236

CICS update

July 2005

In this issue

3 CICS Transaction Gateway – strategic JCA connector with WAS: part 2
8 A CICS exit to count CICS commands
28 CICS TS 3.1 highlights
29 CICS trace analysis program
48 CICS questions and answers
50 CICS news

A tcl PUBLICATION

CICS Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

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

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs $270.00 in the USA and Canada; £175.00 in the UK; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the December 2000 issue, are available separately to subscribers for $24.00 (£16.00) each including postage.

CICS Update on-line

Code from *CICS Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/cics; you will need to supply a word from the printed issue.

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

Printed in England.

CICS Transaction Gateway – strategic JCA connector with WAS: part 2

This month we conclude the article looking at JCA connectors and WebSphere Application Server.

DEPLOYING THE ECI RESOURCE ADAPTER ON WEBSPHERE APPLICATION SERVER FOR Z/OS

CICS J2EE resource adapters are supplied in the *<install_path>/deployable directory* for CTG V5 and V6. There is a single resource adapter file – cicseci.rar. There is no separate resource adapter for WebSphere Application Server (WAS) for z/OS.

You are recommended to completely remove any resource adapter supplied with a previous version when upgrading to a new release of CTG.

CTG V5.1 INTSALLATION TIPS

Create a new file, ctgenvvar, from ctgenvvarsamp in */ctg51/bin* and alter the following variables:

```
CTGSTART_HOME="/ctg51/ctg/bin/"
RRM_NAME="CCL.CTG"
EXCI_OPTIONS="your.CICSTS.SDFHEXCI"
EXCI_LOADLIB=" your.CICSTS.SDFHEXCI"
export DFHJVPIPE="yourPIPEname"
```

Note: this value must be set, or you will get:

```
EXCI error. Function Call = 1, Response = 12, Reason = 403, Subreason
field-1 = 0, subreason ffield-2 = 0 CTG Rc = 2.
```

403 is INVALID_APPL_NAME

Update */ctg51//bin/ctgstart*:

```
#JAVASWITCH="${JAVASWITCH} -Xmx128M -Xrs"
#
JAVASWITCH="${JAVASWITCH} -Xmx350M -Xrs"
```

```
## Increase JVM allocation size from 128M default to 350M

exec $JAVA -Xmx350M  ${JAVASWITCH}  -Dcom.ibm.ctg.cicscli="$CICSCLI"
$jvmoptions com.ibm.ctg.util.CTGLaunch  CTG $otheroptions
```

Create CTG.INI from CTG.INISAMP in */ctg51/bin/* and update the following parameters:

```
# Increase Maximum number of ConnectionManager threads
#    for high-volume systems
# maxconnect=100
maxconnect=200

# Maximum number of Worker threads
maxworker=100

workertimeout=2000
connectionlogging=on
```

Here is a sample CTG PROC:

```
//OEEXCI   EXEC PGM=BPXBATCH,
//        PARM='SH cd /ctg51/bin;ctgstart -noinput ',
//         REGION=500M
//STDIN    DD PATH='/dev/null',
//         PATHOPTS=(ORDONLY)
//STDOUT   DD  PATH='/ctg51/logs/jgateo.log',
//         PATHOPTS=(OWRONLY,OCREAT),
//         PATHMODE=SIRWXU
//STDERR   DD  PATH='/ctg51/logs/jgatee.log',
//         PATHOPTS=(OWRONLY,OCREAT),
//         PATHMODE=SIRWXU
//STDENV DD DISP=SHR,DSN=your.CTLCARD(CTG51)
```

Here are the start-up messages from CTGTEST in */ctg51/logs/jgateo.log*:

```
CTG6111I File 'ctgenvvar' found. Using
the configuration in script 'ctgenvvar'
to start up the application.
CICS Transaction Gateway, Version 5.1, 5724-D12. Build Level c502-
20040302.
(C) Copyright IBM Corporation 1999, 2004. All rights reserved.
CCL8400I: Using ini file /ctg51/bin/CTG.INI.
CCL6577I: Java version is 1.3.1.
CCL6502I: Initial ConnectionManagers = 1, Maximum ConnectionManagers =
200,
CCL6502I: Initial Workers = 1, Maximum Workers = 100, tcp: Port = 2006
CCL6574I: Connection logging has been disabled.
CCL6505I: Successfully created the initial ConnectionManager and Worker
```

```
threads.
CCL6524I: Successfully started handler for the tcp: protocol.
```

CTG V6 INSTALLATION ON Z/OS

CTG V6 now has SMP/E support and you can use the SMP/E installation tape provided to transfer the code to your system. Then proceed with the installation as with previous versions, running the installation script, ctginstall.

MIGRATING FROM CTG V5

Please note that the JCA archives shipped with the CICS TG for z/OS V6 are not in the *<install_path>/deployable directory*, as documented in the *Administration Guide*. The JCA archives (cicseci.rar and ECIDateTime.ear) are in the *<install_path>/classes directory*. They can be deployed from this directory.

HTTP AND HTTPS PROTOCOLS

Support for the HTTP and HTTPS protocols was removed in Version 6. Use one of these protocols instead:

- TCP – use this in place of HTTP.
- SSL – use this in place of HTTPS.

A migration aid, script ctgconvenv, is provided to assist in the migration from earlier versions of CICS Transaction Gateway for z/OS. You can see the CTGCONV member of the SCTGSAMP library as an example of how to use this script.

To run script ctgconvenv from the USS command line against a previous installation in *{old inst dir}* and create an HFS file *{full path name}* use:

```
ctgconvenv [wrap width] {old inst dir}/bin/ctgenvvar
```

where *[wrap width]* is an optional numeric value to set the width of the generated output, which would otherwise default to the terminal screen width.

On z/OS, migration from Version 4 and earlier is not supported. If you are still running unsupported CTG V4 or earlier, you will need to uninstall it before installing this version.

UPDATING JAVA CLASSPATH

If you need to change the Java classpath to point to a newer version of Java, first determine which environment variable sets JAVA CLASSPATH for CTG. In your ctgstart file, add the following line:

```
set > CTGVars.txt
```

When the gateway starts, this file will be created in your CTG bin directory, and will show you which environment variable points to the old Java. It is most likely that your PATH statement is finding the old Java. If so, you'll need to add a line to your ctgenvvar file that inserts the new Java before the other entries:

```
ctgenvvar
export  PATH=/yourjava142classpath/bin:${PATH}

ctgstart
CLASSPATH="/yourjava142classpath/"
LIBPATH="/yourjava142libpath/"
```

Then check your start-up messages to verify that you are now loading Java 1.4.2, as shown below:

```
CTG6111I File 'ctgenvvar' found. Using
the configuration in script 'ctgenvvar'
to start up the application.
02/03/05 : 15:17:04:193 : CICS Transaction Gateway, Version 5.1, 5724-
D12. Build
02/03/05 : 15:17:04:197 : (C) Copyright IBM Corporation 1999, 2004. All
rights r
02/03/05 : 15:17:04:471 : CCL8400I: Using ini file /ctg51/bin/CTG.INI.
02/03/05 : 15:17:04:475 : CCL6577I: Java version is 1.4.2.
```

If you are one of the shops that is still running CTG V5.0.1, please be aware that CTG 5.0.1 was not tested for use on z/OS 1.6, so is not officially supported.

SUPPORTPACS FOR CTG

The following SupportPacs were created to assist with CTG support:

- CA5C – CICS TG sample EAR files for JCA connectivity.

  This SupportPac provides the CTG JCA samples as pre-built EAR files, along with the corresponding documentation on how to deploy the EAR files.

  The SupportPac consists of the following files:

  – ECIDateTime.ear – an enterprise application for testing ECI connectivity to CICS from WebSphere Application Server

  – EPIPlayScript.ear – an enterprise application for testing EPI connectivity to CICS from WebSphere Application Server

  – cclzaw00.pdf – a configuration PDF describing deployment of the EAR files and the CICS ECI and EPI resource adapters.

  This document is written for IBM WebSphere Application Server V5.1 for Windows, and the CICS TG V5.1 and the samples are built at the J2EE V1.3/EJB V2 level.

- CC12 – CICS Transaction Gateway for z/OS environment health check.

  This utility is designed as an environment health check for the CICS Transaction Gateway (CICS TG) for z/OS.

DOWNLOAD DESCRIPTION

The utility is written as a Unix korn shell script (ctgenvcheck.sh) and can be run either from the BPXBATCH environment or from the OMVS or Unix System Services shell environment. It takes an optional file argument as input, which can be the location of a CICS TG ctgenvvar configuration file. It functions by analysing the USS environment variables that affect the running of the CICS TG and reporting on their validity.

Along with the ctgenvcheck.sh script, a Java class (CheckCTGVersion) is supplied. This is used by the script to query the Java and CICS TG version stamps. The script assumes this class is located in the current directory from which the script is executed.

PREREQUISITES

The prerequisite is CICS Transaction Gateway for z/OS V4, V5, or V5.1.

CONCLUSION

CTG is IBM's key e-business-enabling Java connector. CTG has proven, high-performing, secure, and scalable access to CICS, requiring no changes to existing CICS applications. IBM continues to enhance CTG functionality, performance, system management, and support for the latest JCA releases, which define a standard architecture for connecting the J2EE platform to CICS. CTG support for JCA allows the proven qualities of CICS to be exploited by J2EE applications running in WebSphere Application Server. Utilizing the JCA simplifies application development by providing a familiar, standard interface that programmatically manages transactions, connections, and security.

Elena Nanos
IBM Certified Solution Expert in CICS Web Enablement
Zurich NA (USA)

A CICS exit to count CICS commands

A utility for counting any executed CICS API or SPI command is described, using a CICS Global User Exit (GLUE), a service program, and a command table. The article provides a short implementation overview, the required source code, and some operation guidelines.

This utility has been of great practical value during a recent application analysis and redesign effort carried out at R+V Versicherung in Wiesbaden, Germany.

INTRODUCTION

The command proposed here is a useful supplement to the existing CICS analysis tools, ie CICS statistic and the CICS trace.

The CICS statistics function offers a wide scope for analysis, but really displays, more or less, just the resource usage. With the CICS trace function, the execution of an application can be monitored, but the flood of data provided makes it inapplicable in a production environment.

