197

CICS update

April 2002

In this issue

CICS Update

Published by

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

North American office

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

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; $270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the December 1998 issue, are available separately to subscribers for £16.00 ($24.00) 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.

Editor

Trevor Eddolls

Disclaimer

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

Contributions

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

Printed in England.

Changes to EXEC CICS SIGNON for CICS Transaction Server Version 2.2

INTRODUCTION

CICS Transaction Server Version 2.2 alters the behaviour of the EXEC CICS SIGNON command. This article introduces the change and describes some coding techniques to replicate existing behaviour.

SOME BCKGROUND

The EXEC CICS SIGNON command was introduced in CICS/ESA Version 3. This enabled security managers and application programs to change the userid associated with a terminal. EXEC CICS SIGNON did not work for non-terminal transactions.

EXEC CICS SIGNON did two things:

- It altered the security identity for the terminal (updated the userid in the TCTTE, terminal control block) so that the next transaction to be initiated from the terminal ran with the new security identity.
- It changed the security identity of the currently executing transaction.

WHY CHANGE USERIDS?

Once EXEC CICS SIGNON was available, some opportunities for crafty function became available within CICS using the second effect. One could now change the userid associated with a terminal-attached transaction whilst it was running (background transactions could not do this).

Starting a non-terminal transaction

Non-terminal transactions were initiated from terminal-based transactions via EXEC CICS START commands. These background transactions inherited the userid active for the current terminal-based transaction. In those days there was no USERID parameter on the command.

One could get the background transaction to run with a different security identity by temporarily changing the userid around the EXEC CICS START command, eg:

```
/* Save current Authority */
EXEC CICS ASSIGN USERID(u)

/* Go into new Authority */
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID('RHARRI1')

/* Start Transaction under RHARRI1 */
EXEC CICS START TRANSID('RAH1')
                INTERVAL(Ø) NOCHECK

/* Back to proper Authority */
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID(u)
```

Changing security identity within a terminal transaction

Another consequence was that parts of the CICS transaction instance could run under different security authorities. This was useful for signon transactions that changed into the supplied security identity after validating the supplied userid, eg:

```
/* Ask for Userid + Password */
EXEC CICS CONVERSE
/* Validate Security Identity */
.....
.....
/* Set given User */
EXEC CICS SIGNOFF
EXEC CICS SIGNON
/* Continue as this User */
```

This technique was occasionally employed to provide access to some resources under a ‘super-user’ authority. The example below shows this type of sequence. This could engender security violations if the transaction failed.

```
/* Save current Authority */
EXEC CICS ASSIGN USERID(u)

/* Go into Super User */
```

```
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID('RHARRI1')

/* Read Secret file under RHARRI1 */
EXEC CICS READ FILE('SECRET')
               KEY(k) INTO(a)

/* Back to proper Authority */
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID(u)
```

These techniques will not work for CICS TS 2.2

Under CICS Transaction Server Version 2.2 one cannot change the security identity (the userid) of an executing transaction. All CICS transactions, both terminal-attached and background, have a fixed security identity.

Therefore, these methods will not work. The current behaviour can be reactivated for CICS TS 2.2 only – this restoration will not work for future CICS releases.

THE PROBLEM

This use of EXEC CICS SIGNON within a terminal-attached transaction (formally within a transaction instance that has a 3270 terminal as its principal facility) raised problems associated with the consistency of security authorities. It was possible between CICS/ ESA 3.1 and CICS Transaction Server 2.1 for various parts of the transaction instance to be using differing security identities to great confusion. Security violations could inadvertently arise. The second function of EXEC CICS SIGNON (affecting the running transaction) caused these problems.

It was easy enough for an application to manage security identities by knowing what was going on in the CICS environment, because the behaviour of CICS, although formally undefined, was usually predictable enough for problems not to arise.

Consider the case of reading two VSAM files below:

```
EXEC CICS SIGNON USERID('R')
EXEC CICS READ FILE('F1') KEY(k1) INTO(a1)
.....
```

```
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID('A')
EXEC CICS READ FILE('F2') KEY(k2) INTO(a2)
.......
EXEC CICS READ FILE('F1') KEY(k3) INTO(a3)
```

File F1 is read under security identity R and file F2 read under security identity A. The question is: under what security identity is the second access to F1 performed?

It's either R or A, but which one?

Full marks if you replied R and nil points for A.

As far as the application is concerned the EXEC CICS READ FILE('F1') KEY(k3) INTO(a3) command is running under the A security identity because this is what it set. However, CICS File Control actually saves the userid at first usage of a file for subsequent access in the transaction instance. So the second access to F1 is still running under the R security identity.

Somewhat confusing. But if the application programmer understood this phenomenon and CICS did not change its way of working, the behaviour could be managed. Problems arose when either of these conditions did not apply.

Even the behaviour of a distributed CICS transaction (formally, a CICS transaction instance involving a distributed Unit of Work) using MRO or LU6.2 was manageable to avoid security problems. Indeed, some applications specifically took advantage of this ambiguity in order to run a distributed transaction under differing security identities.

This situation changed with the introduction of the Resource Manager Interface for access to DB2 or MQ. Resource managers exist outside of the CICS environment, and have their own rules about security identities. Consequently, the ability of a CICS Unit of Work to change security identity became a problem because RMs could be operating under different security identities. A CICS transaction instance could have many different security identities.

In other words, security processing was undefined for terminal-attached transactions that changed security identity. This problem never arose for background (non-terminal-attached) transactions because they could never execute EXEC CICS SIGNON commands.

THE CHANGE

In CICS Transaction Server Version 2.2, the ability to run a CICS transaction to change its security identity (userid) is removed. Terminal-attached transactions, like non-terminal transactions, have a fixed (at transaction initiation) security identity.

EXEC CICS SIGNON and EXEC CICS SIGNOFF now effect only the terminal definition. They do not alter the security identity of the running transaction, only that of the next transaction initiated at the terminal.

The existing behaviour can be temporarily restored for CICS TS 2.2 only (by running program DFH$SNPI in the PLT) – subsequent releases of CICS will enforce the static userid per transaction instance rule.

MIGRATION PLANNING

The migration plan for moving from CICS/ESA 4.1 and CICS TS 1.3 should consider this issue. Application programs should be checked for the presence of EXEC CICS SIGNON and EXEC CICS SIGNOFF commands and changed to use other techniques to comply with the fixed security identity rule. Taking action now, when the existing behaviour is still available, is preferable to postponing until a future release of CICS enforces the rule.

Finding application programs using the commands

Planning starts by finding all application programs that use EXEC CICS SIGNON or EXEC CICS SIGNOFF commands. This can be done using the Load Module Scanner, which is available for CICS TS 1.3 (this was introduced via an APAR, so, if it's not in your 1.3 code, obtain PQ52323) and CICS TS 2.2. The JCL for this is shown below:

```
//SCANNER  EXEC PGM=DFHEISUP,PARM=('SUMMARY'),REGION=0M
//STEPLIB  DD  DSN=<CICS>.SDFHLOAD,DISP=SHR
//SYSERR   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//DFHIN    DD  DSN=<your library>,DISP=SHR
//DFHFLTR  DD  *
SIGNOFF *
SIGNON *
/*
```

The scanner uses a lot of memory, so don’t forget the REGION=0M setting.

Then you change the found programs!!! This is the more difficult part.

CHANGING PROGRAMS FOR TRANSACTIONS THAT RUN AT TERMINALS

It’s all very well for me to say that once application programs have been located containing EXEC CICS SIGNON and EXEC CICS SIGNOFF commands they have got to be changed. This is not a straightforward thing to do. Each program will have to be examined and the most appropriate technique adopted.

Any modules found by the Scanner must run as terminal-attached transactions (or, at least, part of the code in the program does) as the EXEC CICS SIGNON and EXEC CICS SIGNOFF commands do not work for background transactions.

The cases presented earlier in this article have to be reworked according to what they need to accomplish. Here are some code changes that will be required.

Starting a non-terminal transaction with a given security identity

Our very first example showed code for starting a non-terminal transaction under a different security identity. This is the easiest code change to make as the whole sequence is replaced by the USERID parameter on the EXEC CICS START command.

The code required is shown below. This assumes that the new userid exists and can execute the initiated transaction.

```
/* Start under a new Identity */
EXEC CICS START TRANSID('RAH1')
                USERID('RHARRI1')
                INTERVAL(Ø) NOCHECK
```

Starting a transaction that is to run next at the terminal with another security identity

When starting a transaction that is to run next at the terminal with another security identity, the terminal itself is going to have a new security identity for the next (and all subsequent) activity. This is processing for EXEC CICS SIGNON and EXEC CICS SIGNOFF commands that are unchanged (the first case) in CICS TS 2.2.

The example code below shows the security identity being altered in the terminal control block so subsequent transactions initiated at the terminal acquire this (new, quoted) security identity.

```
/* Assume Current Identity is RAH */

/* Change Security Identity for Terminal */
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID('RHARRI1')
/* Start more Transactions under RHARRI1 */
EXEC CICS START TRANSID('RAH1')
                INTERVAL(Ø) NOCHECK
                TERMID(eibtrmid)

/* Current Transaction still using RAH */
```

Recall that the TERMID and USERID settings on the EXEC CICS START command are mutually exclusive.

Continuing a transaction at the terminal with a different security identity

Below is the coding sequence for starting a new transaction at the terminal without the possibility of an intervening transaction or input interrupting processing. This technique is used for normal transaction continuation – now it is being used to change security identity in the second transaction.

```
/* Assume Current Identity is RAH */

/* Change Security Identity for Terminal */
EXEC CICS SIGNOFF
EXEC CICS SIGNON USERID('RHARRI1')
/* Continue in new Transaction under RHARRI1 */
EXEC CICS RETURN TRANSID('RAH1')
                 IMMEDIATE
```

This technique is very similar to the previous case, except that an EXEC CICS RETURN TRANSID is issued with the IMMEDIATE keyword. As the terminal’s security identity has been changed, this second transaction will use the new userid.

Coding a signon transaction

The code shown in the second example above should be replaced with two transactions based on the technique shown above. The first transaction will:

- Request a userid and password.
- Validate it.
- EXEC CICS SIGNOFF/SIGNON the terminal to the requested security identity.
- EXEC CICS RETURN TRANSID(second) IMMEDIATE.

The second transaction will run under the required security identity to continue initial set-up processing for the user.

Temporarily changing into super-user mode at the terminal

Changing into super-user mode at a terminal is now very difficult to achieve under CICS TS 2.2 because the general philosophy is to run everything for a transaction instance with a fixed security identity.

One way to achieve super-user functionality at the terminal is to use EXEC CICS RETURN TRANSID() IMMEDIATE techniques to split up the existing transaction into several smaller ones, which will run sequentially at the terminal, passing information between themselves. However, this means that instead of a single Unit Of Work there will now be several. This might affect LU6.2 and distributed transaction activity as well as altering recoverability aspects of current processing.

An alternative approach is to spin off separate transactions to do super-user related work in the background, returning information to the original transaction. This technique has its own difficulties associated with synchronization and extended processing. The Business Transaction Service facilities of CICS can be used to assist in the coupling of transactions.

Starting a transaction at a second terminal with a different security identity from that currently being used at the first terminal

Starting a transaction at another terminal with a different security identity from that currently being used at the first terminal has never been available within CICS, and you still cannot do it! On the EXEC CICS START command the TERMID and USERID parameters are mutually exclusive. Therefore, you can only start a transaction at another terminal using the security identity currently associated with that terminal.

However, once you have got a transaction running at this other terminal, it can always change the security identity.

CONCLUSION

The change in behaviour of EXEC CICS SIGNON to affect only a terminal and not the currently executing Transaction for CICS Transaction Server 2.2 requires:

- Changes to application programs.
- Migration planning, which should deal with this problem sooner rather than later.
- Some solutions that have been outlined in this article.

Robert Harris
CICS Technical Planning Team
IBM (UK)

Upper case translation for accented characters

Coding UCTRAN (upper case translation) for a terminal (RDO TYPETERM) or a transaction (RDO PROFILE) brings into play a 256-byte translation table addressed by the TCT prefix. The hexadecimal value of each character, from X'00' to X'FF', is treated as an offset into the table, and is replaced by whatever character is found at that offset. For example, the lower case letter 'a' (X'81') points to offset X'81', where an upper case 'A' (X'C1') is found.

Users of TMON for CICS can issue the following command string to look at a running CICS region's UCTRAN table; once found, it can also be altered:

```
=5;3;SEL=TCTFX;FLD=TCTVTRTA;B
```

Or, in other words, the field TCTVTRTA in the TCT prefix contains the address of the UCTRAN table. Any alteration takes effect immediately but, of course, lasts only until the region comes down.

Why would you want to alter the UCTRAN table? One reason might be that users are entering non-English names containing accented characters, while the default table translates only the standard set of unaccented lower case characters.

The two ‘dummy’ TCTs which follow show how macros can be used to override the default UCTRAN table. Note that either macro may also be used in an existing TCT table; in either case, the UCTRAN macro should immediately follow the TYPE=INITIAL DFHTCT macro.

The first, DFHTCT99, translates accented lower case characters to accented upper case; accented characters already in upper case are left unaltered. The second, DFHTCT98, does away with accented characters altogether, converting them to unaccented upper case. Both continue to translate unaccented lower case characters, of course.

