235

CICS

June 2005

In this issue

- 3 Using POVI for zapping TCTTE
- 8 CICS VSAM Recovery
- 20 <u>CICS Transaction Gateway –</u> <u>strategic JCA connector with</u> <u>WebSphere Application Server</u>
- 37 <u>3270 transactions via CICS</u> Web Interface (CWI)
- 46 CICS news

© Xephon Inc 2005



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

ColinSmith 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.

© Xephon Inc 2005. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher. *Printed in England.*

Using POVI for zapping TCTTE

In CICS regions, we find that, for a variety of reasons, terminals are sometimes stuck with the status of 'Being Acquired'. In such cases, inactivating and then reactivating the terminal from VTAM may help to release it, making it operational again. But sometimes the inact/act operation does not help, in which case you need to recycle the CICS region in order to make the terminal operational in that region. If it is a development or test environment CICS region, you can recycle it at any time. However, in a production environment you need to wait for a suitable time and until then the terminal will be unusable in that CICS region.

However, there is another way to release such a terminal from the CICS region. The cause of the problem is that the values at offsets 1CD and 1CF of the TCTTE address are inconsistent. So if these are set to 00, the terminal is released from the CICS region and it no longer has the status of 'Being Acquired'.

One way to change the values of the offsets of the TCTTE is to use POVI (Programmerless Open VTAM Interface) with Omegamon/CICS. POVI is a facility that is supported by AF/ OPERATOR. It lets you automate interactions with VTAM 3270 applications from AF/OPERATOR and provides REXXbased methods to accomplish this task.

The program that uses POVI to change the values at the required offsets of the TCTTE is listed below. This program is invoked by giving a user-defined command from the console. (AF/OPERATOR captures the user-defined command and runs the following program.) In the command, the CICS ID and termID are given. The CICS ID should already be defined to the POVI interface.

The AF/OPERATOR trap for invoking the POVI program is:

TRAP DEL(CHGTCT)
TRAP ADD(CHGTCT) +

```
CMD('CHGTCT *CICSID *TERMID') ENA +
ACT('EX CHGTCTR ''&&CICSID &&TERMID''')
```

CHGTCT – POVI PROGRAM

```
/* REXX */
/* When related command is captured by AF/OPER, this program is
   invoked; and as parameters, the CICS ID and the terminal ID
   (which is stuck in "Being Acquired" status) is passed.
   In the CICS region, the values at offset "1CD" and "1CF" of
   the TCTTE owned by the related terminal are set as "\emptyset \emptyset". */
/* Note1: This program can be used only for LUØ terminals. */
/* Note2: This program uses POVI */
Parse Arg cicsid, termid
Address mvs
NEWSTACK
RET = SYSVGET('AOSYSID')
say 'Sysid is:' AOSYSID
if LENGTH(cicsid) = 8 then cics_sysid = substr(cicsid,5,4)
else cics_sysid = substr(cicsid,1,4)
/* form name of the Omegamon/CICS session ID that is defined
   for POVI. The session ID changes according to CICS ID.
   This session should be defined before at POVI for each CICS. */
omegcics = 'OMCI' || cics_sysid
/* Decide applid of POVI address space according to the sysid at which
   this program runs */
if AOSYSID = 'TD90' then do
   povilu = 'povit90p'
end
if AOSYSID = 'TD91' then do
   povilu = 'povit91p'
end
if AOSYSID = 'PR90' then do
   povilu = 'povip90p'
end
if AOSYSID = 'PR91' then do
   povilu = 'povip91p'
end
Say 'CICS Id ==>' cicsid
```

```
Say 'Terminal Id ==>' termid
Say 'System id ==>' AOSYSID
Say 'CICS sysid ==>' cics_sysid
Say 'Omegcics ==>' omegcics
Say 'povilu
               ==>' povilu
call Omega_Logoff /* Logoff from POVI sessions if logged on */
call Omega_Logon /* Logon Omegamon/CICS through POVI */
call Zap TCTTE termid /* Determine the TCTTE address and zap the
                        related offset */
call Omega_Logoff /* Logoff from POVI sessions */
exit Ø
Omega_Logon:
   /* Subroutine for logging on Omegamon/CICS through POVI */
   /* POVI Playback Manager ACB
                                                 : povilu
                                                                */
   /* Omegamon/CICS session id that is defined
                                        at POVI : OMEGCICS
                                                                */
   /* Username for logging on Omegamon/CICS
                                                : xxuserid
                                                               */
   /* Password for logging on Omegamon/CICS
                                                 : xxpass
                                                               */
   /*
        Userid and password are hardcoded below and the user
         should be able to perform authorized actions. */
   toolsrc =POVILGON(povilu,omegcics)
   IF (toolsrc \langle \rangle \emptyset) then call error_routine
   toolsrc =POVIFIND(TOP,'ENTER USERID ==> ',)
   IF (toolsrc \langle \rangle \emptyset) then call error_routine
   toolsrc=POVISEND('xxuserid',ENTER,'PASSWORD ==> ')
   IF (toolsrc \langle \rangle 0) then call error_routine
   toolsrc=POVISEND('xxpass',ENTER,'ZMENU',,,CONFIDENTIAL)
   IF (toolsrc \langle \rangle \emptyset) then call error_routine
return
Omega_Logoff:
  /* Subroutine for logging off from POVI */
   DELSTACK
   toolsrc =POVILGOF()
return
Zap_TCTTE:
   /* This subroutine determines the TCTTE address of the terminal
```

```
in the CICS region and zaps the related offsets as "\emptyset\emptyset". */
 arg termid
/* Determine the TCTTE address of the terminal that is in
   "Being Acquired" state. */
  toolsrc=POVILOC(ROWCOL,1,2)
  toolsrc=POVISEND('ZTCT',ENTER,' ZTCT',1,18)
  IF (toolsrc \langle \rangle \emptyset) then call error_routine
  toolsrc=POVILOC(ROWCOL,10,2)
  toolsrc=POVISEND('TABL TCT, ID='termid, ENTER, 'Terminal ID', 12, 8)
 IF (toolsrc \langle \rangle \emptyset) then call error_routine
  toolsrc=POVIDATA(12,72,8)
 IF (toolsrc \langle \rangle \emptyset) then call error_routine
 parse pull TCTTE_addr
 Say 'TCTTE_addr ==>' TCTTE_addr
/* Calculate the address that will be zapped */
 TCTTE_addr_sub7 = substr(TCTTE_addr,2,7)
 TCTTE_addr_sub7_dec = X2D(TCTTE_addr_sub7)
  Limit_Addr_Val = X2D('FFFFFFF')-X2D('1CC')
 /* Limit_Addr_Val = 268434995 */
 Say 'Limit_Addr_Val ==>' Limit_Addr_Val
 /* Determine if the address value is beyond the computable value */
  IF TCTTE_addr_sub7_dec >= Limit_Addr_Val then call error_routine
  TCTTE_addr_sub7_dec_new = TCTTE_addr_sub7_dec + X2D('1CC')
  TCTTE_addr_sub7_new = D2X(TCTTE_addr_sub7_dec_new)
  TCTTE_addr_new = substr(TCTTE_addr,1,1) || TCTTE_addr_sub7_new
 Say 'TCTTE_addr_new ==>' TCTTE_addr_new
/* Get the current value at the related offset of the TCTTE.
   It is a 4-byte value and 2 bytes of it will be zapped. */
  toolsrc=POVILOC(ROWCOL,1,2)
  toolsrc=POVISEND('ZUXMLS',ENTER,' ZUXMLS',1,18)
  IF (toolsrc \langle \rangle \emptyset) then call error_routine
  toolsrc=POVILOC(ROWCOL,13,8)
  toolsrc=POVISEND('~, 'TCTTE_addr_new',4',ENTER,'Storage at',14,2)
  IF (toolsrc \langle \rangle \emptyset) then call error_routine
```

```
toolsrc=POVIDATA(15,9,8)
  IF (toolsrc \langle \rangle 0) then call error_routine
  parse pull TCTTE_val
  Say 'TCTTE_val ==>' TCTTE_val
 /* Form the new value for the related offset of the TCTTE */
  TCTTE_val_new = substr(TCTTE_val,1,2) || 'ØØ' ||,
                  substr(TCTTE_val,5,2) || '00'
  Say 'TCTTE_val_new ==>' TCTTE_val_new
 /* Zap the related offset of the TCTTE */.
   toolsrc=POVILOC(ROWCOL,1,2)
   toolsrc=POVISEND('ZUXZAP',ENTER,' ZUXZAP',1,18)
  IF (toolsrc \langle \rangle \emptyset) then call error_routine
  blank='
  zapcmd = '-XMZP ~,' || TCTTE_addr_new || ',' || TCTTE_val || ',' ||,
           TCTTE_val_new || blank
  Say 'zapcmd ==>' zapcmd
  toolsrc=POVILOC(ROWCOL,13,1)
   toolsrc=POVISEND(zapcmd,ENTER,'Memory Zap Successful',14,20)
  IF (toolsrc \langle \rangle \emptyset) then call error_routine
return
ERROR_ROUTINE:
  say 'Error occurred while processing..'
  PARSE SOURCE tso invoke_type exec_name the_rest
  CALL GLBVGET('aopvfnc')
  PARSE PULL feedback
  say 'Omeg CICS report error:'
  Say '
        Trap action =>' exec_name
  Say '
                      =>' sigl
            Line
  Say '
            Function =>' aopvfnc
  Say '
           Return code =>' toolsrc
  Say ' Feedback =>' feedback
  call Omega_logoff
  EXIT -1
return
```