The approach presented here allows the counting and recording

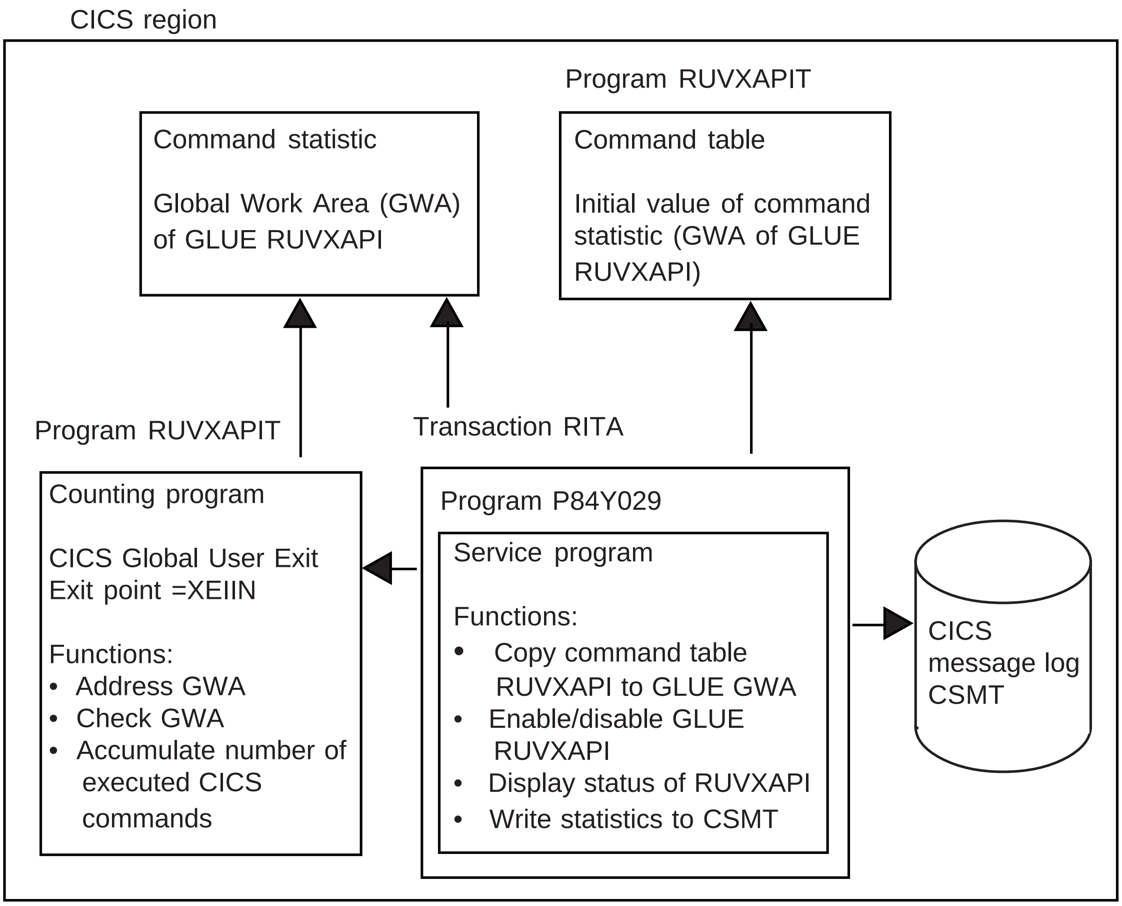
CICS region
Program RUVXAPIT
Command statistic
Global Work Area (GWA) of GLUE RUVXAPI
Command table
Initial value of command statistic (GWA of GLUE RUVXAPI)
Program RUVXAPIT
Transaction RITA
Counting program
CICS Global User Exit
Exit point =XEIIN
Functions:
• Address GWA
• Check GWA
• Accumulate number of executed CICS commands
Program P84Y029
Service program
Functions:
• Copy command table RUVXAPI to GLUE GWA
• Enable/disable GLUE RUVXAPI
• Display status of RUVXAPI
• Write statistics to CSMT
CICS message log CSMT

Figure 1: Overview

of any API and SPI command within an arbitrary time window. We use this at R+V Versicherung for example to determine:

- The number of called COBOL programs within a transaction (by counting the CICS push and pops while the LE run-time option CBLPSHPOP is turned on).
- The usage of any affinity causing CICS commands.
- The usage of SPI commands within business applications.

OVERVIEW

An overview is illustrated in Figure 1.

COMPONENTS

Service transaction RITA:

- Transaction name – RITA
- Program – P84Y029
- TaskDataLocation – any
- TaskDataKey – CICS
- Functions – see Service Program P84Y029
- Input parameter:
  - RITA=ON – enable RUVXAPI and start command counting.
  - RITA=OFF – disable RUVXAPI and stop command counting.
  - RITA=DUMP – write command statistics to CSMT.
  - RITA=INFO – display current status of RUVXAPI.
  - RITA – display current status of RUVXAPI.

Service program P84Y029:

- Programming language – COBOL.

- ExecutionKey – CICS.
- Functions:
  - enable/disable command counting GLUE RUVXAPI
  - copy initial command table RUVXAPIT to GWA of RUVXAPI
  - display current status of counting program RUVXAPI
  - print command statistics on CSMT

Counting program RUVXAPI:

- Programming language – Assembler
- ExecutionKey – CICS
- CICS exit point – XEIIN
- Function – count the number of executed CICS commands defined in the command table.

Command table RUVXAPIT:

- Programming language – Assembler
- Function:
  - definition of CICS commands to be counted.
  - initial value for command statistics.

