 1 DISKW  sysprint (STEM LibName.)"
           "EXECIO Ø DISKW  sysprint (FINIS)"
           End
      Else Do
           Say 'This Dsorg' dsorg ' is not supported !!!'
           rrc=16
           End
      End
Return rrc
/* Get dataset                                                      */
Analyze_program: Procedure Expose Records. Member
Arg Ds_Name, Volume
Call alloc_Ds 'CICS' Ds_Name Volume
rrc=Ø
"EXECIO Ø DISKR  CICS (OPEN)"
If RC <> Ø
Then Do
     Say '>>> Dataset' DS_name ' CANNOT BE OPENED !!!'
     EXIT 4
     End
"EXECIO * DISKR  CICS (STEM Records.)"
"EXECIO Ø DISKR  CICS (FINIS)"
 indCICS=Ø
```

```
 ValueVar.Ø=Ø
 TypePgm='   '
 Call analyze_records
 If TypePgm ¬= '   ' & SubPgm.Ø > Ø
 Then Call Print_Doc
 If indCICS=Ø
 Then Say '***' left(DS_name,44)   TypePgm 'Non CICS'
 Else Say '***' left(DS_name,44)   TypePgm 'CICS'
Return rrc
/* Analyse Record                                                  */
 Analyze_records: Procedure Expose PgmName Records. indCICS Ds_Name,
                            SubPgm. ValueVar. TypePgm
 PgmName      = ''
 Method       = left(' ',1Ø)
 indcomment   = 'N'
 indfirststmt = 'Y'
 label        = ''
 SubPgm.Ø     = Ø
 k=Ø
 Do i=1 to Records.Ø
    record = records.i
    If substr(record,1,1) ¬= '*'
    Then Do
         Do While(record ¬= '')
            PARSE VAR record keyword1 rest
            If substr(keyword1,1,2) = '/*'
            Then indcomment='Y'
            If indcomment='Y' & index(keyword1, '*/') > Ø
            Then indcomment='N'
            If indcomment = 'N'
            Then Do
                    Select
                    When keyword1 = 'EXEC'
                    Then indCICS  = 1
                    When indCICS = 1 & keyword1 = 'CICS'
                    Then Do
                         PARSE VAR rest Method rest
                         If Method = 'LINK' | Method = 'XCTL'
                         Then Do
                              indCICS  = 2
                              Method =left(Method,1Ø)
                              End
                         End
                    When indCICS  = 2 & SUBSTR(keyword1,1,4) = 'PROG'
                    Then Do
                         subparm = Get_subparm(keyword1, Method)
                         If subparm ¬= ''
                         Then Do
                              k=k+1
                              SubPgm.k = Subparm
                              SubPgm.Ø=k
                              End
```

```
                    End
/*----------------- PLI Programs ------------------------------- */
                    When index(keyword1,':PROC') > Ø ,
                       | substr(rest,1,4) = 'PROC'
                    Then Do
                         j=index(keyword1,':');
                         If j = Ø
                         Then j = length(keyword1)
                         label=substr(keyword1,1,j);
                         End
                    When keyword1 = ';' | right(keyword1,1) = ';'
                    Then Do
                         If indfirststmt = 'Y' & label ¬= ''
                         Then Do
                              TypePgm = 'PLI'
                              indfirststmt = 'N'
                              PgmName = label
                              End
                         End
/*----------------- Asm Programs ------------------------------- */
                    When SUBSTR(rest,1,5) = 'CSECT'
                    Then Do
                         TypePgm = 'ASM'
                         PgmName = keyword1
                         End
                    When TypePgm = '' & ,
                         (keyword1 = 'DS' | keyword1 = 'EQU')
                    Then TypePgm = 'ASM'
/*----------------- COBOL Programs ------------------------------ */
                    When keyword1 = 'PROGRAM-ID.'
                    Then Do
                         PARSE VAR rest keyword1 rest
                         TypePgm = 'COB'
                         PgmName = keyword1
                         End
                    OTHERWISE
                    End
                End
                record = rest
            End
      End
 End
 Return
/* Get subparameter                                                 */
 Get_subparm: Procedure Expose SubPgm. Records. ValueVar. TypePgm
 Arg parm, method
 i=index(parm,'(')
 j=index(parm,')')
 If j = Ø
 Then j=length(parm)
 If i > Ø & j > i
 Then Do
```



```
      If substr(parm,i+1,1) = "'"
      Then type='C'
      Else type='V'
      Pgm = substr(parm,i+1,j-i-1)
      subparm = type method  Pgm
      Do j=1 To SubPgm.Ø
         If SubPgm.j = subparm
         Then Do
              subparm = ''
              Leave
              End
      End
      Do j=1 To SubPgm.Ø
         If substr(SubPgm.j,8) = Pgm
         Then Leave
      End
      If  j > SubPgm.Ø & type = 'V'
      Then Call Get_var_value Pgm
      End
 Else subparm = ''
 return subparm
/* Get variable value                                                */
 Get_var_value: Procedure Expose Records. ValueVar. TypePgm
 Arg Var
 Do i=1 to Records.Ø
    j = index(records.i,Var)
    If j > Ø
    Then Do
         recrest = substr(records.i,j+length(Var))
         select
         When TypePgm = 'ASM'
         Then value = Get_var_value_ASM(recrest)
         When TypePgm = 'PLI'
         Then value = Get_var_value_PLI(recrest)
         When TypePgm = 'COB'
         Then value = Get_var_value_COB(var,records.i)
         Otherwise
         End
         If value ¬= ''
         Then Do
              k = Valuevar.Ø
              k=k+1
              Valuevar.k = value var
              Valuevar.Ø = k
              End
         End
 End
 Return
/* Get variable value for ASM program                                */
 Get_var_value_ASM: Procedure Expose Records. ValueVar.
 Arg record
 PARSE VAR record dcds val rest
```

```
 If dcds ¬= 'DC'
 Then val  = ''
 Else Do
      i=index(val,"'")
      val=Substr(val,i)
      End
 Return val
/* Get variable value for PLI program                              */
 Get_var_value_PLI: Procedure Expose Records. ValueVar.
 Arg record
 If substr(record,1,1) ¬= '='
 Then record = strip(record,'L')
 If substr(record,1,1) = '='
 Then Do
      record = substr(record,2)
      PARSE VAR record val rest
      If right(val,1) = ';'
      Then val = strip(val,'T',';')
      End
 Else Do
      PARSE VAR record k1 k2 k3
      init=''
      If SUBSTR(k1,1,4) = 'INIT'
      Then init=SUBSTR(k1,5)
      Else
      If SUBSTR(k2,1,4) = 'INIT'
      Then init=SUBSTR(k2,5)
      PARSE VAR init '(' val ')' rest
      End
 Return val
/* Get variable value for ASM program                              */
 Get_var_value_COB: Procedure Expose Records. ValueVar.
 Arg cicsvar,record
 PARSE VAR record move rest
 If move = 'MOVE'
 Then Do
      PARSE VAR rest val to var rest
      var=strip(var,'T','.')
      if cicsvar  ¬= var
      Then val=''
      End
 Else Do
      PARSE VAR rest k1 k2 k3 k4 k5 rest
      if k4 = 'VALUE'
      Then val = strip(k5,'T','.');
      Else val=''
      End
 Return val
/* print Doc                                                       */
 Print_Doc: Procedure Expose PgmName SubPgm. Ds_Name ValueVar. TypePgm
 k=Ø
 Do i=1 To SubPgm.Ø
```

```
    k=k+1
    If substr(SubPgm.i,1,1) = 'C'
    Then row.k = left(TypePgm,4) left(PgmName,1Ø),
                 left(substr(SubPgm.i,3),21),
                 left(' ',2Ø) left(Ds_Name,44)
    Else row.k = left(TypePgm,4) left(PgmName,1Ø),
                 left(substr(SubPgm.i,3),1Ø),
                 left(' ',1Ø) left(substr(SubPgm.i,14),2Ø),
                 left(Ds_Name,44)
 End
 Do j=1 To ValueVar.Ø
    k=k+1
    row.k = left(TypePgm,4) left(PgmName,1Ø) left(' ',1Ø),
            left(ValueVar.j,31) left(Ds_Name,44)
 End
 k=k+1
 row.k = copies('-',1Ø5)
 row.Ø=k
 "EXECIO * DISKW cicsdoc (STEM row.)"
 Drop row.
 return
/* Alloc Dataset                                                  */
 Alloc_DS: Procedure
 Arg DD_Name Ds_Name Volume
 msgstat=MSG("OFF")    /* Inhibit the display of TSO/E information */
 signal off error      /* messages */
 "FREE F("DD_Name")"
 signal on error
 t=MSG(msgstat)        /* Returns the previous status of message */.
 If Volume = ''
 Then "ALLOC F("DD_Name") DA("''''Ds_Name'''') SHR "
 Else "ALLOC F("DD_Name") DA("''''Ds_Name'''') SHR ",
              " VOLUME("Volume") UNIT(SYSDA)"
 Return
/* print header                                                   */
 Print_Header: Procedure
 Arg file
 row.1 = copies('-',1Ø5)
 row.2 = 'Type'||'|'||,
         CENTER('PgmName',1Ø)||'|'||,
         CENTER('Method',1Ø)||'|'||,
         CENTER('SubPgm',1Ø)||'|'||,
         CENTER('Variable',2Ø)||'|'||,
         CENTER('CICS DSN',44)
 row.3 = copies('-',1Ø5)
 "EXECIO 3 DISKW " file " (STEM row.)"
 Return
```

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

Creating or modifying BMS sources – part 2

This is the code for a utility to create or modify BMSs.

PANELBMS REXX SOURCE

```
/*  REXX MVS  *=============================================================*/
/*  PANELBMS - Creates or modifies CICS BMS source codes.                  */
/*  Optional argument: BMS source file.                                    */
/*  Outputs generated: BMS source and associated copybook.                 */
/*  This EXEC calls module PANELBØ. Modify variable mainprogram            */
/*  at the beginning of this EXEC to reflect its location.                 */
/*  This module must be created with the following source programs.         */
/*  Their hierarchy is as follows:                                         */
/*  PANELBØ                                                                */
/*     |----- PDISP                                                        */
/*     |----- IKJEFTSR                                                     */
/*     |----- PANELB5                                                      */
/*     |----- PANELB1------| PDISP                                         */
/*     |----- PANELB3      | PANELB2                                       */
/*     |----- PANELB4                                                      */
/*========================================================================*/
arg ficin .
mainprogram = "my.loadlib(PANELBØ)"
f11 = userid()".BMS.TEMP"
f22 = userid()".COPY.TEMP"
xx = msg(off)
call liberta
call alloc_new_file f1 f11
call alloc_new_file f2 f22
address tso "call '"mainprogram"'" ficin
do alpha = Ø
   execio 1 diskr f1
   if rc<>Ø then leave alpha
   pull linha
   out1 = word(linha,1)
   call alloc_file fout1 out1
   do forever
      execio 1 diskr f1
      if rc<>Ø then leave
      execio 1 diskw fout1
      if rc<>Ø then do
         say "Error "rc" writing " out1
         signal saida
      end
   end
```

