239

CICS update

October 2005

In this issue

A xephon PUBLICATION

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

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

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs $270.00 in the USA and Canada; £175.00 in the UK; £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 December 2000 issue, are available separately to subscribers for $24.00 (£16.00) 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/cics; you will need to supply a word from the printed issue.

Contributions

When Xephon is given copyright, articles published in *CICS Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

Printed in England.

Zapping TCTTE by using an Assembler program

As stated in the June 2005 issue of *CICS Update* (see 'Using POVI for zapping TCTTE', issue 235), when terminals are stuck in 'Being Acquired' status for some reason, and there seems to be no way to make them operational other than recycling the CICS region, you have to zap the appropriate bits of the TCTTE in order to make the terminals operational again (without recycling the CICS region). The reason those terminals get stuck in 'Being Acquired' status is the inconsistency in the values at offset 1CD and 1CF of the TCTTE address, and to release the terminals from the CICS region you need to make the values at those offsets 00.

One of the ways that you can zap the appropriate bits of the TCTTE is by using POVI (Programmerless Open VTAM Interface) with Omegamon/CICS – and the source code for this POVI process was listed in the article in the June 2005 issue of *CICS Update*. But some users are not lucky enough to have both AF/OPERATOR and Omegamon/CICS in their environment, so cannot make use of POVI. These users who do not have AF/OPERATOR or Omegamon/CICS in their environment can use an Assembler program that determines the TCTTE address of the terminal and zaps the appropriate offsets.

The source code for such a program, running under CICS TS V2.2, is listed below. The terminal ID is passed to the program as a parameter. The program first determines the address of the CSA and then, by using the address of the CSA, determines the address of the terminal and the appropriate offsets. Afterwards it displays the TCTTE address of the terminal and zaps the appropriate offsets.

ASSEMBLER PROGRAM

```
*
* The terminal id is passed to this program as a parameter.
```

```
* This program displays the TCTTE address of the terminal and
* resets the values at offset "1CD" and "1CF" of this TCTTE
* as "ØØ".
*
*
*  SYNTAX: TTTT XXXX
*          (TTTT: Transaction id)
*           XXXX: Terminal id whose TCTTE will be zapped)
*
ADDRTCT  TITLE ' PGM. ZAPS TCT ENTRY '
***** CICS SUPPLIED DSECT DESCRIPTIONS - BEGIN *****
         DFHTM MF=(D,PARMLIST)
         DFHTM MF=(D,TMSTATIC)
         DFHTM MF=(D,SKTTBLE)
         DFHTM MF=(D,DIRELEM)
         DFHEJECT ,
***** CICS SUPPLIED DSECT DESCRIPTIONS - END *****
DFHEISTG DSECT ,
         DFHREGS                              REGISTER EQUATES
TCTTEAR  EQU   1Ø
TCTPFBAR EQU   11
         COPY  DFHCSADS                       CSA DSECT
         COPY  DFHTCTFX
         COPY  DFHTCTTE
*
ADDRTCT  AMODE 31
ADDRTCT  RMODE ANY
ADDRTCT  DFHEIENT CODEREG=(3,4,5)
RUN      EQU   *
************* RECEIVE TERMID *******
         MVC   PARMLEN,=H'Ø9'
         EXEC CICS RECEIVE INTO(PARMIN) LENGTH(PARMLEN)
**************************************
************* FIND CSA ADDRES *********
         L     R9,=F'28672'                   KCB ANCHOR (DFHKEKCB)
         CLC   2(9,R9),=C'>DFHKEKCB'          CHECK EYE CATCHER
         BNE   CSAERROR
         LA    R9,56Ø(R9)                     KCB + X'23Ø' = DOMAIN TABLE
         CLC   2(9,R9),=C'>DFHKEDOH'          CHECK EYE CATCHER
         BNE   CSAERROR
         A     R9,=F'4128'                    DFHKEDOH + X'1Ø2Ø' = DFHAP
         CLC   Ø(5,R9),=C'DFHAP'              CHECK POSITION
         BNE   CSAERROR
         L     R9,16(R9)                      DFHAP + X'1Ø' = CSA
**************************************
         B     ZAPTCT
*
******* INPUT PARAMETER DESC. **********
PARMLEN  DS    H
```

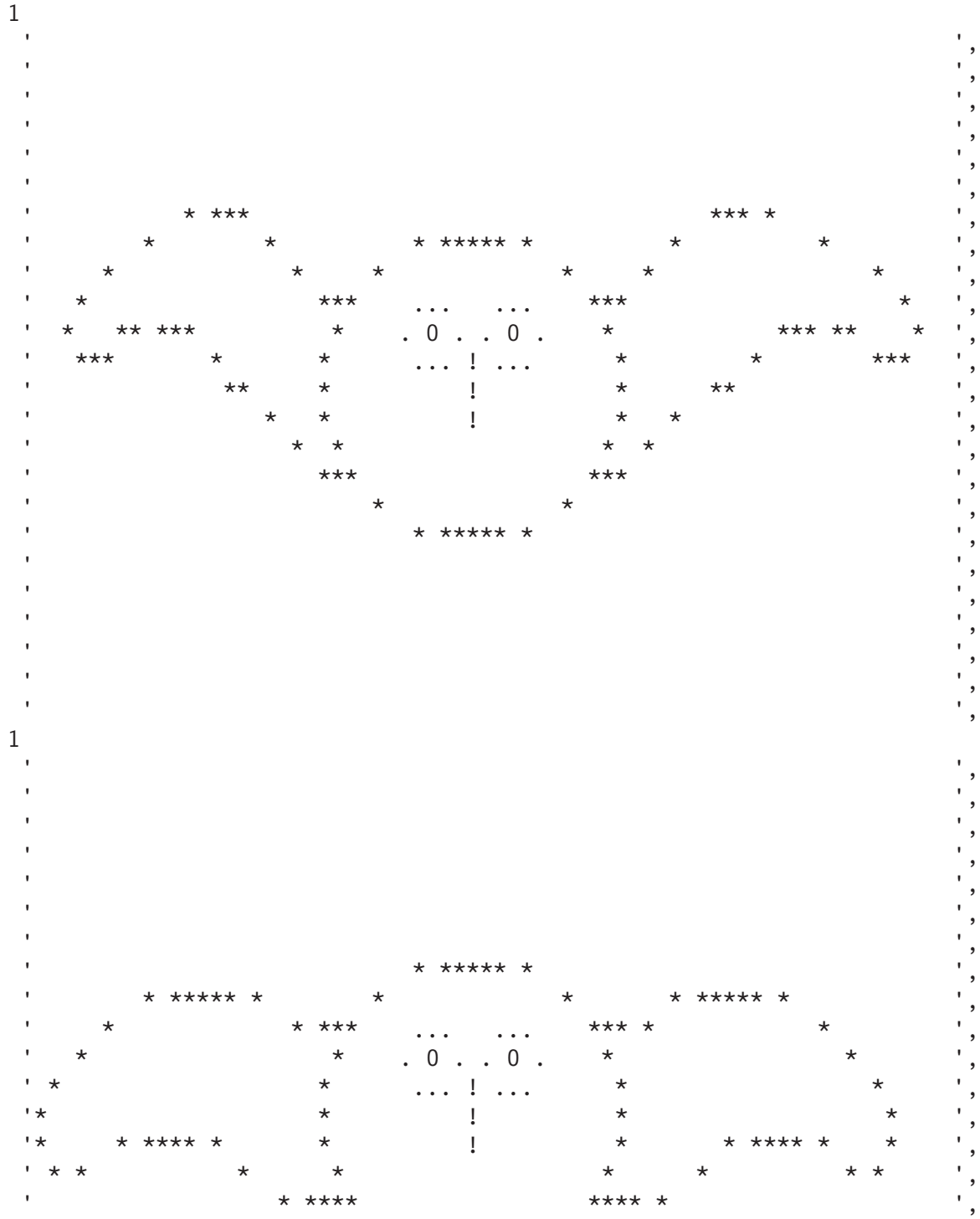
```
PARMIN   DS    ØX
TRANSID  DS    CL4
FILLER1  DS    CL1
TERMID   DS    CL4
****************************************
ERRORMSG DS    CL37
ENDTASK  DS    CL19
CSAERROR MVC   ERRORMSG,=CL37'ERROR IN ADDRESSING CSA, CALL SYSPROG'
         EXEC  CICS SEND FROM(ERRORMSG) ERASE
         B     RETURN
TERMERR  MVC   ERRORMSG,=CL37'ERROR IN ADDRESSING TERMINAL ID..    '
         EXEC  CICS SEND FROM(ERRORMSG) ERASE
         B     RETURN
*
RETURN   EXEC CICS RETURN
         LTORG
*
* ***************************************************************
*  DISPLAY CONTENTS OF R9 ****************
*
DISPLAY  ST    R9,ADDRVAL
         UNPK  MSGTEXT+2Ø(9),ADDRVAL(5)
         TR    MSGTEXT+2Ø(8),TRANSTAB
         MVC   MSGL,=H'28'
         SPACE 1
         MVC   MSGTEXT(4),TERMID
         MVC   MSGTEXT+4(16),MSGCONS
         EXEC  CICS SEND FROM(MSGTEXT) LENGTH(MSGL) ERASE
         SPACE 1
         B     RETURN
*
MSGL     DS H
MSGTEXT  DS CL4Ø
FILLER   DS ØF             FULLWORD ALLIGNMENT
ADDRVAL  DS CL5
MSGCONS  DC CL19' IS LOCATED AT: '
TRANSTAB  DS      ØF Ø 1 2 3 4 5 6 7 8 9 A B C D E F
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
```