Erhan Pasa Senior System Programmer Akbank TAS (Turkey)

© Xephon 2005

CICS VSAM Recovery

OVERVIEW

CICS VSAM Recovery, usually referred to as CICSVR, is an IBM program product that can be used to automate the recovery of corrupt or lost VSAM datasets. It works against datasets used in batch or for batch and online datasets.

CICSVR is aimed at organizations in which the availability and integrity of VSAM data is critical to day-to-day operations. CICSVR provides an ISPF panel interface that can be used to automate the creation of the jobs to be used in recovery. Prior to CICSVR 3.1, the product was normally used as a CICS-only data recovery tool. However, from Version 3.1 of CICSVR onwards, it also provides recovery for VSAM batch applications as well as online CICS files.

This article intends to introduce the basic concepts of the product and also details how to install it on an MVS system.

The programs that are utilized by CICSVR are shown below; the program name is followed by its associated command name or utility:

- DWWAR LOGOFLOGS SCAN command
- DWWCO RECOVER command
- DWWCA CA command
- DWWLC LOGSTREAMCOPY command
- DWWGJCDS RCDS command
- DWWMIW migration utility program.

CICSVR is designed so that it can be used via ISPF. By navigating through a series of panels you can recover VSAM datasets. From the ISPF panels you can select which VSAM datasets you wish to recover and which back-up datasets to use. CICSVR will execute various commands in the background to retrieve the most up-to-date information about the VSAM datasets from the CICS regions they are being used by. DFSMShsm and DFSMSdss can also be called in the background to perform restoration of the relevant VSAM datasets from their latest back-ups. The ISPF dialog will also generate restore and forward recovery JCL.

Before forward recovery, you need to restore your damaged VSAM dataset from its most recent back-up copy. If you use DFSMShsm or DFSMSdss for backing up your VSAM datasets, CICSVR will restore the latest back-up copy. If you use something like FDRABR for back-up, a manual restoration will be required. You can then use the ISPF dialogs to build the forward recovery job. Program DWWCO is called to perform the recovery by reading after-images of VSAM records from the CICS log stream. It then applies them to the restored dataset in the sequence in which they were originally written to the datasets. This means that, on completion, the VSAM dataset should be at the exact point that it was at at the time of the failure.

You can use batch logging to log batch updates to VSAM datasets that are not accessed in record-level sharing mode. Any datasets required for batch logging must be closed to CICS. In order for batch logging to work the CICSVR address space must be active and all VSAM datasets that require batch logging must be SMS-managed and they must be defined with FRLOG(REDO) and LOGSTREAMID(logstreamname) parameters. If a problem occurs with your VSAM datasets after the batch process, you can use CICSVR to forward recover the updates recorded on the MVS system logger log stream.

PRODUCT INSTALLATION INFORMATION

To install the product, a number of base system configuration changes are required. The Link Pack Area requires update. It is essential that the LPALST is set up correctly. Make sure that the dataset library DWW.SDWWLPA (supplied as part of base CICSVR) is placed ahead of dataset hlq.SDWWDLPA (which was supplied by DFSMS PTF UW79809). UW79809 supplies dummy stubs for VSAM batch logging. Obviously the real CICSVR provides the real thing to allow the product to work.

The CICSVR address space will be started at IPL time normally. This is controlled by a parameter named CICSVR_INIT. This parameter is coded in SYS1.PARMLIB(IGDSMS*xx*).

The possible values of this parameter are **YES**, which indicates that CICSVR is to start at system initialization, and **NO**, which defers CICSVR start-up. If you specify CICSVR_INIT(NO), you will have to issue a SETSMS CICSVR_INIT(YES) command before you can activate the CICSVR region. For it to function correctly, every system in a SYSPLEX must have an IGDSMS*xx* set up for the CICSVR server address space. You should also define a prefix for all CICSVR-required datasets (or use the default value of DWW) by using the **CICSVR_DSNAME_PREFIX(hIq. sIq)** IGDSMS*xx* parameter . This prefix is used for the datasets that must be pre-allocated before you can start the CICSVR address space.

These are the datasets that are allocated to the DDnames DWWCON1, DWWCON2, and DWWCON3, and also DWWMSG. The DDnames used by CICSVR and a description of what they do is shown below:

- DWWCON1-3 CICSVR recovery control datasets (RCDS).
- DWWIN dataset containing the CICSVR commands. You can either specify a sequential dataset with 80-byte records or include the CICSVR commands in-stream.
- DWWLOAD load library for the CICSVR exits.
- DWWLOG used to allocate log datasets when not using the CICSVR ALLOCATE command.
- DWWMSG output dataset containing CICSVR

messages. It is used by the CICSVR address space, the CICSVR ISPF dialog, and CICSVR batch jobs.

- DWWSORT output dataset containing sort messages and control statements from the change accumulation job. Usually specified as SYSOUT=*.
- DWWPRINT output dataset containing the reports produced by CICSVR. It is used by the CICSVR ISPF dialog and CICSVR batch jobs.
- DWWDMSG output dataset containing tracing and diagnostic information produced by individual CICSVR subroutines, as requested by CICSVR. It is used by the CICSVR address space and the CICSVR ISPF dialog.
- DWWDUMP output dataset containing tracing and diagnosis information, as requested by CICSVR. It is used by the CICSVR address space and the CICSVR ISPF dialog.

The dataset DWW.SDWWLOAD has to be added to the LNKLST concatenation. This is required for the CICSVR address space. An entry in IFAPRD*xx* is required to enable CICSVR.

The DWWMSG dataset must also have a suffix of systemname. This systemname can be found in SYS1.PARMLIB(IEASYS*xx*) in the SYSNAME parameter. Other datasets that can be preallocated are DWWPRINT, DWWDMSG, and DWWDUMP. The latter are not needed for the CICSVR region, but IBM recommends that you do have them. They must have the same prefix (CICSVR_DSNAME_PREFIX) and the systemname suffix.

The CICSVR_DSNAME_PREFIX parameter is mandatory if you have CICSVR_INIT(YES). It is used to specify the prefix for dynamically-allocated RCDS dataset names.