DFHTCT99

```
         MACRO
         UCACCENT
DFHUCTRT CSECT                         RESUME UCTRAN TABLE CSECT
.* TRANSLATE LOWER-CASE ACCENTED CHARACTERS TO UPPER-CASE ACCENTED.
         ORG  TCZUCTAB+X'42'           REPLACE: HEX 42-49
         DC   X'6263646566676869'         WITH: HEX 62-69
         ORG  TCZUCTAB+X'51'           REPLACE: HEX 51-58
         DC   X'7172737475767778'         WITH: HEX 71-78
         ORG  TCZUCTAB+X'8D'           REPLACE: HEX 8D
         DC   X'AD'                       WITH: HEX AD
         ORG  TCZUCTAB+X'CB'           REPLACE: HEX CB-CF
         DC   X'EBECEDEEEF'               WITH: HEX EB-EF
         ORG  TCZUCTAB+X'DB'           REPLACE: HEX DB-DE
         DC   X'FBFCFDFE'                 WITH: HEX FB-FE
         ORG  ,                        RESTORE THE LOCATION COUNTER.
&SYSLOC  LOCTR                         RESUME PREVIOUS LOCATION
         MEND                          END OF MACRO DEFINITION
*
         DFHTCT TYPE=INITIAL,SUFFIX=99,                               *
               MIGRATE=COMPLETE,                                      *
               ACCMETH=(VTAM),                                        *
               DUMMY=DUMMY
*
```

```
         UCACCENT                      UCTRAN FOR ALL ACCENTED CHARS.
*
         DFHTCT TYPE=FINAL
         END DFHTCTBA
```

DFHTCT98

```
         MACRO
         UCACCENT
DFHUCTRT CSECT                         RESUME UCTRAN TABLE CSECT
.* TRANSLATE LOWER-CASE ACCENTED CHARACTERS TO UPPER-CASE NO ACCENT.
         ORG  TCZUCTAB+X'42'           REPLACE: HEX 42-49
         DC   C'AAAAAACN'                 WITH: CHARACTERS
         ORG  TCZUCTAB+X'51'           REPLACE: HEX 51-58
         DC   C'EEEEIIII'                 WITH: CHARACTERS
         ORG  TCZUCTAB+X'62'           REPLACE: HEX 62-69
         DC   C'AAAAAACN'                 WITH: CHARACTERS
         ORG  TCZUCTAB+X'71'           REPLACE: HEX 71-78
         DC   C'EEEEIIII'                 WITH: CHARACTERS
         ORG  TCZUCTAB+X'8D'           REPLACE: HEX 8D
         DC   C'Y'                        WITH: Y
         ORG  TCZUCTAB+X'9A'           REPLACE: HEX 9A-9B
         DC   C'AO'                       WITH: AO
         ORG  TCZUCTAB+X'AD'           REPLACE: HEX AD
         DC   C'Y'                        WITH: Y
         ORG  TCZUCTAB+X'CB'           REPLACE: HEX CB-CF
         DC   C'OOOOO'                    WITH: CHARACTERS
         ORG  TCZUCTAB+X'DB'           REPLACE: HEX DB-DF
         DC   C'UUUUY'                    WITH: CHARACTERS
         ORG  TCZUCTAB+X'EB'           REPLACE: HEX EB-EF
         DC   C'OOOOO'                    WITH: CHARACTERS
         ORG  TCZUCTAB+X'FB'           REPLACE: HEX FB-FE
         DC   C'UUUU'                     WITH: CHARACTERS
         ORG  ,                        RESTORE THE LOCATION COUNTER.
&SYSLOC  LOCTR                         RESUME PREVIOUS LOCATION
         MEND                          END OF MACRO DEFINITION
*
         DFHTCT TYPE=INITIAL,SUFFIX=98,                                *
               MIGRATE=COMPLETE,                                       *
               ACCMETH=(VTAM),                                         *
               DUMMY=DUMMY
*
         UCACCENT                      UCTRAN FOR ALL ACCENTED CHARS.
*
         DFHTCT TYPE=FINAL
         END DFHTCTBA
```

Taras Wolansky
Technical Consultant (USA)

Migrating from HSM to cryptographic co-processor

At many IBM mainframe sites, an HSM Racal Box is used for PIN verification. A cryptographic co-processor can also be used for this purpose. This facility is supplied within IBM mainframes; to exploit this facility, the cryptographic co-processor should be enabled and ICSF (Integrated Cryptographic Service Facility) should be installed.

To migrate from HSM to a co-processor, the first thing to do after installing ICSF is transfer the Zone PIN Keys (ZPK) from HSM to ICSF. These keys are used as IPINENC (input PIN-encrypting key) or OPINENC (output PIN-encrypting key) in ICSF. If the unencrypted values of the keys are known, they can be directly loaded to ICSF.

However, if you know only the ZPK values that are encrypted under the HSM master key, first you should export the Zone PIN Keys from HSM and then import those keys into ICSF as IPINENC or OPINENC.

The following is an example online program for reformatting a PIN Key by using ICSF APIs.

Note: to compile a program that will use ICSF APIs, the dataset CSF.SCSFMOD0 should be included as the SYSLIB DDname of the link step.

```
*****************************************************************
* THIS PROGRAM; REFORMATS A PIN THAT IS ENCRYPTED UNDER         *
*              AN IPINENC KEY (ACQWK1), ENCRYPTS THE            *
*              REFORMATTED PIN UNDER AN OPINENC KEY (ISSWK1),   *
*              AND DISPLAYS THE OUTPUT.                         *
*                                                               *
*              INPUT  PIN BLOCK FORMAT: HSM PIN BLOK FORMAT Ø1 *
*              OUTPUT PIN BLOCK FORMAT: HSM PIN BLOK FORMAT Ø3 *
*****************************************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. PINTEST.
 AUTHOR.     ERHAN PASA
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.  IBM-Z67.
 OBJECT-COMPUTER.  IBM-Z67.
 DATA DIVISION.
 FILE SECTION.
 WORKING-STORAGE SECTION.
```

```
************* DEFINE SAPI INPUT/OUTPUT PARAMETERS ************
 Ø1  SAPI-REC.
     Ø5  RETURN-CODE-S             PIC       9(Ø8) COMP.
     Ø5  REASON-CODE-S             PIC       9(Ø8) COMP.
     Ø5  EXIT-DATA-LENGTH-S        PIC       9(Ø8) COMP.
     Ø5  EXIT-DATA-S               PIC       X(Ø4).
     Ø5  RULE-ARRAY-COUNT-S        PIC       9(Ø8) COMP.
     Ø5  RULE-ARRAY-S.
         1Ø  RULE-ARRAY            PIC       X(Ø8).
     Ø5  IPINENC-ID                PIC       X(64)
         VALUE LOW-VALUES.
     Ø5  OPINENC-ID                PIC       X(64)
         VALUE LOW-VALUES.
     Ø5  PIN-BLOCK-IN              PIC       X(8)
         VALUE LOW-VALUES.
     Ø5  PIN-BLOCK-OUT             PIC       X(8)
         VALUE LOW-VALUES.
     Ø5  INP-PIN-PROFILE.
         1Ø  INP-PIN-PROF-A1       PIC       X(8)
             VALUE LOW-VALUES.
         1Ø  INP-PIN-PROF-A2       PIC       X(8)
             VALUE LOW-VALUES.
         1Ø  INP-PIN-PROF-A3       PIC       X(8)
             VALUE LOW-VALUES.
     Ø5  PAN-DATA-IN               PIC       X(12)
         VALUE LOW-VALUES.
     Ø5  OUT-PIN-PROFILE.
         1Ø  OUT-PIN-PROF-A1       PIC       X(8)
             VALUE LOW-VALUES.
         1Ø  OUT-PIN-PROF-A2       PIC       X(8)
             VALUE LOW-VALUES.
         1Ø  OUT-PIN-PROF-A3       PIC       X(8)
             VALUE LOW-VALUES.
     Ø5  PAN-DATA-OUT              PIC       X(12)
         VALUE LOW-VALUES.
     Ø5  SEQ-NO                    PIC       9(Ø8) COMP.
*
 PROCEDURE DIVISION.
 MAIN-RTN.
*
***************************************************************
* ONLY THE FOLLOWING PARAMETERS CAN BE CHANGED FOR THE TESTS, *
* BECAUSE THESE ARE THE VARIABLES THAT WILL BE INPUT FROM     *
* OUTSIDE OF THE SYSTEM. THE OTHER PARAMETERS ARE STATIC.     *
*           PIN-BLOCK-IN, PAN-DATA-IN                         *
*                                                             *
* THE OTHER PARAMETERS SHOULD NOT BE CHANGED EVEN IN A REAL   *
* PRODUCTION PROGRAM.                                         *
***************************************************************
        MOVE Ø                   TO  EXIT-DATA-LENGTH-S.
        MOVE LOW-VALUES          TO  EXIT-DATA-S.
```

```
*---------------------------------------------------------------
* IPINENC-ID : THE PIN ENCRYPTING KEY LABEL
*              UNDER WHICH THE INCOMING PIN IS ENCRYPTED.
*
        MOVE 'ACQWK1'            TO  IPINENC-ID.
*---------------------------------------------------------------
* OPINENC-ID : THE PIN ENCRYPTING KEY LABEL THAT WILL BE USED
*              FOR ENCRYPTING THE INCOMING PIN AFTER IT IS
*              REFORMATTED.
*
        MOVE 'ISSWK1'            TO  OPINENC-ID.
*---------------------------------------------------------------
* PIN-BLOCK-IN : INCOMING PIN BLOCK
        MOVE X'345989AB84F984CF' TO  PIN-BLOCK-IN.
*---------------------------------------------------------------
* RULE-ARRAY-... : THE PARAMETERS THAT ARE NEEDED FOR
*                  REFORMATTING AND ENCRYPTING THE INCOMING
*                  PIN UNDER A DIFFERENT KEY.
        MOVE 1                   TO  RULE-ARRAY-COUNT-S.
        MOVE 'REFORMAT'          TO  RULE-ARRAY-S.
*---------------------------------------------------------------
* INP-PIN-PROF-XX  :  THE PARAMETERS THAT ARE NEEDED FOR
*                     HSM PIN BLOK FORMAT Ø1. THIS IS
*                     THE INPUT PIN BLOCK FORMAT.
*                     (THE VALUE OF 'INP-PIN-PROF-A3'
*                      PARAMETER IS NOT IMPORTANT.
        MOVE 'ISO-Ø   '          TO  INP-PIN-PROF-A1.
        MOVE 'NONE    '          TO  INP-PIN-PROF-A2.
        MOVE '       F'          TO  INP-PIN-PROF-A3.
*---------------------------------------------------------------
* PAN-DATA-IN : THE RIGHTMOST 12 DIGIT OF THE ACCOUNT NUMBER
*               EXCEPT THE CHECK DIGIT.
*               Example: PAN: '5534FB9879ACØ262'
*                        PAN-DATA-IN : '4FB9879ACØ26'
        MOVE '4FB9879ACØ26'      TO  PAN-DATA-IN.
*---------------------------------------------------------------
* OUT-PIN-PROF-XX  :  THE PARAMETERS THAT ARE NEEDED FOR
*                     HSM PIN BLOK FORMAT Ø3. THIS IS
*                     THE OUTPUT PIN BLOCK FORMAT.
        MOVE 'VISA-3  '          TO  OUT-PIN-PROF-A1.
        MOVE 'NONE    '          TO  OUT-PIN-PROF-A2.
        MOVE '       F'          TO  OUT-PIN-PROF-A3.
*---------------------------------------------------------------
* PAN-DATA-OUT : THE VALUE OF THIS PARAMETER IS NOT IMPORTANT.
        MOVE 'ØØØØØØØØØØØØ'      TO  PAN-DATA-OUT.
*---------------------------------------------------------------
* SEQ-NO : THE VALUE OF THIS PARAMETER IS NOT IMPORTANT.
        MOVE Ø                   TO  SEQ-NO.
*************  BEGIN - TRANSLATE PIN *************************
        CALL 'CSFPTR' USING RETURN-CODE-S
                            REASON-CODE-S
                            EXIT-DATA-LENGTH-S
                            EXIT-DATA-S
                            IPINENC-ID
                            OPINENC-ID
                            INP-PIN-PROFILE
                            PAN-DATA-IN
                            PIN-BLOCK-IN
                            RULE-ARRAY-COUNT-S
                            RULE-ARRAY-S
                            OUT-PIN-PROFILE
                            PAN-DATA-OUT
                            SEQ-NO
                            PIN-BLOCK-OUT.
        IF RETURN-CODE-S  NOT = Ø OR
           REASON-CODE-S > 4 THEN
           EXEC CICS
              WRITE OPER TEXT('*** TRANSLATE UNSUCCESSFULL ***')
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT('*** RETURN-CODE = ')
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT(RETURN-CODE-S)
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT('*** REASON-CODE = ')
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT(REASON-CODE-S)
           END-EXEC
        ELSE
           EXEC CICS
              WRITE OPER TEXT('*** TRANSLATE SUCCESSFULL ***')
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT('*** COMP TEXT ==>')
           END-EXEC
           EXEC CICS
              WRITE OPER TEXT(PIN-BLOCK-OUT)
           END-EXEC.
*************  END - TRANSLATE PIN ***************************
     EXEC CICS
        WRITE OPER TEXT('*** TEST PROGRAM ENDED ***')
     END-EXEC
     DISPLAY '*** TEST PROGRAM ENDED ***'
     STOP RUN.
```

Erhan Pasa
Senior System Programmer
Aknet AS (Turkey)

Use IBM’s Time Machine and the CWA

To change the time and date in a CICS test environment, it’s simplest and most effective to use a tool. So we looked for a suitable product, and the best and simplest tool we could find was IBM’s Time Machine. The tool is available via the Web as a CICS SupportPac.

However, a lot of our applications use the CWA to obtain the date and time, and they don’t work with ‘EXEC CICS ASKTIME......’. This seemed to mean that we couldn’t use Time Machine from IBM. Fortunately this wasn’t the case!