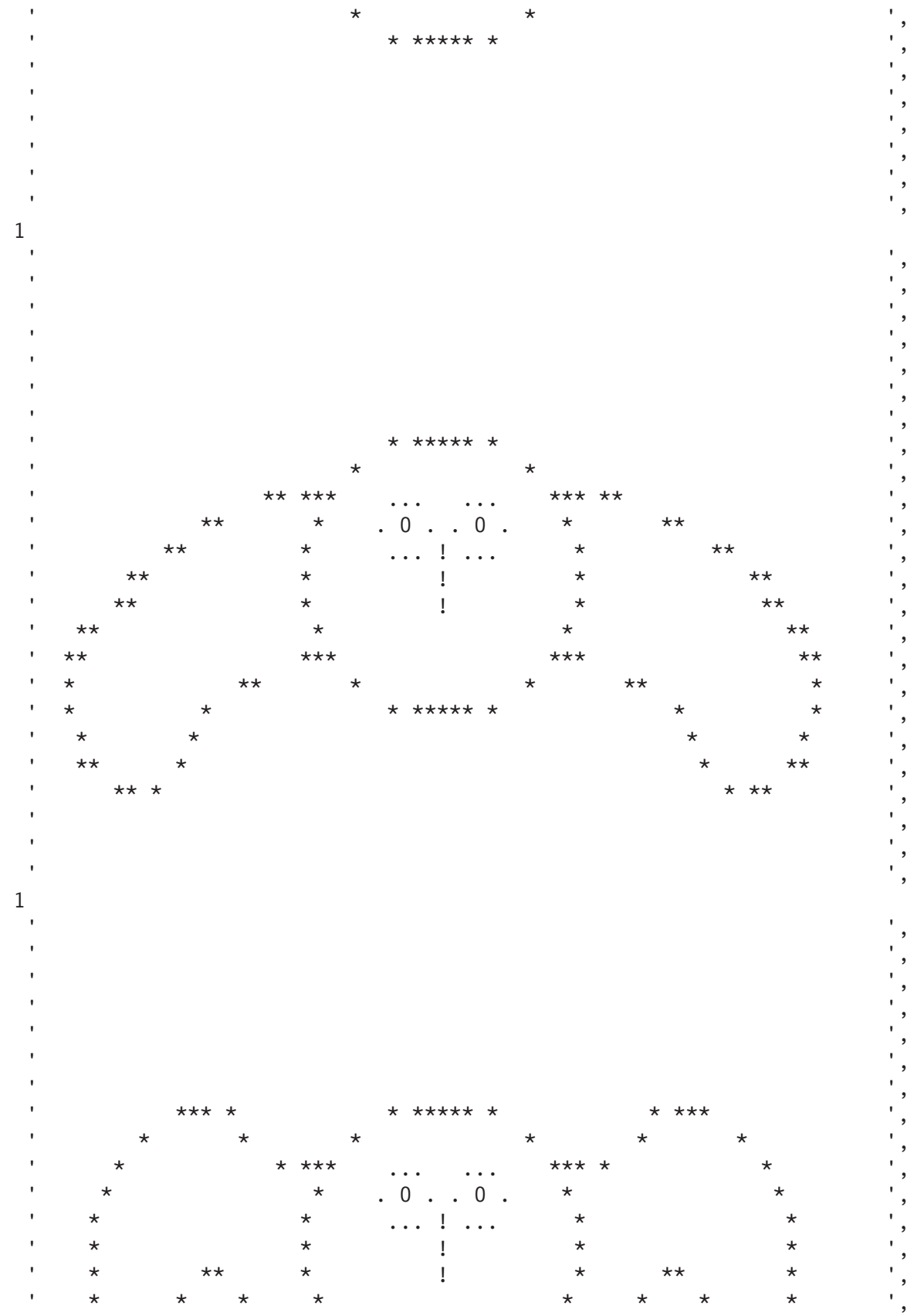
```
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'
          DC      X'F0F1F2F3F4F5F6F7F8F9C1C2C3C4C5C6'
*
*  ***********************************************
*  ************************************************************
*
*********************
*********************
***** BEGIN ZAP *****
*********************
*********************
*
ZAPTCT   DS   0H
*
         USING DFHCSADS,R9
         DFHTM TYPE=(GETFIRST,INLINE,(R5),(TCTTEAR)),                  *
               TABLE=TCTE,ERROR=TERMERR        GET FIRST TERMID
*
CHKTRMID DS    0H
*
         CLC   0(4,TCTTEAR),TERMID      CHECK TERMID
         BNE   NEXTTCTE                 NO, GET NEXT TCTTE
*
RESTERM  DS 0H
         MVI   461(TCTTEAR),B'00'    ZAP TCTTE OFFSETT X'1CD'
         MVI   463(TCTTEAR),B'00'    ZAP TCTTE OFFSETT X'1CF'
         LR    R9,TCTTEAR
         B     DISPLAY
*
NEXTTCTE DS    0H
*
         DFHTM TYPE=(GETNEXT,INLINE,(R5),(TCTTEAR)),                   *
               TABLE=TCTE,ERROR=TERMERR        GET NEXT TERMID
         B     CHKTRMID            GO CHECK THE TERMINAL.
*
*******************
*******************
***** END ZAP *****
*******************
*******************
         END
```

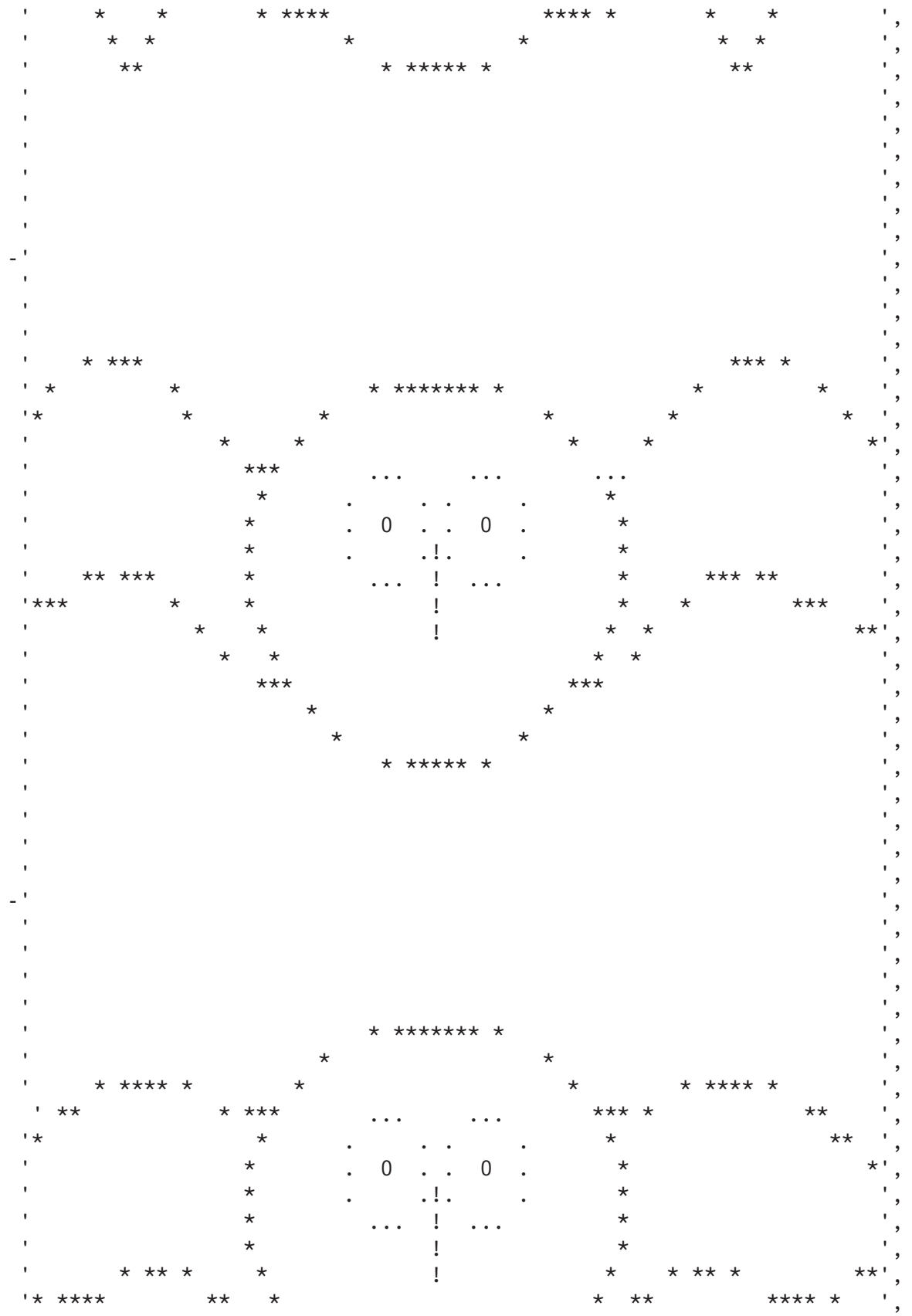
Erhan Pasa
Senior Systems Programmer
Akbank TAS (Turkey)