SOURCE CODE

P84Y029

```
 Id Division.
 Program-Id.      P84YØ29.
 Author.          Rita Backstein.
 Date-Written.    26.Ø4.2ØØ4.
 Date-Compiled.
***------------------------------------------------------------
*** Functions:
*** - Enable/Disable CICS-Api-Exit RUVXAPI
*** - Copy CICS Function Code Table to GWA of RUVXAPI
*** - Display status of CICS-Api-Exit
*** - Display CICS-Command upon CICS Message Log CSMT
***--------------------------------------------------------------
 Environment     Division.
 Configuration   Section.
 Special-names.
     Decimal-point is comma.
 Data            Division.
 Working-Storage Section.
***--------------------------------------------------------------
 Ø1  Alpha-Constants.
     Ø5  My-Pgm-Id                 pic X(Ø8)  value 'P84YØ29'.
     Ø5  RUVXAPI                   pic X(Ø8)  value 'RUVXAPI'.
     Ø5  RUVXAPIT                  pic X(Ø8)  value 'RUVXAPIT'.
     Ø5  XEIIN                     pic X(Ø8)  value 'XEIIN'.
     Ø5  Doublepoint               pic X(Ø1)  value ':'.
     Ø5  CSMT                      pic X(Ø4)  value 'CSMT'.
     Ø5  lower-case-init.
        1Ø          pic x(26) value 'abcdefghijklmnopqrstuvwxyz'.
        1Ø          pic x(Ø1) value low-value.
     Ø5  lower-case  redefines lower-case-init pic x(27).
     Ø5  upper-case pic x(27) value 'ABCDEFGHIJKLMNOPQRSTUVWXYZ '.
***
 Ø1  Num-Constants.
     Ø5  One                       pic s9(Ø8) binary value +1.
***
 Ø1  Num-Variables.
     Ø5  CICS-Response             pic s9(Ø8) binary value zero.
     Ø5  StartStatus               pic s9(Ø8) binary value zero.
     Ø5  CICS-Abstime              pic s9(18) binary value zero.
     Ø5  Receive-Length            pic s9(Ø4) binary value zero.
     Ø5  Count-Disp                pic ZZZ.ZZZ.ZZZ.ZZ9 value zero.
     Ø5  Tab-Ind                   pic s9(Ø8) binary value zero.
     Ø5  Tab-Ptr                   usage pointer.
     Ø5          redefines         Tab-Ptr.
         1Ø Tab-Ptr-Num            pic s9(Ø8) binary.
     Ø5  RUVXAPIT-Len              pic s9(Ø4) binary value zero.
     Ø5  RUVXAPIT-Ptr              usage pointer.
     Ø5  GA-Len                    pic s9(Ø4) binary value zero.
     Ø5  GA-Ptr                    usage pointer.
***
 Ø1  Alpha-Variables.
     Ø5  CICS-Time                 pic x(ØØ8) value space.
     Ø5  CICS-UserId               pic x(ØØ8) value space.
     Ø5  CICS-TermId               pic x(ØØ4) value space.
***
 Ø1  IO-Buffer                     pic x(Ø8Ø) value low-value.
***
 Ø1  Parameter-P84YØ29.
     Ø5 TX-Name                    pic x(ØØ4) value space.
     Ø5 Blank1                     pic x(ØØ1) value space.
     Ø5 IO-Flag                    pic x(ØØ4) value space.
        88 IO-Flag-ON                         value 'ON  '.
        88 IO-Flag-OFF                        value 'OFF '.
        88 IO-Flag-DUMP                       value 'DUMP'.
        88 IO-Flag-INFO                       value 'INFO'.
        88 NO-Flag                            value space
                                                    low-value.
     Ø5 Blank2                     pic x(ØØ6) value space.
 Ø1         redefines Parameter-P84YØ29.
     Ø5                            pic x(Ø3).
     Ø5 Parameter-P84YØ29-Format-2 pic x(12).
***
 Ø1  Terminal-Flag                 pic x(Ø8) value space.
     88 With-Terminal                        value 'WithTerm'.
     88 Without-Terminal                     value space.
***
 Ø1  Error-flag                    pic x(Ø8) value space.
     88 No-Error                             value space.
     88 RUVXAPIT-Load-Error                  value 'LoadFail'.
     88 RUVXAPIT-Not-Active                  value 'NotActiv'.
     88 RUVXAPIT-Damage-Error                value 'Damage  '.
     88 RUVXAPI-Enable-Error                 value 'Enable  '.
     88 RUVXAPI-Disable-Error                value 'Disable '.
     88 RUVXAPI-Inquire-Error                value 'Inquire '.
     88 RUVXAPI-Extract-Error                value 'Extract '.
***
 Ø1  Exit-Status-Flag              pic x(12) value space.
     88 Exit-Status-started                  value 'started  '.
     88 Exit-Status-stopped                  value 'stopped  '.
     88 Exit-Status-not-active               value 'not active '.
***
 Ø1  Send-Text                     pic x(8Ø) value space.
 Ø1            redefines send-text pic x(8Ø).
     88 Text-Load-Error                      value
        ' P84YØ29: Error during Load of Cmd-Tab RUVXAPIT.'.
     88 Text-Set-Enable                      value
        ' P84YØ29: CICS-API-Exit RUVXAPI is now enabled.'.
     88 Text-Set-Disable                     value
        ' P84YØ29: CICS-API-Exit RUVXAPI is now disabled.'.
     88 Text-Enable                          value
        ' P84YØ29: CICS-API-Exit RUVXAPI is enabled.'.
     88 Text-Disable                         value
        ' P84YØ29: CICS-API-Exit RUVXAPI is disabled.'.
     88 Text-Not-Active                      value
        ' P84YØ29: CICS-API-Exit RUVXAPI is not active.'.
     88 Text-Inquire-Error                   value
        ' P84YØ29: Error during Inquire of Exit RUVXAPI.'.
     88 Text-Damage-Error                    value
        ' P84YØ29: Error: Cmd-Tab RUVXAPIT is damaged.'.
     88 Text-Enable-Error                    value
        ' P84Y029: Error during Enable of Exit RUVXAPI.'.
     88 Text-Disable-Error                   value
        ' P84Y029: Error during Disable of Exit RUVXAPI.'.
     88 Text-Extract-Error                   value
        ' P84Y029: Error during Extract of Exit RUVXAPI.'.
     88 Text-Dump                            value
        ' P84Y029: CICS-Cmd-Statistic was written to Queue CSMT.'.
***
 01  CSMT-Text                     pic x(100).
/**
 Linkage Section.
 01  DFHCOMMAREA                   pic x(015).
***
 01  RUVXAPIT-Layout.
     10 RUVXAPIT-Header.
        20 RUVXAPIT-Start-Eyecatch1   pic x(032).
           88 RUVXAPIT-Start-Eyecatch1-ok      value
              '******** Start RUVXAPIT ********'.
        20 RUVXAPIT-Start-Eyecatch2   pic x(028).
           88 RUVXAPIT-Start-Eyecatch2-ok      value
              ' No. of CICS-API-Commands = '.
        20 RUVXAPIT-No-of-Cmds        pic s9(08) binary.
     10 RUVXAPIT-Cmd-Tab-x.
        20 RUVXAPIT-Fc-x              pic x(002).
        20 RUVXAPIT-Fc-Decr-x         pic x(010).
        20 RUVXAPIT-Fc-Count-x        pic s9(08) binary.
***
 01  RUVXAPIT-Cmd-Tab.
     20 RUVXAPIT-Fc                   pic x(002).
        88 RUVXAPIT-Fc-End                     value x'FFFF'.
     20 RUVXAPIT-Fc-Decr              pic x(010).
        88 RUVXAPIT-Fc-Decr-End                value '***End****'.
     20 RUVXAPIT-Fc-Count             pic s9(08) binary.
***
 01  GA-Area.
     10                               pic x(001).
/**
 Procedure Division.
***
     Perform Start-Proc
***
     evaluate true
         when IO-Flag-ON
              perform Enable-Exit
         when IO-Flag-OFF
              perform Disable-Exit
         when IO-Flag-INFO
              perform Inquire-Exit
         when IO-Flag-DUMP
```

```
              perform Dump-RUVXAPIT
         when other
              perform Inquire-Exit
     end-evaluate
***
     Perform End-Proc
***
     Goback
     .
/**
 Start-Proc                          Section.
***
     move length of Parameter-P84Y029 to Receive-Length
     move low-value                   to Parameter-P84Y029
***
     if eibcalen > zero
        move dfhcommarea (1:eibcalen) to Parameter-P84Y029
     else
        exec cics retrieve into   (Parameter-P84Y029)
                           length (receive-length)
                           resp   (cics-response)
        end-exec
***
        if cics-response not = dfhresp(normal)
           exec cics receive into    (io-buffer)
                             length  (receive-length)
                             nohandle
           end-exec
           inspect io-buffer
                   converting lower-case to upper-case
           move io-buffer                to Parameter-P84Y029
           if tx-name = eibtrnid
              continue
           else
              move Parameter-P84Y029-Format-2
                                         to Parameter-P84Y029
           end-if
        end-if
     end-if
     Exec CICS Asktime    AbsTime (CICS-AbsTime)
                          NoHandle
     End-Exec
     Exec CICS FormatTime AbsTime (CICS-AbsTime)
                          Time    (CICS-Time   )
                          TimeSep (Doublepoint )
                          NoHandle
     End-Exec
     Exec CICS Assign     UserId  (CICS-UserId )
                          NoHandle
     End-Exec
     Exec CICS Assign     Facility(CICS-TermId)
                          Resp    (CICS-Response)
     end-exec
     if cics-response = dfhresp(normal)
        set With-Terminal              to true
     else
        set Without-Terminal           to true
     end-if
***
     .
/**
 Enable-Exit                        Section.
***
     exec cics load program (RUVXAPIT)
                    set     (RUVXAPIT-Ptr)
                    length  (RUVXAPIT-Len)
                    resp    (cics-response)
     end-exec
     if cics-response = dfhresp(normal)
        exec cics enable program  (RUVXAPI)
                         exit     (XEIIN)
                         GALength (RUVXAPIT-Len)
                         start
                         resp     (cics-response)
        end-exec
        if cics-response = dfhresp(normal)
           exec cics extract exit program  (RUVXAPI)
                                  GALength (GA-Len)
                                  GASet    (GA-Ptr)
                                  resp     (cics-response)
           end-exec
           if cics-response = dfhresp(normal)
              set address of RUVXAPIT-Layout to RUVXAPIT-Ptr
              set address of GA-Area         to GA-Ptr
              move RUVXAPIT-Layout(1:RUVXAPIT-Len)
                                             to GA-Area(1:GA-Len)
           end-if
        else
           set RUVXAPI-Enable-Error          to true
        end-if
     else
        set RUVXAPIT-Load-Error              to true
     end-if
***
     .
/**
 Disable-Exit                       Section.
***
     exec cics disable program  (RUVXAPI)
                       exitall
                       stop
                       resp     (cics-response)
     end-exec
     if cics-response = dfhresp(normal)
        continue
     else
        set RUVXAPI-Disable-Error   to true
     end-if
***
     .
/**
 Inquire-Exit                       Section.
***
     exec cics inquire exitprogram  (RUVXAPI)
                       exit         (XEIIN)
                       startstatus  (startstatus)
                       resp         (cics-response)
     end-exec
     if cics-response = dfhresp(normal)
        evaluate startstatus
            when dfhvalue(started)
                 set Exit-Status-started to true
            when dfhvalue(stopped)
                 set Exit-Status-stopped to true
        end-evaluate
     else
        if cics-response = dfhresp(pgmiderr)
           set RUVXAPIT-Not-Active       to true
           set Exit-Status-not-active    to true
        else
           set RUVXAPI-inquire-Error     to true
        end-if
     end-if
***
     .
/**
 Dump-RUVXAPIT                       Section.
***
     exec cics extract exit program  (RUVXAPI)
                            GALength (RUVXAPIT-Len)
                            GASet    (RUVXAPIT-Ptr)
                            resp     (cics-response)
     end-exec
     if cics-response = dfhresp(normal)
        set address of RUVXAPIT-Layout to RUVXAPIT-Ptr
        if RUVXAPIT-Start-Eyecatch1-ok and
           RUVXAPIT-Start-Eyecatch2-ok
           perform write-Cmd-Statistic
        Else
           set RUVXAPIT-Damage-Error to true
        End-if
     else
        set RUVXAPI-Extract-Error    to true
     end-if
***
     .
/**
 Write-Cmd-Statistic                 Section.
***
     move space                      to CSMT-Text
     string ' *** ' CICS-Time
            ' *** ' my-pgm-id
            ' *** Start of CICS Command-Statistic **************'
            delimited by size        into csmt-text
     end-string
     perform write-csmt
***
     if RUVXAPIT-No-of-Cmds > one
***     Set Address of Cmd-Tab to 1rst Table Entry
        set tab-ptr                  to address of RUVXAPIT-Layout
        add length of RUVXAPIT-Header
                                     to tab-ptr-num
        set address of RUVXAPIT-Cmd-Tab
                                     to tab-ptr
***
        perform varying tab-ind from one by one
          until tab-ind >= RUVXAPIT-No-of-Cmds
             or RUVXAPIT-Fc-End
             or RUVXAPIT-Fc-Decr-End
***
                move RUVXAPIT-Fc-Count
                                     to Count-Disp
                move space           to CSMT-Text
                String RUVXAPIT-Fc-Decr ' ' Count-Disp
                       delimited by size
                                     into csmt-text
                end-string
                perform write-csmt
***             Set Address of Cmd-Tab to next Table Entry
                add length of RUVXAPIT-Cmd-Tab
                                     to tab-ptr-num
                set address of RUVXAPIT-Cmd-Tab
                                     to tab-ptr
        end-perform
     else
        move space                   to CSMT-Text
        move 'No CICS-Commands in RUVXAPIT-Table found.          '
                                     to csmt-text
        perform write-csmt
     end-if
```

```
***
     move space                      to CSMT-Text
     string ' *** ' CICS-Time
            ' *** ' my-pgm-id
            ' *** End   of CICS Command Statistic ***************'
            delimited by size        into csmt-text
     end-string
     perform write-csmt
***
     .
/**
 End-Proc                            Section.
***
     evaluate true
         when RUVXAPIT-load-Error
              set text-Load-Error    to true
         when RUVXAPIT-Not-Active
              set Text-Not-Active    to true
         when RUVXAPIT-Damage-Error
              set Text-Damage-Error  to true
         when RUVXAPI-Enable-Error
              set Text-Enable-Error  to true
         when RUVXAPI-Disable-Error
              set Text-Disable-Error to true
         when RUVXAPI-inquire-Error
              set Text-Inquire-Error to true
         when RUVXAPI-Extract-Error
              set Text-Extract-Error to true
         when no-error
              evaluate true
                  when IO-Flag-ON
                       set Text-Set-enable          to true
                  when IO-Flag-OFF
                       set Text-Set-Disable         to true
                  when IO-Flag-DUMP
                       set Text-Dump                to true
                  when other
***                    Pgm Function = Info or No Function entered
                       evaluate true
                           when Exit-Status-started
                                set Text-Enable     to true
                           when Exit-Status-stopped
                                set Text-Disable    to true
                           when Exit-Status-not-active
                                set Text-not-active to true
                       end-evaluate
              end-evaluate
     end-evaluate
***
     move space                            to CSMT-Text
     string CICS-Time    ' '  cics-userid  ' '
            eibtrnid     ' '  send-text
            delimited by size into csmt-text
     end-string
     perform write-csmt
***
     if With-Terminal
        exec cics send text
                       from   (send-text)
                       length (length of send-text)
                       erase
                       freekb
                       nohandle
        end-exec
     end-if
***
     .
/**
 write-csmt                         section.
***
     exec cics writeq td queue   (csmt)
                         from    (csmt-text)
                         length  (length of csmt-text)
                         nohandle
     end-exec
***
```