Because CICSVR asks HSM to do accumulation processing for datasets that HSM has dumped, you need to be able to issue HSM authorized commands. Any users who will invoke the CICSVR ISPF dialogs will need an entry in the HSM ARCCMD*xx* SYS1.PARMLIB member coding, as follows:

AUTH userid DATABASEAUTHORITY(CONTROL)

You also need to define the Recovery Control Datasets (RCDS). These are a repository containing all the information necessary to perform a forward recovery for VSAM datasets. The RCDS are actually three linear VSAM datasets. They contain identical copies of the information that CICSVR requires for VSAM recovery. IBM recommends that the RCDS datasets be allocated on different volumes and use different disk controllers and channels. This will minimize the possibility of a hardware failure making all three datasets unusable. CICSVR can determine whether any of the three RCDS datasets are empty or have been corrupted and can copy the current information into a newly-created dataset. The JCL to define the RCDS is shown below:

//JXB7884S JOB	(7884),CLASS=A,MSGCLASS=H	I,MSGLEVEL=(1,1)
//STEP1 EXEC	PGM=IDCAMS	
//SYSPRINT DD	SYSOUT=*	
//SYSIN DD	*	
DEF CL (NAME(DWW	.DWWCON1)	-
	VOLUME(CICØØ1)	-
	CYLINDERS(5Ø 1Ø)	-
	LINEAR	-
	SHAREOPTIONS(3 3))	-
DEF CL (NAME(DWW.	DWWCON2)	-
	VOLUME(CICØ1Ø)	-
	CYLINDERS(5Ø 1Ø)	-
	LINEAR	-
	SHAREOPTIONS(3 3))	-
DEF CL (NAME(DWW.	DWWCON3)	-
	VOLUME(TOTCI3)	-
	CYLINDERS(150 30)	-
	LINEAR	-
	SHAREOPTIONS(3 3)	-

The size of the RCDS will depend on how much batch logging activity is taking place. A high number of datasets being logged and frequent open and close processing requires more RCDS space.

CICSVR uses an ISPF JCL skeleton to submit a job from ISPF.

This can be accessed from the ISPF CICSVR menu by selecting Option 5. The first time that the ISPF dialogs are accessed, this JCL should be altered to match your environment. Change the dataset name in the STEPLIB DDname and the RCDS dataset names in DWWCON1, DWWCON2, and DWWCON3. You may also wish to add DDnames for DWWDUMP, DWWDMSG, SYSMDUMP, and SYSUDUMP for problem determination.

When you save the new JCL skeleton, it will be stored in the library allocated to the ISPFILE DDname as member DWWUJOB. An example of the skeleton from our site is shown below:

//CIC7882&CJOBCHA	A JOB (78	882),MSGLEVEL=(1,1),NOTIFY=&&SYSUID,
//	MSGCL/	ASS=H,CLASS=A,REGION=8M
//DWW	PROC	
//STEP1	EXEC	PGM=DWWCO
//STEPLIB	DD	DSN=DWW.SDWWLOAD,DISP=SHR
//DWWMSG	DD	SYSOUT=*
//DWWPRINT	DD	SYSOUT=*
//DWWDUMP	DD	SYSOUT=*
//DWWDMSG	DD	SYSOUT=*
//SYSMDUMP	DD	SYSOUT=*
//SYSUDUMP	DD	SYSOUT=*
//DWWCON1	DD	DSN=DWW.DWWCON1,DISP=SHR
//DWWCON2	DD	DSN=DWW.DWWCON2,DISP=SHR
//DWWCON3	DD	DSN=DWW.DWWCON3,DISP=SHR
//	PEND	

The CICSVR address space is an integral part of CICSVR and it is used for communicating with DFSMShsm and DFSMSdss. It can be started at IPL time if the relevant parameters have been coded in SYS1.PARMLIB.

The address space can be terminated and reactivated by the following MVS system commands:

V SMS,CICSVR,TERMINATESERVER

V SMS,CICSVR.ACTIVE

The output of these commands is displayed in the syslog and may be required for problem determination. Examples of the commands and expected output are shown below. The CICSVR server address space is required only for batch logging or if you are using DFSMSdss for your back-ups.

Below are examples of SMS VARY commands to control the CICSVR address space and the expected output:

V SMS, CICSVR, TERMINATESERVER DWW172I REQUEST TO TERMINATE CICSVR ADDRESS 491 SPACE IS ACCEPTED: CICSVR SERVER TERMINATION SCHEDULED. DWW21ØI CICSVR HISTORY SCAVENGE STARTED 492 ON SYSTEM: CMØ1 DWW21ØI CICSVR HISTORY SCAVENGE FINISHED 493 ON SYSTEM: CMØ1 DWWØØ8I CICSVR SERVER SUCCESSFULLY TERMINATED AT END OF MEMORY. V SMS,CICSVR,ACTIVE DWWØ15I CICSVR SERVER ADDRESS SPACE HAS FAILED AND IS RESTARTING. IEF196I 1 //IEESYSAS JOB MSGLEVEL=1 2 //CICSVR EXEC IEESYSAS, PROG=DWW1SJST IEF196I IEF196I STMT NO. MESSAGE IEF196I 2 IEFCØØ1I PROCEDURE IEESYSAS WAS EXPANDED USING SYSTEM IEF196I LIBRARY SYS1.PROCLIB IEF196I 3 XXIEESYSAS PROC PROG=IEFBR14 IEF196I 00050000 4 XXIEFPROC EXEC PGM=&PROG IEF196I IEF196I 00100000 XX* THE IEESYSAS PROCEDURE IS SPECIFIED IN THE IEF196I IEF196I 00150000 IEF196I XX* PARAMETER LIST TO IEEMB881 BY MVS COMPONENTIEF1961 XX* STARTING FULL FUNCTION SYSTEM ADD ESS SPACES. IEF196I IEF196I 00250000 IEF196I IEFC653I SUBSTITUTION JCL - PGM=DWW1SJST IEF4Ø3I IEESYSAS - STARTED - TIME=21.18.32 - ASID=Ø2FC - CMØ1 IEF196I IEF237I 3DØ2 ALLOCATED TO DWWDMSG IEF196I IEF237I 3C19 ALLOCATED TO DWWDUMP IEF196I IGD1Ø3I SMS ALLOCATED TO DDNAME DWWCON1 IEF196I IGD1Ø3I SMS ALLOCATED TO DDNAME DWWCON2 IEF196I IGD1Ø3I SMS ALLOCATED TO DDNAME DWWCON3 IEF196I IEF237I 3D1A ALLOCATED TO DWWMSG DWW2Ø4I CICSVR DATA SET NAMING CONVENTION SET DURING 521

CICSVR ADDRESS SPACE INITIALIZATION ON SYSTEM: CMØ1 CURRENT VALUE: DWW DWWØ14I CICSVR SERVER ADDRESS SPACE IS NOW ACTIVE.

It is possible to review the status of the CICSVR address space by issuing:

D SMS,CICSVR,ALL

An example of the output from the D SMS,CICSVR,ALL command is shown below:

D SMS,CICSVR, ALL

IEE932I 185 DWWØ2ØI DISPLAY SMS,CICSVR DISPLAY SMS,CICSVR - SERVER STATUS SYSNAME: CMØ1 AVAILABLE ASID: Ø234 STEP: CICSVR_Init_Complete SYSNAME: CMTS AVAILABLE ASID: Ø1FA STEP: CICSVR_Init_Complete DISPLAY SMS,CICSVR - JOB STATUS NUMBER OF JOBS USING BATCH LOGGING: SYSNAME: CMØ1 15 SYSNAME: CMTS 2 DATA SET NAMING CONVENTION IN USE: SYSNAME: CMØ1 DWW SYSNAME: CMTS DWWT

BATCH LOGGING

The CICSVR address space collects information about candidate VSAM datasets and stores it in the RCDS. The information consists of:

• Open and close activity.

Each time the batch logging-enabled VSAM dataset is opened for update, CICSVR creates a new RCDS record containing dataset name, timestamp, and log stream ID information.

• Back-up activity.

The process of recording is different, depending on the back-up utility that you use. If you use DFSMSdss, by using the CICSVRBACKUP option, DFSMS writes the

information to the RCDS directly. If you are using DFSMShsm, CICSVR can get the information from the DFSMShsm back-up control dataset.

• Relative log stream IDs.

OS/390 or z/OS log streams log the after-images of the VSAM dataset updates. CICSVR is able to forward recover the after-images of these VSAM datasets to a point in time. Forward recovery steps would be to:

- restore the VSAM datasets from a back-up.
- apply the after-images from the logs to recover VSAM datasets.

The main benefits from batch logging are as follows. By forward recovering corrupt VSAM datasets you can prevent data loss. You can save time by not having to re-run batch jobs or whole schedules. The system is relatively easy to use. The ISPF dialogs guide you through the creation of recovery jobs. The system is reasonably flexible and the MVS log stream copy and RCDS export/import utilities make it feasible for disaster recovery. By maintaining three copies of the RCDS, as long as guidelines are adhered too, it is reasonably failsafe.