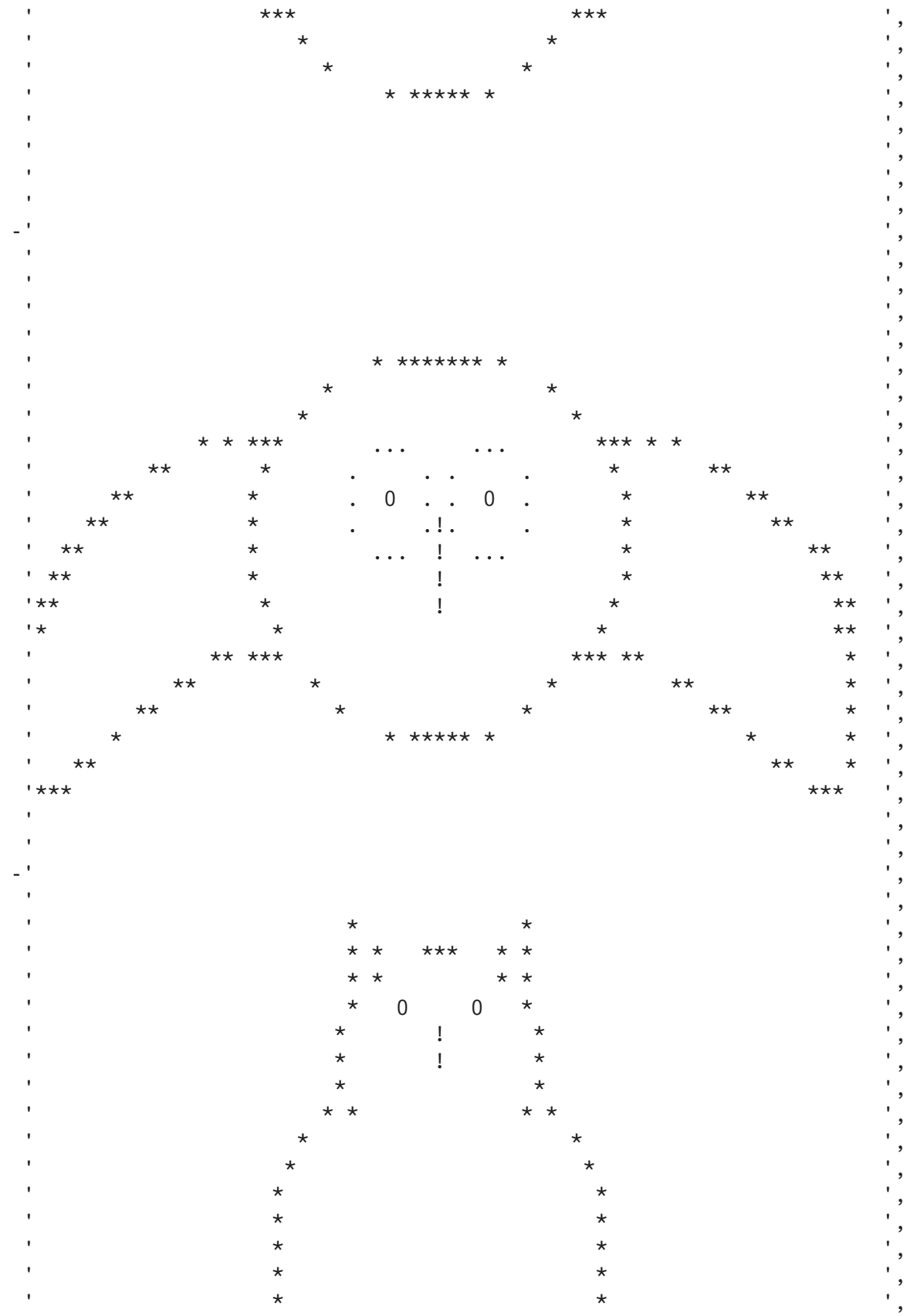
Fly to another planet – part 2

This month we conclude the code to create a more interesting CICS dialog and show that even CICS can be fun sometimes!









```
'                  *                        *                    ',
'                  *                        *                    ',
'                   *                      *                     ',
'                     *     **      **    *                       ',
'                          * ******** *                           ',
'                                                                 ',
'                                                                 ',
'                                                                ')
                                                                  ;
         EXEC CICS HANDLE CONDITION ENDDATA(START);
         EXEC CICS RETRIEVE INTO(PARAM) LENGTH (LNG);
         GOTO INIT;
         /* FIRST RUN */
 START:
         IF EIBCALEN = Ø THEN DO;
            SUBSTR(M1RED,1,STG(M1O)) = LOW(STG(M1O));
            INTO = 'ØØØØØØ';
            BSO  = EIBTRMID;
            IZEITO = EIBTIME;
            INT1O = 'ØØ';
            INT2O = 'ØØ';
            ANZO  = 'Ø2';
            ALARMO = 'N';
            EXEC CICS SEND MAP('M1') MAPSET ('CMFLIG') ERASE;
            EXEC CICS RETURN TRANSID ('FLIG') COMMAREA(MS) LENGTH(1);
         END;
         /* END FIRST RUN */
         /* SECOND RUN */
-        IF EIBAID = DFHCLEAR | EIBAID = DFHPA1 | EIBAID = DFHPA2
            THEN DO;
            EXEC CICS SEND MAP ('ML') MAPSET ('CMFLIG') MAPONLY ERASE;
            EXEC CICS RETURN;
         END;
      /*       */
-        SUBSTR(M1RED,1,STG(M1I)) = LOW(STG(M1I));
         BS    = EIBTRMID;
         EXEC CICS HANDLE CONDITION MAPFAIL(MAPFAIL);
         EXEC CICS RECEIVE MAP('M1') MAPSET ('CMFLIG');
-MAPFAIL:
         IF INTL ¬= Ø THEN DO;
            IF ZEITL ¬= Ø THEN CALL FEHLER(3,1);
            IF VERIFY(INTI,NUM) THEN CALL FEHLER(1,1);
            INT = INTIR;
            MS = '1';
         END;
-        IF ZEITL ¬= Ø THEN DO;
            IF VERIFY(ZEITI,NUM) THEN CALL FEHLER(1,2);
            INT = ZEITIR;
            MS = '2';
         END;
```

```
-         IF BSL ¬= Ø THEN BS = BSI;
-         IF INT1L ¬= Ø THEN DO;
             IF VERIFY(INT1I,NUM) THEN CALL FEHLER (1,3);
             INT1 = INT1IR;
          END;
-         IF INT2L ¬= Ø THEN DO;
             IF VERIFY(INT2I,NUM) THEN CALL FEHLER(1,4);
             INT2 = INT2IR;
          END;
-         IF ANZL ¬= Ø THEN DO;
             IF VERIFY(ANZI,NUM) THEN CALL FEHLER (1,5);
             ANZ = ANZIR;
          END;
-         IF ALARML ¬= Ø THEN DO;
             IF ALARMI ¬= 'N' &
                ALARMI ¬= 'J' THEN CALL FEHLER (2,6);
             ALARM = ALARMI;
          END;
-         IF T1L ¬= Ø THEN T1 = T1I;
-         IF T2L ¬= Ø THEN T2 = T2I;
         /*  STARTEN DER ZEIT-TASK */
-         EXEC CICS HANDLE CONDITION TERMIDERR (FEHL4);
          IF MS = '1' THEN EXEC CICS START TRANSID('FLIG')
                                          INTERVAL(INT)
                                          TERMID(BS)
                                     FROM (PARAM) LENGTH(LNG);
                      ELSE;
                           EXEC CICS START TRANSID('FLIG')
                                          TIME(INT)
                                          TERMID(BS)
                                     FROM (PARAM) LENGTH(LNG);
          SUBSTR(MLRED,1,STG(MLO)) = LOW(STG(MLO));
          EXEC CICS SEND MAP ('ML') MAPSET ('CMFLIG') DATAONLY
                                          CURSOR(EIBCPOSN);
          EXEC CICS RETURN TRANSID ('FLIG') COMMAREA(MS) LENGTH(1);
         /* ENDE SECOND RUN */
         /* ZEIT - TASK */
-INIT:
         ZL (*) = Ø;
         ZA (*) = LOW(1);
         ZO (*) = ' ';
          DO J = 1 TO ANZ;
             IF ALARM = 'N' THEN
                EXEC CICS SEND MAP ('M2') MAPSET ('CMFLIG') ERASE
                                          MAPONLY WAIT;
                           ELSE
                EXEC CICS SEND MAP ('M2') MAPSET ('CMFLIG') ERASE ALARM
                                          MAPONLY WAIT;
             EXEC CICS DELAY INTERVAL (INT1);
```

```
-           DO K = 1 TO 24;
            /* EULEN BILD */
               DO I = 1 TO 24;
                  SUBSTR(ZO(I),5,69) = MA (K,I);
               END;
               SUBSTR ( ZO(1),7,39) =
    'YOUR COMPUTER WISHES YOU A GOOD FLIGHT';
               EXEC CICS ASKTIME;
               ZWPIC = EIBTIME;
               SUBSTR(ZO(1),60,9) = ZWPIC;
               SUBSTR(ZO(1),70,03) = 'UHR';
               SUBSTR(ZO(23),28,20) = T1;
               SUBSTR(ZO(24),28,20) = T2;
               EXEC CICS SEND MAP ('M2') MAPSET ('CMFLIG') ERASE WAIT;
               EXEC CICS DELAY INTERVAL (INT1);
            END;
         EXEC CICS DELAY INTERVAL (INT2);
         END;
         EXEC CICS RETURN;
-FEHL4:  CALL FEHLER (4,7);
-FEHLER: PROC(K,L);
         SUBSTR(M1RED,1,STG(M1O)) = LOW(STG(M1O));
         FMO = FEHLTAB (K);
         IF L = 1 THEN INTL = -1;
         IF L = 2 THEN ZEITL = -1;
         IF L = 3 THEN INT1L = -1;
         IF L = 4 THEN INT2L = -1;
         IF L = 5 THEN ANZL  = -1;
         IF L = 6 THEN ALARML = -1;
         IF L = 7 THEN BSL    = -1;
         EXEC CICS SEND MAP ('M1') MAPSET('CMFLIG') DATAONLY CURSOR;
         EXEC CICS RETURN TRANSID ('FLIG') COMMAREA(MS) LENGTH(1);
         END;  /* FEHLER  */
         END;  /* EULE    */
```

Dieter Maier
CICS Systems Programmer (Germany)

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

The CICS Transaction Server for z/OS Version 3.1 Information Center – an update

The documentation for CICS Transaction Server for z/OS Version 3.1 has moved with the times, and is now presented using the latest Eclipse help system technology. Some common questions my CICS Information Development team here at Hursley get asked are:

- What's this new Eclipse Information Center?
- Is the Information Center on the Internet?
- How do I give feedback or suggestions?

In this article I'll address these questions and, with a few suitable screen captures, give you a flavour of the new world that awaits CICS Transaction Server for z/OS Version 3.1 Information Center users.

WHAT'S THIS NEW ECLIPSE INFORMATION CENTER?

A wealth of information is available at your fingertips, including documentation for CICS Transaction Server and all the CICS Tools, CICS Transaction Gateway, CICS Universal Client, and REXX for CICS.

We now have a much improved search engine.

There's a *What's new* section that mirrors the information included in the traditional hardcopy *Release Guide*.

There are some new ways of grouping information to add value for the reader such as *learning paths*, *information roadmaps*, and *troubleshooting and support* with technotes.

There are lots of informative links to information across the Internet. There are links to *tutorials and demos* that offer Quick Tours on SOAP for CICS and CICSPlex SM; there are more links to *service and support downloads* that include the

CICS support page, our popular SupportPac series, and fixes. Education gets a links section with *training and certification*, *online publications* (white papers, articles, and manuals), plus the campus (your source for IBM eServer and TotalStorage education). There are links to both the *IBM Redbooks* home page and the *CICS Redbooks* home page. *Third-party information* rounds out the links support information for CICS-related products (a site you should already have bookmarked!).

The meat of the CICS Transaction Server for z/OS Version 3.1 information is grouped in *task*, *concept*, and *reference* categorizations.

There is also a library and PDFs section for those who like producing hardcopy.

Figure 1 shows what the CICS Transaction Server for z/OS Version 3.1 Information Center home page looks like in its new

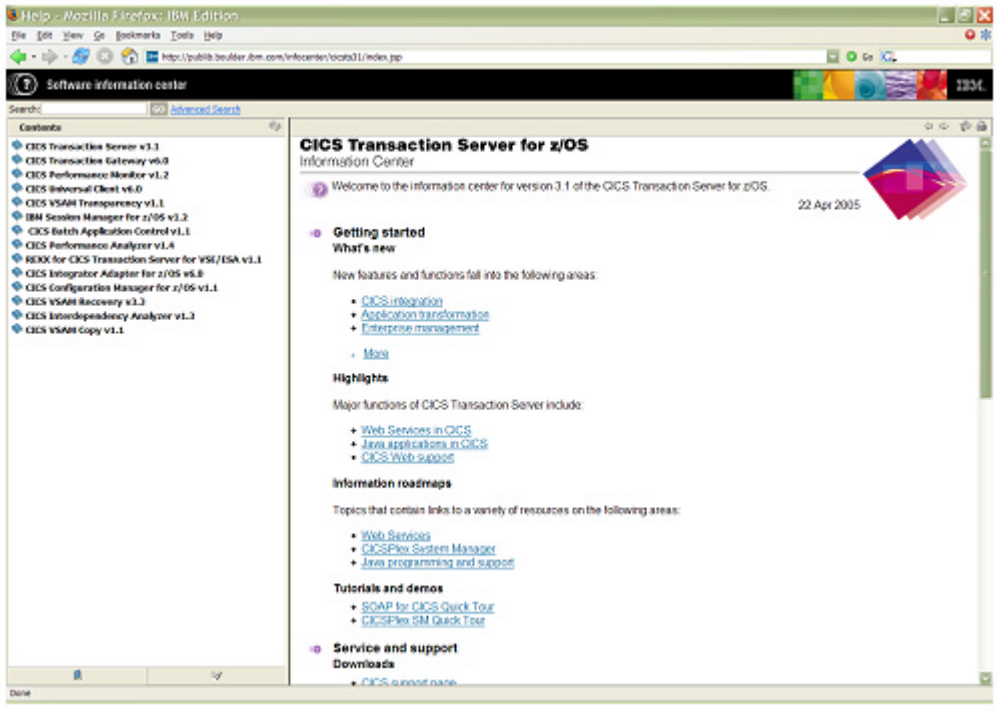


Figure 1: CICS TS 3.1 Information Center home page

IBM corporate livery.

A single click on one of the 14 plug-ins in the left-hand navigation pane will expand its contents.

The CICS Transaction Server for z/OS Version 3.1 *What's new* section includes everything you need to know about new functions in the release grouped by the capability themes of CICS Integration, Application transformation, and Enterprise management. There are also a few other changes that add