What can we do? First let’s have a look at the way we administer the CWA. All relevant data is mentioned in a definition-chained area called CSCWAA.

```
         EJECT
*---------------------------------------------------------------------
*        C I C S       C W A - R E G I O N                           *
*        INCLUDE-ELEMENT  FOR ASM  PROGRAM  CSCWAA                   *
*---------------------------------------------------------------------
*        ADDRESSING   :   EXEC CICS ADDRESS CWA(CWAPTR)              *
*        WARNING      :   NO  CHANGES ALLOWED - READ-ONLY            *
*                         CWAPTR MUST BE DEFINED                     *
*---------------------------------------------------------------------
                SPACE 3
                USING CWADSECT,CWAPTR
                SPACE 3
CWADSECT        DSECT
                SPACE 3
CWAAREA         DS   ØCL1536         CWA-AREA
                SPACE 1
CWAECSA         DC    AL4(Ø)         POINTER ECSA-CTL-AREA
CWAENQA         DC    AL4(Ø)         POINTER ECSA ENQ-AREA
CWATSYS         DC    CL4' '         TERMINAL-SYSID FOR CWACICID
CWASYSID        DC    CL4' '         ORIGINAL SYSTEM-ID
CWAPPLID        DC    CL8' '         ORIGINAL APPLICATION-ID
CWANCVT         DC    AL4(Ø)         POINTER NLV-CVT
CWA_PTR_CUATR   DC    AL4(Ø)         ADDRESS D.CUA-TRANSACTIONSTABELLE
CWA_CICSLEVEL   DC   ØCL4' '         CICS-LEVEL 'Ø311' OR 'Ø33Ø'
CWA_CICSLEV     DC    CL1' '         CICS-LEVEL
CWA_CICSVER     DC    CL1' '         CICS-VERSION
CWA_CICSREL     DC    CL1' '         CICS-RELEASE
CWA_CICSMOD     DC    CL1' '         CICS-MODIFICATION
CWA_CMFSTOP     DC    PL4'Ø'         STOP-TIME FOR CMF-EVENTS HHMMSSTC
                DS    XL56Ø          .........  FREI  .........
                DS    XL12Ø          .........  FREI  .........
                DS    XL22           .........  FREI  .........
CWACICTX        DC    CL4' '         CICS-ID-BESCHREIBUNG
CWACICID        DC    CL1' '         CICS-ID
CWA$PROD        EQU   C'P'            .. PROD
CWA$TEST        EQU   C'T'            .. TEST
CWA$VPRD        EQU   C'V'            .. VORPROD
CWA$BOST        EQU   C'S'            .. SYSTEM-CICS
CWACICNR        DC    CL1' '         CICS-NR
CWA$TERM        EQU   C'T'            .. TERMINAL
CWA$VSAM        EQU   C'V'            .. DATASET VSAM
CWA$PAIS        EQU   C'P'            .. PAISY
CWA$ODM         EQU   C'O'            .. ODM
CWA$PROB        EQU   C'9'            .. APPLICATION 9 / PROBLEMCICS
CWA$APPL        EQU   C'Ø'            .. APPLICATION ØØ-Ø9
*  $APPL        EQU   ????            .. APPLICATION A-C
*  $APPL        EQU   ????            .. APPLICATION E-O
*  $APPL        EQU   ????            .. APPLICATION Q-S
*  $APPL        EQU   ????            .. APPLICATION U-Z
CWADATUM        DC    CL8' '         DATUM FORMAT TT.MM.JJ
CWACTMJ         DC    CL6' '         DATUM TTMMJJ
CWAPTMJ         DC    PL4'Ø'         DATUM ØTTMMJJC
CWACJMT         DC    CL6' '         DATUM JJMMTT
CWAPJMT         DC    PL4'Ø'         DATUM ØJJMMTTC
CWACTMJ4        DC    CL8' '         DATUM TTMMJJJJ
CWAPTMJ4        DC    PL5'Ø'         DATUM ØTTMMJJJJC
CWACJ4MT        DC    CL8' '         DATUM JJJJMMTT
CWAPJ4MT        DC    PL5'Ø'         DATUM ØJJJJMMTTC
CWACMJ          DC    CL4' '         DATUM MMJJ
CWAPMJ          DC    PL3'Ø'         DATUM ØMMJJC
CWACJM          DC    CL4' '         DATUM JJMM
CWAPJM          DC    PL3'Ø'         DATUM ØJJMMC
CWACMJ4         DC    CL6' '         DATUM MMJJJJ
CWAPMJ4         DC    PL4'Ø'         DATUM ØMMJJJJC
CWACJ4M         DC    CL6' '         DATUM JJJJMM
CWAPJ4M         DC    PL4'Ø'         DATUM ØJJJJMMC
CWACT3J         DC    CL5' '         DATUM TTTJJ
CWAPT3J         DC    PL3'Ø'         DATUM TTTJJC
CWACJT3         DC    CL5' '         DATUM JJTTT
CWAPJT3         DC    PL3'Ø'         DATUM JJTTTC
CWACT3J4        DC    CL7' '         DATUM TTTJJJJ
CWAPT3J4        DC    PL4'Ø'         DATUM TTTJJJJC
CWACJ4T3        DC    CL7' '         DATUM JJJJTTT
CWAPJ4T3        DC    PL4'Ø'         DATUM JJJJTTTC
CWAZEIT         DC    CL5' '         UHRZEIT  SS:MM
CWATABLE        DS   ØCL24           ,+Ø123456789-,Ø123456789
CWATAB1         DS   ØCL13           TABELLE 1 /,/+/Ø123456789/-/
CWATAB2         DS   ØCL12           TABELLE 2 /,/+/Ø123456789/
CWACHKO1        DC    C','
CWATAB3         DS   ØCL12           TABELLE 3 /+/Ø123456789/-/
CWATAB4         DS   ØCL11           TABELLE 4 /+/Ø123456789/
```

```
CWACHARP        DC    C'+'
CWATAB5         DS   ØCL12          TABELLE 5 /Ø123456789/-/,/
CWATAB6         DS   ØCL11          TABELLE 6 /Ø123456789/-/
CWATAB7         DS   ØCL1Ø          TABELLE 7 /Ø123456789/
CWACHØ9         DC    C'Ø123456789'
CWACHARM        DC    C'-'
CWATAB8         DS   ØCL11          TABELLE 8 /,/Ø123456789/
CWACHKO2        DC    C','
CWACHØ92        DC    C'Ø123456789'
CWAZEITP        DC    PL4'Ø'        UHRZEIT  HHMMSSTC
CWADAY          DC    CL1Ø' '       WOCHENTAG
CWAMONTH        DC    CL9' '        MONAT
CWA_PTR_FTT     DC    AL4(Ø)        ADDRESS D. FUNKTIONSTASTENTABELLE
CWA_PTR_ANT     DC    AL4(Ø)        ADDRESS THE AKTIONSNAMENTABELLE
CWA_INFOCICS    DC    C' '          INFO-CICS IDENTIFIER
CWA_INFOCICS_Y  EQU   C'Y'          INFO-CICS IDENTIFIER -JA-
CWA_INFOCICS_N  EQU   C' '          INFO-CICS IDENTIFIER -NEIN-
CWA_DATUM_JJJJ  DC    CL1Ø' '       DATE FORMAT TT.MM.JJJJ
                SPACE 1
CWAAREAE        EQU   *             END CWA DEFINITIONS
                SPACE 5
*---------------------------------------------------------------------
*         END OF THE  CICS   CWA_AREAS                          *
*---------------------------------------------------------------------
*         START OF DSECT FOR FUNKTIONSTASTENTABELLE                   *
*         ADDRESS  "CWA_PTR_FTT"                                      *
*---------------------------------------------------------------------
CWAFTTDSECT      DSECT
CWA_FTT_TASTE    DC    XL1'Ø'        TASTENIDENTIFIKATION
CWA_FTT_AKTION   DC    CL16' '       KURZBEZEICHNUNG DER TASTE
*                                    BSP. : HILFE
CWA_FTT_ANZEIGE  DC    CL2Ø' '              TEXT FOR THE FUNKTIONS
*                                           TASTENBLOCK EINES BILDES
*                                    BSP. : F1=HILFE
CWA_FTT_PFKEY    DC    CL4' '        PF-TASTE Z.B. "PF1 "
CWA_FTT_KURZTEXT DC    CL8' '        TASTENKUERZEL FOR POP-UP-MENUS
*                                    BSP. : F12=ABBR
CWA_FTT_TEXT     DS    CL2Ø7         DESCRIPTION DER AKTION
CWAFTTDSECTE     EQU   *
CWAFTTANZAHL     EQU   3Ø            NUMBER OF TABELLENEINTRAEGE FTT
                 SPACE 2
*---------------------------------------------------------------------
*         END OF THE DSECT FOR FUNKTIONSTASTENTABELLE                 *
*---------------------------------------------------------------------
*         BEGIN THE DSECT FOR AKTIONSNAMENTABELLE                     *
*         ADDRESS  "CWA_PTR_ANT"                                      *
*---------------------------------------------------------------------
CWAANTDSECT DSECT
CWA_ANT_HILFE         DS CL16             HELP TEXT
CWA_ANT_TASTEN        DS CL16             DISPLAY THE TASTENBELEGUNG
CWA_ANT_AUSGANG       DS CL16             END A FUNCTION
```

```
CWA_ANT_REFRESH       DS CL16             RESTORE
CWA_ANT_UPDATE        DS CL16             DATA SOURCE
CWA_ANT_RUECKWAERTS   DS CL16             BACKWARDS BROWSE
CWA_ANT_VORWAERTS     DS CL16             FORWARDS BROWSE
CWA_ANT_AKTION        DS CL16             ACTIVATE ACTIONBAR
CWA_ANT_UNTERBRECHEN  DS CL16             VORGANGSUNTERBRECHUNG
CWA_ANT_ABBRUCH       DS CL16             ABORT
CWA_ANT_EINSTIEG      DS CL16             BACK TO EINSTIEGSBILD
CWA_ANT_AUSWAHL       DS CL16             BACK TO AUSWAHLBILD
CWA_ANT_SICHERN       DS CL16             FREEZE THE DATA
CWA_ANT_LINKS         DS CL16             LEFT-SIDE PAGES
CWA_ANT_RECHTS        DS CL16             RIGHT-SIDE PAGES
CWA_ANT_ANFANG        DS CL16             SHOW THE FIRST SIDE
CWA_ANT_SCHLUSS       DS CL16             SHOW THE OTHER SIDE
CWA_ANT_ABMELDEN      DS CL16             ZSS-ABMELDUNG
CWA_ANT_DRUCKEN       DS CL16             PRINT (PA2)
CWA_ANT_LOESCHEN      DS CL16             OUTPUT TO SCREEN
CWA_ANT_DATENFREIGABE DS CL16             DATENFREIGABE
CWA_ANT_HILFE_ANLEGEN DS CL16             BOSHELP HELP START
CWA_ANT_SUCHEN        DS CL16             SEARCH
CWA_ANT_EURODM        DS CL16             CONVERT EURO/DM
CWAANTDSECTE          EQU   *
*--------------------------------------------------------------------
*         END OF THE DSECT FOR AKTIONSNAMENTABELLE                  *
*--------------------------------------------------------------------
```