For batch logging to work, the following criteria must be met:

- CICSVR address space must be active.
- All VSAM datasets have to be SMS managed.
- All VSAM datasets must be defined with FRLOG(REDO) and LOGSTREAMID specified.
- All VSAM datasets must not be accessed in RLS mode or in use by CICS.

You can either define a new VSAM dataset or alter an existing SMS-managed VSAM dataset for batch logging. An example JCL stream for defining a VSAM dataset for batch logging is shown below:

```
//JXB7884D JOB
//STEP1 EXEC
//SYSPRINT DD
DD
                      (7884), CLASS=A, MSGCLASS=H, MSGLEVEL=(1,1)
                       PGM=IDCAMS
                       SYSOUT=*
                       *
        DEF CL(NAME(JXB7884.CICSVR.EXAMPLE) -
               UNIQUE
                CYL(2 1)
                RECSZ(8Ø 8Ø)
                STORCLAS(CICVRSMS)
                FRLOG(REDO)
                LOGSTREAMID(BATCH.LOGGING)
                KEYS (15 Ø)
                SH (3 3)
                VOL(CICAP1)
                FSPC(50 50))
             DATA(NAME(JXB7884.CICSVR.EXAMPLE.D))
             INDEX(NAME(JXB7884.CICSVR.EXAMPLE.I))
```

//

It is important to pay attention to the following parameters:

- STORCLAS(CICVRSMS) for a dataset to be SMSmanaged, it has to be assigned a storage class.
- FRLOG(REDO) specifies that CICSVR VSAM batch logging will be performed for the VSAM dataset. CICSVR will write forward recovery log records to the log specified by the LOGSTREAMID parameter. If you specify FRLOG(REDO), you must also specify the LOGSTREAMID parameter unless it is already defined.
- LOGSTREAMID(BATCH.LOGGING) specifies the MVS log stream name to be used for writing the forward recovery log records.

Obviously, one of the key factors in recovery is to have a good back-up of the dataset to revert to and then to forward recover. You can use any storage management product to back up your VSAM datasets; it does not have to be an IBM product. However, if you do use IBM products, they integrate in a better manner with the CICSVR product. CICSVR provides complete dataset forward recovery automation if you use DFSMShsm or DFSMSdss utilities.

Some other factors to consider if using the DFSMS products are discussed below.

DFSMShsm

The CICSVR address space is not required in order to use DFSMShsm. If the CICSVR server address space is active, all userids running CICSVR must be authorized to issue DFSMShsm authorized commands, as explained earlier. CICSVR gets the back-up information directly from the DFSMShsm back-up control dataset. You can make a DFSMShsm back-up using automatic back-up:

- Via a management class under SMS control.
- By issuing an HBACKDS command from a TSO online session.
- By performing what is termed an 'in-line back-up', using the ARCINBAK batch program.

If your VSAM datasets are backed up using DFSMShsm, the entire VSAM sphere, the base cluster, AIX, and paths will be backed up. Therefore, during the restore process the entire sphere will be restored. The restored VSAM dataset will then be forward recovered, and the AIX will be rebuilt if the AIX is part of the upgrade set.

DFSMSdss

When you use DFSMSdss COPY or DUMP to back up your VSAM datasets, the CICSVR address space will receive a notification every time a back-up is made, by specifying the special keyword CICSVRBACKUP as part of the back-up job. CICSVR stores the DFSMSdss back-up information in its RCDS. In contrast to DFSMShsm, the CICSVR address space must be active for the dump to work correctly. Example JCL to back up a dataset using the CICSVRBACKUP parameter is shown below:

```
//JXB7884D JOB (7884),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COPY DATASET( -
INCLUDE(JXB7884.CICSVR.TESTDUMP)) -
```

```
RENAMEU(** CICSVR.**)
OUTDYNAM((CICDØ1),(CICDØ2))
CICSVRBACKUP
/*
```

By specifying CICSVRBACKUP and RENAMEU(**,CICSVR.**), CICSVR provides DFSMSdss with a new name for each copy using the naming convention prefix.DSOUTPUT.D yyyydd.T hhmmsss, where:

- *prefix* is the value of the CICSVR_DSNAME_PREFIX parameter defined in the IGDSMS*xx* PARMLIB member.
- *Yyyyddd* represents the year and day in Julian date format.
- *Hhmmsss* represents the hour, minute, and second that the VSAM datasets were copied into new VSAM datasets.

An important point to note when using DSS is that the SPHERE parameter is mutually exclusive with the CICSVRBACKUP keyword, therefore only the base VSAM cluster will be backed up. So, during a restore operation of a VSAM dataset from its DFSMSdss back-up, only the base cluster will be restored. The user is responsible for manually recreating and rebuilding any AIXs for the VSAM dataset. Also note that if a new name is specified to restore the VSAM dataset the user again is responsible for recreating and rebuilding the AIXs.

You can use the ISPF dialog to check the information about the back-ups of your VSAM datasets. On the CICSVR main menu panel select Option 1 and enter the search criteria. You will see the list of VSAM datasets that meet the selected criteria.

CONCLUSION

CICSVR can be a useful way of recovering data and can also enhance disaster recovery. There are a number of set-up steps that must be performed, and, as with anything, a degree of manual intervention is still required. A number of key APARs should be reviewed before embarking on a CICSVR install. These are:

- II13131
- OW53432
- OW53881.

There are also a number of dependency APARs that should be reviewed on the IBM Web sites and several problem APARs that are worth reading.

John Bradley Systems Programmer Meerkat Computer Services (UK)

© Xephon 2005

CICS Transaction Gateway – strategic JCA connector with WebSphere Application Server

Many shops today are running their core applications under CICS, and use CICS Transaction Gateway (CTG) as a connector with WebSphere Application Server. CTG is the preferred J2EE connector for CICS TS, and, in conjunction with IBM WebSphere Application Server, it provides a high performing, secure, scalable, and tightly-integrated access method in CICS. CTG has proven high performance and scalability with minimum overhead, and usually does not require any changes to existing CICS applications. This article will give you both an insight into CTG V5.1 and V6.0, focusing on WebSphere on z/OS connectivity to CICS, and an overview of CTG V6 enhancements, planning, and migration information.

CTG consists of a set of client and server software components that provide connectivity into CICS to invoke existing legacy CICS applications. It allows multiple users of Web browsers to access new and existing mission-critical CICS applications on multiple CICS servers. CTG enables tightly-coupled technologies like Java Connector Architecture (JCA) and loosely-coupled Web services to co-exist and exploit the agility of an on-demand operating environment. IBM positions CTG as a strategic JCA connector with WebSphere Application Server and continues to enhance its functionality, improve its performance, and provide support on additional platforms.

The latest CTG, Version 6.0, was announced at the same time as CICS TS V3.1 and went GA on 14 January 2005. CTG V6 supports JCA at the 1.5 specification level and comes with ECI and EPI adapters, enabling both COMMAREA and 3270datastream-based CICS applications to interoperate effectively with WebSphere Application Server V6. You can take advantage of CTG 6.0 if you are running Java SDK 1.4.2 and CICS TS 1.3 or higher.

CTG V5.1 and V6 support the following platforms: z/OS, Linux on zSeries, AIX, HP-UX, Sun Solaris, Microsoft Windows, and Linux on Intel. Connectivity is provided on these platforms from all supported WebSphere Application Server environments to all supported CICS servers. If CTG is deployed on z/OS, then the only supported CICS server is CICS Transaction Server on z/OS.

MAJOR COMPONENTS OF THE CTG

The components of CTG are shown in Figure 1. They consist of:

- CTG classes Java classes used by the application to invoke services in an attached CICS server if the application uses the CTG interfaces directly. The CTG classes implement different interfaces – the External Call Interface (ECI), External Presentation Interface (EPI), and External Security Interface (ESI).
- ECI and EPI resource adapters CICS JCA resource adapters.

 Gateway daemon – a long-running process that functions as a server to network-attached Java client applications by listening on a specified TCP/IP port.