RUVXAPI

```
           TITLE 'RUVXAPI - GLUE for CICS Command Statistic     '
***---------------------------------------------------------------
*** CICS Global API/SPI User Exit
***
*** Exit Point Name: XEIIN (Exec Interface Input Exit)
***
*** Functions:
***    - Count the no. of executed CICS API+SPI Commands
***      defined in the function code table RUVXAPIT
***
*** Author: Rita Backstein, Mai 2004
***---------------------------------------------------------------
           DFHUEXIT TYPE=EP,ID=(XEIIN)
***---------------------------------------------------------------
*** RUVXAPI Control Section
***---------------------------------------------------------------
RUVXAPI    CSECT
RUVXAPI    AMODE 31
RUVXAPI    RMODE ANY
***
           STM   R14,R12,12(R13)            save caller's regs
           LR    R3,R15                      establish base
           USING RUVXAPI,R3
***
           LR    R2,R1                       address standard parms
           USING DFHUEPAR,R2
***---------------------------------------------------------------
*** Check if GWA exists and Cmd-Table ok
***---------------------------------------------------------------
CHECK_GWA  DS    ØH
           L     R1,UEPGAL                   Load A(GWA-Len)
           USING UEPGAL_LEN_DSECT,R1
           CLC   UEPGAL_LEN,=H'Ø112'         GWA-Len < Minimum of 112?
           BL    GOBACK                      yes -> goback
           DROP  R1
***
           L     R5,UEPGAA                   Load A(GWA)
           C     R5,=F'ØØØØØØØØ'             GWA-Addr = Ø ?
           BE    GOBACK                      yes -> goback
***
           USING RUVXAPIT_START,R5
***
           CLC   XAPIT_START_C1,PGM_START_MARK1  Check Start-Eyecatch1
           BNE   GOBACK                          not ok -> goback
           CLC   XAPIT_START_C2,PGM_START_MARK2  Check Start-Eyecatch2
           BNE   GOBACK                          not ok -> goback
***
           CLC   XAPIT_TOTAL_E,ONE           No.of total Entries > 1?
           BNH   GOBACK                      no -> goback
***
           L     R7,XAPIT_TOTAL_E            No.tot.Entries->R7
           M     R6,XAPIT_TAB_SZ             * Tab-Entry-Len
           A     R7,XAPIT_START_SZ           + Tab-Start-Len
           AR    R7,R5                       + Tab Load Address
***
           USING RUVXAPIT_END,R7
           CLC   XAPIT_END_C1,PGM_END_MARK  Check End-Eyecatcher
           BNE   GOBACK                      not ok -> goback
***---------------------------------------------------------------
*** Init for Cmd Loop
***---------------------------------------------------------------
INIT_LOOP  DS    ØH
           LR    R6,R5                       Load A(RUVXAPIT)-> R6
           A     R6,XAPIT_START_SZ           + Tab-Start-Len
           USING RUVXAPIT_TAB,R6             Set first Tab_Entry
           XR    TAB_IND,TAB_IND             Init Index for Start Loop
           L     TAB_MAX,XAPIT_TOTAL_E       Save No.of total Entries
***
           L     R8,UEPARG                   Load A(UEPARG) -> R8
           USING UEPARG_LST,R8
           L     R9,UEPARG_FC_ADDR           Load A(UEPARG_FC) -> R9
           USING UEPARG_FC_DSECT,R9
***---------------------------------------------------------------
*** Loop
***---------------------------------------------------------------
DO_LOOP    DS    ØH
***
           A     TAB_IND,ONE                Index + 1
***
           CLC   XAPIT_FC,UEPARG_FC         RUVXAPIT-Fc = CICS-Fc?
           BNE   CHECK_LOOP                 no -> do next loop
***
           L     R4,XAPIT_COUNT             Hit -> Fc found in Cmd Tab
           A     R4,ONE                     add 1 to fc-counter
           ST    R4,XAPIT_COUNT
           B     GOBACK                     leave loop + goback
***---------------------------------------------------------------
*** Loop Check
***---------------------------------------------------------------
CHECK_LOOP DS    ØH
           CR    TAB_IND,TAB_MAX            Tab_Ind < Tab_Max?
           BNL   GOBACK                     no -> goback
           CLC   XAPIT_DESCR,END_MARK       Description = *End*?
           BE    GOBACK                     yes -> goback
***
           A     R6,XAPIT_TAB_SZ            Set next Tab_Entry
           USING RUVXAPIT_TAB,R6
           B     DO_LOOP                    do next loop
***---------------------------------------------------------------
*** Return to Caller
***---------------------------------------------------------------
GOBACK     DS    ØH
           LA    R15,UERCNORM               Set Return Code OK
           L     R13,UEPEPSA                Address Save Area
           L     R14,12(R13)                Set Return Address
           LM    RØ,R12,2Ø(R13)             Restore Caller's Register
           BR    R14                        Goback
***---------------------------------------------------------------
*** Constants
***---------------------------------------------------------------
                DS    ØD
RUVXAPIT        DC    CLØ8'RUVXAPIT'
END_MARK        DC    CL1Ø'***End****'
PGM_START_MARK1 DC    CL32'******** Start RUVXAPIT ********'
PGM_START_MARK2 DC    CL28' No. of CICS API Commands = '
PGM_END_MARK    DC    CL32'******** End   RUVXAPIT ********'
ONE             DC    F'ØØØ1'
XAPIT_START_SZ  DC    F'ØØ64'
XAPIT_TAB_SZ    DC    F'ØØ16'
***---------------------------------------------------------------
*** RUVXAPIT Layout
```

```
***---------------------------------------------------------------
RUVXAPIT_START  DSECT
                DS  ØF
XAPIT_START_C1  DS  CL32                     ******** Start RUVXAPIT *
XAPIT_START_C2  DS  CL28                     No. of CICS API Commands
XAPIT_TOTAL_E   DS  F
***
RUVXAPIT_TAB    DSECT
                DS  ØF
XAPIT_FC        DS  CL2                      CICS Function Code
XAPIT_DESCR     DS  CL1Ø                     Command Description
XAPIT_COUNT     DS  F                        Command Counter
***
RUVXAPIT_END    DSECT
                DS  ØF
XAPIT_END_C1    DS  CL32                     *********  End  RUVXAPIT *
***---------------------------------------------------------------
*** User-Exit-Parameter Layout
***---------------------------------------------------------------
UEPARG_LST      DSECT
                DS  ØF
UEPARG_FC_ADDR  DS  A
***
UEPARG_FC_DSECT DSECT
                DS  ØF
UEPARG_FC       DS  CL2
***
UEPGAL_LEN_DSECT DSECT
                DS  ØF
UEPGAL_LEN      DS  H
***---------------------------------------------------------------
*** Register Equates
***---------------------------------------------------------------
TAB_IND         EQU Ø
TAB_MAX         EQU 1
RØ              EQU Ø
R1              EQU 1
R2              EQU 2
R3              EQU 3
R4              EQU 4
R5              EQU 5
R6              EQU 6
R7              EQU 7
R8              EQU 8
R9              EQU 9
R1Ø             EQU 1Ø
R11             EQU 11
R12             EQU 12
R13             EQU 13
R14             EQU 14
```

```
R15             EQU 15
***---------------------------------------------------------------
           END   RUVXAPI
```

RUVXAPIT (sample)

```
         TITLE 'Table of CICS API+SPI Function Codes for Cmd Statistic'
***--------------------------------------------------------------------*
*** Table for CICS-Api-Exit RUVXAPI
***
*** The Exit counts the no. of executed CICS API+SPI Commands
*** defined in the function code table RUVXAPIT
***
*** Attention:
*** - CONST_STARTØ4 must define the total No. of table entries
*** - the last entry must contain x'FFFF' (= end marker)
***--------------------------------------------------------------------*
RUVXAPIT CSECT
RUVXAPIT AMODE 31
RUVXAPIT RMODE ANY
***--------------------------------------------------------------------*
***  Start-Eyecatcher
***--------------------------------------------------------------------*
CONST_STARTØ1    DS ØF
CONST_STARTØ2    DC CL32'******** Start RUVXAPIT ********'
CONST_STARTØ3    DC CL28' No. of CICS API Commands = '
CONST_STARTØ4    DC F'ØØØ8'            <<<=== No. of table entries
***                                           incl. end marker
***--------------------------------------------------------------------*
CONST_SØ1_Ø1     DS ØF
CONST_SØ1_Ø2     DC XL2'Ø2ØC'
CONST_SØ1_Ø3     DC CL1Ø'Push      '
CONST_SØ1_Ø4     DC F'Ø'
***--------------------------------------------------------------------*
CONST_SØ2_Ø1     DS ØF
CONST_SØ2_Ø2     DC XL2'Ø2ØE'
CONST_SØ2_Ø3     DC CL1Ø'Pop       '
CONST_SØ2_Ø4     DC F'Ø'
***--------------------------------------------------------------------*
CONST_SØ3_Ø1     DS ØF
CONST_SØ3_Ø2     DC XL2'Ø2ØA'
CONST_SØ3_Ø3     DC CL1Ø'Ignore Con'
CONST_SØ3_Ø4     DC F'Ø'
***--------------------------------------------------------------------*
CONST_SØ4_Ø1     DS ØF
CONST_SØ4_Ø2     DC XL2'Ø2Ø4'
CONST_SØ4_Ø3     DC CL1Ø'Handle Con'
CONST_SØ4_Ø4     DC F'Ø'
***--------------------------------------------------------------------*
CONST_SØ5_Ø1     DS ØF
```

```
CONST_SØ5_Ø2      DC XL2'ØEØE'
CONST_SØ5_Ø3      DC CL1Ø'Handle Abn'
CONST_SØ5_Ø4      DC F'Ø'
***---------------------------------------------------------------------*
CONST_SØ6_Ø1      DS ØF
CONST_SØ6_Ø2      DC XL2'ØEØ2'
CONST_SØ6_Ø3      DC CL1Ø'Link      '
CONST_SØ6_Ø4      DC F'Ø'
***---------------------------------------------------------------------*
CONST_SØ7_Ø1      DS ØF
CONST_SØ7_Ø2      DC XL2'ØEØ4'
CONST_SØ7_Ø3      DC CL1Ø'Xctl      '
CONST_SØ7_Ø4      DC F'Ø'
***---------------------------------------------------------------------*
CONST_S99_Ø1      DS ØF
CONST_S99_Ø2      DC XL2'FFFF'
CONST_S99_Ø3      DC CL1Ø'***End****'
CONST_S99_Ø4      DC F'Ø'
***---------------------------------------------------------------------*
***  End Eyecatcher
***---------------------------------------------------------------------*
CONST_ENDEØ1      DS ØF
CONST_ENDEØ2      DC CL32'******** End   RUVXAPIT ********'
                  END
```

OPERATIONAL CONSIDERATIONS

RITA=ON

Starting the transaction RITA with the input parameter ON turns command counting on.

This is confirmed as follows:

- On the 3270 screen:

```
P84YØ29: CICS-API-Exit RUVXAPI is now enabled.
```

- In the CICS messages log CSMT:

```
Ø7:44:32 XV1Ø853  RITA  P84YØ29: CICS-API-Exit RUVXAPI is now enabled.
```

RITA=OFF

Starting the transaction RITA with the input parameter OFF turns command counting off.

This is confirmed as follows:

- On the 3270 screen:

```
P84YØ29: CICS-API-Exit RUVXAPI is now disabled.
```

- In the CICS messages log CSMT:

```
Ø7:44:Ø9 XV1Ø853  RITA  P84YØ29: CICS-API-Exit RUVXAPI is now disabled.
```

RITA=DUMP

Starting the transaction RITA with the input parameter DUMP writes the CICS command statistic to the CICS messages log CSMT. This function works only when the statistics are turned on. After the function DUMP the statistics stay turned on.