```
   execio Ø diskw fout1 "(finis"
end
do beta  = Ø
   execio 1 diskr f2
   if rc<>Ø then leave beta
   pull linha
   out2 = word(linha,1)
   call alloc_file fout2 out2
   do forever
      execio 1 diskr f2
      if rc<>Ø then leave
      execio 1 diskw fout2
      if rc<>Ø then do
         say "Error "rc" writing " out2
         signal saida
      end
   end
   execio Ø diskw fout2 "(finis"
end
saida:
 call liberta
exit
liberta:
 "free  da('"out1"')"
 "free  da('"out2"')"
 "free  da('"f11"')"
 "free  da('"f22"')"
 "free  dd(f1)"
 "free  dd(f2)"
 "free  dd(fout1)"
 "free  dd(fout2)"
return
alloc_new_file:
 arg ddname dsname
 "alloc da('"dsname"') dd("ddname") new reuse,
        blksize(8ØØØ) lrecl(8Ø) recfm(f,b),
        dsorg(ps) space(4 2) tracks delete "
 if rc<>Ø then do say "Error "rc" allocating " dsname
    "Error "rc" allocating file " dsname
    signal saida
 end
return
alloc_file:
 arg ddname dsname
 "alloc da('"dsname"') dd("ddname") shr"
 if rc<>Ø then do
    "Error "rc" allocating file " dsname
    signal saida
 end
return
```

PANELB0 COBOL SOURCE

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PANELBØ.
* PANELBMS: Program to display the initial and final screen   *
*           and call the remaining programs.                  *
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77   K    PIC S9(4) COMP VALUE +Ø.
 77   I    PIC S9(4) COMP VALUE +Ø.
 77   X    PIC S9(4) COMP VALUE +Ø.
 77   Y    PIC S9(4) COMP VALUE +Ø.
 77   Z    PIC S9(4) COMP VALUE +Ø.
 77   PA4S PIC S9(4)      VALUE +Ø.
 77   PA4D PIC  9(4)      VALUE  Ø.
 COPY PANELZØ.
 Ø1  MSG-INP.
     Ø2   FILLER PIC X(22) VALUE "ENTER: Drawing  area  ".
     Ø2   FILLER PIC X(22) VALUE "        PF14: Defaults".
     Ø2   FILLER PIC X(22) VALUE "      PF15: Exit      ".
 Ø1  MSG-OUT.
     Ø2   FILLER PIC X(22) VALUE "ENTER: SAVE           ".
     Ø2   FILLER PIC X(22) VALUE SPACES.
     Ø2   FILLER PIC X(22) VALUE "PF3/15: CANCEL        ".
 Ø1  MSG-1.
     Ø2   FILLER PIC X(1Ø) VALUE SPACES.
     Ø2   FILLER PIC X(22) VALUE "===>> ERROR ALLOCATING".
     Ø2   FILLER PIC X(22) VALUE " INPUT FILE           ".
 Ø1  MSG-2.
     Ø2   FILLER PIC X(1Ø) VALUE SPACES.
     Ø2   FILLER PIC X(22) VALUE "===>> ERROR ALLOCATING".
     Ø2   FILLER PIC X(22) VALUE " BMS FILE             ".
 Ø1  MSG-3.
     Ø2   FILLER PIC X(1Ø) VALUE SPACES.
     Ø2   FILLER PIC X(22) VALUE "===>> ERROR ALLOCATING".
     Ø2   FILLER PIC X(22) VALUE " COPY FILE            ".
 Ø1  FTEMPØ1.
     Ø2   FTEMPR PIC X(44) VALUE SPACES.
     Ø2   FTEMP  REDEFINES FTEMPR PIC X OCCURS 44.
 Ø1  PARMS-ALLOC.
     Ø2   PA1    PIC S9(8) COMP VALUE +1.
     Ø2   PA3    PIC S9(8) COMP VALUE +Ø.
     Ø2   PA4    PIC S9(8) COMP VALUE +Ø.
     Ø2   PA5    PIC S9(8) COMP VALUE +Ø.
     Ø2   PA6    PIC S9(8) COMP VALUE +Ø.
     Ø2   PA2.
       Ø4 FILLER  PIC X(25) VALUE "ALLOC DD(FICIN1) SHR DA('".
       Ø4 FINP-1R PIC X(44).
       Ø4 FINP-1  REDEFINES FINP-1R PIC X OCCURS 44.
 Ø1  PARMS-DEALLOC.
     Ø2   PD1    PIC S9(8) COMP VALUE +1.
```

```
       Ø2   PD2    PIC X(15) VALUE "FREE DD(FICIN1)".
       Ø2   PD3    PIC S9(8) COMP VALUE +15.
       Ø2   PD4    PIC S9(8) COMP VALUE +Ø.
       Ø2   PD5    PIC S9(8) COMP VALUE +Ø.
       Ø2   PD6    PIC S9(8) COMP VALUE +Ø.
    COPY PANATRIB.
    COPY PANELTAB.
    LINKAGE SECTION.
    Ø1  ARGUMENTO.
       Ø2  ARGLEN     PIC S9(4) COMP.
       Ø2  ARGVALUE   PIC X(1ØØ).
    PROCEDURE DIVISION USING ARGUMENTO.
       IF ARGLEN > Ø
          MOVE ARGVALUE  TO FINPI.
       PERFORM INICIO-POSICAO
               VARYING I FROM 1 BY 1 UNTIL I > MXF.
       PERFORM DISPLAY-PANELZØ-INPUT.
       IF ( FINPI = SPACES OR = LOW-VALUES )
          GO TO DESENHO.
       MOVE SPACES TO FTEMPØ1 FINP-1R
       MOVE FINPI  TO FTEMPØ1
       PERFORM MOVE-NAME
       CALL "IKJEFTSR" USING PD1 PD2 PD3 PD4 PD5 PD6
       CALL "IKJEFTSR" USING PA1 PA2 PA3 PA4 PA5 PA6
       IF PA4 NOT = Ø
          MOVE PA4  TO PA4S
          MOVE PA4S TO PA4D
          DISPLAY MSG-1  PA4D
          GO TO TERMINAR
       ELSE
          CALL "PANELB5" USING TAB-CAMPOS.
    DESENHO.
       CALL "PANELB1" USING ATRIBUTOS TAB-CAMPOS.
       MOVE  +162Ø      TO  CUROUT
       MOVE  SPACES     TO  MSGBI
       MOVE  ALARME-OFF TO  ALARME.
    OUTPUT-SELECT.
       PERFORM DISPLAY-PANELZØ-OUTPUT
       IF NOT ( FBMSI = SPACES OR = LOW-VALUES )
          PERFORM OUTPUT-BMS.
       IF NOT ( FCOPYI = SPACES OR = LOW-VALUES )
          PERFORM OUTPUT-COPY.
       CALL "IKJEFTSR" USING PD1 PD2 PD3 PD4 PD5 PD6.
    TERMINAR.
       STOP RUN.
   *==================== Subroutines =========================*
    INICIO-POSICAO.
       MOVE SPACES TO CAMPOS(I)
       MOVE  5ØØØ  TO TAB-POS(I)
       MOVE  Ø     TO TAB-STOP(I).
    OUTPUT-BMS.
       MOVE SPACES TO FTEMPØ1 FINP-1R TITMAP
```

```
    MOVE FBMSI  TO FTEMPØ1
    PERFORM MOVE-NAME
    CALL "PANELB3" USING TAB-CAMPOS FBMSI.
OUTPUT-COPY.
    MOVE SPACES TO FTEMPØ1 FINP-1R TITMAP
    MOVE FCOPYI TO FTEMPØ1
    PERFORM MOVE-NAME
    CALL "PANELB4" USING TAB-CAMPOS FCOPYI.
MOVE-NAME.
    MOVE Ø TO K Y Z
    PERFORM MOVE-NAME-1 THRU MOVE-NAME-1-FIM
            VARYING X FROM 1 BY 1 UNTIL X > 44
    ADD   1  TO Y
    MOVE "'" TO FINP-1(Y)
    ADD   1  TO Y
    MOVE ")" TO FINP-1(Y).
    ADD  25  TO Y
    MOVE Y TO PA3.
MOVE-NAME-1.
    IF FTEMP(X) = SPACE OR FTEMP(X) = LOW-VALUE
       OR  FTEMP(X) = "'"
       GO TO MOVE-NAME-1-FIM.
    ADD 1 TO Y
    MOVE FTEMP(X) TO FINP-1(Y)
    IF Z = 1 AND FTEMP(X) NOT = ")"
       ADD 1 TO K
       MOVE FTEMP(X) TO TITMAP-R(K).
    IF FTEMP(X) = "("
       MOVE 1 TO Z.
MOVE-NAME-1-FIM.
    EXIT.
DISPLAY-PANELZØ-INPUT.
    MOVE PROT-DARK TO  F-OUTA  F-BMSA  F-COPYA
    MOVE  MSG-INP  TO MSGAI
    CALL "PDISP" USING PANELZØ.
    IF AIDKEY = PF3 OR AIDKEY = PF15
       GO TO TERMINAR.
DISPLAY-PANELZØ-OUTPUT.
    MOVE  PROT-DARK   TO  F-INPA  FINPA
    MOVE  PROT-FRSET  TO  F-OUTA  F-BMSA  F-COPYA
    MOVE  UNPROT      TO  FBMSA   FCOPYA
    MOVE  MSG-OUT     TO  MSGAI
    MOVE  FINPI       TO  FBMSI
    CALL "PDISP" USING PANELZØ.
    IF AIDKEY = PF3 OR AIDKEY = PF15
       GO TO TERMINAR.
```

PANELB1 COBOL SOURCE

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PANELB1.
```

```
* PANELBMS: Program to handle the drawing screen and          *
*           manipulate fields and their attributes.           *
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77    X              PIC S9(4)  COMP   VALUE +Ø.
 77    X1             PIC S9(4)  COMP   VALUE +Ø.
 77    X2             PIC S9(4)  COMP   VALUE +Ø.
 77    K              PIC S9(4)  COMP   VALUE +Ø.
 77    J              PIC S9(4)  COMP   VALUE +Ø.
 77    I              PIC S9(4)  COMP   VALUE +Ø.
 77    I-LIVRE        PIC S9(4)  COMP   VALUE +Ø.
 77    I-LAST         PIC S9(4)  COMP   VALUE +Ø.
 77    T              PIC S9(4)  COMP   VALUE +Ø.
 77    Y              PIC S9(4)  COMP   VALUE +Ø.
 77    Z              PIC S9(4)  COMP   VALUE +Ø.
 77    LNOME          PIC S9(4)  COMP   VALUE +7.
 77    LMDT           PIC S9(4)  COMP   VALUE +1.
 77    XMAX           PIC S9(4)  COMP   VALUE +Ø.
 77    BOT            PIC S9(4)  COMP   VALUE +Ø.
 77    ERRO           PIC S9(4)  COMP   VALUE +Ø.
 77    LTABLEN        PIC S9(4)  COMP   VALUE +1896.
 77    FC             PIC S9(4)  COMP   VALUE +Ø.
 77    C1             PIC 99            VALUE  Ø.
 77    C2             PIC 99            VALUE  Ø.
 77    PREVKEY        PIC X             VALUE SPACE.
 77    TIPO           PIC X             VALUE SPACE.
 77    CASEØ          PIC X             VALUE "M".
 77    CHARØ          PIC X             VALUE SPACE.
 77    CHAR1          PIC X             VALUE SPACE.
 77    FLAG-ATRIB     PIC 9             VALUE Ø.
 77    FIELD-FOUND    PIC 9             VALUE Ø.
 77    R              PIC 9             VALUE Ø.
 77    U              PIC 9             VALUE Ø.
 77    CAMPOS-TROCA   PIC X(125)        VALUE SPACES.
       COPY PANELZ1.
 LINKAGE SECTION.
       COPY PANATRIB.
       COPY PANELTAB.
 PROCEDURE DIVISION  USING ATRIBUTOS TAB-CAMPOS.
     MOVE  PROT  TO  DINIA DFIMA
     MOVE  " 1"  TO  DINII
     MOVE  "15"  TO  DFIMI
     MOVE  FILØ  TO  FILLI
     MOVE  TXTØ  TO  CP55I
     MOVE  NUMØ  TO  CP35I
     MOVE  UNPØ  TO  CP43I
     MOVE  PROØ  TO  CP49I
     MOVE  CASEØ TO  CASEI
     PERFORM PROTECT-BOT
     PERFORM UNPROTECT-TOP
             VARYING I FROM 1 BY 1 UNTIL I > 15
```

```
    PERFORM LOAD-INICIAL THRU LOAD-INICIAL-FIM
            VARYING X FROM 1 BY 1 UNTIL X > MXF.
    PERFORM TOP-PAGE.
DISPLAY-SCREEN.
    CALL "PDISP" USING PANELZ1.
    IF AIDKEY = CLEAR
       STOP RUN.
    MOVE AIDKEY TO PREVKEY
    IF ERRO = 1
       MOVE Ø TO ERRO
       MOVE ALARME-OFF TO ALARME
       MOVE PRIMEIRA-LINHA TO PRIMEIRA.
    PERFORM MOVE-LINES
            VARYING K FROM 1 BY 1 UNTIL K > 15
    IF FLAG-ATRIB = Ø
       CALL "PANELB2" USING CASEI LMDT
       IF NOT (CASEI = "M" OR CASEI = "U" )
          GO TO ERRO-CASE.
    IF FLAG-ATRIB = Ø AND CASEI = "U"
       CALL "PANELB2" USING L2479 LTABLEN
       PERFORM UNMOVE-LINES
               VARYING K FROM 1 BY 1 UNTIL K > 15.
    IF FILLI NOT = FILØ
       MOVE FILØ  TO CHARØ
       MOVE FILLI TO CHAR1
       PERFORM CHARØ-CHANGE-FILLER
       MOVE FILLI TO FILØ.
    IF CP55I NOT = TXTØ
       MOVE TXTØ  TO CHARØ
       MOVE CP55I TO CHAR1
       PERFORM CHARØ-CHANGE
       MOVE CP55I TO TXTØ.
    IF CP35I NOT = NUMØ
       MOVE NUMØ  TO CHARØ
       MOVE CP35I TO CHAR1
       PERFORM CHARØ-CHANGE
       MOVE CP35I TO NUMØ.
    IF CP43I NOT = UNPØ
       MOVE UNPØ  TO CHARØ
       MOVE CP43I TO CHAR1
       PERFORM CHARØ-CHANGE
       MOVE CP43I TO UNPØ.
    IF CP49I NOT = PROØ
       MOVE PROØ  TO CHARØ
       MOVE CP49I TO CHAR1
       PERFORM CHARØ-CHANGE
       MOVE CP49I TO PROØ.
    IF ERRO = 2
       PERFORM UNMOVE-LINES
               VARYING K FROM 1 BY 1 UNTIL K > 15.
    IF FLAG-ATRIB = 1
       PERFORM ATTRIBUTE-LOAD.
    IF AIDKEY = PF3 OR AIDKEY = PF15
```