For inbound requests, the Gateway daemon supports a variety of TCP/IP-based network protocols, including Secure Sockets Layer (SSL) and HTTP. For outbound requests to the CICS server, the Gateway daemon uses the Java Native Interface (JNI) to invoke the Client daemon, which, in turn, uses one of the transport drivers to forward the request to CICS. It should be noted that a Gateway daemon is not required when the invoking Java application executes on the same machine as the one on which the CTG is installed. For example, when running WebSphere Application server for z/OS, the CTG local protocol can be used, directly invoking the underlying transport mechanism (using the JNI) and bypassing the Gateway daemon.

 Client daemon – used with CTG on all distributed platforms. It provides the CICS connectivity as the CICS Universal Client. On z/OS, the External CICS Interface (EXCI) is used in place of the Client daemon.



• Transport drivers – these are transport mechanisms used to connect the Client daemon to the CICS server.

A variety of network protocols are supported for connections as follows:

- J2EE component to CICS Transaction Gateway: TCP/ IP, SSL, or local bindings
- CICS Transaction Gateway to CICS: SNA, TCP62, TCP/IP, or EXCI.

JCA AND CICS ECI RESOURCE ADAPTER

J2EE Connector Architecture (JCA) is part of the J2EE standard and specifies the system contracts to be implemented by a resource adapter. These system contracts define the qualities of service that a resource adapter provides for transaction management, connection management, and security.

Utilizing the JCA simplifies application development by providing a familiar standard interface that programmatically manages transactions, connections, and security. This allows for faster development of better applications.

The CICS ECI resource adapter implements the Local Transaction interface and has limited support for global transactions. It can participate in transactions that are local to the resource manager, but not in two-phase commit transactions. However, when running within WebSphere Application Server for z/OS, the CICS ECI resource adapter supports global transactions, if a local Gateway is used. The benefit of using CICS ECI resource adapter with WebSphere for z/OS is that CTG can run within the WebSphere Application Server address space and use internal functions of the MVS Resource Recovery Services (RRS). This provides RRS global transaction support, and allows the CICS ECI resource adapter to participate in a global transaction in WebSphere Application Server for z/OS with any number of other twophase capable resource managers. Figure 2 shows Global transaction coordination under WebSphere on z/OS.

If a remote Gateway is used on WebSphere Application Server for z/OS, then the ECI Connection becomes only onephase commit capable. In this circumstance, WebSphere Application Server for z/OS permits the usage of a single onephase commit capable resource with any RRS-capable resources in the same transaction.

LAST PARTICIPANT SUPPORT

Last participant support is an extension to the transaction service to allow a single one-phase resource to participate in a two-phase transaction with one or more two-phase resources.

Last participant support allows the use of a single one-phase commit resource in a global transaction along with any number of two-phase commit resources. At transaction commit, the two-phase commit resources will first be prepared. If this is successful, the one-phase commit resource will be called to commit; this is followed by the call to commit for two-phase commit resources.



Within the J2EE transactional environment, the last participant support function of WebSphere Application Server Enterprise extends the global transaction model to allow a single onephase commit resource to participate in a global transaction with any number of two-phase commit capable resources. At transaction commit, the application server first prepares the two-phase commit resource managers and, if this is successful, the one-phase commit-resource is then called to commit. The two-phase commit resources are then committed or rolled back depending on the response of the one-phase commit resource, effectively delegating transaction coordination to the one-phase commit resource.

You need to be aware that with a one-phase commit resource, there is no recovery from a communication failure and roll back may not occur if a communication failure occurs during a commit. Your application has to be configured to accept the additional risk of such an outcome.

Outside of WebSphere Application Server for z/OS, the CICS ECI resource adapter can be used only as a local transactioncapable resource adapter, and so can be used with the last participant support function of WebSphere Application Server. With the use of last participant support, the J2EE component does not need to be concerned with the order in which updates are made within the transaction because the commit process is handled entirely by the application server container.

JCA AND CTG V6

CTG V6 ECI and EPI resource adapters support only JCA 1.5, at J2EE V1.4 spec level, and only WebSphere Application Server V6.

CTG JCA 1.0 resource adapters can be used in remote mode from a J2EE 1.3 application server (such as WebSphere Application Server V5.1) to connect to a remote CTG V6 Gateway daemon. To do this you need to use a version of the CTG resource adapter that is supported by the version of the J2EE application server. However, using the JCA 1.0 resource adapters in a J2EE 1.3 application server, with a local CICS Transaction Gateway V6 on the same machine, is not supported.

It is important to point out that you cannot use CTG V6 in local mode with WebSphere V5.1 on z/OS using the JCA connector. This is because the connector supplied with CTG V6 is a JCA 1.5 connector, which is not supported in WebSphere V5.1. It is also not possible to use the 1.0 connector, supplied with CTG V5.1 in local mode, with CTG V6. In order to use WebSphere 5.1 with CTG V6, you would have to use the JCA 1.0 connector supplied with CTG V5.1, connecting via remote mode (ie over TCP/IP or SSL) to a CTG V6 Gateway daemon.

CTG V6 has performance enhancements as a result of using JCA 1.5 architectural connection optimizations. The following enhancements are implemented:

- Lazy Transaction enlistment:
 - optimizes transaction enlistment calls so that RA participates in a transaction only if it is actually invoked.
 - provides potentially better performance if JCA applications are used in an enterprise application.
 - is not applicable to WAS z/OS local mode (which uses RRS).
- Lazy Connection association:
 - provides improved connection pooling in WebSphere.
 - allows the get-use-cache model to be used efficiently with WebSphere Connection Pool manager.

In WebSphere Application Server Version 5, connection manager has implemented smart handle support. This technology enables the allocation of a connection handle to an application while the managed connection associated with that connection handle is used by others. This concept is part of the JCA 1.5 specification. Smart handle support introduces the use of a method on the ConnectionManager object – the LazyAssociatableConnectionManager() method.

Transaction management under J2EE server

The JCA offers the distinct advantage that if the application is deployed into the managed environment of WebSphere Application Server, the application will benefit from the connection pooling, transaction, and security management provided by the application server. J2EE server components structure is shown in Figure 3.

Transaction manager defines the scope of transactional integration between the J2EE application server and an EIS that supports transactional access. With global transaction support, J2EE application server can use a transaction manager to manage transactions across multiple resource managers, like CICS and DB2. Global transactions can also be created within the Web container by using the javax.transaction.UserTransaction interface to begin and end a transaction. If the ECI resource adapter is being used in



local mode on WebSphere Application Server for z/OS, the CICS Transaction Gateway RRS support does allow an ECI request to participate in a bean-managed global transaction.

Transaction manager also supports the local transaction interface, which refers to transactions that are managed internally by a resource manager without the involvement of an external transaction manager. Within WebSphere Application Server these transactions are known as resource manager local transactions.

Local transactions can be controlled by invoking the method getLocalTransaction() on the ECI resource adapter connection object; this provides a programmatic transactional context specific to the instance of the JCA connection factory. An example of using a local transaction to run as one unit-of-work within CICS is shown below:

```
Context ic = new InitialContext();
cxnf = (ConnectionFactory) ic.lookup("java:comp/env/eis/ECICICS1");
Connection cxn = cxnf.getConnection();
Interaction ixn = cxn.createInteraction();
ECIInteractionSpec ixnSpec = new
    ECIInteractionSpec(SYNC_SEND_RECEIVE, "myCICSPGM");
JavaStringRecord jsr = new JavaStringRecord();
LocalTransaction tran = cxn.getLocalTransaction();
tran.begin();
jsr.setText("TEXT1");
1 ixn.execute(ixnSpec, jsr, jsr);
jsr.setText("TEXT2");
2 ixn.execute(ixnSpec, jsr, jsr);
tran.commit();
ixn.close();
cxn.close();
```

EXCI PIPE USAGE

Within WebSphere Application Server on z/OS the CTG uses the facilities of the external CICS interface (EXCI) to flow ECI requests into CICS. Based on the maximum number of EXCI pipes allowed, CICS limits each client address space to a maximum number of simultaneous sessions attached to CICS regions.