This is confirmed as follows:

- On the 3270 screen:

```
P84YØ29: CICS-Cmd-Statistic was written to Queue CSMT.
```

- In the CICS messages log CSMT:

```
Ø7:43:5Ø XV1Ø853  RITA  P84YØ29: CICS-Cmd-Statistic was written to Queue
CSMT.
```

The statistics themselves are written to the CICS messages log CSMT and look like the following (example):

```
*** Ø7:43:5Ø *** P84YØ29  *** Start of CICS Command Statistic **********
Push           1Ø5.279.142
Pop            1Ø5.279.142
Ignore Con           4.945
Handle Con         291.794
Handle Abn             547
Link             1.237.1Ø5
Xctl                14.847
*** Ø7:43:5Ø *** P84YØ29  *** End   of CICS Command Statistic **********
```

RITA=INFO

Starting the transaction RITA with the input parameter INFO shows the status of the CICS command statistics exit.

This can be as follows:

- On the 3270 screen:

```
P84YØ29: CICS-API-Exit RUVXAPI is enabled.
```

Or:

```
P84Y029: CICS-API-Exit RUVXAPI is not active.
```

- In the CICS messages log CSMT:

```
07:43:26 XV10853  RITA  P84Y029: CICS-API-Exit RUVXAPI is enabled.
```

Or:

```
07:44:10 XV10853  RITA  P84Y029: CICS-API-Exit RUVXAPI is not active.
```

RITA

If the transaction RITA is started without an input parameter the function INFO is assumed.

The same applies to any input parameter other than ON, OFF, INFO, or DUMP.

Installing a new command table

The following describes how to install a new command table (RUVXAPIT):

1 Change the command table, RUVXAPIT, assemble and link it.

2 Turn the command counting off: RITA=OFF (this is necessary only if command counting is still active).

3 Request a new copy of the command table, RUVXAPIT.

4 Turn command counting on: RITA=ON.

REFERENCES

- *CICS Transaction Server for z/OS Customization Guide*, ‘Global user exit programs’ and ‘EXEC interface program exits XEIIN, XEIOUT, XEISPIN, and XEISPOUT’ describe the parameter lists passed to these exits.
- *CICS Transaction Server for z/OS Application Programming Reference*, ‘Appendix A. Exec interface block, EIB fields’ contains a list of the API function codes.

- *CICS Transaction Server for z/OS System Programming Reference*, ‘Appendix B. Exec interface block, Function codes of Exec CICS commands’ contains a list of the SPI function codes.

Rita Backstein (Rita.Backstein@ruv.de)
Systems Programmer
R+V Versicherung (Germany)

CICS TS 3.1 highlights

CICS provides users with a number of very useful new facilities. This article is a simple reminder of what those are.

The big (and that’s capital letters, double-underlined) feature is the Service-Oriented Architecture (SOA) deliverables. CICS now really is SOA-ready and has increased ease of integration with Web services. It’s now bi-directional, which means that existing CICS applications can make use of external Web services as well as being re-used as Web services. The one downside to all this is that some users may not want a whole CICS application as a Web service, they may want only part of its business logic.

There’s also enhanced application transformation. This brings increased links with WebSphere Studio Enterprise Developer, which can be used as a development tool for new applications and enhancing existing applications. There’s also a COMMAREA-independent way to transfer large volumes of data between applications. Gone is the old 32KB limit.

Extensions to the CICSPlex System Manager Web User Interface, improved workload throughput, and enhanced performance of C/C++ programs are the main parts of the improved performance and system management announcements.

Nick Nourse
Independent Consultant (UK)

CICS trace analysis program

INTRODUCTION

The CICS trace facility has been enhanced over the years. With each new release of CICS, users of the CICS trace facility and the batch reporting program, DFHTUP, have been given more and more ways to see what is going on inside a specific CICS transaction or region. Not only has CICS itself been evolving, but the applications (transactions) themselves, which are running within CICS, have become more and more complex. Today it isn't unusual to find transactions using 300 or more EXEC CICS LINKs during one execution path.

Analysing a CICS trace, however, is usually a very cumbersome but sophisticated task. This is not only because an increasing number of trace entries aren't fully documented by IBM in the CICS books; many application programmers, and even system programmers, aren't experienced enough to find this 'needle in a haystack'.

To support application and system programmers in their effort to understand what is happening in a CICS trace, I wrote the REXX procedure CITRLINK, which summarizes the activities of all transactions found in a CICS trace – a CICS AUX (auxiliary) trace written to DFHAUXT/DFHBUXT and formatted with DFHTUvrm, a CICS GTF trace formatted with the IPCS GTFTRACE function, or an internal trace formatted with the CICS IPCS verb exit DFHPDvrm out of a system dump. (vrm means Version, Release, Modification Level of CICS, eg 630 for CICS Transaction Server for z/OS 2.3.0.)

The main goal of the REXX utility was to visualize the program link hierarchy on a transaction basis, as well as showing the used EXEC CICS and EXEC SQL commands on a program basis. All other CICS trace entries are eliminated because they are not necessary in the understanding of the transaction and program logic. Now, we can see the program logic in a quasi condensed and easy-to-read form. This tool can also be used by performance analysts to detect, for instance, loops – the same EXEC or series of EXECs multiple times – which could eventually be optimized or omitted.

The second main goal of CITRLINK is to give both application developers and system programmers a tool to simplify the task of analysing a transaction's behaviour in an OTE (Open Transaction Environment). Thus CITRLINK is an additional tool when migrating to CICS Transaction Server 2.2 or higher. Its aim is to recognize TCB switches from L8 to QR and *vice versa*, thus reducing CPU costs for transactions.

WHAT REXX CITRLINK DOES FOR YOU

What does CITRLINK for you? It shows you:

1 All transactions found in the input dataset – CICS formatted trace (AUX trace, GTF trace, internal trace out on a SDUMP) – will be shown in a short summary at the beginning of the output.

2 The TASKID, TRANSID, and USERID for each transaction using one line per transaction.

3 How far the logic within a specific transaction has proceeded within the scope of the CICS trace, perhaps until an error occurred or perhaps to a normal end.

4 The flow and nesting of EXEC CICS LINKs visualized in a pseudo-graphical manner. CICS module calls can be shown only when they received control via EXEC CICS LINK. If you bind together multiple programs into one CICS load module, you have no chance of getting information on a specific program within the LOAD module because this is not visible in the CICS trace.

All programs using non-threadsafe CICS commands are marked with an '*'.

5 Which execution path a transaction took during its lifetime

until its successful end, for evaluation purposes. For example: is it correct that my transaction called subroutine XYZ *n* times? etc.

6 Which EXEC CICS and EXEC SQL commands have been used by each individual program.

7 Whether the EXEC CICS commands being used are threadsafe or not. Non-threadsafe CICS commands will be marked for this with an ‘!’.

8 A link from the CITRLINK information to the original trace dataset by trace entry numbers (eg =004711=), which is shown on the right side of every line. Because the output from CITRLINK is an extract of the whole CICS trace, you may need this later to find the right position in the original trace for further and more detailed analysis.

9 A summary of all EXEC CICS and EXEC SQL commands used by the transaction, giving you an overview of all EXEC CICS and EXEC SQL used in alphabetical order. Additional information about CICS commands is whether they are threadsafe or not.

10 The calculated percentage of threadsafe and non-threadsafe CICS commands used.

11 The calculated number of TCB switches from QR to L8 and *vice versa*. So you have an excellent overview of whether the number of TCB switches can be reduced dramatically when migrating to CICS Transaction Server 2.2 or higher and using CONCURRENCY (THREADSAFE) on the program definition. Because this number is calculated, you can use CITRLINK as an analysis tool for finding out how the number of TCB switches could be reduced by eliminating non-threadsafe CICS commands or by grouping together EXEC SQLs. This function of CITRLINK will work even if the trace analysed is from a CICS TS 1.3.

The program prints out two program indexes – one in alphabetical order by program name (also indicating the number of times each program has been used), the other in descending order by program use count and program name.

SPECIAL CONSIDERATIONS FOR REXX CITRLINK

REXX CITRLINK has been developed and tested with CICS Transaction Servers 1.3 and 2.3.

At the end of the coding of program CITRLINK there is a table containing all threadsafe CICS API and SPI (System Programming Interface) commands. This table is at the CICS TS 2.3 level.

Installations using CICS TS 2.2 or CICS TS 3.1 should use the CICS books *CICS Application Programming Reference*, ‘Appendix L. Threadsafe command list’ and *CICS System Programming Reference*, ‘Appendix D. Threadsafe SPI commands’ to change the table to suit their needs.

Here is some additional information for specific CICS commands using the detailed statistics of CITRLINK:

- EXEC CICS LOAD/RELEASE will show the name of the loaded and released module.
- The EXEC CICS API for VSAM commands will show the VSAM file name for the first command to a specific file. Subsequent calls to the same VSAM file cannot show the file name because this information is in the CICS trace records only for the first VSAM call.
- Some threadsafe CICS commands will become non-threadsafe when used in conjunction with function shipping, eg EXEC CICS LINK, READQ TS, WRITEQ TS. The CICS command will be marked for this – like any other non-threadsafe command – with an ‘!’.

  This restriction with function shipping is documented in the Redbook *Threadsafe considerations for CICS* – SG24-6951.

REXX CITRLINK formats trace entries only when the start of the task can be found in the trace. Tasks for which there is no task end are marked as incomplete tasks. For such tasks there are no overview reports or program indexes.

PREPARATIONS TO GET REXX CITRLINK WORKING CORRECTLY

Set the right CICS trace level on-line via the CETR transaction, F4 components, or at CICS start-up in PARM=, or SYSIN with the STNTR*xx*= parameters (*xx* represents the trace domain, eg STNTRAP=1 for AP domain). For details about CETR see *CICS Supplied Transactions*, Chapter 18, 'CETR – trace control'. For details about CICS start-up parameters see *CICS System Definition Guide*, Chapter 17, 'Specifying CICS system initialization parameters'.

As a minimum, to allow REXX CITRLINK to do its work correctly, you must activate trace level 1 for the following CICS domains:

- AP – Application domain
- EI – EXEC interface
- IS – Inter-system communication
- PG – Program manager domain
- RI – Resource manager interface
- XS – Security manager domain.

Produce a CICS trace – remember to reset all trace settings after you get your CICS trace dataset. CICS trace functions, especially auxiliary trace, will degrade your overall CICS system performance particularly in a production environment.

There are three ways to get a CICS trace:

- Use CETR to produce a CICS auxiliary trace on DFHAUXT/DFHBUXT:
  - set internal trace status to STARTED

- set auxiliary trace status to STARTED
- set auxiliary trace status to STOPPED after all the transactions you intend to analyse have been run. Only with the STOPPED status will the auxiliary trace dataset be closed and then it can be processed further.

- Use CETR to produce a CICS GTF trace:
  - set internal trace status to STARTED
  - set GTF trace status to STARTED
  - start the GTF MVS started task
  - stop the GTF MVS started task after all the transactions you intend to analyse have been run. Only when the GTF started task has ended and the GTF trace dataset is closed can further processing take place.
- Generate a CICS system dump:
  - set internal trace status to STARTED
  - set internal trace table size to an appropriate size
  - Use CEMT PERFORM SNAP.