This area is supplied via the program CSCWAUPD.

```
*ASM XOPTS(CICS,EDF,NOEPILOG,SP)
CSCWAUPD TITLE '- UPDATE SEVERAL CWA-FIELDS'
         SPACE 1
*--------------------------------------------------------------------*
*               C S C W A U P D                                      *
*--------------------------------------------------------------------*
*  WARNING:   PROGRAM MUST BE LINKED   AMODE=31 / RMODE=ANY          *
*            AT MIDNIGHT - SEVERAL CWA-FIELDS MUST BE UPDATED        *
*            FOR CICS SYSTEMS THAT RUNS LONGER THAN 24 HOURS.        *
*            THIS PROGRAM IS CALLED VIA TRANSACTION 'NCWA',          *
*            IN PLT-PROCESSING OR IF IBM'S TIME MACHINE WAS          *
*            USED.                                                   *
*            AN INFORMATION-MESSAGE DURING THIS PROCESS AT           *
*            SYSTEM-CONSOLE IS ALSO AVAILABLE.                       *
*--------------------------------------------------------------------*
             EJECT
*--------------------------------------------------------------------*
       ++INCLUDE CSCWAA                INCLUDE CICS NLV CWA ASSEMBLER
             EJECT 1
DFHEISTG     DSECT
EIS_RESP     DC    F'Ø'                RESPONSE FIELD FROM E.C.REQUEST
EIS_DOWO     DC    D'Ø'                DOUBLE-WORD
EIS_YYDDD    DC    CL6'YY.DDD'         WORK-FIELD FOR FORMATTIME
```

```
EIS_HHMMSSC  DC    CL6' '              WORK-FIELD FOR FORMATTIME
EIS_HHMMSS   DC    PL8'Ø'              WORK-FIELD FOR FORMATTIME
EIS_MM       DC    AL4(Ø)              WORK-FIELD FOR FORMATTIME
EIS_WORKFLD  DC    CL5' '              WORK-FIELD FOR CONVERT
EIS_TIME     DS    CL7                 VALUE FOR TIME CONVERT
EIS_DAYINDX  DC    F'Ø'                DAY OF WEEK
EIS_MSG      DC    CL1ØØ' '            MESSAGE-FIELD FOR WTO-COMMAND
EIS_SYSID    DC    CL4' '              SYSTEM ID
EIS_APPLID   DC    CL8' '              APPLID FROM DFHSIT..
EIS_YEAR     DS    F                   WORK-FIELD FOR FORMATTIME
CSCWAUPD DFHEIENT CODEREG=R1Ø,DATAREG=R12
CSCWAUPD AMODE ANY
CSCWAUPD RMODE ANY
         EJECT
*---------------------------------------------------------------------*
*              REFRESH NLV CWA VALUES                                 *
*---------------------------------------------------------------------*
         EXEC  CICS ASSIGN     APPLID      (EIS_APPLID)               *
                               SYSID       (EIS_SYSID)                *
         EXEC  CICS ADDRESS    CWA         (CWAPTR)                   *
         EXEC  CICS ASKTIME    ABSTIME     (EIS_DOWO)                 *
         EXEC  CICS FORMATTIME ABSTIME     (EIS_DOWO)                 *
                               DDMMYY      (CWADATUM)                 *
                               YYDDD       (EIS_YYDDD)                *
                               DATESEP     ('.')                      *
                               DAYOFWEEK   (EIS_DAYINDX)              *
                               MONTHOFYEAR (EIS_MM)                   *
                               TIME        (EIS_HHMMSSC)              *
                               YEAR        (EIS_YEAR) ,
         L     R5,EIS_YEAR           YEAR IN BINARY
         CVD   R5,EIS_DOWO           CONVERT TO DECIMAL
         OI    EIS_DOWO+L'EIS_DOWO-1,X'ØF'
         UNPK  EIS_YEAR,EIS_DOWO     YEAR in CHARACTER                *
         EXEC  CICS INQUIRE    SYSTEM                                 *
                               RELEASE     (CWA_CICSLEVEL)
         PACK  EIS_HHMMSS,EIS_HHMMSSC(6) CONVERT CHAR TO PACKED
         MP    EIS_HHMMSS,=P'1Ø'       SET TIME VALUE HHMMSST
MVC   CWAPPLID,EIS_APPLID   SAVE ORIGINAL APPLID IN CWA
         MVC   CWASYSID,EIS_SYSID    SAVE ORIGINAL SYSID  IN CWA
         MVC   CWACICID,EIS_APPLID+4 MRO CICS-ID FROM APPLID
         MVC   CWACICNR,EIS_APPLID+5 MRO CICS-NR FROM APPLID
         LA    R5,SYSIDTAB           CICS-ID-BESCHR.-TABELLE
         LH    R7,=Y((SYSIDTBE-SYSIDTAB)/L'SYSIDTAB) NBR. TAB-ENTRIES
CWACONT2 DS    ØH
         CLC   CWACICID,Ø(R5)        CICS-ID FOUND ?
         BNE   CWACONT3              NO...COMPARE NEXT ONE
         MVC   CWACICTX,1(R5)        CICS-ID-DESCRIPTION
         B     CWACONT4              SKIP OUTSIDE LOOP
CWACONT3 DS    ØH
         LA    R5,L'SYSIDTAB(,R5)    POINT TO NEXT SYSID-ENTRY
         BCT   R7,CWACONT2           LOOP (MAX. SYSIDTAB-ENTRIES)
```

```
CWACONT4 DS    ØH
         MVC   CWACTMJ(2),CWADATUM   DATE CHAR   TTMMJJ
         MVC   CWACTMJ+2(2),CWADATUM+3
         MVC   CWACTMJ+4(2),CWADATUM+6
         PACK  CWAPTMJ,CWACTMJ       DATE PACKED ØTTMMJJC
         ZAP   CWAPTMJ,CWAPTMJ
         MVC   CWACJMT(2),CWADATUM+6 DATE CHAR   JJMMTT
         MVC   CWACJMT+2(2),CWADATUM+3
MVC   CWACJMT+4(2),CWADATUM
         PACK  CWAPJMT,CWACJMT       DATE PACKED ØJJMMTTC
         ZAP   CWAPJMT,CWAPJMT
         MVC   CWACTMJ4(6),CWACTMJ   DATE CHAR TTMMJJJJ
         MVC   CWACTMJ4+6(2),CWACTMJ+4
         MVC   CWACTMJ4+4(2),EIS_YEAR
         PACK  CWAPTMJ4,CWACTMJ4
         ZAP   CWAPTMJ4,CWAPTMJ4
MVC   CWACJ4MT+2(6),CWACJMT DATE CHAR JJJJMMTT
         MVC   CWACJ4MT(2),EIS_YEAR
         PACK  CWAPJ4MT,CWACJ4MT
         ZAP   CWAPJ4MT,CWAPJ4MT
         MVC   CWACMJ,CWACTMJ+2      DATE MMJJ
         PACK  CWAPMJ,CWACMJ
         ZAP   CWAPMJ,CWAPMJ
         MVC   CWACJM,CWACJMT        DATE JJMM
         PACK  CWAPJM,CWACJM
         ZAP   CWAPJM,CWAPJM
         MVC   CWACMJ4,CWACTMJ4+2    DATE MMJJJJ
         PACK  CWAPMJ4,CWACMJ4
         ZAP   CWAPMJ4,CWAPMJ4
         MVC   CWACJ4M,CWACJ4MT      DATE JJJJMM
         PACK  CWAPJ4M,CWACJ4M
         ZAP   CWAPJ4M,CWAPJ4M
         MVC   EIS_WORKFLD+Ø(2),EIS_YYDDD  GET YY FROM FORMATTIME
         MVC   EIS_WORKFLD+2(3),EIS_YYDDD+3 GET DDD FROM FORMATTI
         PACK  CWAPJT3,EIS_WORKFLD      DATE JJTTT
         UNPK  CWACJT3,CWAPJT3
         OI    CWACJT3+4,X'FØ'
         PACK  CWAPJT3,CWACJT3
         ZAP   CWAPJT3,CWAPJT3
         MVC   CWACT3J(3),CWACJT3+2  DATE TTTJJ
         MVC   CWACT3J+3(2),CWACJT3
         PACK  CWAPT3J,CWACT3J
         ZAP   CWAPT3J,CWAPT3J
         MVC   CWACT3J4(3),CWACT3J    DATE TTTJJJJ
         MVC   CWACT3J4+3(2),EIS_YEAR
         MVC   CWACT3J4+5(2),CWACT3J+3
         PACK  CWAPT3J4,CWACT3J4
         ZAP   CWAPT3J4,CWAPT3J4
         MVC   CWACJ4T3+2(5),CWACJT3  DATE JJJJTTT
         MVC   CWACJ4T3(2),EIS_YEAR
         PACK  CWAPJ4T3,CWACJ4T3
         ZAP   CWAPJ4T3,CWAPJ4T3
         UNPK  EIS_TIME,EIS_HHMMSS    TIME SS:MM
         MVC   CWAZEIT(2),EIS_TIME
         MVI   CWAZEIT+2,C':'
         MVC   CWAZEIT+3(2),EIS_TIME+2
         MVI   CWACHKO1,C','          PERFORM NUMERIC TABLES
         MVI   CWACHARP,C'+'
         MVC   CWACHØ9,=C'Ø123456789'
         MVI   CWACHARM,C'-'
         MVI   CWACHKO2,C','
         MVC   CWACHØ92,=C'Ø123456789'
ZAP   CWAZEITP,EIS_HHMMSS     UHRZEIT HHMMSSTC
         ICM   R5,B'1111',EIS_DAYINDX  DAY OF WEEK
         MH    R5,=Y(L'DAYTAB)         OFFSET OF WEEKTAB
         LA    R5,DAYTAB(R5)           ADDR OF WEEK-DAY
         MVC   CWADAY,Ø(R5)            MOVE CURRENT WEEK-DAY
         ICM   R5,B'1111',EIS_MM       MONTH OF YEAR
         BCTR  R5,Ø                    OFFSET-VALUE
         MH    R5,=Y(L'MONTHTAB)       OFFSET OF MONTH
         LA    R5,MONTHTAB(R5)         ADDR OF MONTH
         MVC   CWAMONTH,Ø(R5)          MOVE CURRENT MONTH
         MVC   CWATSYS+Ø(2),EIS_APPLID+3 GET PREFIX FROM APPLID
         MVC   CWATSYS+2(2),=C'TØ'     GET SUFFIX FROM LITERAL
         ZAP   CWA_CMFSTOP,=PL4'Ø8ØØØØØ' STOP-TIME FOR CMF-EVENTS
         MVC   CWA_DATUM_JJJJ+Ø(6),CWADATUM
         MVC   CWA_DATUM_JJJJ+6(2),EIS_YEAR
         MVC   CWA_DATUM_JJJJ+8(2),CWADATUM+6
*
         MVC   EIS_MSG,BLANK
         MVC   EIS_MSG(L'MSGØØ1),MSGØØ1
         BAL   R9,MESSAGE              WRITE MESSAGE
         EJECT 1
*---------------------------------------------------------------------
*              RECURSIVE START OF MIDNIGHT-TRANSACTION 'NCWA'
*---------------------------------------------------------------------
         SPACE 1
         EXEC  CICS START TRANSID  ('NCWA')                           *
                          INTERVAL (24ØØØØ)                           *
                          RESP     (EIS_RESP)
         CLC   EIS_RESP,DFHRESP(NORMAL) ANY ERRORS DETECTED ?
         BNE   RETURN_3                YES...ISSUE INFORMATION-MESSAGE
         EJECT 1
*---------------------------------------------------------------------*
*              R E T U R N     T O   C A L L E R                      *
*---------------------------------------------------------------------*
         SPACE
RETURN   DS    ØH
         EXEC  CICS RETURN
RETURN_2 DS    ØH                      ISSUE INFO-MESSAGE-NBR 2
         MVC   EIS_MSG,BLANK
         MVC   EIS_MSG(L'MSGØØ2),MSGØØ2
         MVC   EIS_MSG+49(4),EIBTRMID  TERMINAL-ID
         BAL   R9,MESSAGE              WRITE MESSAGE
         B     RETURN
RETURN_3 DS    ØH                      ISSUE INFO-MESSAGE-NBR 3
         MVC   EIS_MSG,BLANK
         MVC   EIS_MSG(L'MSGØØ3),MSGØØ3
         BAL   R9,MESSAGE              WRITE MESSAGE
         B     RETURN
RETURN_4 DS    ØH                      ISSUE INFO-MESSAGE-NBR 4
         CP    EIBTIME,=PL4'Ø6ØØØØ'    CURRENT-TIME ABOVE 6 O'CLOCK ?
         BNH   RETURN_5                NO...WRITE INFORMATION MESSAGE
         MVC   EIS_MSG,BLANK
         MVC   EIS_MSG(L'MSGØØ4),MSGØØ4
         BAL   R9,MESSAGE              WRITE MESSAGE
         EXEC  CICS START TRANSID  ('NCWA')                            *
                          TIME     (ØØØØØØ)                            *
                          RESP     (EIS_RESP)
         CLC   EIS_RESP,DFHRESP(NORMAL) ANY ERRORS DETECTED ?
         BNE   RETURN_3                YES...ISSUE INFORMATION-MESSAGE
         B     RETURN
RETURN_5 DS    ØH                      ISSUE INFO-MESSAGE-NBR 5
         MVC   EIS_MSG,BLANK
         MVC   EIS_MSG(L'MSGØØ5),MSGØØ5
         BAL   R9,MESSAGE              WRITE MESSAGE
         B     RETURN
         EJECT
*----------------------------------------------------------------------*
*             MESSAGE  -  WRITE-ROUTINE                                *
*----------------------------------------------------------------------*
         SPACE
MESSAGE  DS    ØH
         EXEC  CICS WRITE OPERATOR TEXT (EIS_MSG)                      *
                                   RESP (EIS_RESP)
         BR    R9                      RETURN TO CALLER
         EJECT
*----------------------------------------------------------------------*
*             DEFINITION + LITERAL POOL                                *
*----------------------------------------------------------------------*
         SPACE 1
BLANK    DC    CL256' '
         SPACE 1
DAYTAB   DS   ØCL1Ø                    DAY-DESCRIPTION-TABLE
         DC    CL1Ø'Sunday    '        DAYTAB (Ø) = SUNDAY
         DC    CL1Ø'Monday    '        DAYTAB (1) = MONDAY
         DC    CL1Ø'Tuesday   '        DAYTAB (2) = TUESDAY
         DC    CL1Ø'Wednesday '        DAYTAB (3) = WEDNESDAY
         DC    CL1Ø'Thursday  '        DAYTAB (4) = THURSDAY
         DC    CL1Ø'Friday    '        DAYTAB (5) = FRIDAY
         DC    CL1Ø'Saturday  '        DAYTAB (6) = SATURDAY
MONTHTAB DS   ØCL9                     MONTH-DESCRIPTION-TABLE
         DC    CL9'January   '         MONTHOFYEAR (Ø1) = JANUARY
         DC    CL9'February  '         MONTHOFYEAR (Ø1) = FEBRUARY
         DC    CL9'March     '         MONTHOFYEAR (Ø1) = MARCH
         DC    CL9'April     '         MONTHOFYEAR (Ø1) = APRIL
         DC    CL9'May       '         MONTHOFYEAR (Ø1) = MAY
         DC    CL9'June      '         MONTHOFYEAR (Ø1) = JUNE
         DC    CL9'July      '         MONTHOFYEAR (Ø1) = JULY
         DC    CL9'August    '         MONTHOFYEAR (Ø1) = AUGUST
         DC    CL9'September '         MONTHOFYEAR (Ø1) = SEPTEMBER
         DC    CL9'October   '         MONTHOFYEAR (Ø1) = OCTOBER
         DC    CL9'November  '         MONTHOFYEAR (Ø1) = NOVEMBER
         DC    CL9'December  '         MONTHOFYEAR (Ø1) = DECEMBER
HEXØØ    DC    XL4'Ø'                  ERASE OR COMPARISON-VALUE
MSGØØ1   DC    C'CSCWAUPD-ØØ1 Refresh of CWA-Fields is being processed'
MSGØØ2   DC    C'CSCWAUPD-ØØ2 Refresh of CWA-Fields from terminal xxxx*
               denied'
MSGØØ3   DC    C'CSCWAUPD-ØØ3 Recursive start of next midnight-process*
               ing impossible'
MSGØØ4   DC    C'CSCWAUPD-ØØ4 Refresh of CWA-Fields will be started at*
               midnight'
MSGØØ5   DC    C'CSCWAUPD-ØØ5 Recursive refresh of CWA-Fields ignored'
SYSIDTAB DS    ØCL5                    START OF SYSID-BESCHR.-TABELLE
         DC    AL1(CWA$PROD)           CICS-ID C'P' = PROD
SYSPROD  DC    CL4'PROD'               CICS-ID C'P' = PROD
         DC    AL1(CWA$TEST)           CICS-ID C'T' = TEST
SYSTEST  DC    CL4'TEST'               CICS-ID C'T' = TEST
         DC    AL1(CWA$VPRD)           CICS-ID C'V' = VORPROD
SYSVPRD  DC    CL4'VPRD'               CICS-ID C'V' = VORPROD
         DC    AL1(CWA$BOST)           CICS-ID C'S' = SYSTEM-CICS
SYSBOST  DC    CL4'BOST'               CICS-ID C'S' = SYSTEM-CICS
SYSIDTBE EQU   *
         LTORG
         EQUREG                        REGISTER EQUATES
CWAPTR   EQU   R8                      REGISTER FOR CWA-DSECT
         END   CSCWAUPD
```

For applications that will use the CWA, the date and the time-of-day in the CWA for the exit program CSXPCFTC are always kept current.

```
 CSXPCFTC TITLE 'XPCFTCH : CICS USER-EXIT IN DFHPCP'
         SPACE 1
*---------------------------------------------------------------------
*        MODULE MUST BE LINKED   AMODE=31,RMODE=ANY
*---------------------------------------------------------------------
         SPACE 1
***********************************************************************
*              XPCFTCH : CICS USER-EXIT IN DFHPCP                    *
***********************************************************************
*          THIS EXIT STORES THE ADDRESS OF THE PPT ENTRY INTO        *
*                    THE FIELD TPPTADR OF THE TCTUA                  *
*          STORE WILL ONLY BE DONE IF PROGRAM CONTROL ACTIVATES A    *
*          NLV USER PROGRAM (CI.... ) IN THE HIGHEST LOGICAL LEVEL (Ø)*
*          AND FACILITY IS TERMINAL CONTROL.                          *
*          UPDATE TIME VALUE WITHIN CWA.                              *
*        ---- ABEND-CODES ------------------- *
*                 ABENDØØ1         INVALID GLOBAL WORK AREA LENGTH    *
***********************************************************************
         SPACE 3
***********************************************************************
*               P R O G R A M - C H A N G E S                         *
***********************************************************************
         EJECT
DFHUEXIT TYPE=EP,ID=XPCFTCH
         EJECT
         DFHUEXIT TYPE=XPIENV
         EJECT
         COPY  DFHPCUE
         EJECT
         COPY  DFHSAIQY
         EJECT
       ++INCLUDE CSGWA
         EJECT
       ++INCLUDE CSCWAA
         EJECT
CSXPCFTC CSECT
CSXPCFTC AMODE ANY
CSXPCFTC RMODE ANY
         SAVE  (14,12)             SAVE CALLING PROGRAM'S REGISTERS
         LR    BASEREG,R15         SET UP USER EXIT PGM BASE REGISTER
         USING CSXPCFTC,BASEREG
         LR    R2,R1               SET UP PARAMETER LIST ADDR
         USING DFHUEPAR,R2
         B     START
         DC    CL16'*** CSXPCFTC ***'
DC    CL8'&SYSDATE'
         DC    CL8'&SYSTIME'
START    DS    ØH
         ICM   DSECTREG,B'1111',UEPGAA  GET ADDR OF GLOBAL WORK AREA
         USING GWADSECT,DSECTREG
         BZ    ABENDØØ1                 ZERO...GLOBAL WORK AREA NOT FND
         L     R15,UEPGAL               GET ADDR OF GLOBAL WORKAREA LEN
         CLC   =Y(GWADSECT_E-GWADSECT),Ø(R15) GWA-LENGTH OK ?
BNE   ABENDØØ1                 MISMATCH...ANYTHING GOES WRONG
         ICM   CWAPTR,B'1111',GWA_CWA   GET CWA-ADDRESS
         USING CWADSECT,CWAPTR          ESTABLISH DSECT
         L     XPI_REG,UEPXSTOR         SET UP ADDRESSING FOR XPI
         USING DFHSAIQ_ARG,XPI_REG      MAP PARAMETER LIST
         L     R13,UEPSTACK             ADDRESS KERNEL STACK
         DFHSAIQX CALL,                                               *
               CLEAR,                                                 *
               IN,                                                    *
               FUNCTION(INQUIRE_SYSTEM),                              *
               OUT,                                                   *
               TIMEOFDAY(GWA_XPCFTCH_TIMEP),                          *
               RESPONSE(*),                                           *
               REASON(*)
*
         CLI   SAIQ_RESPONSE,SAIQ_OK    GETMAIN SATISFIED ?
         BNE   RETURN_PURG              NO...ABEND THIS TASK
         MVC   GWA_XPCFTCH_TIME,MASK    TIME VALUE HH:MM
         ED    GWA_XPCFTCH_TIME,GWA_XPCFTCH_TIMEP
*                                       TIME OF DAY X'HHMMSSTC'
         MVC   CWAZEIT,GWA_XPCFTCH_TIME+1
         ZAP   CWAZEITP,GWA_XPCFTCH_TIMEP
*                                       TIME OF DAY X'HHMMSSTC'
RETURN_NORM    DS    ØH
         L     R13,UEPEPSA
         L     R12,UEPCRCA              ADDRESS OF CURRENT RETURN-CODE
         CLC   Ø(2,R12),=H'Ø'           CURRENT RETURN-CODE IS ZERO ?
         BE    RETNORM_1ØØØ              YES: RETURN WITH RC=ZERO
         LH    R15,Ø(R12)                NO : RETURN WITH CURRENT RC
         RETURN (14,12),RC=(15)
RETNORM_1ØØØ   DS    ØH
         RETURN (14,12),RC=UERCNORM
RETURN_PURG    DS    ØH
         L     R13,UEPEPSA
         RETURN (14,12),RC=UERCPURG
         EJECT
*---------------------------------------------------------------------*
*              MESSAGE  -  WRITE-ROUTINE                              *
*---------------------------------------------------------------------*
         SPACE
MESSAGE  DS    ØH
         WTO   MF=(E,(R15))        ISSUE MESSAGE
         BR    SUBREG              RETURN TO CALLER
         EJECT
*---------------------------------------------------------------------*
*              ERROR-ROUTINE                                          *
*---------------------------------------------------------------------*
         SPACE
ABENDØØ1 EQU   *                   INVALID GLOBAL WORK AREA LENGTH
         LA    R15,MSGØØ1          MESSAGE-AREA
         BAS   SUBREG,MESSAGE      WRITE MESSAGE
         LA    R15,1               ABEND-NUMBER
         B     ABEND
ABEND    EQU   *
         ABEND (R15),,STEP
         EJECT
*---------------------------------------------------------------------*
*              MESSAGES                                               *
*---------------------------------------------------------------------*
         SPACE
MSGØØ1   WTO  'CSXPCFTC-ØØ1 invalid global work area length detected',*
               MF=L,                                                  *
               ROUTCDE=11
MSGØØ1L  EQU   *-MSGØØ1            MESSAGE-LENGTH
         EJECT
BLANK    DC    CL4'    '
MASK     DC    X'FØ212Ø7A2Ø2Ø7A2Ø2Ø2Ø'
         SPACE 3
CWAPTR   EQU   R4
XPI_REG  EQU   R5
DSECTREG EQU   R8                   DSECT-REGISTER FOR GLOBAL WORK AREA
SUBREG   EQU   R9                   BAS-SUBROUTINE-REGISTER
BASEREG  EQU   R11                      BASE-REGISTER
         SPACE 3
         DFHREGS
         END
```