Previous releases of CTG had a 100 pipes limit, which can now be increased, if you need to do so. CICS Transaction Server for z/OS Version 2 will provide an APAR (PQ92943) to increase the maximum number of pipes from 100 to 250. PQ92943/UQ95449 adds support for a new CICS subsystem initialization parameter, LOGONLIM. This parameter specifies the maximum number of pipes that can be allocated in an EXCI address space. The default limit is 100 pipes. A maximum of 250 pipes may be allocated in an EXCI address space.

With CTG V6 there is a new environment variable, CTG_PIPE_REUSE=ONE, that allows one pipe per thread. This allows existing allocated pipes to de-allocate if the thread needs to allocate to a new applid. This creates predictable usage, which ensures that the maximum number of pipes is equal to the number of threads.

Please be aware that when the maximum limit of EXCI pipes has been reached, the next EXCI Allocate_Pipe call, made from a particular address space, fails with a SYSTEM_ERROR response code and a reason code 608, and the ECI application receives a return code -9 (ECI_ERR_SYSTEM_ERROR).

JCA connection pool – JVM threads allocation

The JCA connection pool managed by WebSphere Application Server is a set of local connection objects that map to the CTG JavaGateway connections. These connection objects do not map onto the EXCI pipes allocated by the CTG. EXCI pipes are allocated directly by the JVM threads within the J2EE servant region. The maximum number of JVM threads within a J2EE servant region is defined in the server workload profile and has an upper maximum of either 40 or 96, depending on the setting of the application server workload profile. For reasons of performance, once a call has been made to a CICS region from a given thread, the CTG will keep the pipe allocated to that particular CICS region for the lifetime of the thread, or until the CICS region terminates or closes IRC.

An EXCI pipe shortage situation may occur if multiple JCA connection factories are used for naming different CICS regions. This has the potential to cause pipes to become allocated to multiple CICS regions from a single thread. If you are using this type of configuration, you should configure the J2EE servant region to have fewer threads, and use multiple servant regions to obtain the desired throughput.

WEBSPHERE APPLICATION SERVER THREAD USAGE

You can configure the number of threads in a J2EE servant region to be used by WebSphere Application Server. There is a trade-off between a thread bottleneck and over-commitment of resources if too many threads are used. The recommendation is to have a low number of threads to get the best throughput when the calls to the CICS server execute quickly. Better throughput can be achieved with higher numbers of threads when using a multi-CPU architecture. In WebSphere Application Server for z/OS V5, the number of threads in a J2EE servant region is controlled by the server workload profile.

Here is the general recommendation for setting the thread values:

- IOBOUND three times the number of CPUs (max of 3x32=96).
- CPUBOUND the number of CPUs online (max 24).
- ISOLATE 1 thread.
- LONGWAIT 40 threads.

WEBSPHERE APPLICATION SERVER AND THE CTG ON Z/OS

The most common z/OS configuration makes use of a local CTG under WebSphere and CICS ECI resource adapter. With this configuration you can make use of a direct cross-memory EXCI connection between the application server and CICS. Figure 4 shows the application deployed to WebSphere Application Server using a local CTG.



REMOTE CONNECTIVITY

CTG V5.1 and V6 also support remote connections for this topology, which allows a connection from the application server to a z/OS Gateway daemon. With this support, it is possible to use CTG for z/OS on the other LPAR and establish a remote connection from WebSphere Application Server into the CTG using the EXCI interface. If your shop is considering this configuration, it should be noted that the best performance can be achieved using a local connection between the application server and the CICS region. Another consideration for using remote versus local connection is the two-phase commit capability provided by Resource Recovery Services (RRS). Only local topology provides thread identity support and two-phase commit transactions.

The unique thread identity support in WebSphere Application Server for z/OS allows the application server to automatically pass the user ID of the thread to CICS when using the ECI resource adapter. This allows the automatic propagation of the authenticated caller's user ID from WebSphere Application Server to CICS.

WEBSPHERE APPLICATION SERVER AND THE CTG ON A DISTRIBUTED PLATFORM

WebSphere Application Server can also be deployed on one



of the distributed platforms. It is possible to access CICS through a Gateway daemon running on z/OS – as shown in Figure 5.

SOFTWARE REQUIREMENTS

The CICS Transaction Gateway is supported by the following

CICS servers:

- CICS Transaction Server for z/OS V1.3:
 - apply z/OS APAR PQ38644, for EXCI, if you are using the CICS Transaction Gateway for z/OS.
- CICS Transaction Server for z/OS V2.2:
 - if connecting over TCP/IP, you are recommended to apply APAR PQ75803 and PQ82124.
 - apply APAR PQ92943 for improved EXCI scalability.
- CICS Transaction Server for z/OS V2.3.
 - If connecting over TCP/IP, you are recommended to apply:
 - APAR PQ81772/ UQ83755 DFHSM0002 code('0316'). When a PING request from an ECI over a TCP/IP client results in a PURGE of the mirror task, CICS rejects every ping_abend request, except for the first conversation on a connection. A fix was written for the CTG to put the correct value in the ping_abend sequence number field.
 - APAR PQ82124 PING_REQUEST (PING_ABEND flow) from a CTG client using ECI over TCP/IP may result in a reply of CONV_PING_REPLY_NOT_KNOWN from CICS if the mirror task is defined as SPURGE(No). DFHIEIE has been altered to ignore IECCBs with the ieccb_abend flag on when searching for an IECCB to match the flow from the client.
- Apply APAR PQ92943 for improved EXCI scalability.

ENHANCEMENTS IN CTG V6

The new version of CTG has an impressive number of enhancements, as follows:

- Improvements in performance, resulting in reduced CPU overhead. For example, the improved performance of data processing on z/OS provides significant CPU usage savings of up to 40% when null padded COMMAREAs are used, saving up to 220KB of instructions. Also, you can see a CPU reduction of up to 15% with improved run-time trace performance of the CTG.
- Support for zAAP off-load in z/OS V1.6, with potential for off-loading up to 50% of CTG instructions to zAAP CP. This allows for a reduction in the CPU cost of Java processing, without affecting the throughput. For more details on ZAAP performance, please refer to the white paper z/OS Performance: Capacity Planning Considerations for zAAP Processors.
- Improved garbage collection and more optimized JNI code for CTG running on z/OS.
- Improved handling of error messages the output of all run-time messages can now be directed to JES. This is a great improvement in system management on z/OS. In the past, all messages were written to the HFS log and crucial error messages could go undetected. This enhancement gives you an opportunity to use your current automation tool to alert you should problems occur.
- Improved systems management by providing an extension of the CICSPlex System Manager Web User Interface.
- Integrated administration interfaces by providing the ability to manage CTG from SDSF (on z/OS only). CICS Transaction Gateway for z/OS now provides installation via SMP/E, vastly simplifying the task of installing, upgrading, and applying maintenance. You can use SDSF to set the trace level for the Gateway daemon as follows:
 - set the trace file size limit
 - set the amount of data from the hex dump to trace
 - set the offset within a hex dump

- do a full hex dump
- turn JNI trace on or off
- query trace settings.
- For example, to enable JNI trace, enter the following command:

/F JOB_NAME, APPL=TRACE, JNILEVEL=1

To provide more control over tracing, the com.ibm.connector2.cics.tracelevel system property is available.

This allows you to override the deployed trace level for the resource adapters without having to redeploy or deploy another CICS resource adapter. These JVM system properties can be passed to the JVM upon start-up. The option com.ibm.connector2.cics.tracelevel is equivalent to the managed environment property 'tracelevel', which is set as a custom property on the connection factory.

- Improved workload throughput.
- Improved data integrity a normal shutdown mechanism ensures the integrity of all outstanding units of work.
- Improvements in availability and scalability, by providing a new option to limit the number of EXCI pipes to one per thread. This results in more accurate predictions of the number of pipes required.
- Enhanced support for the configuration of multiple Gateway daemons.
- Better security by providing a number of SSL supportrelated enhancements:
 - SSL private keys and certificates can be stored in a RACF database.
 - improved support for hardware cryptography cards when using SSL – new external configuration option

for SSL encryption, allowing the SSL cipher suite to be specified.

Also it should be noted that CICS TS 3.1 includes the CICS Transaction Gateway Version 3.1, which enables users to access business-critical applications running on CICS servers from a Web browser or network computer.