    Note: producing a CICS system dump will stop processing work within the specific CICS region for some time. This time will vary from installation to installation but can be expected to be in the range of one to two minutes.
- Format the CICS trace depending on how you produced it and write the formatted trace to an output dataset. See the sample jobs below.
- Run the REXX program CITRLINK in interactive or in batch mode using the output dataset from the previous step.

FORMATTING CICS TRACE

REXX CITRLINK accepts three types of CICS formatted trace datasets:

- CICS auxiliary trace – use CICS utility DFHTRvrm (eg DFHTR630).
- CICS GTF trace – use the GTFTRACE subcommand of IPCS.
- CICS internal trace out of a CICS system dump – use the CICS IPCS verb exit DFHPDvrm.

Regardless of what kind of CICS trace you are using as input for CITRLINK, the trace always has to be formatted as an abbreviated trace.

A CICS abbreviated trace shows only one line of information for each trace record. This is the format CITRLINK uses.

CICS trace formatting routines allow functions such as selecting specific transactions or task numbers etc. You can use all the functions you need. For details see the chapters ‘Trace utility print program’ and ‘Dump utility program’ in *CICS Operations and Utilities Guide*.

The following three sample jobs have been used and proved for testing CITRLINK.

Formatting CICS auxiliary trace dataset

This is the simplest form of formatting a CICS auxiliary trace without additional select parameters:

```
//PRTRACE  EXEC  PGM=DFHTU630
//STEPLIB   DD DSN=your.cics.SDFHLOAD,DISP=SHR
//DFHAUXT   DD DSN=your.cics.DFHAUXT,DISP=SHR
//DFHAXPRT  DD DSN=formatted.cics.trace.dataset.as.input.for.citrlink,
//             DISP=(NEW,CATLG),
//             SPACE=(CYL,(50,10),RLSE),
//             DCB=(LRECL=133,BLKSIZE=23408)
//DFHAXPRM  DD *
ABBREV
```

Formatting CICS GTF trace dataset

In this example, TASKIDs 91217 to 91240 only will be selected for formatting:

```
//IPCS       EXEC PGM=IKJEFTØ1,REGION=4M
//STEPLIB     DD DISP=SHR,DSN=your.cics.SDFHLINK
//IPCSDDIR    DD DISP=SHR,DSN=your.IPCS.dump.directory
//DFHTRACE    DD DISP=SHR,DSN=your.cics.GTF.trace.dataset
//IPCSPRNT    DD DSN=formatted.cics.trace.dataset.as.input.for.citrlink,
//               DISP=(,CATLG,DELETE),UNIT=DISK,
//               SPACE=(5Ø,(5Ø,2),RLSE),AVGREC=M,
//               DSORG=PS,
//               RECFM=VBA,LRECL=137
//SYSTSPRT    DD SYSOUT=*
//SYSTSIN     DD *
PROF NOPREFIX
IPCS NOPARM
DROPDUMP DD(DFHTRACE)
SETDEF   DD(DFHTRACE)
GTFTRACE +
  CICS((ABBREV,TASKID=(91217-9124Ø))) USR(F6C)
END
```

Formatting CICS internal trace in a system dump

The TR=1 keyword below has the same meaning as ABBREV in the above mentioned samples:

```
//IPCSDUMP    EXEC PGM=IKJEFTØ1,REGION=4M
//STEPLIB     DD DISP=SHR,DSN=your.cics.SDFHLINK
//            DD DISP=SHR,DSN=your.cics.SDFHPARM
//DFHSDUMP    DD DISP=SHR,DSN=your.cics.sdump.dataset
//DFHSNAP     DD SYSOUT=*
//IPCSDDIR    DD DISP=SHR,DSN=your.IPCS.dump.directory
//IPCSTOC     DD SYSOUT=*
//IPCSPRNT    DD DSN=formatted.cics.trace.dataset.as.input.for.citrlink,
//               DISP=(,CATLG,DELETE),UNIT=DISK,
//               SPACE=(2Ø,(2,2),RLSE),AVGREC=M,
//               DSORG=PS,
//               RECFM=VBA,LRECL=137
//SYSPROC     DD DISP=SHR,DSN=your.ipcs.clists
//            DD DISP=SHR,DSN=SYS1.SBLSCLIØ
//SYSTSPRT    DD SYSOUT=*
//IPCSDUMP.SYSTSIN   DD *
PROF NOPREFIX
IPCS NOPARM
DROPDUMP DD(DFHSDUMP)
SETDEF DD(DFHSDUMP) NOPROBLEM NOCONFIRM NOTERMINAL PRINT LIST
```

```
VERBEXIT CICS53Ø 'DEF=Ø,IND=1,TR=1'
END
```

RUNNING REXX CITRLINK

You have the options of running CITRLINK on-line in ISPF or as a batch job. In either case it is recommended to run CITRLINK as compiled REXX. This will reduce CPU consumption by about 60%.

CITRLINK can recognize two positional parameters. The first always has to be the dataset name of the CICS formatted trace dataset. It can also be **?** to get short help information. The second parameter that can be given is **-d**. This will give you the detailed version of CITRLINK statistics. 'Detailed version' means that all used EXEC CICS and EXEC SQL will be formatted.

Syntax:

```
[TSO]      CITRLINK      datasetname | ? [-d]
```

The output that CITRLINK produces will be printed into an 80-byte- wide sequential dataset (RECFM=FBA) with the name userid.CITRLINK.TEMP.

CITRLINK allocates this dataset dynamically and reuses it if it already exists.

When CITRLINK has been run in ISPF, this dataset will automatically be displayed in ISPF BROWSE mode at the end of the run.

A note to using apostrophes with dataset names:

- CITRLINK behaves like an ISPF standard module. This means that using dataset names with apostrophes will process the dataset as given.
- Using dataset names without apostrophes will substitute the user's USERID as a dataset prefix when the user has set TSO PROFILE PREFIX. To prevent this behaviour, the user can use TSO PROFILE NOPREFIX. With this setting

there is no difference when using datasets with or without apostrophes.

RUNNING CITRLINK IN ISPF

ISPF dataset list menu (DSLIST)

Use the ISPF DSLIST menu to display the formatted CICS trace dataset. The ISPF DSLIST menu can be reached using the following methods:

- From the ISPF Workplace using a dataset name or using a Workplace dataset list (Reflist).
- Using the ISPF old-style menu Option 3.4.

In the dataset list, type in CITRLINK in front of the CICS formatted trace dataset name. This will produce the normal CITRLINK statistics.

To get detailed statistics, the easiest way is to type into the line where the dataset name was before you erased it:

```
CITRLINK / -d
```

'/' is a placeholder for the dataset name.

CITRLINK TSO command

Using CITRLINK as a TSO command type into the ISPF command line (or ISPF menu 6) the TSO command:

```
TSO CITRLINK datasetname
```

Bear in mind that for both methods CITRLINK has to be found in one of the concatenated ISPFLIBs. If not, you can also use the command:

```
TSO EXECUTE "rexx.library(CITRLINK)" datasetname
```

Running CITRLINK as a batch job

To process big CICS formatted trace datasets it makes sense to run the analysis done by CITRLINK as a batch job.

Use the following JCL to do so:

```
//CITRLINK EXEC PGM=IKJEFTØ1
//SYSEXEC  DD DSN=your.rexx.ISPELIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//CITROUT  DD SYSOUT=*
```

or:

```
//CITROUT  DD DISP=(NEW,CATLG),UNIT=SYSDA,
//            DSN=userid.CITRLINK.TEMP,
//            SPACE=(CYL,(1,2),RLSE),LRECL=8Ø,RECFM=FBA
//SYSTSIN  DD *
   CITRLINK datasetname [-d]
```

SAMPLE OUTPUT FROM REXX CITRLINK

The following sample output shows how CITRLINK formats a CICS trace.