```
           PERFORM TERMINAR.
        IF ERRO = 1
           GO TO DISPLAY-SCREEN.
        IF AIDKEY = PF4 OR AIDKEY = PF16
           PERFORM  ATTRIBUTE-OFF.
        IF AIDKEY = PF6 OR AIDKEY = PF18
           PERFORM  ATTRIBUTE-NEXT.
        IF AIDKEY = PF5 OR AIDKEY = PF17
           PERFORM  ATTRIBUTE-PREV.
        IF AIDKEY = PF7 OR AIDKEY = PF19
           PERFORM TOP-PAGE.
        IF AIDKEY = PF8 OR AIDKEY = PF2Ø
           PERFORM BOT-PAGE.
        GO TO DISPLAY-SCREEN.
     *=================  S U B R O U T I N E S ======================*
       ATTRIBUTE-ON.
          PERFORM  ATTRIBUTE-1 THRU  ATTRIBUTE-FIM
                   VARYING FC FROM 1 BY 1 UNTIL FC > 1895
          IF FIELD-FOUND = 1
             PERFORM  ATTRIBUTE-OLD
                      VARYING I FROM 1 BY 1 UNTIL I > MXF
             PERFORM ORDENAR
                      VARYING I FROM 1 BY 1 UNTIL I > MXF
                      AFTER   J FROM I BY 1 UNTIL J > MXF
             PERFORM PROTECT-TOP
                      VARYING I FROM 1 BY 1 UNTIL I > 15
             PERFORM UNPROTECT-BOT
             MOVE Ø TO I.
       ATTRIBUTE-OFF.
          MOVE 81 TO CUROUT
          MOVE Ø  TO FLAG-ATRIB  FIELD-FOUND
          PERFORM PROTECT-BOT
          PERFORM UNPROTECT-TOP
                   VARYING I FROM 1 BY 1 UNTIL I > 15.
       ATTRIBUTE-1.
          IF L1896(FC) = TXTØ OR L1896(FC) = NUMØ OR
             L1896(FC) = UNPØ OR L1896(FC) = PROØ
             MOVE 1 TO FIELD-FOUND
             PERFORM  ATTRIBUTE-2
                      VARYING I FROM 1 BY 1 UNTIL I > MXF.
       ATTRIBUTE-FIM.
          EXIT.
       ATTRIBUTE-2.
          IF TAB-POS(I) = FC
             GO TO ATTRIBUTE-FIM.
          IF TAB-POS(I) = 5ØØØ
             MOVE FC        TO TAB-POS(I)
             MOVE L1896(FC) TO TAB-TIPO(I)
             IF TAB-TIPO(I) = TXTØ
                MOVE  MDTØ-TXT TO  TAB-MDT(I)
                MOVE  BRTØ-TXT TO  TAB-BRT(I)
                MOVE  CORØ-TXT TO  TAB-COR(I)
                MOVE  EXTØ-TXT TO  TAB-EXT(I)
```

```
            GO TO ATTRIBUTE-FIM
         ELSE
            MOVE  MDTØ     TO  TAB-MDT(I)
            MOVE  BRTØ     TO  TAB-BRT(I)
            MOVE  CORØ     TO  TAB-COR(I)
            MOVE  EXTØ     TO  TAB-EXT(I)
            GO TO ATTRIBUTE-FIM.
 ATTRIBUTE-OLD.
    IF TAB-POS(I) NOT = 5ØØØ
       MOVE TAB-POS(I) TO FC
       IF NOT ( L1896(FC) = TXTØ OR L1896(FC) = NUMØ OR
                L1896(FC) = UNPØ OR L1896(FC) = PROØ )
          MOVE SPACES TO CAMPOS(I)
          MOVE 5ØØØ   TO TAB-POS(I).
ORDENAR.
    IF J > I AND TAB-POS(I) > TAB-POS(J)
       MOVE CAMPOS(I)    TO CAMPOS-TROCA
       MOVE CAMPOS(J)    TO CAMPOS(I)
       MOVE CAMPOS-TROCA TO CAMPOS(J).
 ATTRIBUTE-PREV.
    IF FLAG-ATRIB = Ø
       PERFORM  ATTRIBUTE-ON.
    IF FIELD-FOUND = Ø
       GO TO ERRO-NOFIELDS.
    MOVE 1 TO FLAG-ATRIB
    SUBTRACT 1 FROM I
    IF I < 1
       MOVE 1 TO I.
    PERFORM ATTRIBUTE-DISPLAY.
 ATTRIBUTE-NEXT.
    IF FLAG-ATRIB = Ø
       PERFORM  ATTRIBUTE-ON.
    IF FIELD-FOUND = Ø
       GO TO ERRO-NOFIELDS.
    MOVE 1 TO FLAG-ATRIB
    ADD  1 TO I.
    IF I > MXF
       MOVE 1 TO I.
    IF TAB-POS(I) > 3ØØØ
       MOVE 1 TO I.
    IF TAB-TIPO(I) NOT = SPACE
       PERFORM ATTRIBUTE-DISPLAY.
 ATTRIBUTE-DISPLAY.
    DIVIDE TAB-POS(I) BY 79 GIVING C1 REMAINDER C2
    ADD 1 TO C1
    MOVE  C1           TO  LINXI
    MOVE  C2           TO  COLXI
    MOVE  TAB-MDT (I)  TO  MDTI
    MOVE  TAB-NOME(I)  TO  NOMEI
    MOVE  TAB-BRT (I)  TO  BRTI
    MOVE  TAB-COR (I)  TO  CORI
    MOVE  TAB-EXT (I)  TO  EXTI
    MOVE  TAB-LENG(I)  TO  LENXI
```

```
    IF TAB-TIPO(I) = TXTØ
       MOVE 1554 TO CUROUT
       MOVE PROT-FRSET TO NOMEA MDTA
       MOVE PROT-DARK  TO CP27A CP29A CP62A
    ELSE
       MOVE 1448 TO CUROUT
       MOVE UNPROT-FRSET TO NOMEA MDTA
       MOVE PROT         TO CP27A CP29A CP62A.
 ATTRIBUTE-LOAD.
    IF TAB-TIPO(I) NOT = TXTØ
       IF NOMEI = SPACES OR NOMEI = LOW-VALUES
          GO TO ERRO-NOME
       ELSE
          CALL "PANELB2" USING NOMEI LNOME
          PERFORM NOME-REPETIDO
                  VARYING X FROM 1 BY 1 UNTIL X > MXF.
    CALL "PANELB2" USING MDTI LMDT
    CALL "PANELB2" USING BRTI LMDT
    CALL "PANELB2" USING CORI LMDT
    CALL "PANELB2" USING EXTI LMDT
    IF NOT (MDTI = "Y" OR MDTI = "N" OR MDTI = " ")
       GO TO ERRO-MDT.
    IF NOT (BRTI = "B" OR BRTI = "D" OR BRTI = " ")
       GO TO ERRO-BRT.
    IF NOT (EXTI = "B" OR EXTI = "R" OR EXTI = "U"
      OR EXTI = " ")
       GO TO ERRO-EXT.
    IF NOT (CORI = "B" OR CORI = "R" OR CORI = "P"
      OR CORI = "G" OR CORI = "T" OR CORI = "Y"
      OR CORI = "W" OR CORI = " " )
       GO TO ERRO-COR.
    MOVE  MDTI  TO  TAB-MDT (I)
    MOVE  NOMEI TO  TAB-NOME(I)
    MOVE  BRTI  TO  TAB-BRT (I)
    MOVE  CORI  TO  TAB-COR (I)
    MOVE  EXTI  TO  TAB-EXT (I)
    IF TAB-TIPO(I) = TXTØ
       MOVE 1554 TO CUROUT
    ELSE
       MOVE 1448 TO CUROUT.
NOME-REPETIDO.
    IF ( I NOT = X ) AND NOMEI = TAB-NOME(X)
       GO TO ERRO-NOME-REP.
TOP-PAGE.
    MOVE  Ø   TO BOT
    MOVE " 1" TO DINII
    MOVE "15" TO DFIMI
    PERFORM UNMOVE-LINES
            VARYING K FROM 1 BY 1 UNTIL K > 15.
BOT-PAGE.
    MOVE  9   TO BOT
    MOVE "1Ø" TO DINII
    MOVE "24" TO DFIMI
```

```
    PERFORM UNMOVE-LINES
            VARYING K FROM 1 BY 1 UNTIL K > 15.
MOVE-LINES.
    ADD K BOT GIVING T
    MOVE LTABI(K) TO LT(T).
UNMOVE-LINES.
    ADD K BOT GIVING T
    MOVE LT(T) TO LTABI(K).
CHARØ-CHANGE-FILLER.
    IF CHAR1 = SPACE OR CHAR1 = LOW-VALUE
       GO TO ERRO1.
    PERFORM CHARØ-PESQUISA
            VARYING Y FROM 1 BY 1 UNTIL Y > 1896.
    PERFORM CHARØ-ALTER
            VARYING Y FROM 1 BY 1 UNTIL Y > 1896.
CHARØ-CHANGE.
    IF CHAR1 = SPACE OR CHAR1 = LOW-VALUE
       GO TO ERRO1.
    PERFORM CHARØ-PESQUISA
            VARYING Y FROM 1 BY 1 UNTIL Y > 1896.
    PERFORM CHARØ-ALTER
            VARYING Y FROM 1 BY 1 UNTIL Y > 1896.
    PERFORM CHARØ-ALTER-TAB
            VARYING Y FROM 1 BY 1 UNTIL Y > MXF.
CHARØ-PESQUISA.
    IF L1896(Y) = CHAR1
       GO TO ERRO2.
CHARØ-ALTER.
    IF L1896(Y) = CHARØ
       MOVE CHAR1 TO L1896(Y).
       MOVE 2 TO ERRO.
CHARØ-ALTER-TAB.
    IF TAB-TIPO(Y) = CHARØ
       MOVE CHAR1 TO TAB-TIPO(Y).
PROTECT-BOT.
    MOVE PROT-DARK    TO CP27A  CP29A  CP31A  CP36A  CP38A
                  CP63A CP24A  CP4ØA  CP44A  CP6ØA  CP61A
                        CP5ØA  CP62A  NOMEA  MDTA   CORA
                        BRTA   LINXA  COLXA  LENXA  EXTA
    MOVE UNPROT-FRSET TO CASEA CP55A CP35A CP43A CP49A FILLA.
UNPROTECT-BOT.
    MOVE PROT         TO CP27A  CP29A  CP31A  CP36A  CP38A
                         CP24A  CP4ØA  CP44A  CP6ØA  CP61A
                         CP62A  CP63A
    MOVE UNPROT-FRSET TO NOMEA  MDTA   CORA   EXTA   BRTA
    MOVE PROT-FRSET   TO CASEA  CP55A  CP35A  CP43A  CP49A
                         CP5ØA  FILLA  LINXA  COLXA  LENXA.
PROTECT-TOP.
    MOVE PROT-FRSET TO LTABA(I).
UNPROTECT-TOP.
    MOVE UNPROT-FRSET TO LTABA(I).
TERMINAR.
    PERFORM ATTRIBUTE-ON
```

```
    IF FIELD-FOUND = 1
       PERFORM TERMINAR-1
               VARYING X FROM 1 BY 1 UNTIL TAB-POS(X) = 5000
       PERFORM STOP-BYTES
               VARYING X FROM 1 BY 1 UNTIL TAB-POS(X) = 5000.
    GOBACK.
TERMINAR-1.
    MOVE  TAB-POS(X) TO FC
    MOVE  0 TO I
    PERFORM TERMINAR-2 THRU TERMINAR-2-FIM.
TERMINAR-2.
    ADD 1 TO FC I
    IF     L1896(FC) = TXT0 OR L1896(FC) = NUM0
        OR L1896(FC) = UNP0 OR L1896(FC) = PRO0
        OR FC = 79   OR FC = 158  OR FC = 237  OR FC = 316
        OR FC = 395  OR FC = 474  OR FC = 553  OR FC = 632
        OR FC = 711  OR FC = 790  OR FC = 869  OR FC = 632
        OR FC = 711  OR FC = 790  OR FC = 869  OR FC = 948
        OR FC = 1027 OR FC = 1106 OR FC = 1185 OR FC = 1264
        OR FC = 1343 OR FC = 1422 OR FC = 1501 OR FC = 1580
        OR FC = 1659 OR FC = 1738 OR FC = 1817 OR FC = 1896
        GO TO TERMINAR-2A.
    MOVE L1896(FC) TO TAB-TEXT-R (X, I)
    GO TO TERMINAR-2.
TERMINAR-2A.
    IF TAB-TEXT-R(X, I) = SPACE
    OR TAB-TEXT-R(X, I) = LOW-VALUE
       SUBTRACT 1 FROM I
       GO TO TERMINAR-2A.
TERMINAR-2-FIM.
    MOVE I TO TAB-LENG(X).
STOP-BYTES.
    IF TAB-TIPO(X) NOT = TXT0
       ADD TAB-POS(X) TAB-LENG(X) GIVING X1
       ADD 1 X GIVING X2
       ADD 2 TO X1
       IF X1 < TAB-POS(X2)
          SUBTRACT 1 FROM X1
          MOVE X1 TO TAB-STOP(X).
LOAD-INICIAL.
    IF TAB-POS(X) = 5000
       MOVE 3000 TO X
       GO TO LOAD-INICIAL-FIM.
    MOVE TAB-POS(X)  TO X1
    MOVE TAB-TIPO(X) TO L1896(X1)
    MOVE TAB-LENG(X) TO X2
    PERFORM LOAD-INICIAL-TEXTO
            VARYING K FROM 1 BY 1 UNTIL K > X2
    GO TO LOAD-INICIAL-FIM.
LOAD-INICIAL-TEXTO.
    ADD 1 TO X1
    MOVE TAB-TEXT-R(X, K) TO L1896(X1).
LOAD-INICIAL-FIM.
```

```
    EXIT.
ERRO-NOFIELDS.
    PERFORM PREPARA-ERRO.
    MOVE "There are no fields to give attributes to" TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-CASE.
    PERFORM PREPARA-ERRO.
    MOVE +13Ø4 TO CUROUT
    MOVE "Case must be Mixed or Upper"  TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-MDT.
    PERFORM PREPARA-ERRO.
    MOVE +1474 TO CUROUT
    MOVE "MDT must be Yes or space" TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-BRT.
    PERFORM PREPARA-ERRO.
    MOVE +1554 TO CUROUT
    MOVE "BRT must be Bright Dark or space"   TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-EXT.
    PERFORM PREPARA-ERRO.
    MOVE +1714 TO CUROUT
    MOVE "Hilight: U(underline) R(reverse) B(blink) or space"
       TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-COR.
    PERFORM PREPARA-ERRO.
    MOVE +1634 TO CUROUT
    MOVE
    "Colors: Blue Red Pink Green Turq Yellow White or space"
       TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-NOME.
    PERFORM PREPARA-ERRO.
    MOVE +1448 TO CUROUT
    MOVE "Name is mandatory for data fields"  TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO-NOME-REP.
    PERFORM PREPARA-ERRO.
    MOVE +1448 TO CUROUT
    MOVE "Name already given to another field" TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO1.
    PERFORM PREPARA-ERRO.
    MOVE "Field indicator cannot be space" TO CP1I.
    GO TO DISPLAY-SCREEN.
ERRO2.
    PERFORM PREPARA-ERRO.
    PERFORM CURSOR-POSICAO.
    MOVE "Field indicator already exists at cursor position"
       TO CP1I.
```

