December 1999

In this issue

3  RDO displays sorted by date
16  Cold start next time – revisited
21  NEWCOPY of programs in an MRO environment
37  National language sort in CICS
48  CICS news

© Xephon plc 1999
Contributions

Articles published in *CICS Update* are paid for at the rate of £170 ($250) per 1000 words and £90 ($140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

*CICS Update* on-line

Code from *CICS Update* can be downloaded from our Web site at http://www.xephon.com/cicsupdate.html; you will need the user-id shown on your address label.

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.

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.

© Xephon plc 1999. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs $36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*
RDO displays sorted by date

‘Sort changed’ is one of the most commonly used commands in TSO. When using RDO, the ability to list RDO entities by descending date and timestamp can be useful. Support staff often need to check whether a known change has been made and, during problem determination, whether a change has been made in error. This is particularly useful at sites where access to CEDA is widely available (which is often the case in development systems).

The utility presented here provides this function for entries within RDO Groups and Lists. It was developed under CICS Version 4 in an OS/390 environment.

The first map allows selection of the CSD (see Figure 1). This particular implementation assumes that a single CSD is shared across an environment. The three allowable environments are DEV, ACC, and PRD.

The RDO entry is assumed to be a Group Name. Lists are recognized by using the convention that list names are of the form xxxCICS. The logic in program SYDRDQ close to the label GDISP will need to be changed to suit your own List naming convention. The RDO Group Name entered can be generic – for example ‘PDLA*’ would expand all groups beginning with PDLA within the sorted display.

Figure 1: Selection screen
The ‘From Date’ is provided to restrict the displayed entries to those entries changed or added after the specified date. There is an upper limit on the number of RDO entries that can be selected because the response area from DFHEDAP is restricted to 30,000 bytes.

A detail screen is given in Figure 2, showing the RDO entries in descending date and timestamp order.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>GROUP</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYLONG</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.24</td>
</tr>
<tr>
<td>SYSHUT</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.24</td>
</tr>
<tr>
<td>SYG8</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.23</td>
</tr>
<tr>
<td>SYG9</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.23</td>
</tr>
<tr>
<td>SYMSYG8</td>
<td>MAPSET</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.23</td>
</tr>
<tr>
<td>SYMSYG9</td>
<td>MAPSET</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.23</td>
</tr>
<tr>
<td>SYMLONG</td>
<td>MAPSET</td>
<td>P###SYS</td>
<td>1999.179</td>
<td>09.27.03</td>
</tr>
<tr>
<td>DFHXTENF</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.175</td>
<td>16.06.54</td>
</tr>
<tr>
<td>SYSGMT</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1999.154</td>
<td>15.31.16</td>
</tr>
<tr>
<td>SYCPLTCO</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.335</td>
<td>16.54.01</td>
</tr>
<tr>
<td>SYSDMPCO</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.247</td>
<td>13.06.44</td>
</tr>
<tr>
<td>SYC24SL</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.216</td>
<td>15.16.37</td>
</tr>
<tr>
<td>SYC24EP</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.141</td>
<td>17.08.04</td>
</tr>
<tr>
<td>SYC24EU</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.141</td>
<td>17.08.04</td>
</tr>
<tr>
<td>SYC24EZ</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.141</td>
<td>17.08.04</td>
</tr>
<tr>
<td>+ SYC24EB</td>
<td>PROGRAM</td>
<td>P###SYS</td>
<td>1998.141</td>
<td>17.08.04</td>
</tr>
</tbody>
</table>

Results: 1 To 16 Of 68

To exit press Clear or PF3

PF8(FWD)

**Figure 2: Sorted RDO displays – detail screen**

The current Julian date is shown in the heading for comparison with the group entries. The year is shown in YYYY format, although the data returned from DFHEDAP contains only a two-digit year. This is to enable the sorted displays to continue beyond the millennium (the code assumes that years before 50 are after 2000).

PF7 and PF8 can be used to page backwards and forwards through the
display. PF7 and PF8 only appear as options on the bottom line when they are available.

ERROR MESSAGES
The following error messages may be encountered:

• ‘Group xxxxx not found’ – a group name has been entered that does not exist.
• ‘! Maximum reached. Narrow your search’ – the 30,000 byte limit has been exceeded.
• ‘Someone is using CEDA – try later’ – the requested CSD is already in use and cannot be allocated.

INSTALLATION INSTRUCTIONS
To install this utility:

• Assemble the maps SYMDRDP and SYMDRDQ and programs SYDRDP and SYDRDQ into an appropriate library in your CICS RPL chain.
• Add RDO definitions for the above maps and programs. The programs are DATALLOCATION(ANY).
• Add RDO transaction definitions for SYDP and SYDQ, pointing to programs SYDRDP and SYDRDQ respectively.

PROGRAM SYDRDQ – SORTED RDO DISPLAY TOP LEVEL
Adjust the constants at the bottom of this program to your own CSD dataset names. The logic will need slight adjustment if you use a convention other than a single CSD for DEV, ACC, and PRD.

The LPAR name arrived at in section SNDMAP0 is also site-dependent.

*ASM XOPTS(SP)
SYDRDQ RMODE ANY
        TITLE 'SYDRDQ - RDO FRONT END'
*
* R1  EQU 1
R2  EQU 2
R3  EQU 3
R4  EQU 4 WORK REGISTER
R5  EQU 5
R6  EQU 6
R7  EQU 7
DATAREG  EQU 8 DATA REGISTER
EIBREG  EQU 9 EIB REGISTER
RA  EQU 10 LINK REGISTER
BASE  EQU 11 PROGRAM BASE REGISTER
COPY DFHAID
DFHEISTG DSECT
APPL  DS  CL8
ATIME  DS  PL8
TSTAT  DS  CL1
SUBMENV  DS  CL4
SUBMENT  DS  CL28
CSDDSN  DS  CL44
COPY SYMDRO
SYDRdq  DFHEIENT CODEREG=(BASE),EIBREG=(EIBREG),DATAREG=(DATAREG)
B  BEGIN
DC  CL12'PROGRAM ID: '
DC  CL8'SYDRdq '
DC  CL4': '
DC  CL24'ASSEMBLY TIME AND DATE: '
DC  CL8'&SYSTIME'
DC  CL8'&SYSDATE'
BEGIN  DS  ØH
CLC  EIBCALEN,=H'0' Any COMMAREA?
BNE  BEGINØ Been here before
B  SNDMAPØ Send Map
BEGINØ  DS  ØH
L  R1,DFHEICAP Address COMMAREA
MVC  TSTAT,Ø(R1) Overlay TSTAT
EXEC  CICS HANDLE AID PF3(RETURN1) CLEAR(RETURN1)
BEGIN1  DS  ØH
MVI  TSTAT,X'00'
EXEC  CICS RECEIVE MAP('SYMDRdq')
EXEC  CICS HANDLE CONDITION INVREQ(BUSY)
*
* Display by Date Processing
*
MVC  CSDDSN,DEVCSd
CLC  ENVREI,=C'DEV'
BE  ASSCSd
MVC  CSDDSN,ACCsCSd
CLC  ENVREI,=C'ACC'
* RETURN1 DS OH
* Reset CSD to ALF on Exit
  EXEC CICS HANDLE CONDITION INVREQ(BUSY)
  MVC CSDdsn,ACCCSD
  EXEC CICS SET FILE('DFHCSO') DSNAME(CSDdsn)
  EXEC CICS SEND CONTROL ERASE FREEKB
  EXEC CICS RETURN
*
* Return when CEDA in use
*
BUSY DS OH
  EXEC CICS SEND TEXT
  FROM(QMESS)
  ERASE WAIT
  EXEC CICS RETURN
*
* Constants
*
COMMAL DC H'1'
DRDPCLEN DC H'32'
SUBMENTG DC CL16'GROUP(    ) '
SUBMENTL DC CL16'LIST(    ) '
SPACES DC CL32' '
DEVCSD DC CL44'xxx.xxxxxxxx.DFHCSO' <===
ACCCSD DC CL44'xxx.xxxxxxxx.DFHCSO' <=== Change to appropriate DSNs
PRDCSD DC CL44'xxx.xxxxxxxx.DFHCSO' <===
QMESS DC CL50'Someone is using CEDA - try later    ',

LTORG
END SYDRDQ

MAP SYMDRDQ – USED BY PROGRAM SYDRDQ
You should add your own company name where indicated.

MAPSET3 DFHMSD TYPE=&SYSPARM,MODE=INOUT,CTRL=FREEKB,LANG=ASM, TIOAPFX=YES
SYMDRDQ DFHMDF SIZE=(24,80),LINE=1,COLUMN=1,MAPATTS=(COLOR)
  DFHMDF POS=(01,1),LENGTH=4,COLOR=GREEN,
  INITIAL='SYDQ'
  DFHMDF POS=(01,06),LENGTH=11,COLOR=NEUTRAL,
  INITIAL='Partition :'
PARTI DFHMDF POS=(01,18),LENGTH=4,COLOR=NEUTRAL,
  INITIAL='rrrr'
  DFHMDF POS=(01,23),LENGTH=17,COLOR=NEUTRAL,
  INITIAL='-- CICS Region :'
REGI DFHMDF POS=(01,41),LENGTH=7,COLOR=NEUTRAL,
  INITIAL='rrrrrrrr'
  DFHMDF POS=(01,49),LENGTH=5,COLOR=BLUE,INITIAL='DATE:'
PROGRAM SYDRDP – SORTED RDO DISPLAY DETAIL

The LPAR name arrived at in section SNDMAP2 is site-dependent.