The result is an overview report that shows the EXEC CICS LINK structure of the transactions in the trace.

```
CICS - AUXILIARY TRACE FROM Ø1/Ø2/Ø5 - APPLID CICSTSØB

CITRLINK Statistics
  Run Date / Time..: Ø3.Ø2.Ø5 / 21:16:Ø9
  Input Dataset....: SY.CICSTSØB.DØ5Ø2Ø1.T2ØØ114.TRACEA.TEMPW


The following (24) tasks have been found:
  91217  Tranid=UDIA, Called programs: 41
  91218  Tranid=W2$$, Called programs: 34
  91219  Tranid=W2$$, Called programs: 34
  9122Ø  Tranid=CSMI, Called programs: 1
  …
  …
  9124Ø  Tranid=W2$$, Called programs: 41

All programs marked with (*) contain non-threadsafe CICS commands!

Taskid=91217, Tranid=UDIA, Userid=TESTØ1

HUKDYP                                                          =ØØ7388=
! TP5ØØSI                                                       =ØØ74ØØ=
CI5ØØHM(*)                                                      =ØØ7442=
! TP5Ø1TZ(*)                                                    =ØØ7466=
! ! TP5Ø1TZ in CICSTST2                                         =ØØ7491=
! ZS51ØSE                                                       =ØØ7574=
! ! TP5ØØSI                                                     =ØØ7593=
! ZS51ØGD(*)                                                    =ØØ7644=
! AL5ØØID                                                       =ØØ774Ø=
! ! TP5ØØSI                                                     =ØØ7753=
! ! TP5Ø1TZ(*)                                                  =ØØ7786=
! SE7ØØBI                                                       =ØØ7833=
! ! SE7Ø1IB                                                     =ØØ7846=
! ! ! AP5ØØTI                                                   =ØØ7859=
! ! ! TP5Ø1TZ(*)                                                =ØØ7882=
! ! ! DM7Ø1MM(*)                                                =ØØ7917=
! ! ! ! DM765AL                                                 =ØØ7966=
! ! ! DM7Ø1MM                                                   =ØØ8Ø31=
! ! ! ! DM765UI                                                 =ØØ8Ø44=
! TP5ØØSI                                                       =ØØ8149=
! ZS51ØSE                                                       =ØØ8182=
! ! DM799ZS                                                     =ØØ82Ø1=
! ZS51ØST                                                       =ØØ8264=
! AP5ØØTI                                                       =ØØ83Ø1=
! DS5Ø1AN                                                       =ØØ8324=
! ! PM5Ø1KS                                                     =ØØ8337=
! ! DS59ØPR                                                     =ØØ8362=
! VK5ØØT1                                                       =ØØ8399=
! ! ZS52ØDT                                                     =ØØ8412=
! ! VK5Ø4TP                                                     =ØØ8437=
! ! VK5Ø3LS                                                     =ØØ8462=
! ! VK5Ø4TP                                                     =ØØ8487=
! ! ! VK5Ø3LS                                                   =ØØ85ØØ=
! ! ! ZS52ØDT                                                   =ØØ8525=
! ! ! DM765DB                                                   =ØØ855Ø=
! ! AP5ØØTI                                                     =ØØ86Ø7=
! ! VK5Ø3LS                                                     =ØØ863Ø=
! ! ZS52ØDT                                                     =ØØ8655=
! ! ! ZS52ØDL                                                   =ØØ8668=
! ZS51ØGD(*)                                                    =ØØ8717=
! ! AP5ØØTI                                                     =ØØ8734=
HUKDYP_END


OVERVIEW OF USED EXEC CICS COMMANDS AND USE COUNTS

  CICS Commands                  Count Threadsafe
  _____________________________  _____ __________
  EXEC CICS ADDRESS . . . . . .     11 Yes
  EXEC CICS ASSIGN. . . . . . .      5 Yes
  EXEC CICS FREEMAIN. . . . . .      8 Yes
  EXEC CICS GETMAIN . . . . . .     11 Yes
  EXEC CICS INQUIRE-PROGRAM . .      1 No
  EXEC CICS INQUIRE-TERMINAL. .      3 No
  EXEC CICS INQUIRE-TRACEDEST .      1 No
  EXEC CICS LINK. . . . . . . .     39 Yes
  EXEC CICS LOAD. . . . . . . .      1 Yes
  EXEC CICS MONITOR-POINT . . .      3 No
  EXEC CICS READ. . . . . . . .      3 No
  EXEC CICS RECEIVE-TC. . . . .      1 No
  EXEC CICS RELEASE . . . . . .      1 Yes
  EXEC CICS RETURN. . . . . . .     4Ø Yes
  EXEC CICS REWRITE . . . . . .      2 No
  _____________________________  _____ __________
  *Total EXEC CICS*                13Ø
  *Total Threadsafe Cmds*          116  89.2 %
  *Total non-Threadsafe Cmds*       14  1Ø.8 %


OVERVIEW OF USED EXEC SQL COMMANDS AND USE COUNTS

  SQL Commands                   Count
  _____________________________  _____
  EXEC SQL CLOSE. . . . . . . .      1
  EXEC SQL FETCH. . . . . . . .      2
  EXEC SQL OPEN . . . . . . . .      1
  EXEC SQL SELECT . . . . . . .      3
  _____________________________  _____
  *Total EXEC SQL*                   7


OVERVIEW OF CALCULATED TCB SWITCHES QR - L8

  *Total switches in CICS TS 1.3 and below*         14
  *Total switches in CICS TS 2.2 and above*          2
   (when all programs are defined with
    COncurency(Threadsafe).)

  *This means a possible reduction of*            85.7 %


PROGRAM INDEX:
  SORTED BY PROGRAM NAME              SORTED DESCENDING BY PROGRAM USAGE

  Pgm name      Count                     Pgm name      Count
  ___________   _____                     ___________   _____
  AL5ØØID           1                     AP5ØØTI           4
  AP5ØØTI           4                     TP5ØØSI           4
  CI5ØØHM(*)        1                     TP5Ø1TZ(*)        4
  DM7Ø1MM(*)        2                     VK5Ø3LS           3
  DM765AL           1                     ZS52ØDT           3
  DM765DB           1                     DM7Ø1MM(*)        2
  DM765UI           1                     VK5Ø4TP           2
  DM799ZS           1                     ZS51ØGD(*)        2
  DS5Ø1AN           1                     ZS51ØSE           2
  DS59ØPR           1                     AL5ØØID           1
  HUKDYP            1                     CI5ØØHM(*)        1
  PM5Ø1KS           1                     DM765AL           1
  SE7ØØBI           1                     DM765DB           1
  SE7Ø1IB           1                     DM765UI           1
  TP5ØØSI           4                     DM799ZS           1
  TP5Ø1TZ(*)        4                     DS5Ø1AN           1
  VK5ØØT1           1                     DS59ØPR           1
  VK5Ø3LS           3                     HUKDYP            1
  VK5Ø4TP           2                     PM5Ø1KS           1
  ZS51ØGD(*)        2                     SE7ØØBI           1
  ZS51ØSE           2                     SE7Ø1IB           1
  ZS51ØST           1                     VK5ØØT1           1
  ZS52ØDL           1                     ZS51ØST           1
  ZS52ØDT           3                     ZS52ØDL           1
  ___________   _____                     ___________   _____
  *Total*          41

Taskid=91218, Tranid=W2$$, Userid=TESTØ2

… and so on …
…
```

In this little example one could raise the question of why three programs – AL500TI, TP500SI, and TP501TZ – have been called four times each.

The next question could be, why are four programs marked as non-threadsafe? This can be analysed with the detail output.

SAMPLE DETAIL OUTPUT FROM REXX CITRLINK

The information in the detail output report looks similar to the overview output from REXX CITRLINK. The difference is that all found EXEC CICS and EXEC SQL calls are formatted. So it can easily be seen which commands are used by which program. This is much easier to see than in a native CICS trace.

```
Taskid=91217, Tranid=UDIA, Userid=AP3Ø2

HUKDYP                                                          =ØØ7388=
! TP5ØØSI                                                       =ØØ74ØØ=
! ! EXEC CICS ASSIGN                                            =ØØ7413=
! ! EXEC CICS ADDRESS                                           =ØØ7419=
! ! EXEC CICS RETURN                                            =007423=
! ! EXEC CICS RETURN                                            =007433=
CI500HM(*)                                                      =007442=
! EXEC CICS ADDRESS                                             =007456=
! EXEC CICS GETMAIN                                             =007460=
! TP501TZ(*)                                                    =007466=
! ! EXEC CICS ADDRESS                                           =007479=
! ! EXEC CICS INQUIRE-TERMINAL(!)                               =007483=
! ! TP501TZ in CICSTST2                                         =007491=
! ! EXEC CICS INQUIRE-TRACEDEST(!)                              =007550=
! ! EXEC CICS RETURN                                            =007558=
! EXEC CICS ASSIGN                                              =007568=
! ZS510SE                                                       =007574=
! ! EXEC CICS GETMAIN                                           =007587=
! ! TP500SI                                                     =007593=
! ! ! EXEC CICS ASSIGN                                          =007606=
! ! ! EXEC CICS ADDRESS                                         =007612=
! ! ! EXEC CICS RETURN                                          =007616=
! ! EXEC CICS RETURN                                            =007626=
! ! EXEC CICS FREEMAIN                                          =007630=
! ZS510GD(*)                                                    =007644=
! ! EXEC CICS ADDRESS                                           =007657=
! ! EXEC CICS GETMAIN                                           =007661=
! ! EXEC CICS READ(!) UDMZ2VD                                   =007667=
! ! EXEC CICS RETURN                                            =007722=
! EXEC CICS RECEIVE-TC(!)                                       =007734=
! AL500ID                                                       =007740=
! ! TP500SI                                                     =007753=
! ! ! EXEC CICS ASSIGN                                          =007766=
! ! ! EXEC CICS ADDRESS                                         =007772=
! ! ! EXEC CICS RETURN                                          =007776=
! ! TP501TZ(*)                                                  =007786=
! ! ! EXEC CICS ADDRESS                                         =007799=
! ! ! EXEC CICS INQUIRE-TERMINAL(!)                             =007803=
! ! ! EXEC CICS RETURN                                          =007811=
! ! EXEC CICS RETURN                                            =007821=
! SE700BI                                                       =007833=
! ! SE701IB                                                     =007846=
! ! ! AP500TI                                                   =007859=
! ! ! ! EXEC CICS RETURN                                        =007872=
! ! ! TP501TZ(*)                                                =007882=
! ! ! ! EXEC CICS ADDRESS                                       =007895=
! ! ! ! EXEC CICS INQUIRE-TERMINAL(!)                           =007899=
! ! ! ! EXEC CICS RETURN                                        =007907=
! ! ! DM701MM(*)                                                =007917=
! ! ! ! EXEC CICS LOAD TABCICS                                  =007932=
! ! ! ! EXEC CICS INQUIRE-PROGRAM(!)                            =007942=
! ! ! ! EXEC CICS RELEASE TABCICS                               =007954=
…
… and so on …
…
```