And now let’s have a look at the way Time Machine and the CWA can be used:

```
*ASM XOPTS(CICS)
CSTIME   TITLE 'Interface between CICS Time Machine and CWA refresh'
         SPACE
*  Code start                                                        *
*  Macro DFHEIENT is used to obtain working storage.                 *
*  In this program a single register for code (R4) and               *
*  storage (R12) is sufficient.                                      *
         SPACE
DFHEISTG DSECT
         SPACE
*              D E F I N I T I O N S                                 *
         SPACE
RESPONSE DS    F
         SPACE
*              M A I N  P R O G R A M                                *
         SPACE
CSTIME   DFHEIENT CODEREG=(R4),DATAREG=(R12)
CSTIME   RMODE  ANY
CSTIME   AMODE  31
         B     MAIN
         SPACE
         DC    C'CICS Time Machine Interface'
         DC    C'NAME : '
         DC    C'CSTIME'
         DC    C' DATE AND TIME ASSEMBLED : '
         DC    C'&SYSDATC',C','
         DC    C'&SYSTIME '
         DC    C'&SYSPARM '
         SPACE
*  SECTION NAME : MAIN                                               *
*  FUNCTION     : Controls flow of program                           *
*  CALLED BY    : The CTMC transaction                               *
*  CALLS        : Pgm. TMCSCRN and CSCWAUPD                          *
*  RETURN       : EXEC CICS RETURN                                    *
SPACE
MAIN     DS    ØH
         EXEC CICS LINK PROGRAM('TMCSCRN')                             *
               RESP(RESPONSE)
         SPACE
         CLC   RESPONSE,DFHRESP(NORMAL)
         BE    MAIN_1ØØ
         SPACE
         EXEC CICS ABEND ABCODE('TMMA')
         SPACE
         B     RETURN
         SPACE
MAIN_1ØØ DS    ØH
         SPACE
         EXEC CICS LINK PROGRAM('CSCWAUPD')                            *
               RESP(RESPONSE)
         SPACE
         CLC   RESPONSE,DFHRESP(NORMAL)
         BE    RETURN
         SPACE
         EXEC CICS ABEND ABCODE('CWAU')
         SPACE
RETURN   DS    ØH
         EXEC  CICS RETURN
         EJECT
*              L I T E R A L S                                        *
         SPACE
         LTORG
         EJECT
*              R E G I S T E R  E Q U A T E S                         *
         SPACE
         EQUREG
         SPACE
         DC    C' '
         END   CSTIME
```

What happens? The transaction CTMC doesn’t call Time Machine directly, program CSTIME is called! In this program Time Machine (TMCSCRN) is called via ‘EXEC CICS LINK PROGRAM(TMCSCRN)’. After this, program CSCWAUPD will be called, also via ‘EXEC CICS LINK ....’.

Claus Reis
CICS Systems Programmer
Nuernberger Lebensversicherung AG (Germany)

Maintenance of CICS DB2 entries and transactions – part 2

This month we conclude the code for a tool for the administration of DB2 entries and transactions. The REXX EXECs store information from CSD files into DB2 tables, prepare jobs for migration purposes, and use generated ISPF tables allowing the online update of CICS resource definitions.

RCTS1

```
)CM ---------------------------------------------------------
)CM LOAD DB2 ENTRIES AND TRANSACTIONS
)CM ---------------------------------------------------------
//&USERID.X JOB MSGCLASS=X,TIME=1440,REGION=4M,NOTIFY=&USERID
//STEP0001 EXEC PGM=IEFBR14
//FILEDEL  DD DSN=&FOROUT,DISP=(MOD,DELETE,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
/*
//STEP0002  EXEC PGM=DFHCSDUP,
//          PARM='CSD(READONLY),NOCOMPAT',COND=(4,LT)
//STEPLIB   DD DSN=CICSTS12.CICS.SDFHLOAD,DISP=SHR
//DFHCSD    DD DISP=SHR,DSN=&CSD
//FOROUT    DD DSN=&FOROUT,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(15,1)),
//          DCB=(RECFM=F,LRECL=460,BLKSIZE=460)
//SYSPRINT  DD SYSOUT=*
//SYSIN    DD *
    EXTRACT GROUP(DB2*) USERPROGRAM(DFH$FORA) OBJECTS
/*
//STEP0003 EXEC DSNUPROC,PARM='&DSN8SSID,RCTU1',COND=(4,LT)
//DSNTRACE DD SYSOUT=*
//SORTLIB  DD DISP=SHR,DSN=SYS1.SORTLIB
//SORTOUT DD DSN=&USERID..SORTOUT.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(2000,200),,,ROUND),
//     UNIT=3390
//SYSUT1 DD DSN=&USERID..SYSUT1.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(2000,200),,,ROUND),
//     UNIT=3390
//SORTWK01 DD DSN=&USERID..SORTWK01.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(100,10),,,ROUND),
//     UNIT=3390
//SORTWK02 DD DSN=&USERID..SORTWK02.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(1ØØ,1Ø),,,ROUND),
//     UNIT=339Ø
//SORTWKØ3 DD DSN=&USERID..SORTWKØ3.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(1Ø,1Ø),,,ROUND),
//     UNIT=339Ø
//SORTWKØ4 DD DSN=&USERID..SORTWKØ4.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(1ØØ,1Ø),,,ROUND),
//     UNIT=339Ø
//SYSDISC DD DSN=&USERID..SYSDISC.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(25,2),,,ROUND),
//     UNIT=339Ø
//SYSERR DD DSN=&USERID..SYSERR.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(25,2),,,ROUND),
//     UNIT=339Ø
//SYSMAP DD DSN=&USERID..SYSMAP.PRIV,
//     DISP=(NEW,DELETE,CATLG),
//     SPACE=(16384,(25,2),,,ROUND),
//     UNIT=339Ø
//SYSRECØØ DD DISP=SHR,DSN=&FOROUT
//SYSIN    DD *
LOAD DATA
  INDDN SYSRECØØ
  LOG NO
  REPLACE
  INTO TABLE CICS.DB2ENTRY
  WHEN(1:4)='DB2E' (
    DB2ENTRY             POSITION(ØØ5:Ø12)    CHAR(8),
    RDOGROUP             POSITION(Ø13:Ø2Ø)    CHAR(8),
    DESCRIPTION          POSITION(Ø21:Ø78)    CHAR(58),
    TRANSID              POSITION(Ø79:Ø82)    CHAR(4),
    ACCOUNTREC           POSITION(Ø83:Ø86)    CHAR(4),
    AUTHID               POSITION(Ø87:Ø94)    CHAR(8),
    AUTHTYPE             POSITION(Ø95:1ØØ)    CHAR(6),
    DROLLBACK            POSITION(1Ø1:1Ø3)    CHAR(3),
    PLAN                 POSITION(1Ø4:111)    CHAR(8),
    PLANEXITNAME         POSITION(112:119)    CHAR(8),
    PRIORITY             POSITION(12Ø:124)    CHAR(5),
    PROTECTNUM           POSITION(125:128)    CHAR(4),
    THREADLIMIT          POSITION(129:132)    CHAR(4),
    THREADWAIT           POSITION(133:136)    CHAR(4)
    )
LOAD DATA
  INDDN SYSRECØØ
  LOG NO
  RESUME YES
  INTO TABLE CICS.DB2TRAN
  WHEN(1:4)='DB2T' (
    DB2TRAN              POSITION(005:012)   CHAR(8),
    RDOGROUP             POSITION(013:020)   CHAR(8),
    DESCRIPTION          POSITION(021:078)   CHAR(58),
    ENTRY                POSITION(079:086)   CHAR(8),
    TRANSID              POSITION(087:090)   CHAR(4)
    )
/*
//STEP0004 EXEC DSNUPROC,SYSTEM=&DSN8SSID,UID='RCTU2',UTPROC='',
//         COND=(4,LT)
//DSNUPROC.SYSIN    DD  *
REPAIR LOG NO
  SET TABLESPACE DSNDB04.TSCICS01 NOCOPYPEND
//
```

RCTS2

```
)CM ---------------------------------------------------------------
)CM OPEN DFHCSD FILE
)CM DEFINE DB2 ENTRIES AND/OR TRANSACTIONS
)CM ---------------------------------------------------------------
//&USERID.X JOB MSGCLASS=X,TIME=1440,REGION=4M,NOTIFY=&USERID
//STEP0001  EXEC PGM=IEFBR14
// F &CICSNAME,'CEMT SET FILE(DFHCSD) CLO DIS'
/*
//STEP0002  EXEC PGM=DFHCSDUP,
//          PARM='CSD(READWRITE),NOCOMPAT',COND=(4,LT)
//STEPLIB   DD DSN=CICSTS12.CICS.SDFHLOAD,DISP=SHR
//DFHCSD    DD DISP=SHR,DSN=&CSD
//SYSUT1    DD UNIT=SYSDA,SPACE=(1024,(100,100))
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
```

RCTS3

```
)CM ---------------------------------------------------------------
)CM DEFINE DB2 ENTRIES AND/OR TRANSACTIONS (CONTINUE 1)
)CM ---------------------------------------------------------------
&STRCMD
```

RCTS4

```
)CM ---------------------------------------------------------------
)CM OPEN DFHCSD FILE
)CM ---------------------------------------------------------------
//&USERID.Y JOB MSGCLASS=X,TIME=1440,REGION=4M,NOTIFY=&USERID
//STEP0001  EXEC PGM=IEFBR14
// F &CICSNAME,'CEMT SET FILE(DFHCSD) OPE ENA'
/*
```

RCTS5

```
)CM ----------------------------------------------------------
)CM CEDA DELETE/INSERT/ALTER DB2E AND/OR DB2T
)CM ----------------------------------------------------------
//STEPØØØ2  EXEC PGM=IEFBR14,COND=(4,LT)
```