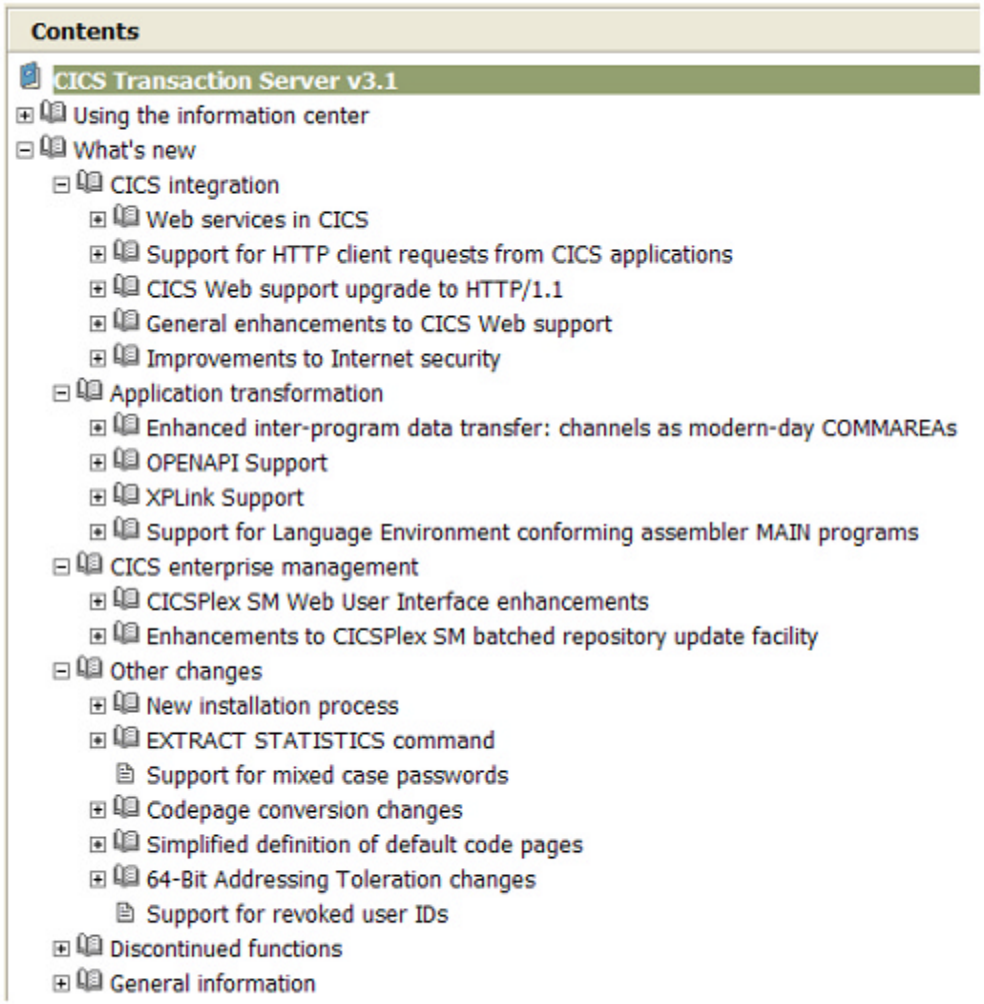


Figure 2: What's new *navigation pane*

value to the product. And finally there are the functions that are discontinued in this release, such as support for OS/VS COBOL, which has caused much activity on the forums!

Figure 2 shows what the *What's new* navigation looks like.

Learning paths have been created in the Information Center to help you learn about a functional area of CICS. Learning paths are a set of topics that should be read in sequence.

There are a couple of learning paths in the CICS Transaction Server for z/OS Version 3.1 Information Center – one for Channels (the modern-day alternative to traditional CICS COMMAREAs) and one for installing CICSPlex SM to use the WUI.

Each learning path contains a list of the steps in the path so that you can move to any topic in the path. Some topics also

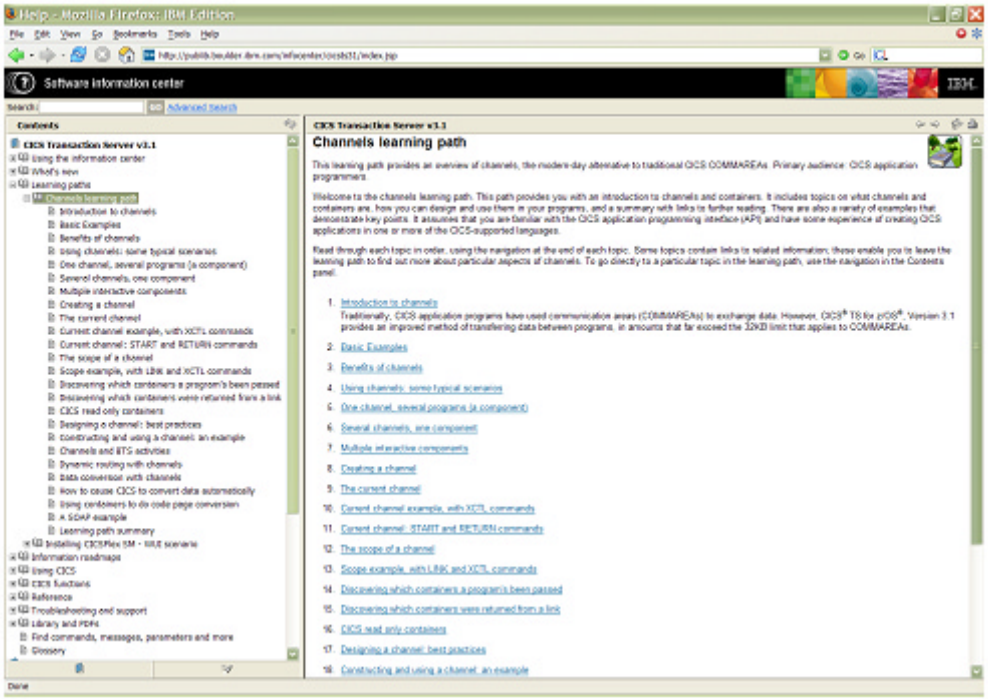


Figure 3: Channels learning path

contain links to more detailed information, so you can move off the path at any point. There is an icon (a book on a tree-lined road) in the top right-hand corner that shows when you are on the path. If the icon disappears when you click a link, you are off the path! The final topic in a learning path is a summary containing links to further reading.

Figure 3 shows what the Channels learning path looks like.

Information roadmaps provide a comprehensive set of links with guidance to information from a variety of sources. There

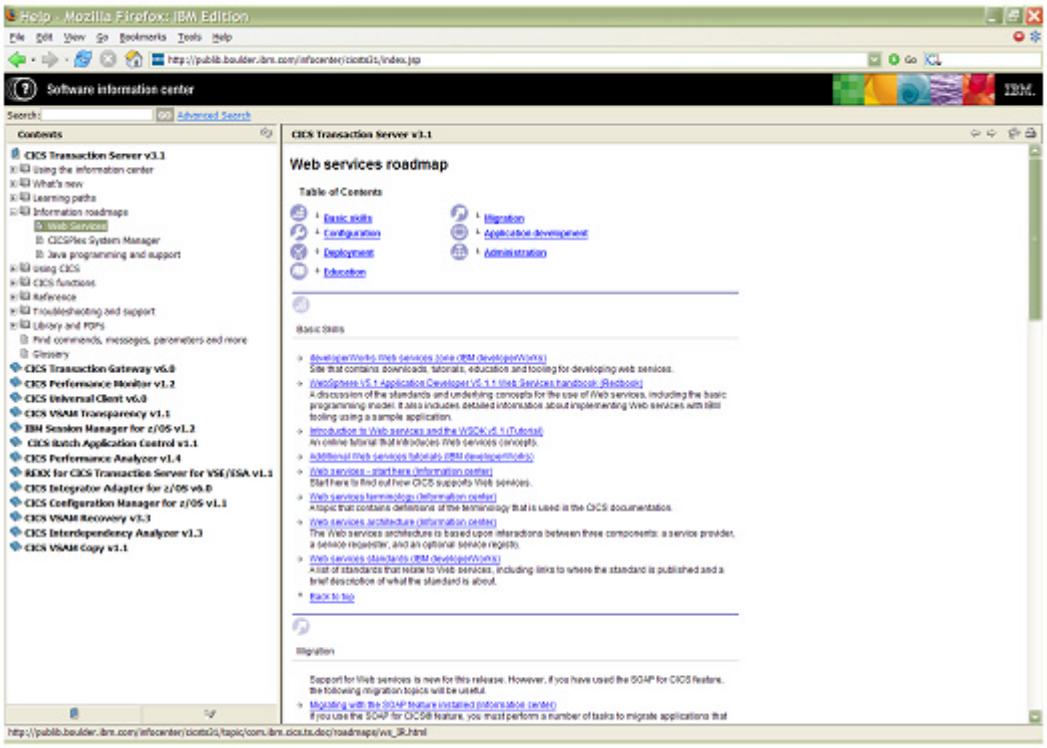


Figure 4: Web services information roadmap

are three information roadmaps in the CICS Transaction Server for z/OS Version 3.1 Information Center – Web services, CICSPlex System Manager, and Java programming and support. This style of information is available in many IBM

Information Centers across a whole range of products.

Figure 4 shows what the Web services information roadmap looks like.

The troubleshooting and support section is a self-help resource including instructions for searching knowledge bases, downloading fixes, and getting support from IBM. You might have heard this referred to as the e-support plug-in. For added convenience, this section includes copies of relevant technotes

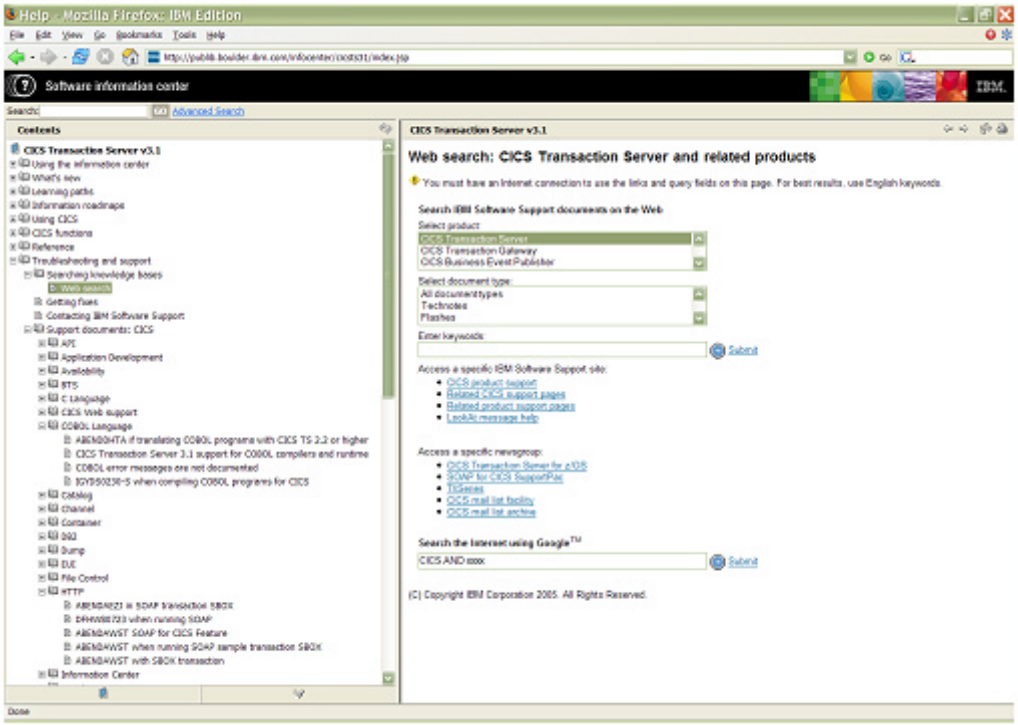


Figure 5: Troubleshooting and support section

and other support documents that have been published on the CICS Support Web site. These documents are included in any Information Center search you might undertake. If you have an Internet connection, you can also search live Web-based support resources by using the customized query fields in the

Web search topic.

Figure 5 shows what the troubleshooting and support section looks like.