*ASM XOPTS(SP)
SYDRDP   RMODE ANY
      TITLE 'SYDRDP - Sorted RDO Displays'
* *
* This program will display the RDO GROUP or LIST
* PASSED in the COMMAREA sorted into change date/time
* order. The display is scrollable.
* *
R4      EQU  4    WORK REGISTER
R5      EQU  5    WORK REGISTER
R6      EQU  6    WORK REGISTER
R7      EQU  7    WORK REGISTER
DATAREG EQU  8    DATA REGISTER
EIBREG  EQU  9    EIB REGISTER
RA      EQU 10  WORK REGISTER
BASE EQU 11 PROGRAM BASE REGISTER
COPY DFHAID
DFHEISTG DSECT
INLENG DS H
WKDET DS CL57
COMMAS DS ØH COMMAREA start for SYDRDP
CURPOS DS H
MOREBWD DS X
MOREFWD DS X
TOTFROM DS PL3
RDOENV DS CL4
RDCCMD DS CL22
RDOAT DS CL8
* COMMAREA end
TOTTO DS PL3
TOTALL DS PL3
ATIME DS PL8
APPL DS CL8
OUTCTR DS CL6
CEDAPARM DS ØF COMMAREA Start FOR CEDA call
CEDAPRM1 DS F RDO Command Address
CEDAPRM2 DS F RDO Command Length Address
CEDAPRM3 DS F RDO Mode Indicator Address
CEDAPRM4 DS F RDO Response Address
CEDAPRM5 DS F RDO Response Length Adress
COPY SYMDRDP
HERE DS F
COMRES DS CL30000
COMEND DS CL6Ø
SYDRDP DFHEIENT CODEREG=(BASE),EIBREG=(EIBREG),DATAREG=(DATAREG)
B BEGIN
DC CL12'PROGRAM ID: '
DC CL8'SYDRDP '
DC CL4': '
DC CL24'ASSEMBLY TIME AND DATE:'
DC CL8'SYSTIME'
DC CL8'SYSDATE'
BEGIN DS ØH
MVC HERE,=C'HERE' Eyecatcher
CLC EIBCALEN,=H'32' COMMAREA of 32?
BE BEGIN1 First time through
CLC EIBCALEN,=H'41' COMMAREA not 41?
BNE RETURN1 Reject entry
L RA,DFHEICAP Valid entry
MVC COMMAS(41),Ø(RA) Move into dynamic
EXEC CICS HANDLE AID PF3(RETURN1) CLEAR(RETURN1) EXEC CICS RECEIVE LENGTH(INLENG)
B SNDMAPØ
BEGIN1 DS ØH
MVC RDOCMD(2),=C'E '  
L RA,DFHEICAP       Address COMMAREA  
MVC RDOENV,Ø(RA)  Passed Environment  
MVC RDOCMD+2(28),4(RA)  Passed Command  
MVI MOREFWD,X'ØØ'  
MVI MOREBWD,X'ØØ'  
XC CURPOS,CURPOS  Zeroize start position  
ZAP TOTFROM,=P'1'  

SNDMAPØ DS ØH  
ZAP TOTTO,TOTFROM  
SP TOTTO,=P'1'  Believe me!  
MVI T1AO,C' '  BWD not available  
CLI MOREBWD,X'FF'  
BNE SNDMAP2  
MVI T1AO,C'+ '  BWD available  

SNDMAP2 DS ØH  
EXEC CICS ASSIGN APPLID(APPL)  
MVC REGIO,APPL  Move Applid to Map  
MVC PARTIO,=C'DEVL'  
CLC APPL+3(1),=C'C'  PROD is C  
BNE GTIME  
MVC PARTIO,=C'PROD'  

GTIME DS ØH  
MVC ENVO,RDOENV  
EXEC CICS ASKTIME ABSTIME(ATIME)  
EXEC CICS FORTAMTIME ABSTIME(ATIME) DDMMYYYY(DATEO)  
TIME(TIMEO) DATESEP TIMESEP  
EXEC CICS FORTAMTIME ABSTIME(ATIME) YYYYDDD(JDATO) DATESEP('.')  

LA R4,RDOCMD  Set up parameter List with...  
ST R4,CEDAPRM1  (1) Address of command  
LA R4,RDOCMDL  
ST R4,CEDAPRM2  (2) Address of command length  
LA R4,MODEIND  
ST R4,CEDAPRM3  (3) Address of mode indicator  
LA R4,COMRES  
ST R4,CEDAPRM4  (4) Address of response area  
LA R4,COMRESL  
ST R4,CEDAPRM5  (5) Address of response length  

*  
* Call RDO to action the command  
*  
EXEC CICS LINK PROGRAM('DFHEDAP') COMMAREA(CEDAPARM)  
*  
* Check for good response  
*  
LA R4,COMRES+16  Contains NAME if good  
CLC Ø(4,R4),=C'NAME'  
BE GOODRET  
MVC TITLO(18),8(R4)  Move response  
CLC TITLO(4),=C'List'
BE SNDMAPF
MVC TITLO(24),8(R4) Move group response
B SNDMAPF
*
* Sort details into Descending Timestamp Order*
*
GOODRET DS 0H
MVC TITLO,HEDR Move heading
MVC TITLO+24(5),22(R4)
MVC FDATE0,SPACES
CLC RDOOAT,SPACES
BE NODATE
MVC FDATE0(22),='Changes Since '
MVC FDATE0+14(8),RDOOAT
NODATE DS 0H
LA R4,COMRES+72 First entry
*
* This is the long-awaited fix for the so-called millennium bug!
* It will stop working in 2051, you have been warned*
*
DFIX DS 0H
CLI 0(R4),X'00' End?
BE DLOOP0 Go to Sort
MVC 33(2,R4),='19' Add 19nn to year
CLC 35(2,R4),='50'
BH DFIX1
CLI 35(R4),='00' Bad date returned
BE DFIX1
MVC 33(2,R4),='20' Year 2000 and beyond
DFIX1 DS 0H
LA R4,57(R4) Next entry
B DFIX
DLOOP0 DS 0H
LA R4,COMRES+72 First entry
ZAP TOTALL,='0'
DLOOP1 DS 0H
CLI 0(R4),X'00' End of sort?
BE SNDMAP1 Send the Map
LA R5,57(R4) Next entry
AP TOTALL,='1' Count them
DLOOP2 DS 0H
CLI 0(R5),X'00' End of pass?
BNE COMPARE No
LA R4,57(R4) Move on
B DLOOP1 Around again
COMPARE DS 0H
CLC 33(18,R4),33(R5) Compare dates
BL SWAP Found a higher one
DLOOP3 DS 0H
LA R5,57(R5) Move on
B    DLOOP2                  Around again
SWAP  DS ØH
   MVC WKDET,Ø(R4)         Save old highest
   MVC Ø(57,R4),Ø(R5)      Replace with new
   MVC Ø(57,R5),WKDET      And put back in pool
B    DLOOP3
SNDMAP1 DS ØH

* Move response to Map
*

LA   R4,COMRES+72          First entry
AH   R4,CURPOS            Add current offset
LA   R5,T10               First output field
LA   R6,T16A0             Last output field
MVI  Ø(R5),C' '           Clear more FWD
MVI  MOREFWD,X'ØØ'       Clear in COMMAREA

MAPLOOP DS ØH
   CLC  33(8,R4),RDODAT    Compare will passed date
   BL   MAPBLOW             Map
   CLI  Ø(R4),X'ØØ'        End of response?
   BE   MAPBLOW             Send the Map
   MVC  Ø(51,R5),Ø(R4)     Move details
   AP   TOTTO,=P'1'
   CR   R5,R6              Last line?
   BH   MAPBLOW             Map
   LA   R4,57(R4)          Next response field
   LA   R5,54(R5)          Next line
   CR   R5,R6              Last line?
   BNE  MAPLOOP             No
   CLI  57(R4),X'ØØ'      More to come?
   BE   MAPLOO1            No
   MVI  Ø(R5),C'+'        Show more FWD
   MVI  MOREFWD,X'FF'     Indicate in COMMAREA

MAPLOO1 DS ØH
   LA   R5,4(R5)          Offset position
   B    MAPLOOP

MAPBLOW DS ØH
   CLC CURPOS,=H'29000'    Prevent overrun
   BNH SNDMAPE
   MVC  4(36,R5),=C'Maximum reached, narrow your search '
   MVI  Ø(R5),C'!'        Show limit reached

SNDMAPE DS ØH
   MVC OUTCTR,EDPAT        OUTCTR
   ED   OUTCTR,TOTFROM
   MVC TLINO,OUTCTR+1      TLINO
   MVC OUTCTR,EDPAT        OUTCTR
   ED   OUTCTR,TOTTO
   MVC BLINO,OUTCTR+1      BLINO
   MVC OUTCTR,EDPAT        OUTCTR
   ED   OUTCTR,TOTAL

MVC TALLO,OUTCTR+1
MVC PF10,=C'PF7(BWD)'
MVC PF20,=C'PF8(FWD)'
CLI MOREBWD,X'FF'
BNE CHKFWD
MVC PF10,=C'PF7(BWD)'
CHKFWD DS ØH
CLI MOREFWD,X'FF'
BNE SNDMAPF
MVC PF20,=C'PF8(FWD)'
SNDMAPF DS ØH
  EXEC CICS SEND MAP ('SYMDRP') ERASE FREEKB
*
* RETURN BUT COME BACK
*
RETURNØ DS ØH
  EXEC CICS RETURN TRANSID(EIBTRNID)
  COMMAREA(COMMAS) LENGTH(COMMAL)
RETURN1 DS ØH
  EXEC CICS SEND CONTROL ERASE FREEKB
RETURN DS ØH
  EXEC CICS RETURN TRANSID('SYDQ') IMMEDIATE
*
PAGEBWD DS ØH
CLI MOREBWD,X'FF'    Backward allowed?
BNE SNDMAPØ         No, so ignore
SP TOTFROM,=P'16'
LH R4,CURPOS
SH R4,=H'912'     Page backward offset
STH R4,CURPOS
LTR R4,R4            Back to start?
BNZ SNDMAPØ
MVI MOREBWD,X'ØØ'    Indicate in COMMAREA
B SNDMAPØ

PAGEFWD DS ØH
CLI MOREFWD,X'FF'    Forward allowed?
BNE SNDMAPØ       No, so ignore
AP TOTFROM,=P'16'
MVI MOREBWD,X'FF'    Indicate in COMMAREA
LH R4,CURPOS
AH R4,=H'912'     Page forward offset
STH R4,CURPOS
B SNDMAPØ

*    Constants
*
COMMAL DC H'41'    COMMAREA length
RDOCMDL DC H'22'    Length of Command Area
MAP SYMDRDP – USED BY PROGRAM SYDRDP

You should add your own company name where indicated.