SQLISPF

```
//SYSADM1 JOB MSGCLASS=X,REGION=4M,NOTIFY=&SYSUID
//************************************************************
//* Customized version of SQLSPDB published in the January   *
//*      1998 issue of DB2 Update                            *
//************************************************************
//JOBLIB   DD  DISP=SHR,DSN=SYS1.DSN51Ø.SDSNEXIT
//         DD  DISP=SHR,DSN=DSN51Ø.SDSNLOAD
//PC      EXEC PGM=DSNHPC,PARM='HOST(ASM),SOURCE',REGION=4Ø96K
//DBRMLIB  DD DSN=SYSADM.DBRMLIB5.DATA(SQLISPF),DISP=SHR
//STEPLIB  DD  DISP=SHR,DSN=SYS1.DSN51Ø.SDSNEXIT
//         DD  DISP=SHR,DSN=DSN51Ø.SDSNLOAD
//SYSCIN   DD  DSN=&&DSNHOUT,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(8ØØ,(1Ø,1Ø))
//SYSLIB DD    DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  SPACE=(8ØØ,(1Ø,1Ø),,,ROUND),UNIT=SYSDA
//SYSIN    DD  *
         TITLE ' REXX INTERFACE WITH DB2 - CAF'
***********************************************************************
* FUNCTION : THE RESULTS FROM THE SPECIFIED ESQL QUERY ARE RETURNED   *
*            AS REXX VARIABLES. THE VARIABLE NAMES ARE THOSE          *
*            ATTRIBUTES RESULTING FROM THE ESQL QUERY.                *
*            IF A NON-SELECT SQL STATEMENT IS ENTERED NO OUTPUT IS    *
*            RETURNED, BUT THE COMMAND IS EXECUTED. A MAXIMUM OF 9    *
*            FUNCTIONS (LIKE -SUM- OR -COUNT-) ARE ALLOWED.           *
*            IN CASE OF A NON-SELECT SQL COMMAND, YOU CAN HAVE ONE    *
*            (1) HOSTVARIABLE IN THE SQL STATEMENT (REPRESENTED BY A  *
*            ?). PUT THE VALUE OF THE HOSTVARIABLE IN REXX VARIABLE   *
*            HOSTVAL. A HOSTVARIABLE IN A SELECT SQL STATEMENT WILL   *
*            RESULT IN SQLCODE -313.                                  *
*            THIS PROGRAM USES THE DB2 CAF INTERFACE AND SO CAN       *
*            EXECUTE OUTSIDE THE DB2 ENVIRONMENT (HOWEVER, SQL MUST   *
*            BE AVAILABLE).                                           *
*            THIS PROGRAM IS WRITTEN FOR THE SP/DB DB2 AIDS ONLY.     *
*            NO GUARANTEES ARE MADE!!!!!!!!!                          *
*            THERE IS A KNOWN BUG IN THIS PROGRAM! DO NOT ATTEMPT     *
*            TO PERFORM ARITHMETIC FUNCTIONS LIKE SUM(COLUMN)/1Ø IN   *
*            THIS PROGRAM. IT WILL MESS UP THE PRECISION!             *
*            TRY AND FIX IT. IF YOU DO, MAIL IT TO ME!                *
* PLANNAME : SQLISPF                                                  *
* INPUT VARIABLE   : <DB2V>     : DB2 SYSTEM (DEFAULT IS DSN)         *
*                  : <SQLQUERY> : SPECIFIC SQL QUERY                  *
*                  : <HOSTVAL>  : OPTIONAL HOST VARIABLE              *
* OUTPUT VARIABLES : COLUMNNAME(J).I (I = ROW)                        *
*                    <_VN.>.J   : COLUMN NAMES                        *
*                    <_VN.Ø>    : # OF COLUMNS                        *
*                    <_NROWS>   : # OF SELECTED ROWS                  *
*                    <_VN1> THRU <_VN9> FOR EVERY FUNCTION THATS USED *
*                    <_REASON>  : REASONCODE (ONLY IF BAD CONNECT)    *
* RETURNCODES      : Ø    -   OK                                      *
*                    <N>  -   REXX OR SQL CODE                        *
*                    99   -   BAD CONNECT TO DB2                      *
***********************************************************************
SQLISPF  CSECT
         REGEQ
         BEGIN SAVE=SA,BASE=(R11,R12)
         B     SA_END             jump over save-area
SA       DS    18A
SA_END   DS    ØH
         LOAD  EP=IRXEXCOM                LOAD IRXEXCOM
         ST    RØ,AIRXEXCOM               SAVE ENTRYPOINT
* OBTAIN DB2 SYSTEM
* INITIALIZE IT TO TEST DB2 (DSN)
* IF NO DB2V PARM, TAKE IT AS DEFAULT
         MVC   SSID,=C'DSN'
         LA    R5,IRX_SHVBLOCK
         USING SHVBLOCK,R5
         MVI   SHVCODE,SHVFETCH
         MVC   SHVBUFL,=A(L'DB2V)
         MVC   SHVVALA,=A(DB2V)
         MVC   VN,=C'DB2V '
         BAL   R14,GETVNL            GET LENGTH OF <VN>
         ST    RØ,SHVNAML            L(<VN>)
         MVC   SHVNAMA,=A(VN)        A(<VN>)
         L     R15,AIRXEXCOM
         CALL  (15),(IRX_IRXEXCOM,Ø,Ø,IRX_SHVBLOCK),VL
         LTR   R15,R15               REXX RETURN CODE
         BNZ   NODBPRM               PARAMETER MISSING, TAKE DEFAULT
         L     R1,SHVVALL            L(PARAMETER)
         STH   R1,DB2VAR
         MVC   SSID(4),DB2V
NODBPRM  EQU   *
* OBTAIN <SQLQUERY>
         LA    R5,IRX_SHVBLOCK
         USING SHVBLOCK,R5
         MVI   SHVCODE,SHVFETCH
         MVC   SHVBUFL,=A(L'SQLQUERY)
         MVC   SHVVALA,=A(SQLQUERY)
         MVC   VN,=C'SQLQUERY '
         BAL   R14,GETVNL            GET LENGTH OF <VN>
         ST    RØ,SHVNAML            L(<VN>)
         MVC   SHVNAMA,=A(VN)        A(<VN>)
         L     R15,AIRXEXCOM
         CALL  (15),(IRX_IRXEXCOM,Ø,Ø,IRX_SHVBLOCK),VL
         LTR   R15,R15               REXX RETURN CODE
         BNZ   BLOWREXX              PARAMETER ERROR
         L     R1,SHVVALL            L(PARAMETER)
         STH   R1,SELECT
*        CONNECT TO DB2
         CALL  DSNALI,(CONNECT,SSID,TECB,SECB,RIBPTR),VL,MF=(E,CAFCALL)
         ST    R15,RETCODE
         ST    RØ,REASCODE
         CLC   RETCODE,=F'Ø'
         BNE   BLOW                  DB2 CONNECT ERROR
         CALL  DSNALI,(OPEN,SSID,PLAN),VL,MF=(E,CAFCALL)
         ST    RØ,REASCODE
         ST    R15,RETCODE
         CLC   RETCODE,=F'Ø'
         BNE   BLOWCON               DB2 PLAN ERROR
* OBTAIN POSSIBLE HOST VARIABLE <HOSTVAR>
         LA    R5,IRX_SHVBLOCK
         USING SHVBLOCK,R5
         MVI   SHVCODE,SHVFETCH
         MVC   SHVBUFL,=A(L'HOSTVAL)
         MVC   SHVVALA,=A(HOSTVAL)
         MVC   VN,=C'HOSTVAL '
         BAL   R14,GETVNL            GET LENGTH OF <VN>
         ST    RØ,SHVNAML            L(<VN>)
         MVC   SHVNAMA,=A(VN)        A(<VN>)
         L     R15,AIRXEXCOM
         CALL  (15),(IRX_IRXEXCOM,Ø,Ø,IRX_SHVBLOCK),VL
         LTR   R15,R15               TEST REXX RETURN CODE
         BNZ   DB2ST                 <> Ø, NO HOSTVAR SUPPLIED
         MVC   HOSTSW,YES            YES, THERE IS
         L     R1,SHVVALL            L(PARAMETER)
         STH   R1,HOSTVAR            STORE LENGTH
* DETERMINE NO OF COLUMNS IN TABLE
DB2ST    DS    ØH
         LA    R9,SQL_CA
         USING SQLDSECT,R9
         LA    R8,SQL_DA
         USING SQLDA,R8
* DUMMY PREPARE
         EXEC  SQL PREPARE S1 FROM :SELECT
         BAL   R14,CHECK_SQL
         MVC   SQLN,=H'Ø'            RETURN COUNT ONLY
* DESCRIBE
         EXEC  SQL DESCRIBE S1 INTO :SQL_DA
         BAL   R14,CHECK_SQL
* ANALYSE DESCRIPTOR AND ACQUIRE STORAGE
* <SQLD>: NO. OF COLUMNS
         LH    R2,SQLD
         MVC   RC,=F'Ø'              RESET RETURN CODE
         LTR   R2,R2
         BZ    A5ØØ                   :NON-SELECT
         LH    R3,SQLD
         MH    R2,=AL2(SQLVARN_SIZE)
         LA    R2,(SQLVAR-SQLDA)(R2)            +L(FIXED HDR)
* R2: TOTAL COLUMN SIZE
         ST    R2,COLSIZE
         GETMAIN EU,LV=(2),A=A_SQLDA
         L     R8,A_SQLDA
* PERFORM DESCRIBE WITH CORRECT LENGTH
         STH   R3,SQLD
         STH   R3,SQLN
         ST    R2,SQLDABC
* DESCRIBE
         EXEC  SQL DESCRIBE S1 INTO :SQLDA
         BAL   R14,CHECK_SQL
* BUILD ROW BUFFER
         LH    R6,SQLD             NO OF OCCURRENCES (COLUMNS)
         LTR   R6,R6
         BZ    EOP                 NO ENTRIES
         LA    R7,SQLVAR
         USING SQLVARN,R7
         SR    R4,R4               ZEROIZE BUFFER SIZE ACCUMULATOR
* OUTPUT NO OF COLUMNS (<_VN.Ø>)
         CNOP  Ø,4
         CVD   R6,D
         MVC   WK,=X'FØ2Ø2Ø2Ø2Ø2Ø2Ø2Ø'
         ED    WK,D+4
         LA    R3,WK                  A(DATA)
         MVC   VL,=A(L'WK)           L(DATA)
         MVC   VNINDEX,=C'.Ø'
         MVC   VLINDEX,=A(2)
         MVC   VN,=C'_VN '
* SET DATA INTO RESS VARIABLE
         BAL   R14,SETVAR
A23Ø     DS    ØH
* COLUMN TYPE
         BAL   R14,GET_TYPE
* R2: DATA WIDTH
* R3: FIELD SIZE
* R1Ø: DSECT_TYPE ENTRY
         USING DSECT_TYPE,R1Ø
         LA    R4,2(R3,R4)          ACCUMULATE TOTAL BUFFER SIZE
         AP    INDEX,=P'1'          INCREMENT INDEX
* CONVERT INDEX TO FORM: .N
         MVC   VNINDEX,=X'4Ø2Ø2Ø2Ø2Ø2Ø212Ø'
         LA    R1,VNINDEX+7
         EDMK  VNINDEX,INDEX
         BCTR  R1,Ø
         MVI   Ø(R1),C'.'
         MVC   VNINDEX,Ø(R1)
         LA    RØ,VNINDEX+L'VNINDEX
         SR    RØ,R1
         ST    RØ,VLINDEX           L(INDEX)
         MVC   VN,=C'_VN '          NAME PREFIX
         LA    R3,SQLNAME+2         COLUMNS NAME
         LH    RØ,SQLNAME
* NEW CODE FOR UNNAMED COLUMNS (EG COUNT(*) )
* TEST WHETHER  NULL COLUMN NAME
         LH    RØ,SQLNAME           LOAD COLUMN NAME
         LTR   RØ,RØ
         BNZ   A231                 NO COLUMN NAME
         MVC   SQLNAME+2(8),=CL8'_VN'
         AP    VNCT,=P'1'
         UNPK  SQLNAME+5(1),VNCT
         OI    SQLNAME+5,C'Ø'
         LA    RØ,5
         STH   RØ,SQLNAME
         LH    RØ,SQLNAME
A231     EQU   *
         ST    RØ,VL
* SET DATA INTO REXX VARIABLE
         BAL   R14,SETVAR
         LA    R7,SQLVARN_SIZE(R7)
         BCT   R6,A23Ø
* END OF SCAN PHASE, ALLOCATE DATA BUFFER
* R5: TOTAL BUFFER SIZE
         ST    R4,BUFFSIZE
         GETMAIN EU,LV=(4),A=A_DBUF
* COMPLETE SQLDA
         L     R5,A_DBUF            PTR(DATA BUFFER)
         LH    R6,SQLD              # OF OCCURRENCES COLUMNS
         LA    R7,SQLVAR            REINIT POINTER
* COLUMN TYPE
A24Ø     BAL   R14,GET_TYPE
* R3: FIELD SIZE
* R1Ø: DSECT_TYPE ENTRY
         ST    R5,SQLIND            A(INDICATOR), IF NEEDED
         LA    R5,2(R5)             UPDATE PTR
         ST    R5,SQLDATA           A(HOST VARIABLE)
         AR    R5,R3                UPDATE PTR
         LA    R7,SQLVARN_SIZE(R7)
         BCT   R6,A24Ø
* RETRIEVE RECORDS
* DECLARE CURSOR
         EXEC  SQL DECLARE CSR CURSOR FOR S1
         BAL   R14,CHECK_SQL
* OPEN CURSOR
         EXEC  SQL OPEN CSR
         BAL   R14,CHECK_SQL
         ZAP   INDEX,=P'Ø'        INITIALIZE ROW INDEX
A4ØØ     DS ØH
* FETCH ROW
         EXEC  SQL FETCH CSR USING DESCRIPTOR :SQLDA
         CLC   SQLCODE,=F'1ØØ'
         BE    EOD                END OF DATA
         AP    INDEX,=P'1'        INCREMENT INDEX
*  CONVERT INDEX TO FORM: .N
         MVC   VNINDEX,=X'4Ø2Ø2Ø2Ø2Ø2Ø212Ø'
         LA    R1,VNINDEX+7
         EDMK  VNINDEX,INDEX
         BCTR  R1,Ø
         MVI   Ø(R1),C'.'
         MVC   VNINDEX,Ø(R1)
         LA    RØ,VNINDEX+L'VNINDEX
         SR    RØ,R1
         ST    RØ,VLINDEX
         LH    R6,SQLD               NO OF OCCURRENCES (COLUMNS)
         LA    R7,SQLVAR             REINITIALIZE POINTER
* WRITE TABLE ROW
A41Ø     DS    ØH
         USING SQLVARN,R7
* DEFAULT (NULL) VALUE
         LA    R2,1
         LA    R3,=C'-'
         L     R1,SQLIND             A(INDICATOR FIELD)
         LH    RØ,Ø(R1)
         CH    RØ,=H'-1'
         BE    A42Ø                  NO DATA
         BAL   R14,GET_TYPE
* R2: DATA WIDTH
* R3: FIELD SIZE
* R1Ø: DSECT_TYPE ENTRY
         MVC   A_DSADDR,DS_ADDR
         L     R15,A_DSADDR          A(ROUTINE)
         BALR  R14,R15
* R3: A(FORMATTED FIELD)
* R2: L(FORMATTED FIELD)
A42Ø     ST    R2,VL
         LH    R1,SQLNAME            L(NAME)
         LA    RØ,L'VN               MAX(L(NAME))
         CR    R1,RØ
         BNH   *+6
         LR    R1,RØ                 TRUNCATE
         BCTR  R1,Ø                  LC(NAME)
         MVC   VN,VN-1
         MVC   VN,SQLNAME+2          COLUMN NAME
         EX    R1,*-6
* SET DATA INTO REXX VARIABLE
         BAL   R14,SETVAR
         LA    R7,SQLVARN_SIZE(R7)
         BCT   R6,A41Ø               GET NEXT COLUMN IN ROW
         B     A4ØØ                  GET NEXT ROW
A5ØØ     DS    ØH                    PROCESS NON-SELECT
         CLC   HOSTSW,YES
         BNE   NOVARS
         EXEC  SQL EXECUTE S1 USING :HOSTVAR
         BAL   R14,CHECK_SQL
         B     EOD
NOVARS   DS    ØH
         EXEC  SQL EXECUTE S1
         BAL   R14,CHECK_SQL
EOD      CALL  DSNALI,(CLOSE,SYNC),VL,MF=(E,CAFCALL)
* END OF PROCESSING
EOP      DS    ØH
* OUTPUT # OF ROWS PROCESSED
         MVC   WK,=X'FØ2Ø2Ø2Ø2Ø2Ø2Ø2Ø'
         ED    WK,INDEX
         LA    R3,WK             A(DATA)
         MVC   VL,=A(L'WK)       L(DATA)
         MVC   VLINDEX,=A(Ø)     NO INDEX
         MVC   VN,=C'_NROWS '
* SET DATA INTO REXX VARIABLE
         BAL   R14,SETVAR
* RELEASE ALLOCATED STORAGE
         L     R2,COLSIZE
         LTR   R2,R2
         BZ    NOCOLS
         L     R3,A_SQLDA
         FREEMAIN R,LV=(2),A=(3)
NOCOLS   L     R2,BUFFSIZE
         LTR   R2,R2
         BZ    NOBUF
         L     R3,A_DBUF
         FREEMAIN R,LV=(2),A=(3)
NOBUF    EQU   *
*        DISCONNECT FROM DB2
         CALL  DSNALI,(DISCON),VL,MF=(E,CAFCALL)
GETOUT   EQU   *
         L     R13,4(R13)            RESTORE A(OLD SAVE AREA)
         L     R15,RC                SET RETURN CODE
         EINDE SAVE=SA,RCR=R15
BLOWREXX EQU   *
         MVC   RC,=F'8'
         B     NOBUF
BLOWCON  EQU   *
         MVC   RC,RETCODE
         MVC   CS4BYTE,REASCODE
         BAS   R14,HEXCONV             CONVERT TO EBCDIC
         MVC   REASWORK,CS8BYTE
         LA    R3,REASWORK             A(DATA)
         MVC   VL,=A(L'REASWORK)       L(DATA)
         MVC   VLINDEX,=A(Ø)           NO INDEX
         MVC   VN,=C'_REASON '
* SET DATA INTO REXX VARIABLE
         BAL   R14,SETVAR
         B     NOBUF
BLOW     EQU   *
         MVC   RC,=F'99'
         B     GETOUT
RC       DS    F                     PROGRAM RETURN CODE
*  CAF VARIABLES
HOSTSW   DC    CL1' '
REASWORK DS    CL8
YES      DC    CL1'Y'
SYNC     DC    CL4'SYNC'
CLOSE    DC    CL12'CLOSE       '
OPEN     DC    CL12'OPEN        '
CONNECT  DC    CL12'CONNECT     '
DISCON   DC    CL12'DISCONNECT  '
RETCODE  DS    F
REASCODE DS    F
RIBPTR   DS    F
SECB     DS    F
TECB     DS    F
SSID     DS    CL4
PLAN     DC    CL8'SQLISPF '
LIALI    DS    F
CAFCALL  CALL ,(*,*,*,*,*),VL,MF=L
CAFLEN   EQU  *-CAFCALL
*  END CAF VARIABLES
         TITLE 'SUBROUTINES'
*---------------------------------------------------------------------*
*  SUBROUTINE HEXCONV - CONVERTS ADDRESSES TO PRINTABLE HEX EBCDIC    *
*  INPUT: CS4BYTE       OUTPUT: CS8BYTE                               *
*  CSECT MUST BE 24Ø BYTES OR LONGER                                  *
*---------------------------------------------------------------------*
HEXCONV  DS    ØH
         B     CODING
SAVEHC   DS    18F                     SAVE AREA HEXCONV
CS5BYTE  DS    ØCL5
CS4BYTE  DS    CL4
         DS    C
CS9BYTE  DS    ØCL9
CS8BYTE  DS    CL8
         DS    C
         DC    C'$'
TRTABLE  ORG   *-X'FØ'
         ORG   TRTABLE+X'FØ'
         DC    C'Ø123456789ABCDEF'
CODING   STM   RØ,R15,SAVEHC
         UNPK  CS9BYTE,CS5BYTE
         TR    CS8BYTE,TRTABLE
EXITHC   DS    ØF
         LM    RØ,R15,SAVEHC
         BR    R14
GETVNL   DS    ØH                    DETERMINE ACTUAL LENGTH OF NAME
```

```
* INPUT: <VN> - NAME
* OUTPUT: RØ - L(NAME)
* R15 - A(FIRST BLANK)
         LA    R1,L'VN
         SR    RØ,RØ                 COUNTER
         LA    R15,VN
GETVNL1  CLI   Ø(R15),C' '
         BER   R14                   END FOUND
         AH    RØ,=H'1'
         LA    R15,1(R15)
         BCT   R1,GETVNL1
* RØ: L(NAME). WITHOUT TRAILING BLANKS
         BR    R14
         DS    A
SETVAR   ST    R14,SETVAR-4          SET REXX VARIABLE
* <VN>: VAR1ABLE NAME, PREFIX
* <VNINDEX>: VARIABLE NAME, SUFFIX
* <VLINDEX>: LENGTH (VARIABLE NAME, SUFFIX)
* <VL>: L(VARIABLE DATA)
* R3: A(VARIABLE DATA)
         BAL   R14,GETVNL        GET L(VN)
* RØ: L(VN), R15: A(FIRST BLANK IN <VN>)
         MVC   Ø(L'VNINDEX,R15),VNINDEX
         A     RØ,VLINDEX
         LA    R5,IRX_SHVBLOCK
         USING SHVBLOCK,R5
         ST    RØ,SHVNAML
         MVC   SHVNAMA,=A(VN)
         MVI   SHVCODE,SHVSTORE
         ST    R3,SHVVALA
         MVC   SHVVALL,VL
         L     R15,AIRXEXCOM           A(IRXEXCOM)
         CALL  (15),(IRX_IRXEXCOM,Ø,Ø,IRX_SHVBLOCK),VL
         L     R14,SETVAR-4
         BR    R14                     RETURN
GET_TYPE DS    ØH                      GET COLUMN TYPE AND SIZE(S)
* INPUT:
* DSECT_SQLVARN ENTRY
* OUTPUT:
* R2: DATA WIDTH
* R3: FIELD SIZE
* DSECT TYPE ENTRY
         LA    R1Ø,T_TYPE-DS_L
         USING DSECT_TYPE,R1Ø
GETTYPE1 LA    R1Ø,DS_L(R1Ø)
         CLC   SQLTYPE,DS_TYPE
         BNE   GETTYPE1
* ENTRY FOUND, GET COLUMN-LENGTH
         LH    R2,SQLLEN
         LR    R3,R2                 PRESET DATA FIELD SIZE
         CLC   DS_CODE,=CL2'P'      (PACKED) DECIMAL?
         BNE   GETTYPE2             :NO
         SR    R2,R2
         IC    R2,SQLPRCSN           PRECISION
         LR    R3,R2
         SRL   R3,1                  NO OF DIGIT PAIRS
         LA    R3,1(R3)              TRUE DATA FIELD SIZE
GETTYPE2 CLC   DS_CODE,=CL2'CV'      CHARACTER (VARIABLE)?
         BNE   GETTYPE3             :NO
         LA    R3,2(R3)              ALLOC ROOM FOR LENGTH
GETTYPE3 BR    R14                   RETURN
         TITLE 'CONVERSION ROUTINES'
T_TYPE   DS    ØH                    ALIGNMENT
         DC    H'384',AL1(@CHAR,Ø),CL2'D ',AL4(D_DATE)
         DC    H'385',AL1(@CHAR,Ø),CL2'D ',AL4(D_DATE)
         DC    H'388',AL1(@CHAR,Ø),CL2'T ',AL4(D_TIME)
         DC    H'389',AL1(@CHAR,Ø),CL2'T ',AL4(D_TIME)
         DC    H'392',AL1(@CHAR,Ø),CL2'TS',AL4(D_TIME)
         DC    H'393',AL1(@CHAR,Ø),CL2'TS',AL4(D_TIME)
         DC    H'448',AL1(@CHAR,Ø),CL2'CV',AL4(D_CHARV)
         DC    H'449',AL1(@CHAR,Ø),CL2'CV',AL4(D_CHARV)
         DC    H'452',AL1(@CHAR,Ø),CL2'C ',AL4(D_CHAR)
         DC    H'453',AL1(@CHAR,Ø),CL2'C ',AL4(D_CHAR)
         DC    H'456',AL1(@CHAR,Ø),CL2'CV',AL4(D_CHAR)
         DC    H'457',AL1(@CHAR,Ø),CL2'CV',AL4(D_CHAR)
         DC    H'484',AL1(@NUM,Ø),CL2'P ',AL4(D_DEC)
         DC    H'485',AL1(@NUM,Ø),CL2'P ',AL4(D_DEC)
         DC    H'496',AL1(@NUM,Ø),CL2'I ',AL4(D_INT)
         DC    H'497',AL1(@NUM,Ø),CL2'I ',AL4(D_INT)
         DC    H'5ØØ',AL1(@NUM,Ø),CL2'I ',AL4(D_SIN)
         DC    H'5Ø1',AL1(@NUM,Ø),CL2'I ',AL4(D_SIN)
         DC    H'Ø'                         EOT
D_DATE   EQU   D_CHAR
D_TIME   EQU   D_CHAR
D_CHARV  EQU   D_CHAR
@NULL    EQU   X'Ø1'
@CHAR    EQU   1
@NUM     EQU   2
D_CHAR   DS    ØH                      CHARACTER
         L     R3,SQLDATA              A(DATA)
         LH    R2,SQLLEN               L(DATA)
         CLC   DS_CODE,=C'CV'
         BNER  R14
         LH    R2,Ø(R3)
         LA    R3,2(R3)
         BR    R14
D_DEC    DS    ØH                      PACKED DECIMAL
         L     R3,SQLDATA              A(DATA)
         SR    R1,R1
         IC    R1,SQLLEN               L(DATA), PRECISION
* R1: NO. OF DECIMAL DIGITS
         SRL   R1,1
```

```
* R1: LENGTH CODE OF PACKED FIELD
         EX    R1,ZAP
         B     FMT_DEC                 FORMAT DECIMAL FIELD
D_SIN    DS    ØH                      BINARY SMALL INTEGER (2 BYTES)
         L     R3,SQLDATA              A(DATA) IN R3
         MVC   H2(2),Ø(R3)             MAKE SURE ITS HALFWORD BOUNDARY
         LH    RØ,H2
         CNOP  Ø,4
         CVD   RØ,D                    CONVERT IT TO DECIMAL
         B     FMT_DEC                 FORMAT DECIMAL FIELD
D_INT    DS    ØH                      BINARY INTEGER
         L     R3,SQLDATA              A(DATA)
         LH    R1,SQLLEN               L(DATA)
* R1: FIELD SIZE
* CREATE MASK FOR ICM INSTRUCTION
         L     R2,=X'ØØØØØØØF'         LOAD MASK
         SLL   R2,Ø(R1)
         SRL   R2,4
         N     R2,=X'ØØØØØØØF'         R2: ICM MASK
         SR    RØ,RØ
         EX    R2,ICM                  LOAD BINARY VALUE INTO RØ
         CNOP  Ø,4
         CVD   RØ,D                    CONVERT IT TO DECIMAL
         B     FMT_DEC                 FORMAT DECIMAL FIELD
FMT_DEC  DS    ØH                      FORMAT DECIMAL FIELD
* INPUT:
* <D>: PACKED DECIMAL FIELD
* R14: RETURN ADDRESS
* OUTPUT:
* R2: L(FORMATTED FLD)
* R3: A(FORMATTED FLD)
         MVC   EDWK,EDMKINT            MOVE MASK
         LA    R1,EDWK+L'EDWK-2
         EDMK  EDWK,D
* R1: 1ST SIGNIFICANT CHARACTER
         LA    R2,EDWK_E             END ADDR
         SR    R2,R1                 L(FORMATTED FLD)
         LR    R3,R1                 A(FORMATTED FLD)
         BR    R14
* EX INSTRUCTIONS
ZAP      ZAP   D,Ø(Ø,R3)
ICM      ICM   RØ,Ø,Ø(R3)
* WORK FIELDS
SCALE    DS    F
H2       DS    H
FILLER2  DS    ØD
D        DS    PL8
WK       DS    CL8
EDWK     DS    CL17
EDWK_E   EQU   *
EDMKINT  DC    X'4Ø',13X'2Ø',X'212Ø',C'-'      EDIT MASK INTEGERS
```

```
         DS A
CHECK_SQL ST   R14,CHECK_SQL-4     CHECK SQLCODE
         L     R15,SQLCODE
         LTR   R15,R15
         BZR   R14                 :SOL OK
         MVC   RC,SQLCODE
         B     EOD
         TITLE 'DATA AREAS'
STARTWRK DS    ØF
A_DEBUG  DS    A                  VALENTINO
A_SQLDA  DS    A                  A(ALLOCATED SQLDA)
A_DBUF   DS    A                  A(DATA BUFFER)
A_DSADDR DS    AL4
COLSIZE  DC    F'Ø'               COLUMN SIZE
BUFFSIZE DC    F'Ø'               BUFFER SIZE
         DC    C' '               CLEAR BYTE
VN       DS    2CL18              VARIABLE NAME
VL       DS    A                  VARIABLE LENGTH
VNCT     DC    PL4'Ø'             GENERATED NAME COUNT (MAX 9)
INDEX    DC    PL4'Ø'             ROW INDEX
VNINDEX  DS    2CL8
VLINDEX  DS    F
SQL_CA   DS    CL(SQLDLEN)        BASIC SQLCA
SQL_DA   DS    CL16               BASIC SQLDA
         EXEC  SQL INCLUDE SQLCA
         EXEC  SQL INCLUDE SQLDA
SQLVARN_SIZE EQU 44
         LTORG
AIRXEXCOM DS A
IRX_IRXEXCOM DC CL8'IRXEXCOM'
         DS    ØA                 ALIGN
IRX_SHVBLOCK DC (SHVBLEN)X'Ø'
DB2VAR   DC    H'4',CL4' '
         ORG   DB2VAR+2
DB2V     DS    CL4
SELECT   DC    H'8Ø',CL8Ø' '
         ORG   SELECT+2
SQLQUERY DS    CL4Ø96
HOSTVAR  DS    H,CL3276Ø
         ORG   HOSTVAR+2
HOSTVAL  DS    CL3276Ø
         TITLE 'DSECTS'
DSECT_TYPE DSECT
DS_TYPE  DS    HL2
DS_GEN   DS    AL1            GENERIC TYPE (NUMERIC, CHARACTER)
         DS    AL1            FILLER
DS_CODE  DS    CL2
DS_ADDR  DS    AL4
DS_L     EQU   *-DS_TYPE
         IRXSHVB              DEFINITION OF REXX SHVB
         END
```

```
//*            ASSEMBLE IF THE PRECOMPILE RETURN CODE
//*            IS 4 OR LESS
//ASM     EXEC PGM=ASMA9Ø,PARM='OBJECT,NODECK',COND=(4,LT,PC)
//SYSIN    DD  DSN=&&DSNHOUT,DISP=(OLD,DELETE)
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR
//         DD  DSN=SYS1.MODGEN,DISP=SHR
//         DD  DSN=SYSADM.USER5.ASM,DISP=SHR
//SYSLIN   DD  DSN=&&LOADSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(8ØØ,(1Ø,1Ø)),DCB=(BLKSIZE=8ØØ)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  SPACE=(8ØØ,(1Ø,1Ø),,,ROUND),UNIT=SYSDA
//*            LINKEDIT IF THE PRECOMPILER AND ASSEMBLER
//*            RETURN CODES ARE 4 OR LESS
//LKED    EXEC PGM=IEWL,PARM='XREF,AMODE=31,RMODE=ANY',
//             COND=((4,LT,ASM),(4,LT,PC))
//SYSLIB   DD  DISP=SHR,DSN=DSN51Ø.SDSNLOAD
//         DD  DSN=SYS1.DSN51Ø.SDSNEXIT,DISP=SHR
//SYSLIN   DD  DSN=&&LOADSET,DISP=(OLD,DELETE)
//         DD  DDNAME=SYSIN
//*SYSLMOD  DD DSN=POSTP.BASEV5.LOAD(SQLISPF),DISP=SHR
//SYSLMOD  DD DSN=SYSADM.USER5.LOAD(SQLISPF),DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSUT1   DD  SPACE=(1Ø24,(5Ø,5Ø)),UNIT=SYSDA
//SYSIN    DD  *
  INCLUDE SYSLIB(DSNALI)
  NAME SQLISPF(R)
/*
//*
//BINDPROD EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)
//DBRMLIB  DD DSN=SYSADM.DBRMLIB5.DATA,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN   DD *
   GRANT BIND, EXECUTE ON PLAN SQLISPF TO PUBLIC;
//SYSTSIN DD *
 DSN SYSTEM(DSN)
 BIND PLAN(SQLISPF) MEMBER(SQLISPF) ACT(REP) ISO(CS) EXPLAIN(NO)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA51) -
      LIB('DSN51Ø.RUNLIB.LOAD')
 END
//*
//BINDTEST EXEC PGM=IKJEFTØ1,DYNAMNBR=2Ø,COND=(4,LT)
//DBRMLIB  DD DSN=SYSADM.DBRMLIB5.DATA,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN   DD *
   GRANT BIND, EXECUTE ON PLAN SQLISPF TO PUBLIC;
```

```
//SYSTSIN DD *
 DSN SYSTEM(DBT)
 BIND PLAN(SQLISPF) MEMBER(SQLISPF) ACT(REP) ISO(CS) EXPLAIN(NO)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA51) -
      LIB('DBT51Ø.RUNLIB.LOAD')
 END
//
```