Search gets an upgrade! An oft called for requirement was to highlight the search argument in the topics that were found, something the CICS Transaction Server for z/OS Version 2 Information Centers did not provide. The box in which to enter your search argument can be found at the top left of the header. Search results appear in the left-hand navigation

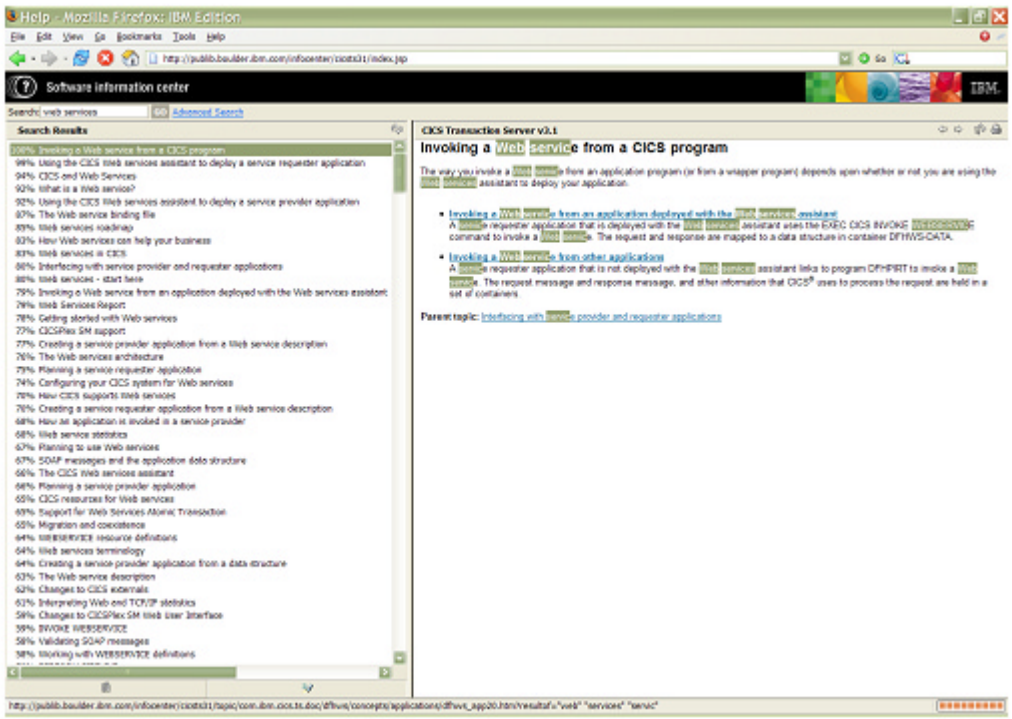


Figure 6: 100% hit for 'web services'

pane. You can toggle between the *Contents* and *Search* results by using, respectively, the 'book' and 'torch' icons in the footer. A word of warning: the default search will look at all the 14 plug-ins in the Information Center, so you could get

results back from CICS Transaction Gateway, CICS tools, etc. Should you wish to trim down the plug-ins searched, click **Advanced search** (to the right of the GO button) and deselect the plug-ins you wish to exclude.

Figure 6 shows what the 100% hit for 'web services' looks like.

The meat of the documentation can be found in the three sections – *Using CICS*, *CICS functions*, and *Reference*. The sections are based on the task/concept/reference categorization that began in earnest with the Version 2 Information Centers.

The *Using CICS* section contains: Planning, Installing, Migrating, Setting up the system, Designing applications, Developing applications, Using Java applications, Running applications, Customizing, Administering, The CICSplex, Securing, Improving performance, and Diagnosing problems.

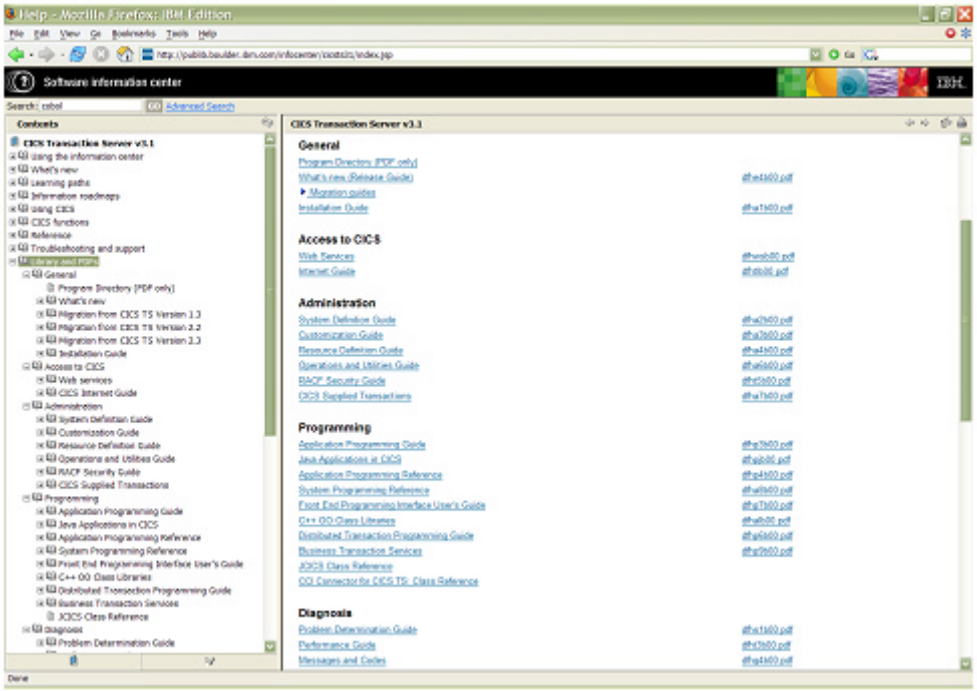


Figure 7: Library and PDFs *section*

The *CICS function* section contains: Application programming, Web services, Web interfaces, Java applications in CICS, Network services, Database services, External interfaces, Customization, System management, Performance and monitoring, Security, and Recovery and restart.

The *Reference* section contains: CICS-supplied transactions, Application programming interfaces, System definition, Resource definitions, System programming, Customization, CICS statistics, Trace, and CICS utilities.

And finally there is the *Library and PDFs* section. Here you can find HTML and PDF versions of the 'books' on the left and right of the main pane respectively. Select the book title for the HTML version, and the file name with the .pdf extension for the PDF.

Should you wish to download a PDF file, right-click on the link to the PDF (or tab to the link and press Shift-F10), and select **Save Target As** or **Save link as**.

Figure 7 shows how some of this section looks – only some because there are too many books to get them all on one screen capture.

Enjoy.

IS THE INFORMATION CENTER ON THE INTERNET?

Is the Information Center on the Internet? The answer is a big yes.

The CICS Transaction Server for z/OS Version 3.1 Information Center is at http://publib.boulder.ibm.com/infocenter/cicsts31/index.jsp.

And, for your convenience, the CICS Transaction Server for z/OS Version 2 Information Centers have been migrated to use the latest Eclipse technology.

You can find them at:

- http://publib.boulder.ibm.com/infocenter/cicsts23/index.jsp
- http://publib.boulder.ibm.com/infocenter/cicsts22/index.jsp.

The URLs will stay constant and the content will always be the latest refresh. The information at these sites is the unlicensed version (so no *Diagnosis Reference* or *Data Areas*). Unfortunately, there is no local bookmarks facility at these sites.

HOW DO I GIVE FEEDBACK OR SUGGESTIONS?

Whatever your thoughts on the CICS Information Center I welcome your comments. And yes, we do get praise from time to time on topics of information that readers think hit the spot. So don't hold back; if you think something is really great, let us know.

You can e-mail me (missenp@uk.ibm.com), talk to any of my team at the various conferences around the world, or use the feedback form link at the bottom of the Information Center welcome page.

Here's the link: http://publib.boulder.ibm.com/infocenter/cicsts31/index.jsp?topic=/com.ibm.cics.ts.doc/prod/feedback.html.

Peter Missen
Transaction User Technologies Manager
IBM Hursley (UK)

Presenting CICS DB2 resource statistics from SMF 110 records

CICS statistics are the simplest tool for monitoring CICS resources and overall CICS performance. The information

they contain can be used for performance tuning and capacity planning. Statistics are collected during CICS online processing in five ways – as interval, end-of-day, requested, requested reset, and unsolicited statistics. They are written to System Management Facility (SMF) dataset records of type 110, subtype 002, that can be later processed and analysed by some offline tools (utility DFHSTUP, user written programs, Tivoli Decision Support).

This article describes an application designed to present the subset of information collected by type 110, subtype 002 SMF records, that deal with CICS DB2 resource statistics. Some information about the same area of interest could also be gained by using the DSNC DISP STAT command, but not all. Although the DFHSTUP utility gives the same information, this application presents the data in a more descriptive form.

CICS statistics SMF type 110 records consist of three components – SMF header, SMF product section, and CICS data section. The SMF header describes the system that creates the output. The SMF product section identifies the subsystem to which the statistics relate; in this case; the CICS region. The SMF header and product section format is given in the SDFHMAC library in macro DFHSMFDS. The statistics data section consists of one or more statistics data records with different formats, except for the common format of the first five bytes. A description of the CICS DB2 resource statistics data section is given in macro DFHD2RDS in the SDFHMAC library.

My application consists of:

- SMFST67 – application start-up REXX EXEC.
- SMFST67P – main input panel.
- SMFST67R – main REXX EXEC.

The required parameters on the main input panel are CICS applid, CICS load module library, SMF dataset names, and the date frame (given in the form yyyy/mm/dd). As input, some

or all of the SYS1.MAN*x* datasets or cumulative SMF dataset could be given, depending on the period for which the report is needed. The main input panel is shown below:

```
--------------------- CICS DB2 RESOURCE STATISTICS --------------------



 CICS applid         ===> PSTEST29

 CICS load library   ===> CICSTS22.CICS.SDFHLOAD

 SMF data set 1      ===> IPOSAV.SMFDUMPW

 SMF data set 2      ===>

 SMF data set 3      ===>

         Date From:  ===> 2005/08/04

                To:  ===> 2005/08/04
```

The main REXX EXEC generates a job that consists in two steps. In the first step, records of type 110 for a specified period are extracted from the chosen SMF datasets and placed in a temporary dataset. This is then input to the DFHJUP utility. In the second step, records are further filtered to include only subtype 002 for the given CICS applid. An example of the JCL is given below:

```
//SYSADMX JOB MSGCLASS=X,REGION=4M,NOTIFY=SYSADM
//*************************************************************
//* STEP 1 - UNLOAD THE SMF DATA SET CONTAINING CICS DATA
//*************************************************************
//SMFDUMP EXEC PGM=IFASMFDP
//INDD1    DD  DSN=IPOSAV.SMFDUMPW,DISP=SHR,AMP=('BUFSP=65536')
//OUTDD1   DD DSN=&&TEMP,DISP=(NEW,PASS),
//         SPACE=(CYL,(50,1)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN    DD  *
       INDD(INDD1,OPTIONS(DUMP))
       OUTDD(OUTDD1,TYPE(110))
       DATE(2005215,2005215)
/*
//*************************************************************
//* STEP 2 - DFHJUP SELECT RECORDS
//*************************************************************
```

Resource statistics - resource information

DB2Entry name	Plan name	PlanExit name	Auth id	Auth type	Account records	Thread wait	Thread prty
ENAR01	ARHI0100	N/A	N/A	USERID	TXID	YES	N/A
ENAU01	AUTOMATP	N/A	N/A	USERID	TXID	YES	N/A
ENDB02	DB2TEST	N/A	N/A	USERID	TXID	YES	N/A
ENDB201	DSN8CP0	N/A	N/A	USERID	NONE	POOL	N/A
ENDB202	DSN8CQ0	N/A	N/A	USERID	TXID	YES	N/A

Figure 1a: Excerpt from a summary report

Resource statistics - request information

DB2Entry name	Call count	Signon count	Partial signon	Commit count	Abort count	Single phase	Thread reuse	Thread terms	Thread waits/overflows
ENARØ1	1254	4	1	Ø	Ø	433	Ø	433	Ø
ENAUØ1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
ENDNØ5	1Ø98	289	3	Ø	Ø	468	Ø	468	Ø
ENDNØ6	555	87	1	Ø	Ø	93	Ø	93	Ø
ENDNØ7	37	2	2	Ø	Ø	15	Ø	15	Ø
ENDNØ8	962	73	Ø	Ø	Ø	98	Ø	98	Ø
ENDNØ9	675	51	1	62	Ø	64	Ø	126	Ø
ENDOØ1	76Ø3	13	11	Ø	Ø	24	Ø	24	Ø
ENTR58	1Ø36	258	75	86	Ø	3Ø3	Ø	389	Ø