CODE

```
/*                         REXX                                 */
 doc=,
"*--------------------------------------------------------------*   "!!,
"*%COPYRIGHT (c) 1997, 2005 HUK Coburg, Systems programmer DB/DC*   "!!,
"*--------------------------------------------------------------*   "!!,
"*  Created by H.-J.Gerdes 1997, 2005                           *   "!!,
"*--------------------------------------------------------------*   "!!,
"*  The purpose of this REXX is to visualize the EXEC CICS LINK *   "!!,
"*  structure of CICS transactions in a pseudo-graphic manner.  *   "!!,
"*                                                              *   "!!,
"*  So the nesting of EXEC LINKing during the execution of tran-*   "!!,
"*  saction can be made visible.                                *   "!!,
"*                                                              *   "!!,
"*  Primarily this REXX can be used by application developers to*   "!!,
"*  check their application during runtime for potential tuning *   "!!,
"*  areas, eg using the same EXEC CICS or EXEC                  *   "!!,
"*  SQL over and over again.                                    *   "!!,
"*                                                              *   "!!,
"*  The other main purpose of this tool is to analyze the poten-*   "!!,
"*  tial positive impact when migration to CICS TS 2.3 and above.*  "!!,
"*  All TCB switches of transactions will be analysed and it    *   "!!,
"*  will be possible to detect why a CICS transaction is        *   "!!,
"*  switching back from L8 to QR. With the detail option, non-  *   "!!,
"*  threadsafe CICS commands can be detected and in which       *   "!!,
"*    circumstances they are being used.                        *   "!!,
"*  The output of this tool will be structured in the           *   "!!,
"*     following way:                                           *   "!!,
"*  1. A list of all transactions which have been found in the  *   "!!,
"*     processed CICS trace will be shown.                      *   "!!,
"*                                                              *   "!!,
"*  2. All found transactions will then be further investigated *   "!!,
"*     regarding their LINK structure. Each shown command is    *   "!!,
"*     associated with the original CICS trace line number so   *   "!!,
"*     that it will be easy to find the line in the original    *   "!!,
"*     CICS trace for further analysis.                         *   "!!,
"*     Programs using non-threadsafe EXEC CICS commands will be *   "!!,
"*     marked with '(*)'.                                       *   "!!,
"*     The available detail option of this tool (-d) additionally*  "!!,
"*     gives you all EXEC CICS and EXEC SQL that have been used *   "!!,
"*     by the transaction. All non-threadsafe commands will be  *   "!!,
"*     marked with an '(!)' as well.                            *   "!!,
"*     Note: Even threadsafe commands can be non-threadsafe, if *   "!!,
"*     they are used in function shipping requests.             *   "!!,
"*                                                              *   "!!,
"*     Some CICS commands will show additional information:     *   "!!,
"*      - Program names for LOAD and RELEASE                    *   "!!,
"*      - File names for VSAM requests. Here you should note,   *   "!!,
"*        that the file name for a specific file only occurs    *   "!!,
"*        with the first request to this file. Subsequent file  *   "!!,
"*        requests to the same file won't show a trace entry    *   "!!,
"*        with the file name so this tool has no chance of      *   "!!,
"*        picking up the name.                                  *   "!!,
"*      - DPL LINKed programs will show the applid of the       *   "!!,
"*        target CICS region.                                   *   "!!,
"*                                                              *   "!!,
"*  3. For each analysed and ended transaction, the tool shows  *   "!!,
"*     an overview of the following:                            *   "!!,
"*                                                              *   "!!,
"*     a) Overview and counts of all used EXEC CICS commands    *   "!!,
"*                                                              *   "!!,
"*     b) Overview and counts of all used EXEC SQL commands     *   "!!,
"*                                                              *   "!!,
"*     c) Calculated TCB switches without and with the use of   *   "!!,
"*        COncurrency(Threadsafe)                               *   "!!,
"*                                                              *   "!!,
"*     d) Two sorted program indexes - one showing all used     *   "!!,
"*        programs in alphabetical order including use counts.  *   "!!,
"*        The second program index shows all used programs in   *   "!!,
"*        descending order by use count.                        *   "!!,
"*        So this is a very good point to check the number of   *   "!!,
"*        times individual programs are being used and whether  *   "!!,
"*        this makes sense from a performance standpoint or not.*   "!!,
"*                                                              *   "!!,
"*  For analysis the program uses a CICS formatted trace with   *   "!!,
"*  ABBREV option. Trace formatted with FULL or SHORT cannot be *   "!!,
"*   processed.                                                 *   "!!,
"*                                                              *   "!!,
"*  The CICS trace has to be active for at least the following  *   "!!,
"*  six domains to allow correct execution of this tool:        *   "!!,
"*  - AP   Application domain                                   *   "!!,
"*  - EI   EXEC interface                                       *   "!!,
"*  - IS   Inter-system communication                           *   "!!,
"*  - PG   Program manager domain                               *   "!!,
"*  - RI   Resource manager interface                           *   "!!,
"*  - XS   Security manager domain                              *   "!!,
"*                                                              *   "!!,
"*  Having activated trace only for the above mentioned domains *   "!!,
"*  will reduce the number of written trace entries by about 4Ø *   "!!,
"*   percent.                                                   *   "!!,
"*                                                              *   "!!,
"*  The formatted trace can be produced with                    *   "!!,
"*  1. DFHTUvrm the CICS trace utility                          *   "!!,
"*                                                              *   "!!,
"*  2. DFHPDvrm the CICS IPCS verbexit or with                  *   "!!,
"*        VERBEXIT DFHPD63Ø 'DEF=Ø,IND=1,TR=1'                  *   "!!,
"*                                                              *   "!!,
"*  3. GTFTRACE and the CICS option.                            *   "!!,
"*        GTFTRACE CICS((ABBREV)) USR(ALL)                      *   "!!,
"*                                                              *   "!!,
"*  (where vrm means version, release, modification level,      *   "!!,
"*  eg 63Ø for TS 2.3).                                         *   "!!,
"*                                                              *   "!!,
"*  Note: The list of threadsafe EXEC CICS and SPI commands at  *   "!!,
"*  the end of the program has to be updated with upcoming CICS *   "!!,
"*  releases. The currently used table is for CICS TS 2.3.      *   "!!,
"*                                                              *   "!!,
"*-------------------------------------------------------------*    "!!,
"*  Using the program:                                          *   "!!,
"*   1. As a TSO command                                        *   "!!,
"*      TSO CITRLINK 'dsn' <-d>                                 *   "!!,
"*                                                              *   "!!,
"*   2. As a line command in the ISPF dataset list panel in     *   "!!,
"*      front of the dataset which contains the CICS formatted  *   "!!,
"*       ABBREV trace.                                          *   "!!,
"*                                                              *   "!!,
"*   3. As a batch job using IKJEFTØ1 and the commands and      *   "!!,
"*      parameters in SYSTSIN input dataset.                    *   "!!,
"*                                                              *   "!!,
"*      Example:                                                *   "!!,
"*      //TSO1     EXEC PGM=IKJEFTØ1                            *   "!!,
"*      //SYSEXEC  DD DSN=your.exec.library,DISP=SHR            *   "!!,
"*      //SYSPRINT DD SYSOUT=*                                  *   "!!,
"*      //SYSTSPRT DD SYSOUT=*                                  *   "!!,
"*      //CITROUT  DD DISP=(NEW,CATLG),UNIT=SYSDA,<== can be spec*  "!!,
"*      //            DSN=analyzed.trace.output,                *   "!!,
"*      //            SPACE=(CYL,(1,2),RLSE),LRECL=8Ø,RECFM=FBA *   "!!,
"*      //SYSTSIN  DD *                                         *   "!!,
"*         CITRLINK 'formatted.cics.trace' <-d>                 *   "!!,
"*                                                              *   "!!,
"*-------------------------------------------------------------*    "!!,
"*  Arguments:                                                  *   "!!,
"*   dsn    - Dataset name which contains the CICS ABBREV trace *   "!!,
"*   ?      - Show comment for this program                     *   "!!,
"*                                                              *   "!!,
"*  optional parameters:                                        *   "!!,
"*   -d     - Detail information for EXEC CICS and EXEC SQL     *   "!!,
"*-------------------------------------------------------------*    "!!,
"*  External references:                                        *   "!!,
"*  REXX procedures:                                            *   "!!,
"*   DEBUG       - Interactive deugging                         *   "!!,
"*  Datasets:                                                   *   "!!,
"*   CITRIN      - Input DSN                                    *   "!!,
"*   CITROUT     - Output DSN                                   *   "!!,
"*-------------------------------------------------------------*    "
/*EOC                                                                */
/*  Changes:                                                         */
/*-------------------------------------------------------------------*/
  proc  = 'CITRLINK'
```

```
u     = 'REXX'
if (sysvar(sysispf) = 'ACTIVE') then
  "ISPEXEC VGET (DEBUG)"
if (debug = 'ON') then
  interpret "trace "debug(u,'INIT',proc)""
  /*----------------------------------------------------------------*/
 /* Output of the REXX documentation, taking parms and checking    */
 /* runtime.                                                        */
  /*----------------------------------------------------------------*/
parse upper arg INPARM
call REXXENV sysvar(sysnest),
            ,sysvar(sysicmd),
            ,MACRO NMACRO,           /* TSO und ISPF Umgebung       */
            ,INPARM
parse var result RCODE ENV dsn p2
if (debug = 'ON') then
  interpret "trace "debug(u,'TOPT',proc)""
if RCODE >= 16 then
do
  call message  'Wrong runtime or error in EXEC REXXENV'
  signal ENDE
end
 /*-----------------------------------------------------------------*/
/* Analyse parameters                                               */
 /*-----------------------------------------------------------------*/
call CHECK_PARMS dsn p2
if (rc > 0) then signal ende
 /*-----------------------------------------------------------------*/
/* Allocate input dataset                                           */
 /*-----------------------------------------------------------------*/
call allok 'INPUT'  dsn              /* Allocating input dataset    */
if (rc > 0) then
do
  call message 'Input dataset cannot be allocated.         ',
               'Error messages:                            ',
               outline.1 outline.2
  signal ende
end
else
  address TSO
  "EXECIO 1 DISKR CITRIN (STEM in.)"
call allok 'OUTPUT'                  /* Allocating outut dataset     */
komma = ','
ist   = '='
kl_auf= '('
kl_zu = ')'
mark = '! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! '
o     = 1
```

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

Hans-Joachim Gerdes
CICS Systems Programmer
HUK-COBURG Versicherungen (Germany)

CICS questions and answers

Q We have a CICS region that we use to hold some central functions (for example: postcode look-up) – all our other CICS regions use DPL to LINK to these functions. When we monitor this region, all we can see is lots of CSMI transactions running. Is there anything we can do to easily determine where requests are coming from and limit certain 'types' of request?

A Using the TRANSID option of the *Remote Attributes* section on a program definition allows you to 'identify' different calling systems and functions. This means that monitoring, reporting, debugging, etc is made easier because the transaction IDs help identify what the transaction is doing. CEMT will give more meaningful displays than just a list of CSMI transactions running.

The 'sending' systems need to have a TRANSID added to each *Remote Program* definition, and the 'receiving' system needs a definition for this transaction (this should be a copy of CSMI). Choose transaction IDs that identify the calling system and the function – for example POSA for postcodes from CICSA, POSB for postcodes from CICSB, and so on.

By using transactions you can limit the requests using TRANCLASSes rather than limit the number of sessions on a link, which gives you better control over what the region does.

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

CICS news

Attachmate has announced Version 2.0 of Synapta Services Builder for CICS 3270 Bridge.

The new tool gives developers direct access to 3270 applications residing in multiple CICS regions, via CICS Transaction Server V1.3 or V2.2. This allows new re-useable services to be built that can be part of a Service-Oriented Architecture (SOA). Synapta uses a lightweight programmatic approach to session and state management, which, the company claims, reduces development time and CPU load.

Synapta is a mainframe-resident adapter that runs in the CICS Transaction Server for rapid processing of CICS applications. It supports Multi-Region Operations (MRO) and terminal ID management for better performance.

Additionally, Synapta Services Builder for CICS 3270 Bridge supports CICS Transaction Server 2.2+, a mechanism of CICS integration known as the 'linkable bridge'. The product also offers centralized Web-based management and control.

The product integrates into the Microsoft Windows, Java, and Microsoft .NET interfaces. It has a data-driven resource adaptor and a stateful EJB to be used when integrating the CICS 3270 Bridge into J2EE application server environments.

For further information contact:
www.synapta.com.

* * *

More information about the recently-announced Version 6.0 of WebSphere MQ is emerging from IBM.

This new release of WebSphere MQ contains support for a broad range of operating system and hardware platforms, including Linux on POWER architecture. The new release also includes integrated standards support, with support for JMS 1.1 and .NET classes. JMS support allows WebSphere MQ to communicate with more messaging formats. Also, SOAP messages can now be transported over WebSphere MQ, enabling Web services over WebSphere MQ to provide reliable messaging, versus the use of more unreliable transports such as HTTP.

WebSphere MQ 6.0 will allow a user to work across multiple computing platforms. The software includes facilities to connect to Oracle, SAP, and Siebel applications. Mainframe systems such as CICS and IMS can be integrated as well.

The new version also includes dynamic configuration of z/OS resources. Changes to system resources such as buffer pools and buffers can be made without restarting the queue manager.

For further information contact:
URL: www.ibm.com/software/integration/wmq.

* * *

ClearNova has announced Version 6.0 of ThinkCAP Professional, its rapid application development platform. ThinkCAP is designed to simplify and accelerate Web development by creating new applications or Web-enabling existing ones. JCA and Web Services adapters are available to more than 50 different systems and formats including CICS, SAP, Seibel, Peoplesoft, and more.

For further information contact:
URL: www.clearnova.com.