```
           IF Z > 128Ø AND BOT = Ø
              PERFORM BOT-PAGE
              SUBTRACT 72Ø FROM CUROUT.
           IF Z NOT > 128Ø AND BOT > Ø
              PERFORM TOP-PAGE.
           GO TO DISPLAY-SCREEN.
       PREPARA-ERRO.
           MOVE 1           TO ERRO
           MOVE SPACES      TO CP1I
           MOVE UNDERLINE   TO CP1E
           MOVE WHITE       TO CP1C
           MOVE ALARME-ON   TO ALARME.
       CURSOR-POSICAO.
           ADD 8Ø Y GIVING Z
           DIVIDE Z BY 79 GIVING Z
           ADD  Y TO Z
           ADD 79 TO Z
           MOVE Z TO CUROUT.
```

PANELB2 ASSEMBLER SOURCE

```
*  PANELBMS: This program translates a string to uppercase.          *
*  Parameters:  P1 - String                                          *
*               P2 - Length (Halfword)                               *
PANELB2  CSECT
         STM   R14,R12,12(R13)
         LR    R12,R15
         USING PANELB2,R12
         LR    R2,R1
         L     R4,Ø(Ø,R2)         R4: string
         L     R5,4(Ø,R2)         R5: length
         LH    R5,Ø(Ø,R5)
         LA    R1Ø,TABCHAR
         SR    R7,R7
MOVCHAR  LR    R11,R1Ø
         IC    R7,Ø(Ø,R4)         Put char in R7
         AR    R11,R7             Add to tabchar base address
         MVC   Ø(1,R4),Ø(R11)     Move from table to buffer
         A     R4,=F'1'           Inc pointer
         BCT   R5,MOVCHAR         Loop for length
         LM    R14,R12,12(R13)    Return
         SR    R15,R15
         BR    R14
TABCHAR  DS    ØF
         DC    X'ØØØ1Ø2Ø3Ø4Ø5Ø6Ø7Ø8Ø9ØA0B0C0D0E0F'
         DC    X'1Ø111213141516171819ØA1B1C1D1E1F'
         DC    X'2Ø212223242526272829ØA2B2C2D2E2F'
         DC    X'3Ø313233343536373839ØA3B3C3D3E3F'
         DC    X'4Ø414243444546474849ØA4B4C4D4E4F'
         DC    X'5Ø515253545556575859ØA5B5C5D5E5F'
         DC    X'6Ø616263646566676869ØA6B6C6D6E6F'
```

```
  DC    X'70717273747576777879 7A7B7C7D7E7F'
  DC    X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
  DC    X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
  DC    X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
  DC    X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
  DC    X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
  DC    X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
  DC    X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
  DC    X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
  YREGS
  END
```

PANELB3 COBOL SOURCE

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PANELB3.
      * PANELBMS: Output BMS source to temporary file                *
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT FICIN1 ASSIGN TO F1
           FILE   STATUS IS FS1.
       DATA DIVISION.
       FILE SECTION.
       FD  FICIN1
           RECORDING MODE IS F
           BLOCK CONTAINS 0 RECORDS
           LABEL RECORD OMITTED.
       01  FICIN1-FD PIC X(80).
       WORKING-STORAGE SECTION.
       77  FS1        PIC 99          VALUE   0.
       77  LASTPOS    PIC 99          VALUE   0.
       77  X          PIC S9(4) COMP  VALUE  +0.
       77  LL         PIC S9(4) COMP  VALUE  +0.
       77  ASTERISCO  PIC X           VALUE  "*".
       01  LINHAS.
           02  LINHA-A.
            04 LINHA       PIC X(71) VALUE SPACES.
            04 LINHA-CONT  PIC X     VALUE SPACES.
            04 FILLER      PIC X(8)  VALUE SPACES.
           02  L1.
            04 FILLER PIC X(28) VALUE "MAPSET   DFHMSD TYPE=&SYSPAR".
            04 FILLER PIC X(28) VALUE "M,MODE=INOUT,CTRL=(FREEKB), ".
           02  L2.
            04 FILLER PIC X(26) VALUE "               LANG=COBOL,".
            04 FILLER PIC X(26) VALUE "TIOAPFX=YES,EXTATT=MAPONLY".
           02  L3.
            04 NOMEMP PIC X(9)  VALUE SPACES.
            04 FILLER PIC X(19) VALUE "DFHMDI SIZE=(24,80)".
           02  LFINAL.
            04 FILLER PIC X(9)  VALUE SPACES.
```

```
        04 FILLER PIC X(17) VALUE "DFHMSD TYPE=FINAL".
      02  LFIM.
        04 FILLER PIC X(9)  VALUE SPACES.
        04 FILLER PIC X(3)  VALUE "END".
      02  C0.
        04 FILLER PIC X     VALUE "*".
      02  L-POS.
        04 NOME   PIC X(9)  VALUE SPACES.
        04 FILLER PIC X(12) VALUE "DFHMDF POS=(".
        04 LIN    PIC 99    VALUE 0.
        04 FILLER PIC X     VALUE ",".
        04 COL    PIC 99    VALUE 0.
        04 FILLER PIC X(9)  VALUE "),LENGTH=".
        04 LEN    PIC 99    VALUE 0.
        04 FILLER PIC X(7)  VALUE ",ATTRB=".
        04 ATRIB  PIC X(25) VALUE SPACES.
      02  L-EXT.
        04 FILLER PIC X(15) VALUE SPACES.
        04 FILLER PIC X(8)  VALUE "HILIGHT=".
        04 EXT    PIC X(15) VALUE SPACES.
      02  L-COR.
        04 FILLER PIC X(15) VALUE SPACES.
        04 FILLER PIC X(6)  VALUE "COLOR=".
        04 COR    PIC X(15) VALUE SPACES.
      02  L-TEXTO1.
        04 FILLER PIC X(15) VALUE SPACES.
        04 FILLER PIC X(9)  VALUE "INITIAL='".
        04 TEXTO1 PIC X(47) VALUE SPACES.
        04 TEXTO1-C REDEFINES TEXTO1 PIC X OCCURS 47.
      02  L-TEXTO2.
        04 FILLER PIC X(15) VALUE SPACES.
        04 TEXTO2 PIC X(47) VALUE SPACES.
        04 TEXTO2-C REDEFINES TEXTO2 PIC X OCCURS 47.
LINKAGE SECTION.
COPY PANELTAB.
01  FBMSI              PIC X(55).
PROCEDURE DIVISION USING TAB-CAMPOS FBMSI.
    MOVE SPACES    TO  LINHA-A
    MOVE FBMSI     TO  LINHA
    OPEN OUTPUT FICIN1
    IF FS1 NOT = 0
       DISPLAY "ERROR OPENING FICIN1 " FS1
       GO TO RETORNAR.
    PERFORM WRITE-LINHA
    MOVE  SPACES    TO  LINHA-A
    MOVE  L1        TO  LINHA
    MOVE  ASTERISCO TO  LINHA-CONT
    PERFORM WRITE-LINHA
    MOVE  L2        TO  LINHA
    PERFORM WRITE-LINHA
    MOVE  C0        TO  LINHA
    PERFORM WRITE-LINHA
```

```
     MOVE  TITMAP    TO  NOMEMP
     MOVE  L3        TO  LINHA
     PERFORM WRITE-LINHA
     MOVE  CØ        TO  LINHA
     PERFORM WRITE-LINHA
     PERFORM WRITE-FIELD THRU WRITE-FIELD-FIM
             VARYING X FROM 1 BY 1 UNTIL X > MXF.
     MOVE  LFINAL    TO  LINHA
     PERFORM WRITE-LINHA
     MOVE  LFIM      TO  LINHA
     PERFORM WRITE-LINHA
     CLOSE FICIN1.
 RETORNAR.
     GOBACK.
*==================== Subroutines ==============================*
 WRITE-FIELD.
     IF TAB-POS(X) = 5ØØØ
        GO TO WRITE-FIELD-FIM.
     MOVE SPACES TO  ATRIB  EXT  COR  TEXTO1  TEXTO2
     DIVIDE TAB-POS(X) BY 79 GIVING LIN REMAINDER COL
     IF COL > Ø
        ADD 1 TO LIN
     ELSE
       MOVE 79 TO COL.
     MOVE  TAB-NOME(X) TO  NOME
     MOVE  TAB-LENG(X) TO  LEN
     MOVE  TAB-EXT(X)  TO  EXT
     MOVE  TAB-COR(X)  TO  COR
     IF (TAB-TIPO(X) = NUMØ AND TAB-MDT(X) = "Y")
        IF TAB-BRT(X) = "B"
           MOVE "(NUM,FSET,BRT)," TO ATRIB
           MOVE 14 TO LASTPOS
        ELSE IF TAB-BRT(X) = "D"
           MOVE "(NUM,FSET,DRK)," TO ATRIB
           MOVE 14 TO LASTPOS
        ELSE IF TAB-BRT(X) = " "
           MOVE "(NUM,FSET)," TO ATRIB
           MOVE 1Ø TO LASTPOS.
     IF (TAB-TIPO(X) = NUMØ AND TAB-MDT(X) NOT = "Y")
        IF TAB-BRT(X) = "B"
           MOVE "(NUM,BRT)," TO ATRIB
           MOVE 9  TO LASTPOS
        ELSE IF TAB-BRT(X) = "D"
           MOVE "(NUM,DRK)," TO ATRIB
           MOVE 9  TO LASTPOS
        ELSE IF TAB-BRT(X) = " "
           MOVE "(NUM)," TO ATRIB
           MOVE 9  TO LASTPOS.
     IF (TAB-TIPO(X) = UNPØ AND TAB-MDT(X) = "Y")
        IF TAB-BRT(X) = "B"
           MOVE "(UNPROT,FSET,BRT)," TO ATRIB
           MOVE 17 TO LASTPOS
```

```
   ELSE IF TAB-BRT(X) = "D"
      MOVE "(UNPROT,FSET,DRK)," TO ATRIB
      MOVE 17 TO LASTPOS
   ELSE IF TAB-BRT(X) = " "
      MOVE "(UNPROT,FSET)," TO ATRIB
      MOVE 13 TO LASTPOS.
IF (TAB-TIPO(X) = UNPØ AND TAB-MDT(X) NOT = "Y")
   IF TAB-BRT(X) = "B"
      MOVE "(UNPROT,BRT)," TO ATRIB
      MOVE 12 TO LASTPOS
   ELSE IF TAB-BRT(X) = "D"
      MOVE "(UNPROT,DRK)," TO ATRIB
      MOVE 12 TO LASTPOS
   ELSE IF TAB-BRT(X) = " "
      MOVE "(UNPROT)," TO ATRIB
      MOVE 8  TO LASTPOS.
IF ((TAB-TIPO(X) = PROØ OR TAB-TIPO(X) = TXTØ)
     AND TAB-MDT(X) = "Y")
  IF TAB-BRT(X) = "B"
     MOVE "(ASKIP,PROT,FSET,BRT)," TO ATRIB
     MOVE 21 TO LASTPOS
  ELSE IF TAB-BRT(X) = "D"
     MOVE "(ASKIP,PROT,FSET,DRK)," TO ATRIB
     MOVE 21 TO LASTPOS
  ELSE IF TAB-BRT(X) = " "
     MOVE "(ASKIP,PROT,FSET)," TO ATRIB
     MOVE 17 TO LASTPOS.
IF ((TAB-TIPO(X) = PROØ OR TAB-TIPO(X) = TXTØ)
     AND TAB-MDT(X) NOT = "Y")
  IF TAB-BRT(X) = "B"
     MOVE "(ASKIP,PROT,BRT)," TO ATRIB
     MOVE 16 TO LASTPOS
  ELSE IF TAB-BRT(X) = "D"
     MOVE "(ASKIP,PROT,DRK)," TO ATRIB
     MOVE 16 TO LASTPOS
  ELSE IF TAB-BRT(X) = " "
     MOVE "(ASKIP,PROT)," TO ATRIB
     MOVE 12 TO LASTPOS.
IF TAB-IC(X) = 1
   MOVE ",IC)," TO ATRIB(LASTPOS:5).
IF TAB-TIPO(X) = TXTØ
  OR NOT (TAB-TIPO(X) = TXTØ OR TAB-TEXT-R(X, 1) = FILØ)
   MOVE TAB-TEXT-1(X) TO TEXTO1
   IF TAB-LENG(X) < 47
      MOVE TAB-LENG(X) TO LL
      ADD 1 TO LL
      MOVE "'" TO TEXTO1-C(LL)
   ELSE IF TAB-LENG(X) > 47
      MOVE TAB-TEXT-2(X) TO TEXTO2
      MOVE TAB-LENG(X) TO LL
      SUBTRACT 47 FROM LL
      ADD 1 TO LL
```

```
           MOVE "'" TO TEXTO2-C(LL)
         ELSE IF TAB-LENG(X) = 47
            MOVE "'" TO TEXTO2-C(1).
     MOVE ASTERISCO TO LINHA-CONT
     MOVE L-POS     TO LINHA
     PERFORM WRITE-LINHA.
 WRITE-EXT.
     IF EXT = SPACE
        MOVE "OFF,       " TO EXT.
     IF EXT = "B"
        MOVE "BLINK,     " TO EXT.
     IF EXT = "R"
        MOVE "REVERSE,   " TO EXT.
     IF EXT = "U"
        MOVE "UNDERLINE,"  TO EXT.
     MOVE ASTERISCO TO LINHA-CONT
     MOVE L-EXT     TO LINHA
     PERFORM WRITE-LINHA.
 WRITE-COR.
     IF TEXTO1 = SPACES
        PERFORM WRITE-COR-SPACE
     ELSE
        MOVE ASTERISCO TO LINHA-CONT
        PERFORM WRITE-COR-COMMA.
     MOVE L-COR TO LINHA
     PERFORM WRITE-LINHA.
 WRITE-COR-SPACE.
     IF COR = SPACE
         MOVE "DEFAULT   " TO COR
     ELSE IF COR = "B"
         MOVE "BLUE      " TO COR
     ELSE IF COR = "R"
         MOVE "RED       " TO COR
     ELSE IF COR = "G"
         MOVE "GREEN     " TO COR
     ELSE IF COR = "T"
         MOVE "TURQUOISE " TO COR
     ELSE IF COR = "Y"
         MOVE "YELLOW    " TO COR
     ELSE IF COR = "P"
         MOVE "PINK      " TO COR
     ELSE IF COR = "W"
         MOVE "NEUTRAL   " TO COR.
 WRITE-COR-COMMA.
     IF COR = SPACE
         MOVE "DEFAULT,  " TO COR
     ELSE IF COR = "B"
         MOVE "BLUE,     " TO COR
     ELSE IF COR = "R"
         MOVE "RED,      " TO COR
```

```
           ELSE IF COR = "G"
               MOVE "GREEN,    " TO COR
           ELSE IF COR = "T"
               MOVE "TURQUOISE," TO COR
           ELSE IF COR = "Y"
               MOVE "YELLOW,   " TO COR
           ELSE IF COR = "P"
               MOVE "PINK,     " TO COR
           ELSE IF COR = "W"
               MOVE "NEUTRAL,  " TO COR.
       WRITE-TEXTO.
           IF TEXTO2 NOT = SPACES
              MOVE ASTERISCO TO LINHA-CONT.
           IF TEXTO1 NOT = SPACES
              MOVE L-TEXTO1 TO LINHA
              PERFORM WRITE-LINHA.
           IF TEXTO2 NOT = SPACES
              MOVE L-TEXTO2 TO LINHA
              PERFORM WRITE-LINHA.
           MOVE CØ TO LINHA
           PERFORM WRITE-LINHA.
           IF TAB-STOP(X) NOT NUMERIC
              GO TO WRITE-FIELD-FIM.
           IF TAB-STOP(X) = Ø
              GO TO WRITE-FIELD-FIM.
           MOVE "(ASKIP,PROT)" TO ATRIB
           MOVE SPACES  TO NOME
           MOVE 1  TO LEN
           DIVIDE TAB-STOP(X) BY 79 GIVING LIN REMAINDER COL
           IF COL > Ø
              ADD 1 TO LIN
           ELSE
              MOVE 79 TO COL.
           MOVE L-POS TO LINHA
           PERFORM WRITE-LINHA.
       WRITE-FIELD-FIM.
           EXIT.
       WRITE-LINHA.
           WRITE FICIN1-FD FROM LINHA-A.
           MOVE SPACES TO LINHA-A.
```