Figure 1b: Excerpt from a summary report

Resource statistics - performance information

DB2Entry name	Thread limit	Thread current	Thread HWM	Pthread limit	Pthread current	Pthread HWM	Task current	Task HWM	Task total	Readyq current	Readyq HWM
ENARØ1	1	Ø	1	Ø	Ø	Ø	Ø	1	433	Ø	Ø
ENAUØ1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
ENBA23	3	Ø	1	Ø	Ø	Ø	Ø	1	258	Ø	Ø
ENBA24	3	Ø	1	Ø	Ø	Ø	Ø	1	26	Ø	Ø
ENBA25	3	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
ENDNØ4	1	Ø	1	Ø	Ø	Ø	Ø	1	13	Ø	Ø
ENDNØ5	3	Ø	2	Ø	Ø	Ø	Ø	2	468	Ø	Ø
ENIN11	1Ø	Ø	4	Ø	Ø	Ø	Ø	4	11193	Ø	Ø
ENJPØ1	1	Ø	1	Ø	Ø	Ø	Ø	1	11Ø	Ø	Ø
ENKAØ1	1	Ø	1	Ø	Ø	Ø	Ø	1	517	Ø	Ø
ENKAØ2	1	Ø	1	Ø	Ø	Ø	Ø	1	1Ø3	Ø	Ø

Figure 1c: Excerpt from a summary report

```
//RECSEL EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICSTS22.CICS.SDFHLOAD,DISP=SHR
//SYSUT1  DD DSN=&&TEMP,DISP=(OLD,PASS)
//SYSPRINT DD DSNAME=SYSADM.SMF110.PRIV,DISP=(NEW,CATLG),
//          SPACE=(CYL,(200,20),RLSE),UNIT=SYSDA,
//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=27930)
//SYSIN DD *
OPTION PRINT OFFSET=23,FLDTYP=X,VALUE=0002,FLDLEN=2,COND=M
OPTION PRINT OFFSET=47,FLDTYP=C,VALUE=PSTEST29,FLDLEN=8,COND=E
END
/*
```

After submitting the generated job, wait for notification of its completion, and then press the PF3 key, the SMFST67R REXX EXEC continues to format output from the DFHJUP utility. The generated summary report for CICS DB2 resource statistics comprises three sections – resource, request, and performance information. The resource information gives details of attribute settings for each DB2ENTRY. The request information reports how many and which types of request have been performed against each DB2ENTRY. The performance information gives detailed thread statistics for each DB2ENTRY. Excerpts from summary report are shown in Figures 1a, b and c.

If the message from SMFST67R REXX EXEC is 'No records found' when SYS1.MAN*x* datasets are used, the CEMT PERFORM STATISTICS DB2 command can be executed to initiate the immediate writing of CICS DB2 statistics to the SMF datasets, rather then wait for the current statistics-gathering interval to expire.

SMFST67

```
/* rexx */
Address ISPEXEC 'select panel(SMFST67P)'
```

SMFST67P

```
)ATTR
%  TYPE(TEXT)
[  TYPE(TEXT) INTENS(LOW)
<  TYPE(INPUT) CAPS(ON)
```

```
+  TYPE(TEXT) INTENS(LOW)
!  TYPE(OUTPUT) INTENS(LOW) CAPS(OFF)
)BODY DEFAULT(]*;)EXPAND($$)
%-$-$- CICS DB2 RESOURCE STATISTICS -$-$-[

                                                                        [
                                                                        [

+CICS applid        ===><Z       [                                      [

+CICS load library  ===><Z                                              [

+SMF data set 1     ===><Z                                              [

+SMF data set 2     ===><Z                                              [

+SMF data set 3     ===><Z                                              [

       +Date From:  ===><Z         [                                    [

              +To:  ===><Z         [                                    [



<Z[

)INIT
  .ZVARS = '(APPLID CICSload MAN1 MAN2 MAN3 DateFrom DateTo ZCMD)'
  .CURSOR = APPLID
  VGET (APPLID CICSload MAN1 MAN2 MAN3 DateFrom DateTo) SHARED
  &APPLID = PSTEST29
  &CICSload = CICSTS22.CICS.SDFHLOAD
  &MAN1 = SYS1.MAN1
  &MAN2 = SYS1.MAN2
  &MAN3 = SYS1.MAN3
  &DateFrom = &ZDATESTD
  &DateTo = &ZDATESTD
)PROC
  VER (&APPLID,NONBLANK)
  VER (&CICSload,NONBLANK)
  VER (&MAN1,NONBLANK)
  VER (&DateFrom,PICT,'9999/99/99')
  VER (&DateTo,PICT,'9999/99/99')
  VPUT (APPLID CICSload MAN1 MAN2 MAN3 DateFrom DateTo) SHARED
  &ZSEL = TRANS(TRUNC(&ZCMD,'.')
               ' ','CMD(SMFST67R)'
               '*','?')
)END
```