The following performance enhancements are applicable to CTG V6 on Windows, Unix, and Linux:

- A simplified installation process.
- Better system administration capabilities.
- Improved run-time tracing enhancements with the Client daemon.
- Around 15% CPU usage saving.
- On Unix and Linux platforms, you can run the CTG as a background process when the system starts.
- A new normal shutdown mechanism ensures the integrity of all outstanding units of work.
- Enhanced support for the configuration of multiple Gateway daemons, provides improved system administration capabilities.
- A new external configuration option for SSL encryption allows the SSL cipher suite to be specified.
- Redesigned and searchable Eclipse-based information centre provides a vastly improved user interface for online documentation.

Editor's note: this article will be concluded in the next issue.

Elena NanosIBM Certified Solution Expert in CICS Web EnablementZurich NA (USA)© Xephon 2005

3270 transactions via CICS Web Interface (CWI)

Editor's note: this interesting article is taken from a PowerPoint presentation.

First the bad news:

• For users with a high transaction volume, 3270 transactions via the CICS Web Interface (CWI) is probably less than ideal because virtual function keys must be used.

However, possibly, real function keys will be able to be used soon.

• Not all BMS functionalities are supported (see the chapter 'Limitations of Web 3270 Support' in the *CICS Internet Guide*).

On the plus side:

- The CWI adds no (extra) costs.
- It is easy to handle.
- It provides a direct way into CICS via the browser.

Furthermore, a 'face-lift' for the older 3270 applications is possible with CWI.

Let's see what access via the CWI to the CTS V2 (demonstrated by a simple application) looks like. Simply call up the transaction via the browser (provided that everything is ready and the user is logged on):

- Log on with your user ID and password, using basic security see Figure 1.
- After the log on, the start-up output is as shown in Figure 2.
- Output from the called transaction looks like it is in a pure 3270-transaction – there is no HTML running under 3270 emulation – see Figure 3.





• It should be noted that some IBM transactions can use the CWI by default (in the example in Figure 4, the CEMT I TAS via the browser).

To summarize, the application remains the same, but:

- It is now called directly via the browser and without emulation.
- So-called HTML templates have been created out of the BMS definitions.

Without using the HTML templates created out of the BMS definitions, drag will not work.

//MYUSRIDX JOB ØØ2665, 'REIS CLAUS', NOTIFY=MYUSRID, //*-----//* JOB SUBMITTED FROM MYUSRID.SJCL.LIB(DFHMAPT) //* DOC: GENERATE HTML-TEMPLATES //*-----CLASS=T,USER=MYUSRID,MSGCLASS=X,REGION=4M.RESTART=* 11 //*-----//DFHMAPT PROC INDEX='CICSTS22.NLV', FOR SDFHMAC MAPLIB='CICS.SYST.PPLOAD', TARGET FOR MAP DSCTLIB='MYUSRID.MAIN.SOURCE', TARGET FOR DSECT TEMPLIB='CICS.SYST.DFHHTML', TARGET FOR TEMPLATES 11 11 11 11 MAPNAME=CMCSGM, NAME OF MAPSET - REQUIRED 11 A=, A=A FOR ALIGNED MAP A=, RMODE=ANY, ASMBLR=ASMA9Ø, REG=2Ø48K, OUTC=*, WORK=SYSDA 11 24/ANY RMODE=ANY, ASMBLR=ASMA9Ø, 11 ASSEMBLER PROGRAM NAME 11 REGION FOR ASSEMBLY PRINT SYSOUT CLASS WORK FILE UNIT 11 11 WORK=SYSDA //COPY EXEC PGM=IEBGENER //SYSPRINT DD SYSOUT=&OUTC //SYSUT2 DD DSN=&&TEMPM,UNIT=&WORK,DISP=(,PASS), // DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=4ØØ), // SPACE=(4ØØ,(5Ø,5Ø)) //SYSIN DD DUMMY //*SYSUT1 DD * NEEDED FOR THE MAP SOURCE //ASMMAP EXEC PGM=&ASMBLR,REGION=®, // PARM='SYSPARM(&A.MAP),DECK,NOOBJECT' //SYSPRINT DD SYSOUT=&OUTC //SYSLIB DD DSN=&INDEX..SDFHMAC,DISP=SHR 11 DD DSN=SYS1.MACLIB,DISP=SHR //SYSUT1 DD UNIT=&WORK,SPACE=(CYL,(5,5)) //SYSUT2 DD UNIT=&WORK,SPACE=(CYL,(5,5)) //SYSUT3 DD UNIT=&WORK,SPACE=(CYL,(5,5))

```
//SYSPUNCH DD DSN=&&MAP,DISP=(,PASS),UNIT=&WORK,
              DCB=(RECFM=FB,LRECL=8Ø,BLKSIZE=4ØØ),
11
11
              SPACE=(400,(50,50))
//SYSIN
           DD DSN=&&TEMPM.DISP=(OLD.PASS)
//LINKMAP EXEC PGM=IEWL,PARM='LIST,LET,XREF,RMODE(&RMODE)'
//SYSPRINT DD SYSOUT=&OUTC
//SYSLMOD DD DSN=&MAPLIB(&MAPNAME),DISP=SHR
           DD UNIT=&WORK, SPACE=(1024, (20, 20))
//SYSUT1
//SYSLIN DD DSN=&&MAP,DISP=(OLD,DELETE)
//ASMDSECT EXEC PGM=&ASMBLR,REGION=&REG,
// PARM='SYSPARM(&A.DSECT),DECK,NOOBJECT'
//SYSPRINT DD SYSOUT=&OUTC
           DD DSN=&INDEX..SDFHMAC,DISP=SHR
//SYSLIB
11
           DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1
           DD UNIT=&WORK, SPACE=(CYL, (5,5))
//SYSUT2
           DD UNIT=&WORK, SPACE=(CYL, (5,5))
//SYSUT3
           DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSPUNCH DD DSN=&DSCTLIB(&MAPNAME),DISP=OLD
          DD DSN=&&TEMPM,DISP=(OLD,PASS)
//SYSIN
//ASMTEMPL EXEC PGM=&ASMBLR,REGION=&REG,
    PARM='SYSPARM(TEMPLATE,CSGM),DECK,NOOBJECT'
//
//SYSPRINT DD SYSOUT=&OUTC
          DD DSN=&INDEX..SDFHMAC,DISP=SHR
//SYSLIB
           DD DSN=SYS1.MACLIB,DISP=SHR
11
//SYSUT1 DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT2 DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSUT3 DD UNIT=&WORK,SPACE=(CYL,(5,5))
//SYSPUNCH DD UNIT=&WORK,SPACE=(CYL,(5,5)),DISP=(,PASS)
           DD DSN=&&TEMPM, DISP=(OLD, DELETE)
//SYSIN
//UPDTEMPL EXEC PGM=IEBUPDTE, REGION=&REG, PARM=NEW
//SYSPRINT DD SYSOUT=&OUTC
           DD DSN=*.ASMTEMPL.SYSPUNCH,DISP=(OLD,DELETE)
//SYSIN
//SYSUT2
           DD DSN=&TEMPLIB, DISP=SHR
//DFHMAPT PEND
//CREAHTML EXEC DFHMAPT
//COPY.SYSUT1 DD * INSERT BMS-DECK
         MACRO
         CSGM
         DFHMDX MAPSET=*,MAP=*,
                                                                        *
                                                                       *
               TITLE='327Ø TRANSACTION (CSGM) IM INTRA- BZW. INTERNET
               BMS MAP (CMCSGM) CONVERTED TO HTML C.REIS ISB-DB/DC',
                                                                        *
                                                                        *
               BGCOLOR=BLUE,
                                                                        *
               TEXT=BLACK,
               RESET=N0
         MEND
         TITLE 'CMCSGM
                         - NLV GOODMORNING
CMCSGM
                                                            MESSAGE'
         DFHMSD TYPE=&SYSPARM, MODE=OUT, LANG=ASM,
                                                                        *
CMCSGM
               STORAGE=AUTO, TIOAPFX=YES
         SPACE 3
```

```
MCSGM
        DFHMDI SIZE=(24,8Ø),LINE=1,COLUMN=1,TIOAPFX=YES,
              HILIGHT=OFF,COLOR=DEFAULT,
              MAPATTS=(COLOR, HILIGHT, OUTLINE)
DFHMDF POS=(1,1),LENGTH=4,INITIAL='ZSSN',ATTRB=(FSET,ASKIP)
        DFHMDF POS=(1,7), LENGTH=1
        DFHMDF POS=(1,24),LENGTH=17,INITIAL='CICS/ESA Version'
Z1VERS
        DFHMDF POS=(1,42), LENGTH=3
        DFHMDF POS=(1,57), LENGTH=8
Z1DATE
Z1TIME
        DFHMDF POS=(1,66), LENGTH=5
        DFHMDF POS=(1,72), LENGTH=4
Z1TERM
        DFHMDF POS=(1,77), LENGTH=3, INITIAL='SYS'
Z2LEFT
        DFHMDF POS=(Ø2,Ø1),LENGTH=2Ø,ATTRB=(ASKIP),COLOR=YELLOW
Z2RIGHT DFHMDF POS=(Ø2,57), LENGTH=23, ATTRB=(ASKIP), COLOR=TURQUOISE
Z3LEFT
        DFHMDF POS=(Ø3,01),LENGTH=20,ATTRB=(ASKIP),COLOR=YELLOW
Z3RIGHT DFHMDF POS=(Ø3,57), LENGTH=23, ATTRB=(ASKIP), COLOR=TURQUOISE
Z4LEFT
        DFHMDF POS=(Ø4,01),LENGTH=20,ATTRB=(ASKIP),COLOR=YELLOW
Z4RIGHT
        DFHMDF POS=(Ø4,57), LENGTH=23, ATTRB=(ASKIP), COLOR=TURQUOISE
Z5LEFT
        DFHMDF POS=(Ø5,01),LENGTH=20,ATTRB=(ASKIP),COLOR=YELLOW
        DFHMDF POS=(Ø5,38),LENGTH=4,INITIAL='NLV>',
                                                                    *
              COLOR=BLUE, HILIGHT=REVERSE
        DFHMDF POS=(Ø5,43),LENGTH=1
?
      My Map...
?
      My Map...
         DFHMDF POS=(19,50), LENGTH=10, INITIAL='NNNNNNNN'
        DFHMDF POS=(20,20), LENGTH=40,
              DFHMDF POS=(21,20),LENGTH=40,
              DFHMDF POS=(23,01), LENGTH=79, ATTRB=(BRT, ASKIP),
              COLOR=PINK, HILIGHT=BLINK,
              INITIAL='>> Via applying the data release key you receiv*
              e the ZSS-log on picture <<'
         DFHMDF POS=(24,01),LENGTH=79,INITIAL='You achieve your user-se*
              rvice-station under telephone number 7777'
        DFHMSD TYPE=FINAL
        END
/*
11
```