PANELB4 COBOL SOURCE

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PANELB4.
      * PANELBMS: Output cobol copybook to temporary file          *
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT FICIN1 ASSIGN TO F2
           FILE STATUS IS FS1.
```

```
DATA DIVISION.
FILE SECTION.
FD  FICIN1
    RECORDING MODE IS F
    BLOCK CONTAINS Ø RECORDS
    LABEL RECORD OMITTED.
Ø1  FICIN1-FD PIC X(8Ø).
WORKING-STORAGE SECTION.
77    FS1      PIC 99          VALUE   Ø.
77    X        PIC S9(4) COMP  VALUE  +Ø.
77    Y        PIC S9(4) COMP  VALUE  +Ø.
77    Z        PIC S9(4) COMP  VALUE  +Ø.
Ø1  FTEMP-GERAL.
    Ø2  FTEMPØ1.
     Ø4 FTEMP  PIC X OCCURS 1Ø.
    Ø2   FTEMPØ2.
     Ø4 FTEMP2 PIC X OCCURS 1Ø.
Ø1  LINHAS.
    Ø2  LINHA.
     Ø4 FILLER      PIC X(6)  VALUE SPACES.
     Ø4 COMENTARIO  PIC X     VALUE SPACES.
     Ø4 ZONA-A      PIC X(1Ø) VALUE SPACES.
     Ø4 ZONA-B      PIC X(63) VALUE SPACES.
    Ø2  L-TIT1.
     Ø4 FILLER PIC X(14) VALUE "       Ø1     ".
     Ø4 TIT1   PIC X(1Ø) VALUE SPACES.
    Ø2  L-TIT2.
     Ø4 FILLER PIC X(14) VALUE "       Ø1     ".
     Ø4 TIT2   PIC X(1Ø) VALUE SPACES.
     Ø4 FILLER PIC X(12) VALUE " REDEFINES  ".
     Ø4 TIT3   PIC X(1Ø) VALUE SPACES.
    Ø2  L-FILLER12.
     Ø4 FILLER PIC X(23) VALUE "Ø5     FILLER          ".
     Ø4 FILLER PIC X(12) VALUE "PIC   X(12).".
    Ø2  L-FILLER3.
     Ø4 FILLER PIC X(23) VALUE "Ø5     FILLER          ".
     Ø4 FILLER PIC X(12) VALUE "PIC   X(Ø3).".
    Ø2  L-L.
     Ø4 FILLER PIC X(Ø7) VALUE "Ø5     ".
     Ø4 NOMEL  PIC X(1Ø) VALUE SPACES.
     Ø4 FILLER PIC X(19) VALUE "COMP  PIC   S9(4).".
    Ø2  L-F.
     Ø4 FILLER PIC X(Ø7) VALUE "Ø5     ".
     Ø4 NOMEF  PIC X(1Ø) VALUE SPACES.
     Ø4 FILLER PIC X(19) VALUE "      PIC   X(Ø1).".
    Ø2  L-R.
     Ø4 FILLER PIC X(24) VALUE "Ø5    FILLER REDEFINES  ".
     Ø4 NOMER  PIC X(1Ø) VALUE SPACES.
    Ø2  L-A.
     Ø4 FILLER PIC X(Ø7) VALUE "   Ø7  ".
     Ø4 NOMEA  PIC X(1Ø) VALUE SPACES.
     Ø4 FILLER PIC X(19) VALUE "      PIC   X(Ø1).".
    Ø2  L-I.
```

```
      04 FILLER PIC X(07) VALUE "05     ".
      04 NOMEI  PIC X(10) VALUE SPACES.
      04 FILLER PIC X(14) VALUE "      PIC   X(".
      04 LENGI  PIC 99.
      04 FILLER PIC X(02) VALUE ").".
     02  L-O.
      04 FILLER PIC X(07) VALUE "05     ".
      04 NOMEO  PIC X(10) VALUE SPACES.
      04 FILLER PIC X(14) VALUE "      PIC   X(".
      04 LENGO  PIC 99.
      04 FILLER PIC X(02) VALUE ").".
     02  C0.
      04 FILLER PIC X(7)  VALUE "      *".
LINKAGE SECTION.
COPY PANELTAB.
01   FCOPYI    PIC X(55).
PROCEDURE DIVISION  USING  TAB-CAMPOS  FCOPYI.
    OPEN OUTPUT FICIN1.
    IF FS1 NOT = 0
       DISPLAY "ERRO DE ABERTURA DE FICIN1 " FS1
       GO TO RETORNAR.
    MOVE  SPACES       TO LINHA
    MOVE  FCOPYI       TO LINHA
    PERFORM WRITE-LINHA
    MOVE  SPACES       TO FTEMP01 FTEMP02
    MOVE  TITMAP       TO FTEMP01
    PERFORM MOVE-NAME
    MOVE  "I"          TO FTEMP2(Y)
    ADD   1  TO Y
    MOVE  "."          TO FTEMP2(Y)
    MOVE  FTEMP02      TO TIT1 TIT3
    MOVE  SPACE        TO FTEMP2(Y)
    SUBTRACT 1 FROM  Y
    MOVE  "O"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO TIT2
    MOVE  L-TIT1       TO LINHA
    PERFORM WRITE-LINHA
    MOVE  C0           TO LINHA
    PERFORM WRITE-LINHA
    MOVE  L-FILLER12   TO ZONA-B
    PERFORM WRITE-LINHA
    PERFORM WRITE-INPUT THRU WRITE-INPUT-FIM
            VARYING X FROM 1 BY 1 UNTIL X > MXF.
    MOVE  C0           TO LINHA
    PERFORM WRITE-LINHA
    MOVE  L-TIT2       TO LINHA
    PERFORM WRITE-LINHA
    MOVE  C0           TO LINHA
    PERFORM WRITE-LINHA
    MOVE  L-FILLER12   TO ZONA-B
    PERFORM WRITE-LINHA
    PERFORM WRITE-OUTPUT THRU WRITE-OUTPUT-FIM
            VARYING X FROM 1 BY 1 UNTIL X > MXF.
```

```
    CLOSE FICIN1.
 RETORNAR.
    GOBACK.
*====================== Subroutines ==============================*
 WRITE-INPUT.
    IF TAB-POS(X)  = 5000
       GO TO WRITE-INPUT-FIM.
    IF TAB-TIPO(X) = TXT0
       GO TO WRITE-INPUT-FIM.
    MOVE  SPACES       TO FTEMP01 FTEMP02
    MOVE  TAB-LENG(X)  TO LENGI
    MOVE  TAB-NOME(X)  TO FTEMP01
    PERFORM MOVE-NAME
    MOVE  "L"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEL
    MOVE  "A"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEA
    MOVE  "I"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEI
    MOVE  "O"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEO
    MOVE  "F"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEF
    ADD 1 TO Y
    MOVE  "."          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMER
    MOVE  L-L          TO ZONA-B
    PERFORM WRITE-LINHA
    MOVE  L-F          TO ZONA-B
    PERFORM WRITE-LINHA
    MOVE  L-R          TO ZONA-B
    PERFORM WRITE-LINHA
    MOVE  L-A          TO ZONA-B
    PERFORM WRITE-LINHA
    MOVE  L-I          TO ZONA-B
    PERFORM WRITE-LINHA
    PERFORM WRITE-LINHA.
 WRITE-INPUT-FIM.
    EXIT.
 WRITE-OUTPUT.
    IF TAB-POS(X)  = 5000
       GO TO WRITE-OUTPUT-FIM.
    IF TAB-TIPO(X) = TXT0
       GO TO WRITE-OUTPUT-FIM.
    MOVE  SPACES       TO FTEMP01 FTEMP02
    MOVE  TAB-LENG(X)  TO LENGO
    MOVE  TAB-NOME(X)  TO FTEMP01
    MOVE  L-FILLER3    TO ZONA-B
    PERFORM WRITE-LINHA
    PERFORM MOVE-NAME
    MOVE  "O"          TO FTEMP2(Y)
    MOVE  FTEMP02      TO NOMEO
```