BEGIN

```
*     REQUIRED KEYWORDS:  SAVE AND BASE
*     DEPENDENCIES : GLOBAL PARAMETERS ALSO USED IN MACRO "DATE"

         MACRO
&NAME    BEGIN &SAVE=,&BASE=,&PARM=,&DATE=,&TIME=,&PRINT=ON
         SPACE 2
***********************************************************************
*                                                                     *
***********************************************************************
         SPACE 2
         AIF   ('&PRINT' NE 'NOGEN').A1
         PRINT &PRINT
.A1      ANOP
         GBLC  &SYDNBØ1,&SYDNBØ2,&SYDNBØ3,&SYDNBØ4
         GBLC  &SYDNBØ5,&SYDNBØ6,&SYDNBØ7
         LCLC  &B,&C,&D,&E,&F,&N,&O,&P
&B       SETC  'B'.'&SYSNDX'.'AA'
&C       SETC  'B'.'&SYSNDX'.'HW'
&D       SETC  'B'.'&SYSNDX'.'AC'
&E       SETC  'B'.'&SYSNDX'.'AB'
&F       SETC  'B'.'&SYSNDX'.'AD'
&N       SETC  'B'.'&SYSNDX'.'AE'
&O       SETC  'B'.'&SYSNDX'.'AF'
&P       SETC  'B'.'&SYSNDX'.'ED'
         AIF   ('&SAVE' EQ '').FT1
         AIF   ('&BASE' EQ '').FT2
         AIF   ('&PRINT' EQ 'NOGEN').AØ
         AIF   ('&PRINT' EQ 'ON').AØ
         PRINT &PRINT
.AØ      ANOP
&NAME    DS    ØH
         B     6Ø(15)            BRANCH AROUND CONSTANTS
MY_CSECT DC    CL1Ø'&SYSECT'     CSECT NAME
         DC    C'&SYSDATE'       ASSEMBLY DATE
         DC    C' '
         DC    C'&SYSTIME'       ASSEMBLY TIME
         DC    C' '
         DC    C' ¸ DE NEDERLANDSCHE BANK N.V.  '
         STM   14,12,12(13)
         CNOP  2,4
         BALR  &BASE(1),Ø
         USING *,&BASE(1)        1ST BASE
.*
         AIF   (N'&BASE LT 5).A
         MNOTE 4,'*** MAXIMUM NUMBER OF BASE-REGISTERS IS FOUR. ***'
.*
.A       ANOP
         AIF   ('&BASE(2)' EQ '').B
&B       EQU   *
         L     &BASE(2),*+8
         USING &B+4Ø96,&BASE(2)    2ND BASE
         B     *+8
         DC    A(&B+4Ø96)
         AIF   ('&BASE(3)' EQ '').B
         L     &BASE(3),*+8
         USING &B+8192,&BASE(3)    3RD BASE
         B     *+8
         DC    A(&B+8192)
         AIF   ('&BASE(4)' EQ '').B
         L     &BASE(4),*+8
         USING &B+12288,&BASE(4)    4TH BASE
         B     *+8
         DC    A(&B+12288)
.B       ANOP
         LA    14,&SAVE
         ST    14,8(13)
         ST    13,&SAVE+4
         LA    13,&SAVE
         AIF   ('&PARM' EQ '').C
         L     14,Ø(1)
         LH    15,Ø(14)
         CH    15,&C
         BE    &D
         BCTR  15,Ø
         EX    15,&E
         B     &D
&E       MVC   &PARM.(1),2(14)
&C       DC    H'Ø'
&D       EQU   *
.C       ANOP
         AIF   ('&DATE' EQ '' AND '&TIME' EQ '').H
.*       AIF   ('&SYDNBØ6' NE '').CNT1Ø
&SYDNBØ6 SETC  'B'.'&SYSNDX'.'FW'
.CNT1Ø   ANOP
         ST    1,&SYDNBØ6                    SAVE REG 1
         LA    1,2(Ø,Ø)                      SPECIFY MAGNITUDE
         SVC   11                            GET TIMER
         B     &F                            BRANCH AROUND CONSTANTS
.*       AIF   ('&SYDNBØ1' NE '').CNT2Ø
&SYDNBØ1 SETC  'B'.'&SYSNDX'.'DB'
&SYDNBØ1 DC    D'Ø'
```

```
.CNT2Ø   ANOP
.*       AIF   ('&SYDNBØ6' NE 'B&SYSNDX.FW').CNT3Ø
&SYDNBØ6 DS    F
.CNT3Ø   AIF   ('&DATE' EQ '').D
.*       AIF   ('&SYDNBØ2' NE '').CNT4Ø
&SYDNBØ2 SETC  'B'.'&SYSNDX'.'P1'
&SYDNBØ2 DC    PL2'1'
.CNT4Ø   ANOP
.*       AIF   ('&SYDNBØ3' NE '').CNT5Ø
&SYDNBØ3 SETC  'B'.'&SYSNDX'.'PØ'
&SYDNBØ3 DC    PL2'Ø'
.CNT5Ø   ANOP
.*       AIF   ('&SYDNBØ4' NE '').D
&SYDNBØ4 SETC  'B'.'&SYSNDX'.'C19'
&SYDNBØ4 DC    C'19'
.*
.D       ANOP
         AIF   ('&TIME' EQ '').E
.*       AIF   ('&SYDNBØ7' NE '').E
&SYDNBØ7 SETC  'B'.'&SYSNDX'.'ED'
&SYDNBØ7 DC    CL14' '
.*
.E       ANOP
         AIF   ('&DATE' EQ '').F
.*       AIF   ('&SYDNBØ5' NE '').CNT6Ø
&SYDNBØ5 SETC  'B'.'&SYSNDX'.'TAB'
&SYDNBØ5 DC    PL2'31',C'JAN.'
         DC    PL2'28',C'FEB.'
         DC    PL2'31',C'MRT.'
         DC    PL2'3Ø',C'APR.'
         DC    PL2'31',C'MEI '
         DC    PL2'3Ø',C'JUNI'
         DC    PL2'31',C'JULI'
         DC    PL2'31',C'AUG.'
         DC    PL2'3Ø',C'SEPT'
         DC    PL2'31',C'OKT.'
         DC    PL2'3Ø',C'NOV.'
         DC    PL2'31',C'DEC.'
.CNT6Ø   ANOP
&F       EQU   *                    SUPPLY DATE
         ST    1,&SYDNBØ1+4
         MVO   &SYDNBØ1,&SYDNBØ1.(6)
         MVC   &DATE+8(2),&SYDNBØ4
         UNPK  &DATE+1Ø(2),&SYDNBØ1
         OI    &DATE+11,X'FØ'
         CVB   14,&SYDNBØ1
         ST    14,&SYDNBØ1
         TM    &SYDNBØ1+3,B'ØØØØØØ11'
         BNZ   *+1Ø
         AP    &SYDNBØ5+6(2),&SYDNBØ2
         ST    1,&SYDNBØ1
         XC    &SYDNBØ1.(2),&SYDNBØ1
         ZAP   &SYDNBØ1+4(4),&SYDNBØ3
         LA    1,&SYDNBØ5
         LA    14,12
&N       EQU   *
         AP    &SYDNBØ1+4(4),Ø(2,1)
         CP    &SYDNBØ1.(4),&SYDNBØ1+4(4)
         BNH   &O
         LA    1,6(1)
         BCT   14,&N
         DC    X'FFFF'
&O       EQU   *
         SP    &SYDNBØ1+4(4),Ø(2,1)
         SP    &SYDNBØ1.(4),&SYDNBØ1+4(4)
         UNPK  &DATE.(2),&SYDNBØ1+2(2)
         OI    &DATE+1,X'FØ'
         MVC   &DATE+3(4),2(1)
.F       ANOP
         AIF   ('&DATE' NE '').G
&F       EQU   *                  SUPPLY TIME
.G       ANOP
         AIF   ('&TIME' EQ '').I
         XC    &SYDNBØ1,&SYDNBØ1
         ST    Ø,&SYDNBØ1+4
         MVC   &SYDNBØ1+3(4),&SYDNBØ1+4
         MVI   &SYDNBØ1+7,X'ØC'
         MVO   &SYDNBØ1.(8),&SYDNBØ1.(7)
         MVC   &SYDNBØ7.(13),=X'4Ø2Ø21214B21214B21216B2121'
         ED    &SYDNBØ7.(13),&SYDNBØ1+3
         MVC   &TIME.(11),&SYDNBØ7+2
.I       ANOP
         L     1,&SYDNBØ6
.H       ANOP
         AIF   ('&PRINT' EQ 'NOGEN').K
         AIF   ('&PRINT' EQ 'ON').K
         PRINT ON
.K       ANOP
*                                      END "BEGIN"
         MEXIT
.FT1     MNOTE 8,'*** KEYWORD SAVE MISSING. ***'
         AGO   .H
.FT2     MNOTE 8,'*** KEYWORD BASE MISSING. ***'
         AGO   .H
         MEND
```