MAPSET3  DFHMSD  TYPE=&SYSPARM,MODE=INOUT,CTRL=FREEKB,LANG=ASM, * TIOAPFX=YES
SYMDRDP  DFHMDI  SIZE=(24,80),LINE=1,COLUMN=1,MAPATTS=(COLOR)  *
        DFHMDF  POS=(01,1),LENGTH=4,COLOR=GREEN, * INITIAL='SYDP'
        DFHMDF  POS=(01,06),LENGTH=11,COLOR=NEUTRAL, *
                  INITIAL='Partition :'
PARTI    DFHMDF  POS=(01,18),LENGTH=4,COLOR=NEUTRAL, *
                  INITIAL='rrrr'
        DFHMDF  POS=(01,23),LENGTH=17,COLOR=NEUTRAL, *
                  INITIAL='-- CICS Region :'
REGI     DFHMDF  POS=(01,41),LENGTH=4,COLOR=NEUTRAL, *
                  INITIAL='rrrrrrr'
        DFHMDF  POS=(01,49),LENGTH=5,COLOR=BLUE,INITIAL='DATE:'
DATE     DFHMDF  POS=(01,55),LENGTH=10,COLOR=BLUE,INITIAL='XX.XX.XXXX'
        DFHMDF  POS=(01,66),LENGTH=5,COLOR=BLUE,INITIAL='TIME:'
TIME     DFHMDF  POS=(01,72),LENGTH=8,COLOR=BLUE,INITIAL='XX.XX.XX'
        DFHMDF  POS=(02,23),LENGTH=20,COLOR=NEUTRAL, *
                  INITIAL='-- Company Name Here'
        DFHMDF  POS=(02,49),LENGTH=7,COLOR=BLUE,INITIAL='JULIAN:'
JDAT     DFHMDF  POS=(02,57),LENGTH=8
        DFHMDF  POS=(03,49),LENGTH=4,COLOR=BLUE,INITIAL='CSD:'
ENV      DFHMDF  POS=(03,54),LENGTH=4
        DFHMDF  POS=(04,23),LENGTH=20,COLOR=NEUTRAL, *
                  INITIAL='RDO Display By Date '
FDATE    DFHMDF  POS=(04,44),LENGTH=22,COLOR=NEUTRAL, *
                  INITIAL=''
TITL     DFHMDF  POS=(05,03),LENGTH=49,COLOR=BLUE,INITIAL=''
T1A      DFHMDF  POS=(06,01),LENGTH=1,COLOR=RED,INITIAL=''
T1       DFHMDF  POS=(06,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T2       DFHMDF  POS=(07,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T3       DFHMDF  POS=(08,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T4       DFHMDF  POS=(09,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T5       DFHMDF  POS=(10,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T6       DFHMDF  POS=(11,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T7       DFHMDF  POS=(12,03),LENGTH=51,COLOR=YELLOW,INITIAL=''
T8       DFHMDF  POS=(13,03),LENGTH=51,COLOR=YELLOW,INITIAL='
Cold start next time – revisited

Cold start next time, CICS Update, Issue 167, October 1999, discussed how to ensure that a CICS region cold starts next time and is also under the control of the CICS systems programmer. It closed with the question ‘So what else is required?’

What follows is a possible answer to this question – a REXX that creates the required COLDNEXT dataset to force this cold start.

The REXX can be passed a mask covering all the required regions, instead of a single one, and also gives you the option to edit the list of regions that it has determined need a COLDNEXT dataset to be created, prior to actually creating them.

Although it has to be site-specific to enable it to determine CICS levels (eg test, live, etc) and the format of the DFHGCD dataset, all the code
that should need changing to enable the REXX to work at another site
has been grouped together in one section.

The REXX is currently designed to work on CICS regions with the format:

\[ \text{CIxpp1A} \]

where ‘x’ is the level (eg T for test) and ‘pp’ are the project letters. In
this way COLDNEXT can be run with a mask of ‘CIT’, to create
datasets for all test regions, or ‘CITpp’, to cold start only a particular
applications region.

A search is done for all DFHGCD datasets to determine the regions
that match this mask. Our GCD datasets have the form:

\[ \text{CIC<l>.<region>.CICS41Ø.DFHGCD} \]

So, using the third character of the supplied mask to get the ‘CIC<l>’
prefix, a search of ‘CIC<l>.<mask>.CICS410.DFHGCD’ will produce
a list of regions for which to create COLDNEXT datasets.

If you change this site-specific section to determine the GCD format
for your standards based in the supplied mask, the rest of the REXX
doesn’t need changing.

The COLDNEXT dataset is created only if it doesn’t already exist and
is deleted when the system is started. Therefore, the existence of the
COLDNEXT dataset indicates that the system hasn’t been started
since its creation date. The REXX highlights this by issuing the
message ‘<region> ignored. COLDNEXT already exists’.

COLDNEXT

```rexx
/* REXX ***************************************************************/
/*                                                                    */
/* EXEC           : COLDNEXT                                          */
/* Called by      :                                                   */
/* Purpose        : create COLDNEXT datasets                          */
/* Parameters     : Mask of which regions need cold starting          */
/*                   must be a minimum of 3 characters                 */
/* Return Codes   : Ø - OK                                            */
/*                  8 - severe error (eg file access problems)        */
/*                                                                    */
/* REXX ***************************************************************/
```
start:
PARSE ARG mask
total = Ø
region.Ø = Ø
/**************************************************************************
/* Site-specific details - may need changing depending on naming          */
/* standards etc.                                                        */
/**************************************************************************/ if LENGTH(mask) < 3 then do
    say 'Invalid mask entered - must be minimum of 3 characters'
    exit Ø
end
list_dsn = userid() '.COLDNEXT.LIST'
level = substr(mask,3,1)
gcdmask = 'CIC' level '.' mask '*.CICS*.DFHGCD.DATA'
temp_start = "start = index(gcddsn,'.',1) + 1"
temp_end = "end = index(gcddsn,'.',start)"
temp_level = "level = substr(temp_reg,3,1)"
temp_cold = "cold_dsn='CIC' level '.' temp_reg '.CICS410.COLDNEXT'"
/**************************************************************************/ /* Does an output list from previous run, if it already exists and      */
/* is waiting to be processed.                                           */
/**************************************************************************/ if SYSDSN('''list_dsn'''') = 'OK' then do
    say 'Output dataset 'list_dsn' already exists'
    say 'Do you wish to process from this? (Y/N)'
PULL answer
    if answer = 'Y' then do
        CALL read
        CALL create
        exit Ø
    end
end
/**************************************************************************/ /* Determine list of regions that match entered mask                   */
/**************************************************************************/ say 'The following regions will be included in this run.'
say 'After which you will have the option to create any required
say 'COLDNEXT datasets or output the list for editing.'
say
address ispexec "LMDINIT LISTID(LISTID) LEVEL("gcdmask")"
address ispexec "LMDLIST LISTID("listid") OPTION(LIST) DATASET(gcddsn)"
do while rc = Ø
    INTERPRET temp_start
    INTERPRET temp_end
    length = end - start
    temp_reg  = substr(gcddsn,start,length)
    INTERPRET temp_level
    INTERPRET temp_cold
total = total + 1
result = SYSDSN("'cold_dsn'")
if result = 'OK' then do
    say temp_reg ' ignored. COLDNEXT dataset already exists.'
end
else do
    say temp_reg ' included. COLDNEXT dataset required.'
    count = region.Ø + 1
    region.Ø = count
    region.count = temp_reg
end
address ispexec "LMDLIST LISTID("listid")OPTION(LIST)DATASET(gcddsn)
end
if region.Ø = Ø then do
    say
    say '*** There are no regions to be cold started ***'
    say
end
else do
/**********************************************************************/
/* Create COLDNEXT datasets immediately or write regions to dataset */
/**********************************************************************/
say 'Do you wish to edit region list before executing (Y/N)?'
PULL answer
if answer = 'N' then
    call CREATE
else if answer = 'Y' then
    call OUTPUT
else
    say 'Processing terminated with no actions'
end
exit Ø
/**********************************************************************/
/* Read in regions from dataset created in previous run of REXX */
/**********************************************************************/
READ:
say 'Do you want to run from this dataset? (Enter 'YES' to con
PULL answer
if answer = 'YES' then exit Ø
say
say 'Data being read in from ' list_dsn '. Please wait.'
address tso "ALLOCATE DA('' list_dsn '') FI(innput) SHR"
address tso "EXECIO * DISKR innput (FINIS stem region."
address tso "FREE FI(innput)"
if rc > Ø then
do
"---------------------------------------------------------------------
    say " ERROR on allocation of file " list_dsn
    say
exit 8
end
say 'Finished reading input'
total = region.Ø
return