```
           MOVE  L-O          TO ZONA-B
           PERFORM WRITE-LINHA.
       WRITE-OUTPUT-FIM.
            EXIT.
       WRITE-LINHA.
           WRITE FICIN1-FD FROM LINHA
           MOVE SPACES TO LINHA.
       MOVE-NAME.
           MOVE Ø  TO Y
           PERFORM MOVE-NAME-1
                   VARYING Z FROM 1 BY 1 UNTIL Z > 8
           ADD 1 TO Y.
       MOVE-NAME-1.
           IF NOT ( FTEMP(Z) = SPACE OR FTEMP(Z) = LOW-VALUE )
              ADD 1 TO Y
              MOVE FTEMP(Z) TO FTEMP2(Y).
```

PANELB5 COBOL SOURCE

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PANELB5.
      * PANELBMS: Read input BMS file.                                 *
       ENVIRONMENT DIVISION.
       INPUT-OUTPUT SECTION.
       FILE-CONTROL.
           SELECT FICIN1 ASSIGN TO FICIN1
           FILE STATUS IS FS1.
       DATA DIVISION.
       FILE SECTION.
       FD  FICIN1
           RECORDING MODE IS F
           BLOCK CONTAINS Ø RECORDS
           LABEL RECORD OMITTED.
       Ø1  FICIN1-FD PIC X(8Ø).
       WORKING-STORAGE SECTION.
       77    MAC       PIC 9           VALUE   Ø.
       77    FS1       PIC 99          VALUE   Ø.
       77    L         PIC S9(4) COMP  VALUE  +Ø.
       77    X         PIC S9(4) COMP  VALUE  +Ø.
       77    Y         PIC S9(4) COMP  VALUE  +Ø.
       77    Z         PIC S9(4) COMP  VALUE  +Ø.
       77    Z1        PIC S9(4) COMP  VALUE  +Ø.
       77    Z2        PIC S9(4) COMP  VALUE  +Ø.
       Ø1   TEMPORARY-FIELDS.
          Ø2   CAMPOS-TEMP.
            Ø4   T-POS       PIC 9999  VALUE  Ø.
            Ø4   T-LENG      PIC 99    VALUE  Ø.
            Ø4   T-NOME      PIC X(7)  VALUE  SPACES.
            Ø4   T-TIPO      PIC X     VALUE  SPACES.
            Ø4   T-MDT       PIC X     VALUE  SPACES.
            Ø4   T-BRT       PIC X     VALUE  SPACES.
            Ø4   T-COR       PIC X     VALUE  SPACES.
```

```
     Ø4   T-EXT       PIC X     VALUE  SPACES.
     Ø4   T-STOP      PIC 9999  VALUE  Ø.
     Ø4   T-TEXT-R    PIC X(78) VALUE  SPACES.
     Ø4   T-TEXT REDEFINES T-TEXT-R  PIC X OCCURS 78.
     Ø4   T-IC        PIC X.
Ø1  LINHAS.
    Ø2  LINHA.
      Ø4   NOME       PIC X(8)  VALUE SPACES.
      Ø4   FILLER REDEFINES NOME.
        Ø6 COMENTARIO PIC X.
        Ø6 FILLER     PIC X(7).
      Ø4   FILLER     PIC X     VALUE SPACES.
      Ø4   MACROS     PIC X(6)  VALUE SPACES.
      Ø4   PARMS      PIC X(57) VALUE SPACES.
      Ø4   PAR REDEFINES PARMS PIC X OCCURS 57.
      Ø4   FILLER     PIC X(8)  VALUE SPACES.
    Ø2  LIN           PIC 99.
    Ø2  FILLER REDEFINES LIN.
      Ø4   L1         PIC X.
      Ø4   L2         PIC X.
    Ø2  COL           PIC 99.
    Ø2  FILLER REDEFINES COL.
      Ø4   C1         PIC X.
      Ø4   C2         PIC X.
LINKAGE SECTION.
COPY PANELTAB.
PROCEDURE DIVISION   USING TAB-CAMPOS.
    OPEN INPUT FICIN1.
    IF FS1 NOT = Ø
       DISPLAY "Error opening ficin1. File Status " FS1
       STOP RUN.
    MOVE SPACES TO CAMPOS-TEMP.
READ-NEXT.
    PERFORM LEITURA.
    IF COMENTARIO = "*"
       GO TO READ-NEXT.
    IF MACROS = "DFHMDI"
       MOVE NOME TO TITMAP
       GO TO READ-NEXT.
    IF NOT (( MACROS = "DFHMDF" OR MACROS = "END") AND MAC = 1)
       GO TO READ-NEXT-1.
    IF (T-NOME = SPACES AND T-TEXT(1) = FILØ AND T-LENG = 1)
       MOVE SPACES TO CAMPOS-TEMP
       GO TO READ-NEXT-1.
    ADD 1 TO X
    IF X > 3ØØ
       DISPLAY "BMS exceeds limit of 3ØØ fields"
       STOP RUN.
    MOVE CAMPOS-TEMP TO CAMPOS(X)
    MOVE SPACES      TO CAMPOS-TEMP
    IF TAB-NOME(X) = SPACES
        MOVE TXTØ    TO TAB-TIPO(X).
```

```
    IF TAB-TIPO(X) = SPACES
        MOVE PROØ    TO TAB-TIPO(X).
READ-NEXT-1.
    IF MACROS = "DFHMDF"
       MOVE NOME TO T-NOME
       MOVE 1    TO MAC.
    IF MACROS = "END   "
       GO TO RETORNAR.
    MOVE 1 TO Z1
    MOVE 2 TO Z2.
ANALISE.
    IF PARMS(Z1:3) = "POS"
       GO TO POSICAO.
    IF PARMS(Z1:3) = "LEN"
       GO TO COMPRIMENTO.
    IF PARMS(Z1:3) = "ATT"
       GO TO ATRIBUTOS.
    IF PARMS(Z1:3) = "COL"
       GO TO COLOUR.
    IF PARMS(Z1:3) = "HIL"
       GO TO HILIGHT.
    IF PARMS(Z1:3) = "INI"
       GO TO VALOR-INICIAL.
    IF PAR(Z1) = SPACE AND Z1 = 1
       ADD 1 TO Z1 Z2
       GO TO ANALISE.
    GO TO READ-NEXT.
POSICAO.
    ADD 5 TO Z1 Z2
    MOVE PAR(Z1) TO L1
    MOVE PAR(Z2) TO L2
    IF L2 = ","
       ADD 2 TO  Z1 Z2
       MOVE L1 TO L2
       MOVE ZERO TO L1
    ELSE
       ADD 3 TO  Z1 Z2.
    MOVE PAR(Z1) TO C1
    MOVE PAR(Z2) TO C2
    IF C2 = ")"
       ADD 3 TO Z1 Z2
       MOVE C1 TO C2
       MOVE ZERO TO C1
    ELSE
       ADD 4 TO Z1 Z2.
    SUBTRACT 1 FROM LIN
    MULTIPLY LIN BY 79 GIVING T-POS
    ADD COL TO  T-POS
    GO TO ANALISE.
COMPRIMENTO.
    ADD  7 TO  Z1 Z2
    MOVE PAR(Z1) TO  L1
```

```
    MOVE PAR(Z2) TO  L2
    IF L2 = ","
       ADD 2 TO Z1 Z2
       MOVE L1 TO L2
       MOVE ZERO TO L1
    ELSE
       ADD 3 TO Z1 Z2.
    MOVE LIN TO T-LENG
    PERFORM DATA-FILL
            VARYING L FROM 1 BY 1 UNTIL L > LIN
    GO TO ANALISE.
DATA-FILL.
    MOVE FILØ TO T-TEXT(L).
ATRIBUTOS.
    ADD 6 TO Z1 Z2.
ATRIBUTOS-1.
    IF PAR(Z1) = "("
       ADD 1 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "AS"
       ADD 6 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "UN"
       MOVE UNPØ TO T-TIPO
       ADD 7 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "PR"
       MOVE PROØ TO T-TIPO
       ADD 5 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "FS"
       MOVE "Y"  TO T-MDT
       ADD 5 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "FR"
       MOVE SPACE TO T-MDT
       ADD 6 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "NU"
       MOVE NUMØ  TO T-TIPO
       ADD 4 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "BR"
       MOVE "B"   TO T-BRT
       ADD 4 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "DR"
       MOVE "D"   TO T-BRT
       ADD 4 TO Z1 Z2
       GO TO ATRIBUTOS-1.
    IF PARMS(Z1:2) = "NO"
       MOVE SPACE TO T-BRT
```

```
           ADD 5 TO Z1 Z2
           GO TO ATRIBUTOS-1.
        IF PARMS(Z1:2) = "DE"
           MOVE SPACE TO T-BRT
           ADD 4 TO Z1 Z2
           GO TO ATRIBUTOS-1.
        IF PARMS(Z1:2) = "IC"
           MOVE 1 TO T-IC
           ADD 3 TO Z1 Z2
           GO TO ATRIBUTOS-1.
        GO TO ANALISE.
    COLOUR.
        ADD 6 TO Z1 Z2
        IF PAR(Z1) = "D"
           MOVE SPACE TO T-COR
           ADD 8 TO Z1 Z2
        ELSE
           MOVE PAR(Z1) TO T-COR
           IF T-COR = "N"
              MOVE "W" TO T-COR.
        IF PAR(Z1) = "B"
           ADD  5 TO Z1 Z2
        ELSE IF PAR(Z1) = "R"
           ADD  4 TO Z1 Z2
        ELSE IF PAR(Z1) = "P"
           ADD  5 TO Z1 Z2
        ELSE IF PAR(Z1) = "G"
           ADD  6 TO Z1 Z2
        ELSE IF PAR(Z1) = "T"
           ADD 1Ø TO Z1 Z2
        ELSE IF PAR(Z1) = "Y"
           ADD  7 TO Z1 Z2
        ELSE IF PAR(Z1) = "N"
           ADD  8 TO Z1 Z2.
        GO TO ANALISE.
    HILIGHT.
        ADD 8 TO Z1 Z2
        IF PAR(Z1) = "O"
           MOVE SPACE   TO T-EXT
           ADD 4 TO Z1 Z2
        ELSE
           MOVE PAR(Z1) TO T-EXT.
        IF PAR(Z1) = "B"
           ADD  6 TO Z1 Z2
        ELSE IF PAR(Z1) = "R"
           ADD  8 TO Z1 Z2
        ELSE IF PAR(Z1) = "U"
           ADD 1Ø TO Z1 Z2.
        GO TO ANALISE.
    VALOR-INICIAL.
        MOVE SPACES TO T-TEXT-R
        MOVE Ø TO Y
```

```
           ADD  8 TO Z1.
       VALOR-INICIAL-1.
           ADD 1 TO Z1 Y
           IF Z1 = 57 AND PAR(57) = "*"
              PERFORM LEITURA
              MOVE Ø TO Z1
              SUBTRACT 1 FROM Y
              GO TO VALOR-INICIAL-1.
           IF PAR(Z1) = "'"
              GO TO ANALISE.
           MOVE PAR(Z1) TO T-TEXT(Y)
           GO TO VALOR-INICIAL-1.
       LEITURA.
           READ FICIN1 INTO LINHA
                AT END GO TO RETORNAR.
           IF FS1 NOT = Ø
              DISPLAY "Error reading Ficin1. File Status " FS1
              STOP RUN.
       RETORNAR.
           CLOSE FICIN1.
           GOBACK.
```