The header is generated via the macro DFHMDX.

The sky's the limit for creativity here...

CICS JCL with the file for the HTML templates is shown below.

For each MRO group, only one library of this kind exists at any one time.

-----//*----------DD DSN=CICSTS22.SSØ1.DFHLRQ,DISP=SHR //DFHLRQ LOCAL REQUEST //*-----DD DSN=CICS.SYST.DFHHTML,DISP=SHR HTML-TEMPLATES //DFHHTML //*-----//* DATASETS FOR JVM SUPPORT //*-----//DFHCJVM DD DUMMY //DFHJVM DD DSN=CICSTS22.IBM.SDFHENV(DFHJDK13),DISP=SHR

TCPIPSERVICES are shown in Figure 5.

There is one definition per port and CICS address space, including the important basic security ('Bas'), which forces users to log on with a userid and password.

Basic authentication is used to obtain a user ID and password



from the client. If an invalid user ID and password are supplied, the process is repeated until valid information is supplied, or until the end user cancels the connection.

When the end user has been successfully authenticated, the user ID supplied identifies the client.

The resources of the IBM group DFHWEB are shown in Figure 6.

The converter program, DFHWBTTA, translates the HTTP forms data into the correct 3270 format required by the CICS terminal-oriented transaction that is the target of this request and the alias transaction (CWBA). An alias transaction is a CICS-supplied transaction that is started by the Web-attached transaction (CWXN) to process a single request.

Some references:

- CICS Internet Guide (IBM SC34-6007-01).
- Redbook CICS Transaction Server for OS/390: Web Interface and 3270 Bridge (IBM SG24-5243-00).
- Redbook CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge (IBM SG24-5480-00).
- And, of course, IBM on the Internet.

In summary, with some restrictions (because of limited BMS functions and virtual function keys) 3270 transactions via CICS Web Interface is ideal for CICS users, who:

- Want to link directly via a browser into CICS.
- Don't have too big a transaction volume.
- Don't have 3270 emulation available.
- Want to save money on the royalties for emulators.
- Want to continue to use legacy applications.

			DATE	TIME
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
			03.087	15.46.0
		SYSID=ST00	APPLID	-CICST00
	TIME:	10.26.56	DATE:	04.147
SR 7 SBH	8 SFH	9 MSG 10 S	B 11 SF	12 CNCL
	SR 7 SBH	TIME: 3R 7 SBH 8 SFH	SYSID=ST00 TIME: 10.26.56 38 7 SBH 8 SFH 9 MSG 10 S	03.087 03.087 SYSID=ST00 APPLID TIME: 10.26.56 DATE: R 7 SBH 8 SFH 9 MSG 10 SB 11 SF

Figure 6: Resources of the IBM group DFHWEB

• Want to (considering security at least theoretically) directly access a 3270 application in CICS via the Internet.

Claus Reis CICS Systems Programmer Nuernberger Lebensversicherung AG (Germany)

© Xephon 2005

The Update family

In addition to CICS Update, the Xephon family of Update publications now includes AIX Update, DB2 Update, MQ Update, MVS Update, RACF Update, and TCP/SNA Update. Details of all of these can be found on the Xephon Web site at www.xephon.com.

In the March 2004 issue of CICS Update, an article entitled "Build a homegrown adapter solution to enable CICS applications for eBusiness" was published over the name and company affiliation, "Vikas Baruah, Senior Technical Specialist, American Management Systems (USA)." This article was inaccurately attributed to Mr Baruah and American Management Systems (now CGI-AMS). Both the article and the software solution it describes were authored by HostBridge Technology, L.L.C. Further, the article mischaracterized HostBridge's software solution as a simple "homegrown" solution. Contrary to statements in the article, CGI-AMS has neither a copyright nor a patent pending in the software solution described in the article. Instead, HostBridge owns the copyright and is awaiting issuance of the patent in the HostBridge software solution, and owns the copyright in the article itself, which is available on HostBridge's website as "XML-Enabling CICS Applications for eBusiness" (www.hostbridge.com).

* * *

GT Software has announced Release 2.1.2 of Ivory Web Services, which now offers support for channel and container processing provided by the recently announced CICS TS Version 3.1.

Ivory Studio and Ivory Server have been enhanced to provide native support for named containers passed via the IBM channel support. Ivory has extended callable services by allowing the choice of COMMAREA or channel containers as the interface method. Ivory channel and container support will allow clients to build new composite applications that take advantage of the flexibility provided with the enhanced storage management process of CICS TS 3.1. For further information contact: URL: www.gtsoftware.com/products/ ivory.php.

* * *

Accelation Software has announced that its BrightUser SecurityView system now helps security and auditing teams manage mainframe access information by making it easy to identify inappropriate rights, accesses, and authorizations.

BrightUser SecurityView is a desktop-based system that uses a snapshot of the CA-Top Secret information to provide visual and printed information about user status, and about users' access to resources, application programs, files, and processes.

BrightUser SecurityView can be used to find obsolete IDs used by jobs, tasks, and application-specific devices (for example CICS terminals).

For further information contact: URL: www.accelation.com/BUSVpress8.htm.

* * *

IBM has announced the general availability of CICS Transaction Server for z/OS Version 3.1, which uses Web services capabilities to extend CICS applications to SOA environments while easing the development, integration, and administration of those applications through a simplified interface.

For further information contact your local IBM representative:

URL: www-306.ibm.com/software/htp/cics/ library/cicstsforzos31.html.



xephon