CREATE:
  count = Ø
  do loop = 1 to region.Ø by 1
      temp_reg = strip(region.loop)
      INTERPRET temp_level
      INTERPRET temp_cold
      if SYSDSN('"cold_dsn"') = 'OK' then do
          say strip(region.loop) ' ignored. COLDNEXT already exists.'
      end else do
          address tso "ALLOCATE DA('"  cold_dsn  ") FI(OUTPUT) NEW",
            "DSORG(PS) RECFM(F,B) LRECL(8Ø) BLKSIZE(Ø)"
          address tso "FREE FILE(OUTPUT)"
          count = count + 1
          say strip(region.loop) ' - COLDNEXT dataset created.'
      end
  end
  say 'Total number of regions processed    = ' total
  say 'Number of COLDNEXT datasets required = ' count
  return

OUTPUT:
  if SYSDSN('"list_dsn"') = 'OK' then
      address tso "DELETE '"list_dsn""
      address tso "ALLOCATE DA('"  list_dsn  ") FI(OUTLIST) NEW",
        "DSORG(PS) RECFM(F,B) LRECL(8Ø) BLKSIZE(Ø)"
      address tso "EXECIO" region.Ø "DISKW OUTLIST (FINIS Stem region.)"
      address tso "FREE FILE(outlist)"
      say 'Region list written to ' list_dsn
  return

Phil Wilton
Systems Programmer
Norwich Union (UK)  © Xephon 1999
NEWCOPY of programs in an MRO environment

Amendments to heavily-used application programs can often result in moments of panic, when a newly-implemented program has to be backed out as quickly as possible and refreshed in the CICS system in which it is active.

In our CICS Version 4.1.0 MRO environment, we typically have many AORs connected to a single TOR. We have provided a method for a user signed on to the TOR to initiate tasks to refresh a program in all attached AORs and get a prompt and visible response to the command on a scrolling list of AORs. This development also permits the CICS systems programmer to devolve the responsibility of issuing NEWCOPY commands to the group responsible for implementing (or backing out) the program.

It consists of three transactions (NCOP, NCO1, and NCO2, two of which run in the TOR only), four programs (MRONCOPY, MRONCOP1, MRONCOP2, and MROIDSYS), plus one BMS screen map (NEWCOPY).

Having entered the transaction NCOP, the user will see a scrolling list of all AORs attached to the TOR. It allows for up to fifty systems to be connected in MRO (although this could easily be extended). The user can enter the name of the program to be refreshed and, by depressing ENTER repeatedly, will see the results of the ‘SET PROGRAM PHASEIN’ command that is issued to all connected AORs. Because we use auto-install for programs at our installation (PGAIPGM=ACTIVE in the SIT), if the program has not been loaded into an application region, the message ‘PROGRAM NOT FOUND’ will be displayed against the region’s SYSID on the screen display. This can also be a useful method for determining the usage of programs across your MRO configuration.

MRONCOPY

*ASM XOPTS(SP)
TITLE 'MRONCOPY - MRO NEWCOPY INITIAL SCREEN SEND'
LCLC &REL
&REL  SETC  '4.1'
  DFHREGS
SYSREG  EQU  6
  DFHEISTG
ACQUIRED EQU  C'A'
RELEASED EQU  C'R'
************************
SYSCNT  DS  F
SYSIDS  DS  50CL5  max number of SYSIDS = (49 + 1)
************************
SAVE14  DS  F
CRES   DS  F
LENF   DS  H
TSMAG  DS  CL(MSSGL)
CMPIR  DS  CL8
NAME   DS  CL8
COPY  DFHAID                 AID key definitions
COPY  DFHBMSCA               BMS attribute definitions
************************
* Screen Map                        *
************************
COPY  NEWCOPY
NEXTSYS EQU  (LINE005L-LINE004L)  length of detail line
MSSGL EQU  L'MSGAR040
MAXSYS EQU  14 maximum number of SYSID lines on screen
MRONCOPY CSECT
  B     START
  DC    C'MRONCOPY '
  DC    C'R: &REL '
  DC    C'&SYSDATE '
  DC    C'&SYSTIME '
*****************************************************************************
* Set up screen area                *
*****************************************************************************
START  DS  ØH                      .
  MVC  QNAME(4),EIBTRMID .
  MVC  QNAME+4(4),=C'NCPY'
 EXEC  CICS READQ TS QUEUE(QNAME) ITEM(1) +
       INTO(PGMD)  +
       RESP(CRES)  .
  BAL   R14,GETSYS  .
*****************************************************************************
* RETRIEVE and display any messages *
*****************************************************************************
  LA    R0,L'MSGLINEO  .
  STH   R0,LENF    .
 EXEC  CICS RETRIEVE INTO(MSGLINEO) LENGTH(LENF) +
       RESP(CRES)  .
*****************************************************************************
* Delete any old TS queue and . *
* SEND ERASE on first screen . *
******************************************************************************
SENDIST DS 0H .
  EXEC CICS DELETEQ QUEUE(QNAME) +
    RESP(CRES)
  EXEC CICS SEND MAP('NEWCOPY') MAPSET('NEWCOPY') +
    WAIT +
    ERASE +
    ALARM
******************************************************************************
* RETURN *
******************************************************************************
RETURN EXEC CICS RETURN TRANSID(NEXTRAN)
******************************************************************************
* Subroutines *
******************************************************************************
**
** ————————————————— GETSYS —————————————————**
* Find attached systems *
******************************************************************************
GETSYS DS 0H
ST R14,SAVE14
LA R0,(4+50*5)
STH R0,LENF
EXEC CICS LINK PROGRAM('MROIDSYS') +
  COMMAREA(SYSCNT) +
  LENGTH(LENF)
L R2,SYSCNT number of MRO systems
LA SYSREG,SYSIDS point to first SYSID
LA R7,SYSID040 point to map start field
CH R2,=Y(MAXSYS) do not exceed screen
BNH SYSLOOP limits
SYSLOOP DS 0H
  MVC Ø(4,R7),Ø(SYSREG) complete SYSID
  MVI TSMG,X'40' clear message area
  MVC TSMG+1(MSSGL-1),TSMSG
  CLI 4(SYSREG),RELEASED is this system available?
  BNE STARTASK
  MVC TSMG(L'NOTAVBL),NOTAVBL no - send unavailable mssg
STARTASK DS 0H
  MVC (MSGARØ40-SYSIDØ40)(MSSGL,R7),TSMSG display message
  LA SYSREG,L'SYSIDS(SYSREG) increment SYSID
  LA R7,NEXTSYS(,R7) increment map pointer
  BCT R2,SYSLOOP loop until last AOR
  L R14,SAVE14.
  BR R14
******************************************************************************
* CONSTANTS *
******************************************************************************
MRONCOP1

*ASM XOPTS(SP)

    TITLE 'MRONCOP1 - MRO NEWCOPY RETURN MESSAGES'
    LCLC &REL
    &REL SETC '4.1'
    DFHREGS

COMMREG EQU 5
SYSREG EQU 6

DFHEISTG

ACQUIRED EQU 'C'A'
RELEASED EQU 'C'R'

************************
SYSCNT DS F
SYSIDS DS 50CL5              max number of SYSIDS = (49 + 1)
************************
SAVE14 DS F
CRES DS F
LENF DS H
ITNUM DS H

************************
TQNAME DS CL8
TSREC DS 0CL(12+MSSGL)
TSSYSID DS CL4
TSPGMID DS CL8
TSMSG DS CL(MSSGL)
************************
CMAREA DS 0CL12
CMPGMID DS CL8
CMSCLINE DS FL4
************************
STRTREC DS 0CL12
STRTSYS DS CL4
STRTSQ DS CL8
STRTRECL EQU *-STRTREC
************************
MSGLINE DS CL35
COPY DFHAID             AID key definitions
COPY DFHBMSCA           BMS attribute definitions
*******************************************************************************
*        Screen Map                          *
*******************************************************************************
COPY NEWCOPY

NEXTSYS EQU (LINE005L-LINE004L) length of detail line
MSSGL EQU L'MSGARØ40
MAXSYS EQU 14               maximum number of SYSID lines on screen
MRONCOP1 CSECT
   B     START
   DC   C'MRONCOP1 '
   DC   C'R: &REL '
   DC   C'&SYSDATE '
   DC   C'&SYSTIME '

*****************************************************
*   Program flow                                       *
*****************************************************
START    DS ØH
   MVC   TQNAME(4),EIBTRMID              set up TS queue name
   MVC   TQNAME+4(4),=C'REMQ'
   MVC   STRTTSQ(8),TQNAME               QNAME passed to AOR

*****************************************************
*    AIDs : ENTER  refresh screen                    *
*           PF3    return to first screen           *
*           PF4    return                           *
*           PF8    scroll forward                   *
*****************************************************
EXEC  CICS HANDLE AID                 +
       PF3 (GOBACK)                       +
       PF4 (RETURN)                       +
       PF8                                +
       ENTER                              +
       ANYKEY (INVKEY)
*
EXEC  CICS IGNORE CONDITION MAPFAIL
EXEC  CICS RECEIVE MAP('NEWCOPY') MAPSET('NEWCOPY') ASIS

*****************************************************
* If returning from screen send, we have COMMAREA    *
*****************************************************
OC    EIBCALEN,EIBCALEN              is there a COMMAREA?
BZ    NOCOMM                         no - first time thru
L     COMMREG,DFHEICAP               address of COMMAREA
MVC   CMAREA(L'CMAREA),Ø(COMMREG)    restore COMMAREA