PANELZ0 COPYBOOK SOURCE

```
       Ø1  PANELZØ.
      *  Panelbms: initial and final screen.
      *  header
           Ø2 TOTLENG PIC S9(8) COMP VALUE +129Ø.
           Ø2 CURRET  PIC S9(4) COMP VALUE +Ø.
           Ø2 CUROUT  PIC S9(4) COMP VALUE +1548.
           Ø2 AIDKEY  PIC X          VALUE SPACE.
           Ø2 UPPER   PIC X          VALUE "*".
           Ø2 FILLER  PIC X(1Ø)      VALUE SPACE.
      *  data: total length 129Ø bytes.
           Ø2 ALARME  PIC X     VALUE SPACE.
           Ø2 FILLER  PIC X(Ø9) VALUE X"114Ø4Ø131140401DFØ".
           Ø2 FILLER  PIC X(Ø9) VALUE X"1140CA290242F2C0FØ".
           Ø2 FILLER  PIC X(29) VALUE "PPPPPPPP    AAAAAAAA    NNN  ".
           Ø2 FILLER  PIC X(22) VALUE "NNN    EEEEEEEE    LLL".
           Ø2 FILLER  PIC X(Ø3) VALUE X"1141DB".
           Ø2 FILLER  PIC X(29) VALUE "PPPPPPPP    AAAAAAAA    NNN  ".
           Ø2 FILLER  PIC X(22) VALUE "NNN    EEEEEEEE    LLL".
           Ø2 FILLER  PIC X(Ø3) VALUE X"1142EB".
           Ø2 FILLER  PIC X(29) VALUE "PPP  PPP    AAA  AAA    NNN  ".
           Ø2 FILLER  PIC X(22) VALUE "NNN    EEE         LLL".
           Ø2 FILLER  PIC X(Ø3) VALUE X"1143FB".
           Ø2 FILLER  PIC X(29) VALUE "PPP  PPP    AAA  AAA    NNNN ".
           Ø2 FILLER  PIC X(22) VALUE "NNN    EEE         LLL".
           Ø2 FILLER  PIC X(Ø3) VALUE X"1145CB".
           Ø2 FILLER  PIC X(29) VALUE "PPP  PPP    AAA  AAA    NNNNN".
           Ø2 FILLER  PIC X(22) VALUE "NNN    EEEEEEEE    LLL".
```

```
Ø2 FILLER  PIC X(Ø3) VALUE X"1146DB".
Ø2 FILLER  PIC X(29) VALUE "PPP  PPP    AAA  AAA    NNNNN".
Ø2 FILLER  PIC X(22) VALUE "NNN    EEEEEEEE    LLL".
Ø2 FILLER  PIC X(Ø3) VALUE X"1147EB".
Ø2 FILLER  PIC X(29) VALUE "PPPPPPPP    AAAAAAAA    NNNNN".
Ø2 FILLER  PIC X(22) VALUE "NNN    EEEEEEEE    LLL".
Ø2 FILLER  PIC X(Ø3) VALUE X"1148FB".
Ø2 FILLER  PIC X(29) VALUE "PPP         AAA  AAA    NNN N".
Ø2 FILLER  PIC X(22) VALUE "NNN    EEE         LLL".
Ø2 FILLER  PIC X(Ø3) VALUE X"114ACB".
Ø2 FILLER  PIC X(29) VALUE "PPP         AAA  AAA    NNN N".
Ø2 FILLER  PIC X(22) VALUE "NNN    EEE         LLL".
Ø2 FILLER  PIC X(Ø3) VALUE X"114BDB".
Ø2 FILLER  PIC X(29) VALUE "PPP         AAA  AAA    NNN N".
Ø2 FILLER  PIC X(27) VALUE "NNN    EEEEEEEE    LLLLLLLL".
Ø2 FILLER  PIC X(Ø3) VALUE X"114CEB".
Ø2 FILLER  PIC X(29) VALUE "PPP         AAA  AAA    NNN N".
Ø2 FILLER  PIC X(27) VALUE "NNN    EEEEEEEE    LLLLLLLL".
Ø2 FILLER  PIC X(Ø9) VALUE X"114FCF29Ø242F5CØFØ".
Ø2 FILLER  PIC X(28) VALUE "BBBBBBBBBB       MM        M".
Ø2 FILLER  PIC X(17) VALUE "M      SSSSSSSSSS".
Ø2 FILLER  PIC X(Ø3) VALUE X"115ØEØ".
Ø2 FILLER  PIC X(28) VALUE "BB       BB      MMM      MM".
Ø2 FILLER  PIC X(Ø8) VALUE "M     SS".
Ø2 FILLER  PIC X(Ø3) VALUE X"1151FØ".
Ø2 FILLER  PIC X(28) VALUE "BBBBBBBBBB      MM M    M M".
Ø2 FILLER  PIC X(17) VALUE "M      SSSSSSSSSS".
Ø2 FILLER  PIC X(Ø3) VALUE X"1153CØ".
Ø2 FILLER  PIC X(28) VALUE "BB       BB      MM  M  M  M".
Ø2 FILLER  PIC X(18) VALUE "M               SS".
Ø2 FILLER  PIC X(Ø3) VALUE X"1154DØ".
Ø2 FILLER  PIC X(28) VALUE "BBBBBBBBBB       MM   MM   M".
Ø2 FILLER  PIC X(17) VALUE "M      SSSSSSSSSS".
Ø2 FILLER  PIC X(5)  VALUE X"1156E329Ø3".
Ø2 F-OUTC  PIC X(2)  VALUE X"42F7".
Ø2 F-OUTE  PIC X(2)  VALUE X"41F2".
Ø2 F-OUTA  PIC X(2)  VALUE X"CØFØ".
Ø2 F-OUTI  PIC X(13) VALUE "OUTPUT  FILES".
Ø2 FILLER  PIC X(2)  VALUE X"1DFØ".
Ø2 FILLER  PIC X(5)  VALUE X"1157FC29Ø3".
Ø2 F-INPC  PIC X(2)  VALUE X"42F7".
Ø2 F-INPE  PIC X(2)  VALUE X"41F2".
Ø2 F-INPA  PIC X(2)  VALUE X"CØFØ".
Ø2 F-INPI  PIC X(18) VALUE " INPUT FILE..:".
Ø2 FILLER  PIC X(5)  VALUE X"1158CB29Ø3".
Ø2 FINPC   PIC X(2)  VALUE X"42F7".
Ø2 FINPE   PIC X(2)  VALUE X"41ØØ".
Ø2 FINPA   PIC X(2)  VALUE X"CØCØ".
Ø2 FINPI   PIC X(44) VALUE SPACES.
Ø2 FILLER  PIC X(5)  VALUE X"1DFØ".
Ø2 FILLER  PIC X(5)  VALUE X"1159C729Ø3".
Ø2 F-BMSC  PIC X(2)  VALUE X"42F6".
```

```
    Ø2 F-BMSE  PIC X(2)  VALUE X"41ØØ".
    Ø2 F-BMSA  PIC X(2)  VALUE X"CØFØ".
    Ø2 F-BMSI  PIC X(11) VALUE "BMS SOURCE:".
    Ø2 FILLER  PIC X(5)  VALUE X"1159D329Ø3".
    Ø2 FBMSC   PIC X(2)  VALUE X"42F4".
    Ø2 FBMSE   PIC X(2)  VALUE X"41ØØ".
    Ø2 FBMSA   PIC X(2)  VALUE X"CØCØ".
    Ø2 FBMSI   PIC X(44) VALUE SPACES.
    Ø2 FILLER  PIC X(5)  VALUE X"1DFØ".
    Ø2 FILLER  PIC X(5)  VALUE X"115AD729Ø3".
    Ø2 F-COPYC PIC X(2)  VALUE X"42F6".
    Ø2 F-COPYE PIC X(2)  VALUE X"41ØØ".
    Ø2 F-COPYA PIC X(2)  VALUE X"CØFØ".
    Ø2 F-COPYI PIC X(11) VALUE "COBOL COPY:".
    Ø2 FILLER  PIC X(5)  VALUE X"115AE329Ø3".
    Ø2 FCOPYC  PIC X(2)  VALUE X"42F4".
    Ø2 FCOPYE  PIC X(2)  VALUE X"41ØØ".
    Ø2 FCOPYA  PIC X(2)  VALUE X"CØCØ".
    Ø2 FCOPYI  PIC X(44) VALUE SPACES.
    Ø2 FILLER  PIC X(5)  VALUE X"1DFØ".
    Ø2 FILLER  PIC X(5)  VALUE X"115BE729Ø3".
    Ø2 MSGAC   PIC X(2)  VALUE X"42F2".
    Ø2 MSGAE   PIC X(2)  VALUE X"41ØØ".
    Ø2 MSGAA   PIC X(2)  VALUE X"CØFØ".
    Ø2 MSGAI   PIC X(62) VALUE SPACES.
    Ø2 FILLER  PIC X(5)  VALUE X"115CE61DFØ".
    Ø2 FILLER  PIC X(5)  VALUE X"115CF729Ø3".
    Ø2 MSGBC   PIC X(2)  VALUE X"42F7".
    Ø2 MSGBE   PIC X(2)  VALUE X"41ØØ".
    Ø2 MSGBA   PIC X(2)  VALUE X"CØFØ".
    Ø2 MSGBI   PIC X(62) VALUE SPACES.
    Ø2 FILLER  PIC X(2)  VALUE X"1DFØ".
```

PANELZ1 COPYBOOK SOURCE

```
Ø1  PRIM-LINHA.
    Ø2  PRIMEIRA-LINHA.
      Ø3 FILLER   PIC X(Ø2) VALUE X"42F6".
      Ø3 FILLER   PIC X(Ø2) VALUE X"41F2".
      Ø3 FILLER   PIC X(Ø2) VALUE X"CØE8".
      Ø3 FILLER   PIC X(25) VALUE "1...+....1....+....2....+".
      Ø3 FILLER   PIC X(25) VALUE "....3....+....4....+....5".
      Ø3 FILLER   PIC X(25) VALUE "....+....6....+....7....+".
      Ø3 FILLER   PIC X(24) VALUE "....".
Ø1  PANELZ.
    Ø2 L2479      PIC X(1896) VALUE SPACES.
    Ø2 FILLER REDEFINES L2479.
      Ø3 LT OCCURS 24.
        Ø4 LTC    PIC X  OCCURS 79.
    Ø2 L1896 REDEFINES L2479 PIC X OCCURS 1896.
Ø1  PANELZ1.
```

```
     Ø2 INI1.
* Panelbms drawing screen.
* header
       Ø3 TOTLENG  PIC S9(8) COMP  VALUE +2272.
       Ø3 CURRET   PIC S9(4) COMP  VALUE +Ø.
       Ø3 CUROUT   PIC S9(4) COMP  VALUE +81.
       Ø3 AIDKEY   PIC X     VALUE SPACES.
       Ø3 UPPER    PIC X     VALUE SPACES.
       Ø3 CHARFILL PIC X     VALUE SPACES.
       Ø3 NUMFILL  PIC X     VALUE ZERO.
       Ø3 FILLER   PIC X(8)  VALUE SPACES.
* data: total length 2272
       Ø3 ALARME   PIC X     VALUE SPACE.
       Ø3 FILLER   PIC X(Ø4) VALUE X"11404Ø13".
       Ø3 FILLER   PIC X(Ø5) VALUE X"1140401DFØ".
       Ø3 FILLER   PIC X(Ø5) VALUE X"1140C02Ø3".
       Ø3 PRIMEIRA.
        Ø5 CP1C    PIC X(Ø2) VALUE X"42F6".
        Ø5 CP1E    PIC X(Ø2) VALUE X"41F2".
        Ø5 CP1A    PIC X(Ø2) VALUE X"CØE8".
        Ø5 CP1I.
         Ø7 FILLER PIC X(25) VALUE "1...+....1....+....2....+".
         Ø7 FILLER PIC X(25) VALUE "....3....+....4....+....5".
         Ø7 FILLER PIC X(25) VALUE "....+....6....+....7....+".
         Ø7 FILLER PIC X(24) VALUE "....".
     Ø2 LTAB.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1141DØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1142EØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1143FØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1145CØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1146DØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1147EØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"1148FØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"114ACØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
       Ø3 FILLER   PIC X(79) VALUE SPACES.
       Ø3 FILLER   PIC X(Ø3) VALUE X"114BDØ".
       Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100CØCØ".
```

```
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"114CEØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"114DFØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"114FCØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"1150DØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"1151EØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø3) VALUE X"1152FØ".
  Ø3 FILLER   PIC X(Ø8) VALUE X"29Ø342F44100C0C0".
  Ø3 FILLER   PIC X(79) VALUE SPACES.
Ø2 LTABR REDEFINES LTAB OCCURS 15.
  Ø3 FILLER   PIC X(5).
  Ø3 LTABC    PIC XX.
  Ø3 LTABE    PIC XX.
  Ø3 LTABA    PIC XX.
  Ø3 LTABI    PIC X(79).
Ø2 LEGENDA.
  Ø3 FILLER   PIC X(Ø5) VALUE X"1154CØ29Ø2".
  Ø3 CP17C    PIC X(Ø2) VALUE X"42F6".
  Ø3 CP17A    PIC X(Ø2) VALUE X"C0F0".
  Ø3 CP17I    PIC X(22) VALUE "===== Case (Mixed,Up):".
  Ø3 FILLER   PIC X(Ø5) VALUE X"1154D729Ø2".
  Ø3 CASEC    PIC X(Ø2) VALUE X"42F7".
  Ø3 CASEA    PIC X(Ø2) VALUE X"C0C0".
  Ø3 CASEI    PIC X(Ø1) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø5) VALUE X"1154D929Ø2".
  Ø3 CP19C    PIC X(Ø2) VALUE X"42F6".
  Ø3 CP19A    PIC X(Ø2) VALUE X"C0F0".
  Ø3 CP19I.
   Ø5 FILLER  PIC X(25) VALUE "=========================".
   Ø5 FILLER  PIC X(14) VALUE " Display lines".
  Ø3 FILLER   PIC X(Ø5) VALUE X"1155C129Ø2".
  Ø3 DINIC    PIC X(Ø2) VALUE X"42F7".
  Ø3 DINIA    PIC X(Ø2) VALUE X"C0C0".
  Ø3 DINII    PIC X(Ø2) VALUE SPACES.
  Ø3 FILLER   PIC X(Ø5) VALUE X"1155C429Ø2".
  Ø3 CP21C    PIC X(Ø2) VALUE X"42F6".
  Ø3 CP21A    PIC X(Ø2) VALUE X"C0F0".
  Ø3 CP21I    PIC X(Ø2) VALUE "to".
  Ø3 FILLER   PIC X(Ø5) VALUE X"1155C829Ø2".
  Ø3 DFIMC    PIC X(Ø2) VALUE X"42F7".
  Ø3 DFIMA    PIC X(Ø2) VALUE X"C0C0".
  Ø3 DFIMI    PIC X(Ø2) VALUE SPACES.
```

```
Ø3 FILLER   PIC X(Ø5) VALUE X"1155CB2ØØ2".
Ø3 CP23C    PIC X(Ø2) VALUE X"42F6".
Ø3 CP23A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP23I    PIC X(Ø5) VALUE " ====".
Ø3 FILLER   PIC X(Ø5) VALUE X"1155DØ2ØØ2".
Ø3 CP24C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP24A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP24I.
 Ø5 FILLER  PIC X(17) VALUE "--- Attributes of".
 Ø5 FILLER  PIC X(23) VALUE " the current field ----".
Ø3 FILLER   PIC X(Ø5) VALUE X"1156C32ØØ2".
Ø3 CP25C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP25A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP25I    PIC X(24) VALUE "Data fields Filler.....:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1156DC2ØØ2".
Ø3 FILLC    PIC X(Ø2) VALUE X"42F7".
Ø3 FILLA    PIC X(Ø2) VALUE X"CØCØ".
Ø3 FILLI    PIC X(Ø1) VALUE SPACES.
Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1156EØ2ØØ2".
Ø3 CP27C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP27A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP27I    PIC X(Ø6) VALUE "Name :".
Ø3 FILLER   PIC X(Ø5) VALUE X"1156E72ØØ2".
Ø3 NOMEC    PIC X(Ø2) VALUE X"42F7".
Ø3 NOMEA    PIC X(Ø2) VALUE X"CØCØ".
Ø3 NOMEI    PIC X(Ø7) VALUE SPACES.
Ø3 FILLER   PIC X(Ø5) VALUE X"1156EF1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1156F12ØØ2".
Ø3 CP29C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP29A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP29I    PIC X(15) VALUE "MDT on........:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157C12ØØ2".
Ø3 MDTC     PIC X(Ø2) VALUE X"42F7".
Ø3 MDTA     PIC X(Ø2) VALUE X"CØCØ".
Ø3 MDTI     PIC X(Ø1) VALUE SPACES.
Ø3 FILLER   PIC X(Ø5) VALUE X"1157C32ØØ2".
Ø3 CP62C    PIC X(Ø2) VALUE X"42F4".
Ø3 CP62A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP62I    PIC X(Ø6) VALUE "(Y, N)".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157D32ØØ2".
Ø3 CP33C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP33A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP33I    PIC X(12) VALUE "Field Start:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157EØ2ØØ2".
Ø3 CP34C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP34A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP34I    PIC X(11) VALUE "Numeric   :".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157EC2ØØ2".
Ø3 CP35C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP35A    PIC X(Ø2) VALUE X"CØCØ".
Ø3 CP35I    PIC X(Ø1) VALUE SPACES.
```

```
Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157FØ29Ø2".
Ø3 CP36C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP36A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP36I    PIC X(Ø6) VALUE "Line.:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1157F729Ø2".
Ø3 LINXC    PIC X(Ø2) VALUE X"42F7".
Ø3 LINXA    PIC X(Ø2) VALUE X"CØCØ".
Ø3 LINXI    PIC X(Ø2) VALUE SPACES.
Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1158C129Ø2".
Ø3 CP38C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP38A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP38I    PIC X(15) VALUE "Bright, Dark..:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1158D129Ø2".
Ø3 BRTC     PIC X(Ø2) VALUE X"42F7".
Ø3 BRTA     PIC X(Ø2) VALUE X"CØCØ".
Ø3 BRTI     PIC X(Ø1) VALUE SPACES.
Ø3 FILLER   PIC X(Ø5) VALUE X"1158D329Ø2".
Ø3 CP63C    PIC X(Ø2) VALUE X"42F4".
Ø3 CP63A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP63I    PIC X(Ø6) VALUE "(B, D)".
Ø3 FILLER   PIC X(Ø5) VALUE X"1158FØ29Ø2".
Ø3 CP42C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP42A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP42I    PIC X(11) VALUE "Unprotect :".
Ø3 FILLER   PIC X(Ø5) VALUE X"1158FC29Ø2".
Ø3 CP43C    PIC X(Ø2) VALUE X"42F7".
Ø3 CP43A    PIC X(Ø2) VALUE X"CØCØ".
Ø3 CP43I    PIC X(Ø1) VALUE SPACES.
Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1159CØ29Ø2".
Ø3 CP44C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP44A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP44I    PIC X(Ø6) VALUE "Col..:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1159C729Ø2".
Ø3 COLXC    PIC X(Ø2) VALUE X"42F7".
Ø3 COLXA    PIC X(Ø2) VALUE X"CØCØ".
Ø3 COLXI    PIC X(Ø2) VALUE SPACES.
Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
Ø3 FILLER   PIC X(Ø5) VALUE X"1159D129Ø2".
Ø3 CP31C    PIC X(Ø2) VALUE X"42F5".
Ø3 CP31A    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP31I    PIC X(15) VALUE "Color.........:".
Ø3 FILLER   PIC X(Ø5) VALUE X"1159E129Ø2".
Ø3 CORC     PIC X(Ø2) VALUE X"42F7".
Ø3 CORA     PIC X(Ø2) VALUE X"CØCØ".
Ø3 CORI     PIC X(Ø1) VALUE SPACES.
Ø3 FILLER   PIC X(Ø5) VALUE X"1159E329Ø2".
Ø3 CP6ØC    PIC X(Ø2) VALUE X"42F4".
Ø3 CP6ØA    PIC X(Ø2) VALUE X"CØFØ".
Ø3 CP6ØI    PIC X(15) VALUE "(G,Y,R,T,B,P,W)".
```

```
           Ø3 FILLER   PIC X(Ø5) VALUE X"115ACØ2ØØ2".
           Ø3 CP48C    PIC X(Ø2) VALUE X"42F7".
           Ø3 CP48A    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 CP48I    PIC X(11) VALUE "Protect   :".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115ACC2ØØ2".
           Ø3 CP49C    PIC X(Ø2) VALUE X"42F7".
           Ø3 CP49A    PIC X(Ø2) VALUE X"CØCØ".
           Ø3 CP49I    PIC X(Ø1) VALUE SPACES.
           Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115ADØ2ØØ2".
           Ø3 CP5ØC    PIC X(Ø2) VALUE X"42F5".
           Ø3 CP5ØA    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 CP5ØI    PIC X(Ø6) VALUE "Leng :".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115AD72ØØ2".
           Ø3 LENXC    PIC X(Ø2) VALUE X"42F7".
           Ø3 LENXA    PIC X(Ø2) VALUE X"CØCØ".
           Ø3 LENXI    PIC X(Ø2) VALUE SPACES.
           Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115AE12ØØ2".
           Ø3 CP4ØC    PIC X(Ø2) VALUE X"42F5".
           Ø3 CP4ØA    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 CP4ØI    PIC X(15) VALUE "Extend Hilite.:".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115AF12ØØ2".
           Ø3 EXTC     PIC X(Ø2) VALUE X"42F7".
           Ø3 EXTA     PIC X(Ø2) VALUE X"CØCØ".
           Ø3 EXTI     PIC X(Ø1) VALUE SPACES.
           Ø3 FILLER   PIC X(Ø5) VALUE X"115AF32ØØ2".
           Ø3 CP61C    PIC X(Ø2) VALUE X"42F4".
           Ø3 CP61A    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 cp61I    PIC X(18) VALUE "(Blink Rev Uline) ".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115BDØ2ØØ2".
           Ø3 CP54C    PIC X(Ø2) VALUE X"42F7".
           Ø3 CP54A    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 CP54I    PIC X(11) VALUE "Text      :".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115BDC2ØØ2".
           Ø3 CP55C    PIC X(Ø2) VALUE X"42F7".
           Ø3 CP55A    PIC X(Ø2) VALUE X"CØCØ".
           Ø3 CP55I    PIC X(Ø1) VALUE SPACES.
           Ø3 FILLER   PIC X(Ø2) VALUE X"1DFØ".
           Ø3 FILLER   PIC X(Ø5) VALUE X"115CFØ2ØØ2".
           Ø3 PF12C    PIC X(Ø2) VALUE X"42F2".
           Ø3 PF12A    PIC X(Ø2) VALUE X"CØFØ".
           Ø3 PF12I.
            Ø5 FILLER  PIC X(25) VALUE "F4:Draw area F5:Attrib pr".
            Ø5 FILLER  PIC X(25) VALUE "ev field F6:Attrib next f".
            Ø5 FILLER  PIC X(25) VALUE "ield F7:Top F8:Bot  F3:Ex".
            Ø5 FILLER  PIC X(Ø2) VALUE "it".
```