SMFST67R

```
/* REXX                                                              */
/*  SMFST67R: INTERPRET DB2 ENTRY STATISTICS from SMF 11Ø RECORDS */
/* ***************************************************************** */
Address ISPEXEC "CONTROL ERRORS RETURN "
Address ISPEXEC ,
  "VGET (APPLID CICSload MAN1 MAN2 MAN3 DateFrom DateTo) SHARED"
NUM_OF_DAYS.1 = 31
NUM_OF_DAYS.3 = 31
NUM_OF_DAYS.4 = 3Ø
NUM_OF_DAYS.5 = 31
NUM_OF_DAYS.6 = 3Ø
NUM_OF_DAYS.7 = 31
NUM_OF_DAYS.8 = 31
NUM_OF_DAYS.9 = 3Ø
NUM_OF_DAYS.1Ø = 31
NUM_OF_DAYS.11 = 3Ø
NUM_OF_DAYS.12 = 31
yyyy = Substr(DateFrom, 1, 4)
mm   = Substr(DateFrom, 6, 2)
dd   = Substr(DateFrom, 9, 2)
If yyyy // 4ØØ <> Ø & yyyy // 4 = Ø Then /* leap year? */
  NUM_OF_DAYS.2 = 29
Else
  NUM_OF_DAYS.2 = 28
ddd = Ø
Do i = 1 To mm - 1
  ddd = ddd + NuM_OF_DAYS.I
End
ddd = ddd + dd
If ddd < 1ØØ Then yyyydddfrom = yyyy || "Ø" || ddd
Else yyyydddfrom = yyyy || ddd
yyyy = Substr(DateTo, 1, 4)
mm   = Substr(DateTo, 6, 2)
dd   = Substr(DateTo, 9, 2)
If yyyy // 4ØØ <> Ø & yyyy // 4 = Ø Then
  NUM_OF_DAYS.2 = 29
Else
  NUM_OF_DAYS.2 = 28
ddd = Ø
Do i = 1 To mm - 1
  ddd = ddd + NuM_OF_DAYS.I
End
ddd = ddd + dd
If ddd < 1ØØ Then yyyydddto = yyyy || "Ø" || ddd
Else yyyydddto = yyyy || ddd
count = Ø
NUMERIC DIGITS 5Ø
```

```
userid   = userid()
tick     = ''''
jcldsn   = tick||userid||".SMFUTIL.PRIV"||tick
If sysdsn(jcldsn) = 'OK' Then
  Address TSO "alloc fi(jcl) da("jcldsn") shr "
Else
  Address TSO "alloc fi(jcl) da("jcldsn") new ",
              "  dsorg(ps) space(1,1) tracks",
              "  lrecl(8Ø) blksize(2792Ø) recfm(f b)"
queue "//"||userid||"X JOB MSGCLASS=X,REGION=4M,NOTIFY="||userid
queue  "//*******************************************************"
queue "//* STEP 1 - UNLOAD THE SMF DATASET CONTAINING CICS DATA"
queue  "//*******************************************************"
queue "//SMFDUMP EXEC PGM=IFASMFDP"
queue "//INDD1    DD  DSN="||MAN1||",DISP=SHR,AMP=('BUFSP=65536')"
Address TSO "execio 6 diskw jcl"
If MAN2 ¬= "" Then Do
  queue "//INDD2    DD  DSN="||MAN2||",DISP=SHR,AMP=('BUFSP=65536')"
  Address TSO "execio 1 diskw jcl"
End
If MAN3 ¬= "" Then Do
  queue "//INDD3    DD  DSN="||MAN3||",DISP=SHR,AMP=('BUFSP=65536')"
  Address TSO "execio 1 diskw jcl"
End
queue "//OUTDD1   DD DSN=&&TEMP,DISP=(NEW,PASS),"
queue "//          SPACE=(CYL,(5Ø,1)),UNIT=SYSDA"
queue "//SYSPRINT DD SYSOUT=*"
queue "//SYSIN    DD  *"
queue "       INDD(INDD1,OPTIONS(DUMP))"
Address TSO "execio 5 diskw jcl"
If MAN2 ¬= "" Then Do
  queue "       INDD(INDD2,OPTIONS(DUMP))"
  Address TSO "execio 1 diskw jcl"
End
If MAN3 ¬= "" Then Do
  queue "       INDD(INDD3,OPTIONS(DUMP))"
  Address TSO "execio 1 diskw jcl"
End
queue "       OUTDD(OUTDD1,TYPE(11Ø))"
queue "       DATE("||yyyydddfrom||","||yyyydddto||")"
queue "/*"
Address TSO "execio 3 diskw jcl"
queue  "//*******************************************************"
queue "//* STEP 2 - DFHJUP SELECT RECORDS"
queue  "//*******************************************************"
queue "//RECSEL EXEC PGM=DFHJUP"
queue "//STEPLIB DD DSN="||CICSload||",DISP=SHR"
queue "//SYSUT1  DD DSN=&&TEMP,DISP=(OLD,PASS)"
queue "//SYSPRINT DD DSNAME="||userid||".SMF11Ø.PRIV,DISP=(NEW,CATLG),"
```

```
queue "//          SPACE=(CYL,(200,20),RLSE),UNIT=SYSDA,"
queue "//          DCB=(RECFM=FBA,LRECL=133,BLKSIZE=27930)"
queue "//SYSIN DD *"
queue "OPTION PRINT OFFSET=23,FLDTYP=X,VALUE=0002,FLDLEN=2,COND=M"
queue "OPTION PRINT OFFSET=47,FLDTYP=C,VALUE="||APPLID||,
      ",FLDLEN=8,COND=E"
queue "END"
queue "/*"
queue "//"
Address TSO "execio 15 diskw jcl"
Address TSO "execio 0 diskw jcl(finis"
Address TSO "free fi(jcl)"
Address ISPEXEC "Edit Dataset("jcldsn")"
Address TSO "Delete "jcldsn
indsn   = tick||userid||".SMF110.PRIV"||tick
outdsn   = tick||userid||'.SMF67.PRIV'||tick
If sysdsn(outdsn) = 'OK' Then
  Address TSO "alloc fi(indd2) da("outdsn") shr "
Else
  Address TSO "alloc fi(indd2) da("outdsn") new ",
              "  dsorg(ps) space(6000,1000) tracks release",
              "  recfm(F B) lrecl(312) blksize(27768)"
Address TSO "alloc fi(indd1) da("indsn") shr"
Address TSO "delstack"
next:
Do Forever
  Address TSO "execio 1 diskr indd1"
  If rc = 2 Then Leave
  Pull inrec
  If Index(inrec, 'DFHJC4531') > 0 Then Leave
  If Substr(inrec, 3, 6) = "000000" Then Do
    lrec = X2D(Substr(inrec, 12, 4))
    Address TSO "execio 1 diskr indd1"
    If rc = 2 Then Leave
    Pull inrec
    If Index(inrec, 'DFHJUP - JOURNAL PRINT') > 0 Then Do
      Address TSO "execio 1 diskr indd1"
      If rc = 2 Then Leave
      Pull inrec
    End
    offd = X2D(Substr(inrec, 25, 4))
    ld = X2D(Substr(inrec, 31, 4))
    ldpom = 0
    srec = (offd % 32) - 1
    rrec = (offd // 32) * 2 + 1
    Do i = 1 To srec
      Address TSO "execio 1 diskr indd1"
      If rc = 2 Then Leave
      Pull inrec
```

```
    If Index(inrec, 'DFHJUP - JOURNAL PRINT') > Ø Then Do
      Address TSO "execio 1 diskr indd1"
      If rc = 2 Then Leave
      Pull inrec
    End
  End
  inrec1 = Space(Substr(inrec, 12, 75),Ø)
  inrec1 = Substr(inrec1, rrec)
Strt:
  inrec2 = inrec1
  ld1 = X2D(Substr(inrec1, 1, 4))
  If ld1 = Ø Then Signal next
  l = Length(inrec1)
  Do While (l < ld1 * 2)
    Address TSO "execio 1 diskr indd1"
    If rc = 2 Then Leave
    Pull inrec
    If Index(inrec, 'DFHJUP - JOURNAL PRINT') > Ø Then Do
      Address TSO "execio 1 diskr indd1"
      If rc = 2 Then Leave
      Pull inrec
    End
    If (l + 64) > ld1 * 2 Then Do
      inrec1 = Space(Substr(inrec, 12, 75),Ø)
      inrec2 = inrec2 || Substr(inrec1, 1, ld1 * 2 - l)
      inrec1 = Substr(inrec1, ld1 * 2 - l + 1)
      l = ld1 * 2
    End
    Else Do
      inrec1 = Space(Substr(inrec, 12, 75),Ø)
      inrec2 = inrec2 || inrec1
      inrec1 = ""
      l = l + 64
    End
  End
  If Substr(inrec2, 5, 4) = "ØØ67" Then Do
    count = count + 1
    Queue inrec2
    Address TSO "Execio 1 Diskw indd2"
  End
  ldpom = ldpom + ld1
  If ldpom < ld Then Do
    If inrec1 = "" Then Do
      Address TSO "execio 1 diskr indd1"
      If rc = 2 Then Leave
      Pull inrec
      If Index(inrec, 'DFHJUP - JOURNAL PRINT') > Ø Then Do
        Address TSO "execio 1 diskr indd1"
        If rc = 2 Then Leave
```

```
          Pull inrec
        End
        inrec1 = Space(Substr(inrec, 12, 75),Ø)
      End
      Signal Strt
    End
    Else Signal next
  End   /* end if */
End   /* do forever */
Address TSO "execio Ø diskr indd1(finis"
Address TSO "Execio Ø Diskw indd2(finis"
Address TSO "free fi(indd1)"
Address TSO "free fi(indd2)"
If count = Ø Then Do
  Say "No records found"
  Address TSO "Delete "indsn
  Address TSO "Delete "outdsn
  Exit 4
End
Address TSO "alloc fi(indd2) da("outdsn") shr"
outdsn1 = tick||userid||'.SMF67OUT.PRIV'||tick
If sysdsn(outdsn1) = 'OK' Then
  Address TSO "alloc fi(outdd1) da("outdsn1") shr "
Else
  Address TSO "alloc fi(outdd1) da("outdsn1") new ",
              "  dsorg(ps) space(1Ø,1) tracks release",
              "  recfm(F B) lrecl(132) blksize(27984)"
Address TSO "delstack"
resourceinfo.Ø = Ø
requestinfo.Ø = Ø
performanceinfo.Ø = Ø
Do Forever
  Address TSO "execio 1 diskr indd2"
  If rc = 2 Then Leave
  Pull inrec
  Parse var inrec,
      1 D2RLEN 5 ,
      5 D2RID 9 ,
      9 D2RDVERS 11 ,
     17 D2R_DB2ENTRY_NAME 33 ,
     33 D2R_PLAN_NAME 49 ,
     49 D2R_PLANEXIT_NAME 65 ,
     65 D2R_AUTHID 81 ,
     81 D2R_AUTHTYPE 83 ,
     83 D2R_ACCOUNTREC 85 ,
     85 D2R_THREADWAIT 87 ,
     87 D2R_PRIORITY 89 ,
     89 D2R_CALLS 97 ,
     97 D2R_SIGNONS 1Ø5 ,
```

```
  105 D2R_COMMITS 113 ,
  113 D2R_ABORTS 121 ,
  121 D2R_SINGLE_PHASE 129 ,
  129 D2R_THREAD_REUSE 137 ,
  137 D2R_THREAD_TERM 145 ,
  145 D2R_THREAD_WAIT_OR_OVERFL 153 ,
  153 D2R_THREAD_LIMIT 161 ,
  161 D2R_THREAD_CURRENT 169 ,
  169 D2R_THREAD_HWM 177 ,
  177 D2R_PTHREAD_LIMIT 185 ,
  185 D2R_PTHREAD_CURRENT 193 ,
  193 D2R_PTHREAD_HWM 201 ,
  201 D2R_TASK_CURRENT 209 ,
  209 D2R_TASK_HWM 217 ,
  217 D2R_TASK_TOTAL 225 ,
  225 D2R_READYQ_CURRENT 233 ,
  233 D2R_READYQ_HWM 241 ,
  241 D2R_PARTIAL_SIGNONS 249
L1 = resourceinfo.0 + 1
resourceinfo.0 = L1
L2 = requestinfo.0 + 1
requestinfo.0 = L2
L3 = performanceinfo.0 + 1
performanceinfo.0 = L3
resourceinfo.L1 = X2C(D2R_DB2ENTRY_NAME)
requestinfo.L2 = X2C(D2R_DB2ENTRY_NAME)
performanceinfo.L3 = X2C(D2R_DB2ENTRY_NAME)
L1 = resourceinfo.0 + 1
resourceinfo.0 = L1
If D2R_PLAN_NAME = "0000000000000000" Then resourceinfo.L1 = "N/A"
Else resourceinfo.L1 = X2C(D2R_PLAN_NAME)
L1 = resourceinfo.0 + 1
resourceinfo.0 = L1
If D2R_PLANEXIT_NAME = "0000000000000000" Then resourceinfo.L1 = "N/A"
Else resourceinfo.L1 = X2C(D2R_PLANEXIT_NAME)
L1 = resourceinfo.0 + 1
resourceinfo.0 = L1
If D2R_AUTHID = "0000000000000000" Then resourceinfo.L1 = "N/A"
Else resourceinfo.L1 = X2C(D2R_AUTHID)
L1 = resourceinfo.0 + 1
resourceinfo.0 = L1
Select
  When X2D(D2R_AUTHTYPE) = 1 Then Do
    resourceinfo.L1 = "USERID"
  End
  When X2D(D2R_AUTHTYPE) = 2 Then Do
    resourceinfo.L1 = "OPID"
  End
  When X2D(D2R_AUTHTYPE) = 3 Then Do
```

```
    resourceinfo.L1 = "GROUP"
  End
  When X2D(D2R_AUTHTYPE) = 4 Then Do
    resourceinfo.L1 = "SIGNID"
  End
  When X2D(D2R_AUTHTYPE) = 5 Then Do
    resourceinfo.L1 = "TERM"
  End
  When X2D(D2R_AUTHTYPE) = 6 Then Do
    resourceinfo.L1 = "TXID"
  End
  Otherwise Do
    resourceinfo.L1 = "N/A"
  End
End
L1 = resourceinfo.Ø + 1
resourceinfo.Ø = L1
Select
  When X2D(D2R_ACCOUNTREC) = 1 Then Do
    resourceinfo.L1 = "NONE"
  End
  When X2D(D2R_ACCOUNTREC) = 2 Then Do
    resourceinfo.L1 = "TXID"
  End
  When X2D(D2R_ACCOUNTREC) = 3 Then Do
    resourceinfo.L1 = "TASK"
  End
  Otherwise Do
    resourceinfo.L1 = "UOW"
  End
End
L1 = resourceinfo.Ø + 1
resourceinfo.Ø = L1
Select
  When X2D(D2R_THREADWAIT) = 1 Then Do
    resourceinfo.L1 = "YES"
  End
  When X2D(D2R_THREADWAIT) = 2 Then Do
    resourceinfo.L1 = "NO"
  End
  Otherwise Do
    resourceinfo.L1 = "POOL"
  End
End
L1 = resourceinfo.Ø + 1
resourceinfo.Ø = L1
Select
  When X2D(D2R_PRIORITY) = 1 Then Do
    resourceinfo.L1 = "HIGH"
  End
```

```
    When X2D(D2R_PRIORITY) = 2 Then Do
      resourceinfo.L1 = "EQUAL"
    End
    When X2D(D2R_PRIORITY) = 3 Then Do
      resourceinfo.L1 = "LOW"
    End
    Otherwise Do
      resourceinfo.L1 = "N/A"
    End
  End
  L2 = requestinfo.0 + 1
  requestinfo.0 = L2
  requestinfo.L2 = X2D(D2R_CALLS)
  L2 = requestinfo.0 + 1
  requestinfo.0 = L2
  requestinfo.L2 = X2D(D2R_SIGNONS)
  L2 = requestinfo.0 + 1
  requestinfo.0 = L2
  requestinfo.L2 = X2D(D2R_PARTIAL_SIGNONS)
```

Editor's note: this article will be concluded next month.

Nikola Lazovic, Gordana Kozovic
DB2 System Administrators
Postal Savings Bank (Serbia and Montenegro)

Abend management

Here's a DFHPEP that will capture the 3270 screen buffer for the abending task and write it to Transient Data along with the basic task information, termid, tranid, program, userid, and username with a time stamp.

The program is derived from IBM's DFHPEP sample, but DFHPEP is just a generic shell program, so I removed any comment material remaining from that sample.

The AIDTABLE is based on IBM's DFHAID macro, but it is significantly modified from that.

The Help Desk finds the program useful for clarifying user

actions leading to an abend, and it provides valuable information when you don't have a comprehensive abend management product.

The application attempts to determine whether the CICS region is under stress or at max task, and bypasses retrieving the terminal buffer in these cases.