NOCOMM DS ØH
   BAL   R14,GETSYS                     fill in system-ids
   OC    CMPGMID,CMPGMID                have we a program-id?
   BZ    GETPGMID                       no
   MVC   CMDINPO(L'CMDINPO),CMPGMID     restore screen prog-id
   MVI   CMDINPA,DFHBMPRO               protect program-id field
   BAL   R14,READTSQ                    read message queues
   B    SENDMAP

GETPGMID DS ØH
   CLI   EIBAID,DFHPF8                  are we scrolling?
   BE    SENDMAP                        go to next screen
   OC    CMDINPI,CMDINPI                PGMID entered?
   BZ    NOINPUT                        tell them if not
MVI CMDINPA,DFHBMPRO  protect program-id
MVC CMPGMID,CMDFNPI  and store it
BAL R14,STRTALL  start NEWCOPY tasks
XC CMSCLINE,CMSCLINE  reset scroll to zero

********************************************
*        Send map and return here          *
********************************************
SENDMAP DS @H
EXEC CICS SEND MAP('NEWCOPY') MAPSET('NEWCOPY')
FROM(NEWCOPYA)
LENGTH(AL2(NEWCOPYL))
*
EXEC CICS RETURN TRANSID(EIBTRNID)
COMMAREA(CMAREA) LENGTH(12)

********************************************
*        Error conditions                   *
********************************************
NOINPUT DS @H
MVI MSGLINE,X'40'
MVC MSGLINE+1(L'MSGLINE-1),MSGLINE
MVC MSGLINE(L'NOPROG),NOPROG
B GOBACK
*
INVKEY DS @H
MVI MSGLINE,X'40'
MVC MSGLINE+1(L'MSGLINE-1),MSGLINE
MVC MSGLINE(L'IKMSG),IKMSG
B GOBACK

********************************************
* Go back to initial screen with optional message *

GOBACK DS @H
EXEC CICS START TRANSID('NCOP')
FROM(MSGLINE) LENGTH(35)
TERMID(EIBTRMID)

********************************************
*        Return                                 *

RETURN DS @H
EXEC CICS RETURN

********************************************
*        SUBROUTINES                         *

**                                          **
**———————————————— GETSYS ——————————————————**
*        Find attached systems               *

GETSYS DS @H
ST R14,SAVE14
LA R0,(4+50*5)
STH R0,LENF
EXEC CICS LINK PROGRAM('MROIDSYS') +
COMMAREA(SYSCNT) +
LENGTH(LENF)
L R2,SYSCNT number of MRO systems
L R14,CMSCLINE current start line
CLI EIBAID,DFHPF8 are we scrolling?
BNE GETSYS1
LA R14,MAXSYS(R14) add 1 pageful
ST R14,CMSCLINE save start line
SR R2,R14 is there a next page to go to?
BP GETSYS1 yes - so go scroll
L R2,SYSCNT get number of systems
XC CMSCLINE,CMSCLINE reset scroll amount to zero
GETSYS1 DS ØH
L R14,CMSCLINE get current start line
MH R14,=Y(L'SYSDIDS) get disp into SYSDIDS
LA SYSREG,SYSDIDS(R14)
LA R7,SYSIDØ40 point to map start field
CH R2,=Y(MAXSYS) do not exceed screen
BNH SYSLOOP limits
LA R2,MAXSYS
SYSLOOP DS ØH
MVC Ø(4,R7),Ø(SYSREG) complete SYSID
MVI TMSG,'40' clear message area
MVC TMSG+1(MSSGL-1),TMSG
CLI 4(SYSREG),RELEASED is this system available?
BNE STARTASK
MVC TMSG(L'NOTAVBL),NOTAVBL no - send unavailable mssg
STARTASK DS ØH
MVC (MSGARØ40-SYSDIDØ40)(MSSGL,R7),TMSG display message
LA SYSREG,L'SYSDIDS(SYSREG) increment SYSID
LA R7,NEXTSYS(.R7) increment map pointer
BCT R2,SYSLOOP loop until last AOR
L R14,SAVE14
BR R14

**                                             **
**———————————————————— STRTALL ————————————————**
* Start tasks to refresh program in all regions *
******************************************************************************
STRTALL DS ØH
ST R14,SAVE14
L R2,SYSCNT get number of systems
LA SYSREG,SYSDIDS get first SYSID
LA R5,1 set up counter reg
LA R7,SYSIDØ40 point to map start field
MVC STRTSYS,SYSDIDS
STLOOP DS ØH
XC TSREC,TSREC clear TS rec
CLI 4(SYSREG),RELEASED is this system available?
BE STRTEND no further action
MVC TMSG(L'REFMSG),REFMSG set up default message

** Delete TS queue **
EXEC CICS DELETEQ TS QUEUE(TQNAME) +
    SYSID(Ø(SYSREG)) +
    RESP(CRES)

** Write new TS queue **
EXEC CICS WRITEQ TS QUEUE(TQNAME) FROM(TSREC) +
    LENGTH(70) +
    SYSID(Ø(SYSREG))

** Start transaction **
EXEC CICS START TRANSID('NCO2') +
    INTERVAL(2) +
    FROM(STRTREC) +
    LENGTH(=Y(STRTRECL)) +
    SYSID(Ø(SYSREG))

** Read TS queues and send messages to screen **
* Read TS queues and send messages to screen *

```
READTSQ  DS  ØH
ST  R14,SAVE14
LA  RØ,1
STH RØ,ITNUM          get first item
L  R2,SYSCNT          get number of systems
XR  R14,R14
CLI EIBAID,DFHPFB     are we scrolling?
BNE NOSCRL
L  R14,CMSCLINE      get current start line
SR  R2,R14
MH  R14,=Y(L'SYSDS)  get disp into SYSDS

NOSCRL  DS  ØH
LA  SYSREG,SYSDS(R14)
LA  R7,SYSDØ40        point to map start field
CH  R2,=Y(MAXSYS)     do not exceed screen
BNH READLOOP         limits
LA  R2,MAXSYS

READLOOP  DS  ØH
CLI 4(SYSREG),RELEASED is this system available?
BNE STRTREAD
MVC TSMSG(L'NOTAVBL),NOTAVBL send unavailable mssg
B READEND          no further action

STRTREAD  DS  ØH
XC  TSREC,TSREC      clear TS rec
```
EXEC CICS READQ TS QUEUE(TQNAME) +
  ITEM(ITNUM) +
  INTO(TSREC) +
  SYSID(Ø(SYSREG)) +
  RESP(CRES)
READEND DS ØH
MVC (MSGARØ40-SYSIDØ40)(MSSGL,R7),TSMSG display message
LA SYSREG,L'SYSIDS('SYSREG) next SYSID
LA R7,NEXTSYS('R7) next message line
BCT R2,READLOOP process next record
L R14,SAVE14
BR R14

***********************************************************************
*          CONSTANTS          *
***********************************************************************
NOPROG   DC C'Enter PROGRAM to NEWCOPY'
IKMSG    DC C'Function not available    '
NOTAVBL  DC C' — SYSTEM NOT AVAILABLE    — '
REFMSG   DC C' — NEWCOPY COMMAND ISSUED — '
END
*********************************************************************** Bottom of Data *****************************************************

MRONCOP2

*ASM XOPTS(SP)
  TITLE 'MRONCOP2 - MRO NEWCOPY INVOKE PHASEIN COMMAND'
LCLC &REL
&REL SETC '4.1'
DFHREGS
DFHEISTG
LSYSID DS CL4
STRTREC DS ØCL12
STRTSYS DS CL4
STRTSQ DS CL8
STRTRECL EQU *-STRTREC
LENSTA DS H
TSREC DS ØCL(12+MSSGL)
TSSYSID DS CL4
TSPGMID DS CL8
TSMSG DS CL(MSSGL)
TSRECL EQU *-TSREC
CRES DS F
LENTSQ DS H
ITNUM DS H

***********************************************************************
* Screen Map     *
***********************************************************************
COPY NEWCOPY
MSSGL EQU L'MSGARØ40
MRONCOP2 CSECT
  B START
  DC C'MRONCOP2 '
  DC C'R: &REL '
  DC C'&SYSDATE '
  DC C'&SYSTIME '
******************************************************************************
  *Retrieve commands                          *
******************************************************************************
START DS ØH
  LA R6,STRTRECL
  STH R6,LENSTA
  EXEC CICS RETRIEVE INTO(STRTREC) LENGTH(LENSTA)
  EXEC CICS ASSIGN SYSID(LSYSID)
  XC TSREC,TSREC clear input
  LA R6,TSRECL
  STH R6,LENTSQ
  EXEC CICS HANDLE CONDITION ITEMERR(TSEXIT)
  EXEC CICS READQ TS QUEUE(STRTTSQ) INTO(TSREC) + LENGTH(LENTSQ) + ITEM(1)
  *
  XC TSMMSG(MSSGL),TSMG clear input
******************************************************************************
  *Let's refresh the program           *
******************************************************************************
REFRESH DS ØH
  EXEC CICS SET PROGRAM(TSPGMID) PHASEIN + RESP(CRES)
  *
  CLC CRES,DFHRESP(PGMIDERR) program not found
  BE NOPGM
  CLC CRES,DFHRESP(NOTAUTH) unauthorized user
  BE NOTAUTH
  CLC CRES,DFHRESP(NORMAL)
  BNE BADCOPY unspecified error
  MVC TSMMSG(L'GDINPUT),GDINPUT
  B GOAWAY
BADCOPY DS ØH
  MVC TSMMSG(L'MSG1),MSG1
  B GOAWAY
NOPGM DS ØH
  MVC TSMMSG(L'MSG2),MSG2
  B GOAWAY
NOTAUTH DS ØH
  MVC TSMMSG(L'MSG3),MSG3
  B GOAWAY
OTHERR DS ØH
  MVC TSMMSG(L'MSG4),MSG4
  B GOAWAY