PANATRIB COPYBOOK SOURCE

```
       Ø1   ATRIBUTOS.
```

```
*          Colors: Fields with suffix C.
     Ø2    CORES.
        Ø4   BLUE          PIC XX     VALUE X"42F1".
        Ø4   RED           PIC XX     VALUE X"42F2".
        Ø4   PINK          PIC XX     VALUE X"42F3".
        Ø4   GREEN         PIC XX     VALUE X"42F4".
        Ø4   TURQ          PIC XX     VALUE X"42F5".
        Ø4   YELLOW        PIC XX     VALUE X"42F6".
        Ø4   WHITE         PIC XX     VALUE X"42F7".
*          Extend hilight: Fields with suffix E.
     Ø2    EXTEND-HILIGHT.
        Ø4   NORMALH       PIC XX     VALUE X"41ØØ".
        Ø4   BLINK         PIC XX     VALUE X"41F1".
        Ø4   REVERSE       PIC XX     VALUE X"41F2".
        Ø4   UNDERLINE     PIC XX     VALUE X"41F4".
*          Attribute: Fields with suffix A.
     Ø2    ATRIBUTO.
        Ø4   UNPROT        PIC XX     VALUE X"CØC1".
        Ø4   UNPROT-FRSET  PIC XX     VALUE X"CØCØ".
        Ø4   UNPROT-BRT    PIC XX     VALUE X"CØC9".
        Ø4   UNPROT-DARK   PIC XX     VALUE X"CØ4D".
        Ø4   PROT          PIC XX     VALUE X"CØF1".
        Ø4   PROT-FRSET    PIC XX     VALUE X"CØFØ".
        Ø4   PROT-BRT      PIC XX     VALUE X"CØF9".
        Ø4   PROT-DARK     PIC XX     VALUE X"CØ7C".
        Ø4   NUM           PIC XX     VALUE X"CØD1".
        Ø4   NUM-BRT       PIC XX     VALUE X"CØD9".
        Ø4   NUM-DARK      PIC XX     VALUE X"CØ5D".
*          Key pressed returned in the KEY field.
     Ø2    KEYLIST.
        Ø4   PF1           PIC  X     VALUE "1".
        Ø4   PF2           PIC  X     VALUE "2".
        Ø4   PF3           PIC  X     VALUE "3".
        Ø4   PF4           PIC  X     VALUE "4".
        Ø4   PF5           PIC  X     VALUE "5".
        Ø4   PF6           PIC  X     VALUE "6".
        Ø4   PF7           PIC  X     VALUE "7".
        Ø4   PF8           PIC  X     VALUE "8".
        Ø4   PF9           PIC  X     VALUE "9".
        Ø4   PF1Ø          PIC  X     VALUE X"7A".
        Ø4   PF11          PIC  X     VALUE X"7B".
        Ø4   PF12          PIC  X     VALUE X"7C".
        Ø4   PF13          PIC  X     VALUE "A".
        Ø4   PF14          PIC  X     VALUE "B".
        Ø4   PF15          PIC  X     VALUE "C".
        Ø4   PF16          PIC  X     VALUE "D".
        Ø4   PF17          PIC  X     VALUE "E".
        Ø4   PF18          PIC  X     VALUE "F".
        Ø4   PF19          PIC  X     VALUE "G".
        Ø4   PF2Ø          PIC  X     VALUE "H".
        Ø4   PF21          PIC  X     VALUE "I".
```

```
         Ø4   PF22          PIC  X     VALUE X"4A".
         Ø4   PF23          PIC  X     VALUE X"4B".
         Ø4   PF24          PIC  X     VALUE X"4C".
         Ø4   CLEAR         PIC  X     VALUE X"6A".
*           Sound alarm.
      Ø2    ALARME-ON       PIC  X     VALUE X"F5".
      Ø2    ALARME-OFF      PIC  X     VALUE SPACE.
```

PANELTAB COPYBOOK SOURCE

```
* PANELTAB - This copybook holds a table with BMS fields.
 Ø1  TAB-CAMPOS.
* MXF - Maximun number of fields supported.
     Ø2   MXF           PIC 999   VALUE 3ØØ.
     Ø2   TITMAP        PIC X(8)  VALUE SPACES.
     Ø2   TITMAP-R REDEFINES TITMAP PIC X OCCURS 8.
* Default symbols for Text, Numeric, Unprot, Prot and Filler
     Ø2   TXTØ          PIC X   VALUE "#".
     Ø2   NUMØ          PIC X   VALUE "?".
     Ø2   UNPØ          PIC X   VALUE "&".
     Ø2   PROØ          PIC X   VALUE "%".
     Ø2   FILØ          PIC X   VALUE "+".
     Ø2   MDTØ          PIC X   VALUE "Y".
     Ø2   BRTØ          PIC X   VALUE " ".
     Ø2   CORØ          PIC X   VALUE " ".
     Ø2   EXTØ          PIC X   VALUE " ".
     Ø2   MDTØ-TXT      PIC X   VALUE " ".
     Ø2   BRTØ-TXT      PIC X   VALUE " ".
     Ø2   CORØ-TXT      PIC X   VALUE " ".
     Ø2   EXTØ-TXT      PIC X   VALUE " ".
*  BMS fields table.
     Ø2  CAMPOS OCCURS  3ØØ.
      Ø4  TAB-POS       PIC 9999.
      Ø4  TAB-LENG      PIC 99.
      Ø4  TAB-NOME      PIC X(7).
      Ø4  TAB-TIPO      PIC X.
      Ø4  TAB-MDT       PIC X.
      Ø4  TAB-BRT       PIC X.
      Ø4  TAB-COR       PIC X.
      Ø4  TAB-EXT       PIC X.
      Ø4  TAB-STOP      PIC 9999.
      Ø4  TAB-TEXT.
       Ø6 TAB-TEXT-1    PIC X(47).
       Ø6 TAB-TEXT-2    PIC X(31).
      Ø4  TAB-TEXT-R REDEFINES TAB-TEXT OCCURS 78 PIC X.
      Ø4  TAB-IC        PIC X.
```

Luis Paulo Figueiredo Sousa Ribeiro
Edinfor (Portugal)

IBM has announced Version 1.1 of its CICS Performance Analyzer for OS/390 for optimizing system and application performance.

Benefits are said to be better transaction response times, system resource usage, and application performance analysis, greater availability of resources, more productivity of system and application programmers, and ongoing system management and measurement reports.

CICS PA reports analyse transaction response time, CICS system resource usage, transaction groups, cross-system performance (including MRO and APPC) Business Transaction Services, CICS Web support, external subsystems including DB2 and IMS, and exception events that cause performance degradation.

Other elements include an ISPF dialogue to build, maintain, and submit report requests and an extract capability to help analyse CICS system performance from PC spreadsheets such as Lotus and Excel.

There's also a report formatting capability to help tailor the content of reports, and a record selection and sorting capability for requesting information required.

Finally, there's support for CICS/ESA Version 4.1 and CICS TS for OS/390 Version 1.

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

* * *

Level 8 Systems has announced Version 2.0 of its Geneva AppBuilder, formerly known as Seer*HPS, enabling developers to build and deploy applications across platforms including AIX, HP-UX, OS/390, OS/400, Solaris, and Windows NT/2000.

Version 2.0 adds support for creating J2EE applications from specifications stored in the product's repository by generating Java, HTML Java Servlets, and Enterprise Java Beans, which enable both thin HTML and downloadable Java clients to be created from the same specification.

The software helps Java sites exploit OS/390 mainframes by connecting the EJB and HTML Servlet applications to AppBuilder-created applications on an OS/390 system running CICS or IMS COBOL.

An enhanced construction workbench is available on Windows NT/2000 with development repositories available on AIX, NT, and OS/390. Application generation supports IBM Websphere 3.5, IBM Apache, and BEA Weblogic 5.1 and above.

For further information contact:
Level 8, 8000 Regency Parkway , Cary, NC 27511, USA.
Tel: (919) 380 5000.
Level 8, Harman House, Ground Floor, George Street, Uxbridge, Middlesex, UB8 1QQ, UK.
Tel: (01895) 206 700.
URL: http://www.level8.com and http://www.AppBuilder.com.

* * *