CODE

```
*ASM XOPTS(NOEPILOG)
DFHPEP   AMODE 31
DFHPEP   RMODE ANY
DFHPEP   DFHEIENT CODEREG=11,DATAREG=1Ø,EIBREG=9
         DFHREGS ,                 EQUATE REGISTERS
         XR    R1,R1
         ICM   R1,B'ØØ11',EIBCALEN GET COMMAREA LENGTH
         BZ    RETURNX             ...NO COMMAREA; EXIT
         EXEC CICS ADDRESS COMMAREA(R8)
         USING DFHPEP_COMMAREA,R8
*
         EXEC CICS HANDLE ABEND LABEL(RETURN)
         MVC  IEHIBALL,=C'IEHIBALL'
         EXEC CICS INQUIRE SYSTEM CICSSTATUS(CICSOK) SOSSTATUS(SOS)    $
                MAXTASKS(MAXT)
         CLC  CICSOK,DFHVALUE(ACTIVE)    IS CICS COMING UP/DOWN?
         BNE  RETURN                     YES LEAVE
* IF WE ARE AT SHORT ON STORAGE LEAVE
         CLC  SOS,DFHVALUE(NOTSOS)
         BNE  RETURN                     DONT COMPLICATE THE MESS
* IF WE ARE AT MAXT TASKS LEAVE
         EXEC  CICS INQUIRE TASK LIST LISTSIZE(NOTASKS)
         CLC  NOTASKS,MAXT               ARE WE AT MAXTASKS
         BNL  RETURN                     TOO BUSY FOR DETAILS
*DISPLAY HEADER
         MVC   OUTLINE,=8ØC'>'
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
         CLC   RESP,DFHRESP(NORMAL)
         BNE   RETURN
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (ABNDHDR) RESP(RESP)
         CLC   RESP,DFHRESP(NORMAL)
         BNE   RETURN
         MVC   OUTLINE,=8ØC' '
         EXEC CICS ASKTIME ABSTIME(ABST)
         EXEC CICS FORMATTIME ABSTIME(ABST)                            $
                   TIMESEP(':') TIME(ABNDTIME)                         $
                   DATESEP('/') DATE(ABNDDATE)
```

```
         MVC   ABCODE,PEP_COM_CURRENT_ABEND_CODE
         UNPK  ABTASK,EIBTASKN
         OI    ABTASK+4,X'FØ'
         MVC   ABTERM,EIBTRMID
         MVC   ABTRAN,EIBTRNID
         MVC   ABPROG,PEP_COM_ABPROGRAM
         EXEC CICS ASSIGN STARTCODE(STARTCODE) DS3270(TERM3270)        $
              USERNAME(ABNDNAME) USERID(ABNDUSER)                      $
              RESP(RESP)
         CLC  RESP,DFHRESP(NORMAL)          PROBABLY NO USER
         BNE  WRITEMSG                      WRITE THE LINE AND RETURN
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
         CLC   RESP,DFHRESP(NORMAL)
         BNE   RETURN
         MVC   OUTLINE,=80C'_'
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
         CLC   RESP,DFHRESP(NORMAL)
         BNE   RETURN
         CLI  TERM3270,X'FF'             ONLY 3270 DEVICES
         BNE  RETURN
         CLC =C'TD',STARTCODE            WE WANT ABENDS
         BNE  RETURN                     AFTER TERMINAL INPUT
*
* SKIP BUFFER RETRIEVAL FOR SESSION RELATED ERRORS
         CLC   =C'AZ',PEP_COM_CURRENT_ABEND_CODE
         BE    RETURN
         CLC   =C'ZC',PEP_COM_CURRENT_ABEND_CODE
         BE    RETURN
*
         CLI  EIBTRNID,C'C'              SKIP CICS XACT
         BE   RETURN
         EXEC CICS INQUIRE TASK(EIBTASKN) PRIORITY(PRIORITY)
* WE"LL TRY NOT TO BE INTERRUPTED WHILE WE FINISH
         EXEC CICS SET TASK(EIBTASKN) PRIORITY(250)
* BUT WE"LL RETURN TO INITIAL PRIORITY AT EOT
*
RSTOR    EQU   7
CAPTURE  EQU *
*>>>>>>>>>>>>>>>>
* CAPTURE ABENDED TASK SCREEN BUFFER
         EXEC CICS RECEIVE BUFFER ASIS SET(RSTOR) LENGTH(TSLENGTH)     $
               MAXLENGTH(8192) RESP(RESP)
         CLC   RESP,DFHRESP(EOC)
         BNE   RETURN
         LR    R3,RSTOR
         LH    R5,TSLENGTH
         CH    R5,=H'5000'               LOOKS TOO BIG ?
         BNL   RETURN
         LTR   R5,R5                     DID WE GET DATA
```

```
         BNH   RETURN
         LA    R4,OUTBUFR
MOVEBUFR EQU *
         CLI   Ø(R3),X'1D'              INSERT ATTRIBUTE CONTROL ?
         BE    SKIPSFLD                 HAVE TO SKIPIT
         CLI   Ø(R3),X'Ø8'              SFE RESIDUE   ?
         BE    SKIPSFLD                 HAVE TO SKIPIT
         MVC   Ø(1,R4),Ø(R3)            MOVE BYTE
         CLI   Ø(R4),X'ØØ'              BINZER FIELD ?
         BNE   NOTBINZER
         MVI   Ø(R4),X'4Ø'              CONVERT BZ TO 4Ø
NOTBINZER EQU   *
         LA    R4,1(R4)
         LA    R3,1(R3)
         BCT   R5,MOVEBUFR
         B     FINISHED
SKIPSFLD EQU   *                        CONVERT ATTRIB TO SPACE
         MVI   Ø(R4),C' '               OUTPUT BUFR + 1
         LA    R4,1(R4)                 INPUT BUFR + 2
         LA    R3,2(R3)                 DECR XTRA BYTE COUNT
         BCTR  R5,Ø
         BCT   R5,MOVEBUFR
FINISHED EQU   *       FOR XECUTED MOVE
         LA    R6,79                    GET END OF OUTPUT
         LR    R5,R4
         LA    R4,OUTBUFR       LENGTH OF OUTPUT
         SR    R5,R4
PRINTLOOP   EQU *
         MVC   OUTLINE,=CL8Ø' '
         EX    R6,MOVELINE
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
         CLC   RESP,DFHRESP(NORMAL)
         BNE   RETURN
*-------------------------------------------------------------------
         LA    R4,8Ø(R4)
         S     R5,=F'8Ø'
         BNP   PRNTOVER
         C     R5,=F'8Ø'
         BH    PRINTLOOP
         LR    R6,R5                    LAST LINE REMNANT LENGTH
         BCTR  R6,Ø                     DECR FOR EXECUTED MOVE
         BNM   PRINTLOOP
PRNTOVER EQU   *
         MVC   OUTLINE,=8ØC'_'
         MVC   OUTLINE+3Ø(16),=CL16' DFHAID = '
         LA    R1,AIDTABLE
         LA    R2,AIDLEN
AIDLOOP  EQU   *
         CLC   EIBAID,Ø(R1)
```

```
         BE    AIDFOUND
         LA    R1,6(R1)
         BCT   R2,AIDLOOP
AIDFOUND EQU   *
         MVC   OUTLINE+4Ø(5),1(R1)
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
         B     RETURN
WRITEMSG EQU   *
         EXEC CICS WRITEQ TD QUEUE('MSGS') FROM (OUTLINE) RESP(RESP)
RETURN   EQU   *
         LA    R1,PEP_COM_RETURN_OK
         ST    R1,PEP_COM_RETURN_CODE
RETURNX  DS    ØH
         EXEC CICS SET TASK(EIBTASKN) PRIORITY(PRIORITY)
         EXEC CICS RETURN
*
MOVELINE MVC   OUTLINE(Ø),Ø(R4)
         DFHEIRET
ABNDHDR  DS    ØCL8Ø
         DC    CL4Ø' DATE     TIME     ABND TASK  TERM TRAN '
         DC    CL4Ø'PROGRAM  USERID   USER NAME'
         DS    ØD
**/
AIDTABLE EQU   *
         DC    X'ØØ',CL5'NULL '
         DC    C'''',CL5'ENTER'
         DC    C'_',CL5'CLEAR'
         DC    X'6A',CL5'CLRP '    CLEAR PARTITION
         DC    C'=',CL5'PEN  '
         DC    C'W',CL5'OPID '
         DC    C'X',CL5'MSRE '
         DC    X'88',CL5'STRF '    INBOUND STRUCTURED FIELD
         DC    C'"',CL5'TRIG '     TRIGGER
         DC    C'%',CL5'PA1  '
         DC    C'>',CL5'PA2  '
         DC    C',',CL5'PA3  '
         DC    C'1',CL5'PF1  '
         DC    C'2',CL5'PF2  '
         DC    C'3',CL5'PF3  '
         DC    C'4',CL5'PF4  '
         DC    C'5',CL5'PF5  '
         DC    C'6',CL5'PF6  '
         DC    C'7',CL5'PF7  '
         DC    C'8',CL5'PF8  '
         DC    C'9',CL5'PF9  '
         DC    C':',CL5'PF1Ø '
         DC    C'#',CL5'PF11 '
         DC    C'@',CL5'PF12 '
         DC    C'A',CL5'PF13 '
```

```
         DC    C'B',CL5'PF14 '
         DC    C'C',CL5'PF15 '
         DC    C'D',CL5'PF16 '
         DC    C'E',CL5'PF17 '
         DC    C'F',CL5'PF18 '
         DC    C'G',CL5'PF19 '
         DC    C'H',CL5'PF2Ø '
         DC    C'I',CL5'PF21 '
         DC    X'4A',CL5'PF22 '
         DC    C'.',CL5'PF23 '
         DC    C'<',CL5'PF24 '
AIDLEN   EQU   (*-AIDTABLE)/6
NOAID    DC    CL6' NONE'
*
DFHEISTG DSECT
IEHIBALL DS    CL8
PRIORITY DS    F
RESP     DS    F
RESP2    DS    F
ABST     DS    F
MAXT     DS    F
SOS      DS    F
NOTASKS  DS    F
CICSOK   DS    F
TSLENGTH DS    F
STORPTR  DS    F
STARTCODE DS   H
TERM327Ø  DS   X
DUBLSIGN DS    C
OUTLINE  DS    ØCL8Ø
         DS    X
ABNDDATE DS    CL8
         DS    X
ABNDTIME DS    CL8
         DS    X
ABCODE   DS    CL4
         DS    X
ABTASK   DS    CL5
         DS    X
ABTERM   DS    CL4
         DS    X
ABTRAN   DS    CL4
         DS    X
ABPROG   DS    CL8
         DS    X
ABNDUSER DS    CL8
         DS    X
ABNDNAME DS    CL2Ø
         DS    CL(8Ø-(*-OUTLINE))
```

```
OUTBUFR  DS    CL8100
         DFHPCOM TYPE=DSECT
         END   DFHPEP
```

Robert Heaslip
Systems Programming Support (USA)

End of an era

Most people working with CICS will be familiar with Bob Yelavich. Back in July he announced that he had decided to fully retire by the end of the year. Bob, whose Web site (www.yelavich.com) tells us his surname is pronounced 'yel-a-vic', worked for IBM for forty years. While there he produced a CICS newsletter. After he retired, he continued producing the CICS newsletter for another ten years. As of July, he has stopped producing the newsletter, which was sent out to over 4,000 people globally. It will be missed by CICS aficionados. Thanks Bob, for all your work for the CICS community, from all the people who have 'followed the Yelavich road'.

NEON Systems has announced that its mainframe Web services solution, Shadow z/Services (part of its Shadow RTE mainframe integration suite) fully supports CICS Transaction Server Version 3.1.

Shadow RTE supports both Web services and real-time mainframe event integration, which can simplify the deployment of Service-Oriented Architectures for organizations that rely on CICS to run their mission-critical business services.

Shadow RTE extends and capitalizes on the new features of CICS TS 3.1, while providing customers with Shadow's ease of development, implementation, and operation, the company claims.

For further information contact:
URL: www.neonsys.com/newsroom/press_releases/2005/20050627.asp.

* * *

Acucorp has announced Version 7 of its extend Interoperability Series. With this version, companies can enhance their legacy applications and reduce their maintenance costs by taking advantage of new interoperability and compatibility features, which include expanded support for distributed CICS; facilities for integrating COBOL with Java, C, and C++; and improved compatibility with other COBOL dialects.

Using extend7, developers can exploit the flexibility and cost savings inherent in distributed environments while preserving their CICS investments, claims the company. The latest release supports distributed CICS including IBM's TXSeries for Multiplatforms running on AIX, HP-UX, Windows, and Solaris, as well as Sun Microsystem's Mainframe Transaction Processing (MTP) and Mainframe Batch Manager (MBM) running on Solaris.

Facilities for interoperating with Java make it possible to call COBOL from Java and Java from COBOL.

For further information contact:
URL: www.acucorp.com/company/newsletter/eupdate/2005/eupdate_july_2005.php.

* * *

webMethods has announced a new reseller agreement and strategic partnership with NEON Systems.

For current webMethods users, additional benefits of the integrated offering include the ability to access mainframe databases via JDBC and SQL, making mainframe data transparently available to applications that require it. These capabilities complement webMethods' existing mainframe integration functionality, which provides a non-invasive solution for enabling online CICS and IMS processes to be called as Web services.

For further information contact:
URL: www.webmethods.com/meta/default/folder/0000005139?pressReleaseDetails%5Fparam0=6652..

* * *