© 1999. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (940) 455 7050, fax (940) 455 2492.
MROIDSYS

*ASM XOPTS(SP)
  TITLE 'MROIDSYS - FIND NUMBER OF ATTACHED AORS'
  LCLC &REL
  &REL SETC '4.1'
  DFHREGS
  COMMREG EQU 4
  SYSREG EQU 6
  DFHEISTG
  ACQUIRED EQU C'A'
  RELEASED EQU C'R'
  CONNECT DS F
  CONACC DS F
  STATUS DS F
  SAVE14 DS F

MROIDSYS CSECT
  B START
  DC C'MROIDSYS '
  DC C'R: &REL '
  DC C'&SYSDATE '
  DC C'&SYSTIME '

  **********************************************
  * Retrieve any COMMAREA *
  **********************************************

START DS ØH
DC EIBCALEN,EIBCALEN is there a COMMAREA?
**——GETSYS——**

* Get SYSIDs of all connected AORs *

*******************************************************************************
* Subroutines                    *
*******************************************************************************

GETSYS EQU *
ST R14,SAVE14
LA R2,1

** get the SYSID of the region we are in **
EXEC CICS ASSIGN SYSID(SYSIDS)
LA SYSREG(SYSIDS+L'SYSIDS
XC CONACC,CONACC
EXEC CICS HANDLE CONDITION END(CONEND)
EXEC CICS INQUIRE CONNECTION START

CONLOOP EXEC CICS INQUIRE CONNECTION(CONNECT) +
   ACCESSMETHOD(CONACC) +
   CONNSTATUS(STATUS) +
   NEXT
CLC CONACC,DFHVALUE(XM)
BNE CONLOOP
LA R2,1(),R2
MVC Ø(4,SYSREG),CONNECT save SYSID
MVI 4(SYSREG),ACQUIRED set system indic
CLC STATUS,DFHVALUE(ACQUIRED) system available?
BE CONLOOP1
MVI 4(SYSREG),RELEASED it seems not

CONLOOP1 DS ØH
LA SYSREG,L'SYSIDS(),SYSREG
B CONLOOP

CONEND EXEC CICS INQUIRE CONNECTION END
ST R2,SYSCNT
L R14,SAVE14
BR R14

*******************************************************************************
* Record maps                     *
*******************************************************************************

COMMDSCT DSECT
SYSCNT DS F
SYSIDS DS CL5
END

******************************************************************************* Bottom of Data *******************************************************************************
NEWCOPY

NEWCOPY DFHMSD TYPE=DESCT,MODE=INOUT,TERM=3270-2,
CTRL=(PRINT,L80,FREEKB,FRSET),
TIOAPFX=YES

NEWCOPY DFHMDI SIZE=(24,80),LINE=1,COLUMN=1,JUSTIFY=LEFT,
MAPATT=COLOR,DSATT=COLOR,COLOR=BLUE

MSGLINE DFHMDF POS=(1,3),ATTRB=(PROT,BRT),LENGTH=35,
INITIAL='                    ',COLOR=DEFAULT

TITLE1 DFHMDF POS=(2,30),ATTRB=(PROT,BRT),LENGTH=20,
INITIAL='MRO NEWCOPY FACILITY',COLOR=DEFAULT

TITLE2 DFHMDF POS=(3,30),ATTRB=PROT,LENGTH=20,COLOR=BLUE,
INITIAL='——————————'

CMDLIN DFHMDF POS=(5,1),ATTRB=(PROT,BRT),LENGTH=13,
INITIAL='PROGRAM ==>',COLOR=DEFAULT

CMDINP DFHMDF POS=(5,14),ATTRB=(UNPROT,IC),LENGTH=8,
INITIAL='  ',COLOR=RED

CMDEND DFHMDF POS=(5,23),ATTRB=(ASKIP,PROT),LENGTH=1,
INITIAL='*',COLOR=DEFAULT

LINE001 DFHMDF POS=(6,1),ATTRB=PROT,LENGTH=80,
INITIAL='  ————————————————————————————————————',COLOR=BLUE

LINE002 DFHMDF POS=(7,1),ATTRB=PROT,LENGTH=17,
INITIAL='SYSID  |  RESULTS',COLOR=BLUE

LINE003 DFHMDF POS=(8,1),ATTRB=PROT,LENGTH=80,
INITIAL='  ————————————————————————————————————',COLOR=BLUE

SYSID04 DFHMDF POS=(9,2),ATTRB=PROT,LENGTH=4,
COLOR=BLUE

INPUT04 DFHMDF POS=(9,10),ATTRB=(PROT,BRT),LENGTH=5,
INITIAL='  ',COLOR=DEFAULT

END004 DFHMDF POS=(9,16),ATTRB=(ASKIP,PROT),LENGTH=1,
INITIAL='  ',COLOR=DEFAULT

MSGAR04 DFHMDF POS=(9,20),ATTRB=(PROT,BRT),LENGTH=58,
INITIAL='  ',COLOR=DEFAULT

LINE005 DFHMDF POS=(10,1),ATTRB=PROT,LENGTH=17,
INITIAL='                        ',COLOR=BLUE

SYSID05 DFHMDF POS=(10,2),ATTRB=PROT,LENGTH=4,
COLOR=BLUE

INPUT05 DFHMDF POS=(10,10),ATTRB=(PROT,BRT),LENGTH=5,
INITIAL='                        ',COLOR=DEFAULT

END005 DFHMDF POS=(10,16),ATTRB=(ASKIP,PROT),LENGTH=1,
John Hall
CICS Systems Programmer
Cooperative Insurance Society (UK) © J D Hall 1999
National language sort in CICS

Temporary storage queues are usually used in CICS programs to store the results of the read/fetch of two or more rows from a file or database, so the end user can browse through the list of records retrieved according to his selection criteria.

If we want to present the result to the end user in sorted form (usually by surname, first name, primary key code, etc), the ORDER BY statement (DB2) must be part of the EXEC SQL statement that fetches records, or the records should be retrieved by using the index (index is always a sorted structure). It sounds so simple. But can you always do so and is the record order always correct? In fact, you cannot always use the system sort.

In DB2, there are cases when the ORDER BY statement cannot be used. There are also cases when the ORDER BY statement leads to internal execution of your query producing so called ‘view materialization’ and your query executes for ages. In DL/I and VSAM, you can only use sort by retrieving the records using the index and/or using the bank definition (segments are defined to be sorted).

Can you sort records if they are retrieved from different sources or different objects? And can you sort records by using, for example, the second column in the index?

NATIONAL LANGUAGES AND CODE PAGES

It is really easy if English is your national language – in every code page (EBCDIC, ASCII) the letters are positioned according to their position in the English alphabet. So, if you want to sort two words, the system easily translates characters to their numeric values and compares the numeric representation of the characters on specific positions in the words.

But what about other national languages? When code pages were adapted to non-English languages the same logic for positioning non-English letters in the code page was not applied to the non-English alphabets. English letters were kept in the same places in the code page and positions of infrequently-used characters (eg square bracket)
were used to record non-English letters. Because those characters are randomly spread across the code page, it is not possible to sort by simply translating characters to their numeric values.

IBM has proposed some solutions for enabling a correct sort (eg FIELDPROC in DB2) but this is far from being a general and easy solution, especially if you retrieve records from different sources. Anyone who has tried to get a correct non-English sort soon comes to the same opinion – it is just not possible! DB2, for example, doesn’t have as nice an interface as Oracle, where you can specify in which language the sort should be done and the system retrieves records in the correct order.

Another possible problem is that some letters in non-English alphabets actually consist of two or more characters (eg ‘NJ’ in Croatian). And what if two characters should be treated as if they are the same character even though they have different numeric representations (eg ‘u’ and ‘ü’ in German)?

SORT USING THE CROATIAN ALPHABET

Although I will describe my solution for sorting words according to the Croatian alphabet, anyone who has a similar problem can easily translate the solution for their own national language sort. The Croatian alphabet consists of the following letters:


Where ‘?’ represents letters that do not exist in the English alphabet. There are also some letters in the English alphabet that don’t exist in the Croatian alphabet (eg Q, W), and other letters that consist of two characters (eg LJ).

The numeric values for ‘C’ and ‘D’ are whole numbers and the difference is equal to 1. So how can we specify that letters ‘?’ and ‘?’ are between those letters? The only way is to assign not the actual numeric value from the code page, but, instead, a decimal value that is greater than the numeric value of the letter ‘C’ and smaller than the numeric value of letter ‘D’. Also, the decimal value of the first letter ‘?’ must be less than the decimal value of the second letter ‘?’, if we want to have the correct order.
What about the letter ‘NJ’, which consists of two characters and is positioned just after the letter ‘N’? This means that whenever we find an occurrence of ‘N’ in a string, we have to check whether the next character is ‘J’. If it is not ‘J’, we use the numeric value of letter ‘N’; if it is ‘J’, we assign to the letter ‘N’ a decimal value that is between the numeric value of ‘N’ and the numeric value of ‘O’.

Finally, what about the English letter ‘W’, which is not part of the Croatian alphabet but is sometimes found in names? According to the Croatian sort, it should be treated as ‘V’. This means that whenever we find an occurrence of the character ‘W’ in the string, we must not use the numeric value of ‘W’ but instead use the numeric value of the character ‘V’.

THE SOLUTION
The following requirements are fulfilled by the solution given in this article:

• When the user specifies search criteria, rows and records are fetched under CICS from different sources (DB2, DL/I, VSAM) and stored in a CICS temporary storage queue.

• The items in the temporary storage queue should be sorted before being displayed to the user.

• Sort criteria can be by item (full sort) or by specific item fields.