EINDE

```
         MACRO
&NAME    EINDE &SAVE=,&RC=,&RCR=,&PARMLST=
         AIF   ('&SAVE' EQ '').FT
         AIF   ('&RCR' NE '').B
&NAME    L     13,&SAVE+4
         LM    14,12,12(13)
         AIF   ('&PARMLST' EQ '').AØ
         LA    1,&PARMLST
.AØ      AIF   ('&RC' EQ '').A1
         LA    15,&RC.(Ø,Ø)
.A1      ANOP
         BR    14
         MEXIT
.B       ANOP
&NAME    DS    ØH
         LR    15,&RCR                              LOAD RETURNCODE
         L     13,&SAVE+4
         L     14,12(13)
         LM    Ø,12,2Ø(13)
         AIF   ('&PARMLST' EQ '').BØ
         LA    1,&PARMLST
.BØ      ANOP
         BR    14
         MEXIT
.FT      MNOTE 8,'KEYWORD SAVE MISSING'
         MEND
```

Nikola Lazovic, Gordana Kozovic, Ivan Bugarinovic
DB2 System AdministratorS
Postal Savings Bank (Yugoslavia)

CICS questions and answers

Q Is it possible to get CTG/390 to automatically re-connect to CICS? If we load CTG before CICS we need to reload the CTG to get it to connect to CICS.

A You need to add the CICS SVC number to DFHXCOPT. Specifying CICSSVC=0 (the default) indicates that EXCI should obtain the SVC number using an MVS VERIFY command. If CTG is up before any other CICS regions have logged on to DFHIRP the *obtain* command will fail.

If you have any CICS-related questions, then please send them in and we will do our best to find answers. Alternatively, e-mail them directly to cicsq@xephon.net.

CICS news

Computer Associates has announced the immediate availability of a range of management applications designed to support implementation of IBM CICS Transaction Server for z/OS Version 2 Release 2 to coincide with the general availability of the latest version of the mainframe transaction processor.

CA's software for managing high-volume CICS TS environments spans its Unicenter, eTrust, BrightStor, CleverPath, Advantage, and AllFusion brands.

For further information contact:
Computer Associates International, One Computer Associates Plaza, Islandia, NY 11749, USA.
Tel: (631) 342 6000.
URL: http://www.ca.com.

* * *

IBM has announced Debug Tool for z/OS and OS/390 Version 1 Release 3, for compiled applications, which has support for Assembler via disassembly view, plus a range of new functions. It's the renamed version of CoOperative Development Environment/370, or CODE/370.

Besides Assembler, the debugger currently supports applications written in C/C++, COBOL, High Performance Java (HPJ), and PL/I.

Debug Tool works in CICS, IMS, DB2, WebSphere, TSO, JES/Batch, Unix System Services and CMS. Via the full-screen interface, users can interactively debug any application as it runs, including batch applications. The tool can be invoked when an application is initialized, or it can be started dynamically when a condition occurs. The application itself can also invoke Debug Tool.

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

* * *

ClientSoft has announced the release of its Tanit Objects development platform and runtime environment for CICS and IMS integration on the S/390 platform.

Tanit Objects provides three means of access to CICS and IMS programs by furnishing a CICS interface via distributed program link (DPL) programs, the 3270 Bridge, and the front-end programming interface (FEPI).

This is the first offering in a new family of products resulting from the acquisition of the French-based company, TanitObjects.

For further information contact:
ClientSoft, 8323 Northwest 12 Street, Suite 216, Miami, FL 33126, USA.
Tel: (305) 716 1007.
URL: http://www.clientsoft.com/products/tanit.htm.

* * *