• The sort order can be ascending or descending.

• Records with the same value of sort criteria can be kept or removed from the temporary storage queue.

• The sort technique that is used should be efficient and should require as few system resources as possible.

• All programmers should use a standard program that will cover all requirements and can be called from their CICS programs.

To satisfy these requirements, I have created the CICS program called SRTC0010 that programmers link to from their CICS programs.

The usual user program that links sort programs looks like the following:
As you can see from the program source, two members should be included before linking the sort program. They should be stored in the PDS where they can be read-only accessed by programmers. Member SRTK0001 contains the structure describing the COMMAREA that is an interface between the user and the sort program. Member SRTR0001 contains sort defaults that can be overridden before linking to the sort program.

SORT COMMAREA SRTK0001

/********************************************************************/
/* SORT TS COMMAREA DEFINITION **************************************/
/********************************************************************/
DCL SORT_PTR PTR;
DCL SORT_STR CHAR(128) BASED(SORT_PTR);
DCL 1 SORT_VARS BASED(SORT_PTR),
  2 TS_NAME CHAR(8),
  2 ITEM_NUMBER BIN FIXED(15),
  2 ITEM_LENGTH BIN FIXED(15),
  2 SORT_ORDER CHAR(1),
  2 DELETE_DUPLICATE CHAR(3),
  2 FIELD_SORT CHAR(3),
  2 FIELD_NUMBER BIN FIXED(15),
  2 START_POSITION(10) BIN FIXED(15),
  2 FIELD_LENGTH(10) BIN FIXED(15),
  2 ITEMNO_PRESERVED BIN FIXED(15),
  2 ITEMNO_DELETED BIN FIXED(15);

SORT DEFAULTS SRTR0001

/********************************************************************/
/* SORT VARIABLES INITIALIZATION - DEFAULTS *************************/
/********************************************************************/
DCL SORT_TMP CHAR(128) INIT(' ');SORT_PTR=ADDR(SORT_TMP);
SORT_VARS.FIELD_SORT='NO '; /* TS ITEM FULL SORT */
After including these members, you can specify sort parameters before linking the sort program. The sort parameters are:

- **SORT_VARS.TS_NAME** – the temporary storage queue of items that should be sorted. This is mandatory with no default.
- **SORT_VARS.ITEM_NUMBER** – the number of items in temporary storage that should be sorted. This is mandatory with no default.
- **SORT_VARS.ITEM_LENGTH** – the length of items in temporary storage that should be sorted. This is mandatory with no default.
- **SORT_VARS.SORT_ORDER** – the required sort order of the items in temporary storage. This is mandatory. Permitted values are ‘A’ (Ascending) and ‘D’ (Descending). The default is A.
- **SORT_VARS.DELETE_DUPLICATE** – whether to delete or keep items that have the same sort criteria. This is mandatory. The permitted values are ‘NO’ (keep duplicates) and ‘YES’ (delete duplicates). The default is NO.
- **SORT_VARS.FIELD_SORT** – whether to sort items or specific field(s) in the items. This is mandatory. The permitted values are ‘NO’ (full item sort) and ‘YES’ (field(s) sort). The default is NO.
- **SORT_VARS.FIELD_NUMBER** – the number of fields that describe the sort criteria. This is mandatory if SORT_VARS.FIELD_SORT = YES. The permitted values are 1 to 10. There is no default.
- **SORT_VARS.START_POSITION(n)** – the starting position of the ‘nth’ field in the item. This is mandatory if SORT_VARS.FIELD_SORT = YES and n is less than or equal to SORT_VARS.FIELD_NUMBER. The permitted values are 1 to SORT_VARS.ITEM_LENGTH. There is no default.
- **SORT_VARS.FIELD_LENGTH(n)** – the length of the ‘nth’ field in the item. This is mandatory if SORT_VARS.FIELD_SORT = YES and n is less than or equal to SORT_VARS.FIELD_NUMBER. The permitted values are 1 through to
SORT_VARS.ITEM_LENGTH – SORT_VARS.START_POSITION(n) + 1. There is no default.

- **SORT_VARS.ITEMNO_PRESERVED** – the sort program returns the number of items after the sort. This can be different from SORT_VARS.ITEM_NUMBER if SORT_VARS.DELETE_DUPLICATE *= YES.

- **SORT_VARS.ITEMNO_DELETED** – the sort program returns the number of items that are deleted during the sort. This can be other than zero if SORT_VARS.DELETE_DUPLICATE *= YES.

This might look complicated, but if the sort to be performed is ‘simple’, as it usually is, it is enough to specify the temporary storage queue name, the number of items, and the item length. After the sort, the temporary storage queue will contain all items sorted in ascending order.

**SRTC0010**

The source of the sort program SRTC0010 follows:

```plaintext
* PROCESS INCLUDE, XOPTS(CICS);
SORT:PROC(SORT_PTR) OPTIONS(MAIN,NOEXECOPS) REORDER;

/* INCLUDE STATMENTS OF COMMAREA DEFINITION */
%INCLUDE SRTKØØØ1;

/* DECLARATIONS OF BUILTIN FUNCTIONS */
DCL (ADDR,DECIMAL,CSTG,STG,STG,STG) BUILTIN;

/* DECLARATIONS OF BASE ENGLISH LANGUAGE CHARACTERS */
/* THAT IMMEDIATELY PRECEED THE NATIONAL CHARACTERS */
DCL CCC CHAR INIT('C');DCL CC BIT(8) BASED(ADDR(CCC));
DCL CCD CHAR INIT('D');DCL CD BIT(8) BASED(ADDR(CCD));
DCL CCL CHAR INIT('L');DCL CL BIT(8) BASED(ADDR(CCL));
DCL CCN CHAR INIT('N');DCL CN BIT(8) BASED(ADDR(CCN));
DCL CCS CHAR INIT('S');DCL CS BIT(8) BASED(ADDR(CCS));
DCL CCV CHAR INIT('V');DCL CV BIT(8) BASED(ADDR(CCV));
DCL CCZ CHAR INIT('Z');DCL CZ BIT(8) BASED(ADDR(CCZ));

/* DECLARATIONS OF VARIABLES AND CONSTANTS */
DCL (I,J,K,GRAY,RECPOM,POS1) BIN FIXED(15);
DCL BFNUL BIN FIXED(15,3) INIT(Ø.Ø);
DCL (CMPLN) BIN FIXED(15) INIT(Ø);
DCL 1 SRT,
  2 SORT_LENGTH BIN FIXED(15) INIT(Ø),
  2 NO_CHANGES BIN FIXED(15) INIT(Ø).
```
2 NO_WRITEQ               BIN FIXED(15) INIT(Ø);
/* CALCULATING THE LENGTH OF SORT FIELDS */
IF SORT_VARS.FIELD_SORT='YES' THEN DO;
   DO I=1 TO 5;
      IF (SORT_VARS.FIELD_NUMBER>=I) THEN
         SRT.SORT_LENGTH=SRT.SORT_LENGTH+SORT_VARS.FIELD_LENGTH(I);
   END;
END;
ELSE SRT.SORT_LENGTH=SORT_VARS.ITEM_LENGTH;
/* SORT AREA DECLARATION AND ALLOCATION */
DCL TSREC(Ø:SORT_VARS.ITEM_NUMBER-1) CHAR(SORT_VARS.ITEM_LENGTH) CTL;
DCL TSRECNO(Ø:SORT_VARS.ITEM_NUMBER-1) BIN FIXED(15) CTL;
DCL (CMPS1,CMPS2) CHAR(SRT.SORT_LENGTH) CTL;
DCL STRPOM CHAR(SORT_VARS.ITEM_LENGTH) CTL;
ALLOCATE TSREC;
ALLOCATE TSRECNO;
ALLOCATE CMPS1;ALLOCATE CMPS2;
ALLOCATE STRPOM;
/* STORING TS ITEMS INTO SORT AREA */
DO I=1 TO SORT_VARS.ITEM_NUMBER;
   J=I-1;
   EXEC CICS READQ TS QUEUE(SORT_VARS.TS_NAME) ITEM(I)
      INTO(TSREC(J));
   TSRECNO(J)=J;
END;
/* SORTING */
SORT_VARS.ITEMNO_PRESERVED=SORT_VARS.ITEM_NUMBER;
SORT_VARS.ITEMNO_DELETED =Ø;
RASPON=SORT_VARS.ITEM_NUMBER/2;
DO WHILE(RASPON>Ø);
   I=RASPON;
   DO WHILE(I<SORT_VARS.ITEM_NUMBER);
      J=I-RASPON;
      DO WHILE(J>=Ø & J+RASPON<=SORT_VARS.ITEM_NUMBER-1 &
               (SORT_VARS.SORT_ORDER='A' & STRCMP(J,J+RASPON)>BFNUL |
                SORT_VARS.SORT_ORDER='D' & STRCMP(J,J+RASPON)<BFNUL));
         SRT.NO_CHANGES=SRT.NO_CHANGES+1;
         STRPOM=TSREC(J);
         TSREC(J)=TSREC(J+RASPON);
         TSREC(J+RASPON)=STRPOM;
         RECNOPOM=TSRECNO(J);
         TSRECNO(J)=TSRECNO(J+RASPON);
         TSRECNO(J+RASPON)=RECNOPOM;
         J=J-RASPON;
      END;
   I=I+1;
END;
RASPON=RASPON/2;
END;

/* DELETING DUPLICATES IF SPECIFIED AND OVERWRITING TS */
IF (SORT_VARS.DELETE_DUPLICATE='YES') THEN DO;
    EXEC CICS DELETEQ TS QUEUE(SORT_VARS.TS_NAME);
    CMPS2="";
    DO I=1 TO ITEM_NUMBER;
        J=I-1;
        IF (SORT_VARS.FIELD_SORT='YES') THEN DO;
            POS1=1;
            DO K=1 TO FIELD_NUMBER;
                SUBSTR(CMPS1,POS1,SORT_VARS.FIELD_LENGTH(K))=
                    SUBSTR(TSREC(J),
                        SORT_VARS.START_POSITION(K),
                        SORT_VARS.FIELD_LENGTH(K));
                POS1 = POS1 + SORT_VARS.FIELD_LENGTH(K);
            END;
            ELSE CMPS1=TSREC(J);
        IF (CMPS1≠CMPS2) THEN DO;
            SRT.NO_WRITEQ=SRT.NO_WRITEQ+1;
            EXEC CICS WRITEQ TS QUEUE(SORT_VARS.TS_NAME)
                FROM(TSREC(J));
            CMPS2=CMPS1;
        END;
    END;
    SORT_VARS.ITEMNO_PRESERVED=SRT.NO_WRITEQ;
    SORT_VARS.ITEMNO_DELETED=SORT_VARS.ITEM_NUMBER-
        SORT_VARS.ITEMNO_PRESERVED;
END;
ELSE DO;
    DO I=1 TO ITEM_NUMBER;
        J=I-1;
        IF (TSRECNO(J) ≠ J) THEN DO;
            SRT.NO_WRITEQ=SRT.NO_WRITEQ+1;
            EXEC CICS WRITEQ TS QUEUE(SORT_VARS.TS_NAME) ITEM(I)
                FROM(TSREC(J)) REWRITE;
        END;
    END;
END;

/* PROCEDURE THAT COMPARES TWO STRINGS ACCORDING TO THE */
/* NATIONAL ALPHABET                          */
STRCMP: PROCEDURE(RN1, RN2) RETURNS(BIN FIXED(15,3));
DCL (RN1, RN2) BIN FIXED(15);
DCL (CH1, CH2) BIN FIXED(15,3);
DCL (NXTC1,NXTC2) CHAR;
DCL (C1,C2) CHAR;
DCL J BIN FIXED(15);
DCL POSITION BIN FIXED(15) INIT(1);
IF (SORT_VARS.FIELD_SORT='YES') THEN DO;
    DO J=1 TO FIELD_NUMBER;
        ...
SUBSTR(CMPS1,POSITION,SORT_VARS.FIELD_LENGTH(J))=
    SUBSTR(TSREC(RN1),
    SORT_VARS.START_POSITION(J),
    SORT_VARS.FIELD_LENGTH(J));
SUBSTR(CMPS2,POSITION,SORT_VARS.FIELD_LENGTH(J))=
    SUBSTR(TSREC(RN2),
    SORT_VARS.START_POSITION(J),
    SORT_VARS.FIELD_LENGTH(J));
POSITION=POSITION+SORT_VARS.FIELD_LENGTH(J);
END;
END;
ELSE DO;
    CMPS1=TSREC(RN1);
    CMPS2=TSREC(RN2);
END;
DO J=1 TO SRT.SORT_LENGTH;
    IF (J=SRT.SORT_LENGTH) THEN DO;
        NXTC1='-';
        NXTC2='-';
    END;
    ELSE DO;
        NXTC1=SUBSTR(CMPS1,J+1,1);
        NXTC2=SUBSTR(CMPS2,J+1,1);
    END;
    C1=SUBSTR(CMPS1,J,1);
    C2=SUBSTR(CMPS2,J,1);
    CH1=Ø;CH2=Ø;
    IF (C1¬=C2 | NXTC1¬=NXTC2) THEN DO;
        CH1=CTN(C1,NXTC1);
        CH2=CTN(C2,NXTC2);
        IF (CH1 ¬= CH2) THEN RETURN(CH1-CH2);
    END;
END;
RETURN(BFNUL);
END STRCMP;
/* PROCEDURE THAT ASSIGN VALUES TO THE CHARACTERS ACCORDING TO THE */
/* NATIONAL ALPHABET                                      */
CTN: PROCEDURE(CH,NXTC) RETURNS(BIN FIXED(15,3));
    DCL (CH,NXTC) CHAR;
    DCL V BIT(8) BASED(ADDR(CH));
    DCL BF BIN FIXED(15,3);
    BF=V;
    SELECT(CH);
    WHEN ('?') BF=CC+.5;                 /* C < ? (¬) <  D */
    WHEN ('?') BF=CC+.7;         /* C < ? (¬) < ? (?) <  D */
    WHEN ('D') DO;                        /* D < D? (D|) < E */
        IF (NXTC='?') THEN BF=CD+.5;
    END;
    WHEN ('?') BF=CD+.7;        /* D < D? (D|) < ? (?) < E */
    WHEN ('L') DO;                           /* L < LJ < M */
IF (NXTC='J') THEN BF=CL+.5;
END;
WHEN ('N') DO;        /* N < NJ < M */
    IF (NXTC='J') THEN BF=CN+.5;
END;
WHEN ('?') BF=CS+.5;  /* C < (?) < D */
WHEN ('W') BF=CV;      /* W = V */
WHEN ('?') BF=CZ+.5;  /* C < (|) < D */
OTHERWISE;
END;
RETURN (BF);
END CTN;

/* FREEING SORT AREA */
FREE TSREC;
FREE TSRECNO;
FREE CMPS1;FREE CMPS2;
FREE STRPOM;

/* UNCOMMENT THE FOLLOWING IF YOU WANT TO TEST AND TRACE SORT */
/* DCL ISPVAR CHAR(255) INIT(' '); */
DCL 1 ISP BASED(ADDR(ISPVAR)),
  2 HD CHAR(21),
  2 BØ1 CHAR(Ø1),
  2 VØ1 CHAR(Ø8),
  2 LØ1 CHAR(Ø8),
  2 BØ2 CHAR(Ø1),
  2 VØ2 CHAR(12),
  2 LØ2 PIC'(5)9',
  2 BØ3 CHAR(Ø1),
  2 VØ3 CHAR(12),
  2 LØ3 PIC'(5)9',
  2 BØ4 CHAR(Ø1),
  2 VØ4 CHAR(11),
  2 LØ4 CHAR(3),
  2 BØ5 CHAR(Ø1),
  2 VØ5 CHAR(13),
  2 LØ5 PIC'(5)9',
  2 BØ6 CHAR(Ø1),
  2 VØ6 CHAR(11),
  2 LØ6 CHAR(1),
  2 BØ7 CHAR(Ø1),
  2 VØ7 CHAR(17),
  2 LØ7 CHAR(3),
  2 BØ8 CHAR(Ø1),
  2 VØ8 CHAR(17),
  2 LØ8 PIC'(5)9',
  2 BØ9 CHAR(Ø1),
  2 VØ9 CHAR(15),
  2 LØ9 PIC'(5)9';
ISP.HD='** SORT VARIABLE **' ;
The sort program uses the bubble-sort procedure. If you want to change the sort program for use with your national language alphabet, it is sufficient to specify at the beginning of the program the English language characters that immediately precede the national language characters and change the procedure CTN that assigns the decimal value to the characters.

IMPORTANT
It might be useful to program the sort program in Assembler – I think this might work even faster.

You should check the CICS initialization parameter that specifies how much processor time can be spent in the program between two EXEC CICS statements. If the time is too short, the sort program will abend when the sort is very big. You are free to increase it.

Josip Ivancic
Database Administrator (Croatia)
IBM has announced Version 3.1 of its CICS Universal Clients and CICS Transaction Gateway, with new hardware support for IBM Network Stations, Netfinity Servers, and Sun Sparc Ultra 5 and Ultra 10 machines.

The Universal Clients provide the basis for IBM’s Application Mining initiative, providing the means to access CICS applications users’ desktops via CICS servers on all platforms. The Transaction Gateway, used where multiple users are required, replaces the CICS Internet Gateway and the CICS Gateway for Java.

Besides new hardware support, the Universal Clients get enhancements to C++ and COM programming interfaces, EPI extensions to support extended terminal attributes and sign-on capable terminals, and improved supplied sample programs, including support for PL/I for NT.

The gateway gets the same improvements plus enhancements to Java programming interfaces and Common Connector Framework classes for CICS. It can also be enabled to run as an NT service and there’s new support for OS/390 Unix with enhanced security and performance. Also new is support for OS/390 System Secure Sockets Layer (SSL), user authentication via an X.509 client certificate, and mapping of client certificates to RACF user-ids.

For further information contact your local IBM representative.

***

HDS has announced its iSuite set of integrated e-business solutions. The suite’s built around Skyline Trinium and Pilot Series mainframes, as well as Freedom Storage 7700E disk arrays.

Its IntraPlex creates a complete Parallel Sysplex configuration within a single Skyline Trinium. Key elements include dynamic transaction routing, application data sharing for CICS, DB2, IMS, and VSAM environments, and the development and implementation of an overall strategy.

For further information contact your local HDS representative.

***

Candle has announced immediate support for OS/390 Version 2.8 in all relevant products. It also has expanded participation in the IBM SystemPac programme.

Candle also unveiled a new version of its Candle Command Center product for CICS Version 200, including enhanced usability and additional support for Parallel Sysplex and data sharing analysis, such as support for VSAM record-level sharing.

There’s also OMEGAMON II Version 500 for CICS with a range of new features including analysis of various types of Web-based connections in CICS, and new flexible user profile controls.

For further information contact:
Candle, 2425 Olympic Blvd, Santa Monica, CA 90404, USA.
Tel: (310) 829 5800.
Candle Services, 1 Archipelago, Lyon Way, Frimley, Camberley, Surrey, GU16 5ER, UK.
Tel: (01276) 414700.