



219

CICS

February 2004

In this issue

- 3 Using message DFHLG0760 in problem determination
 - 11 TCP/IP printers under CICS: using a TSO netstat command from a CICS transaction
 - 19 CICS Transaction Gateway V5.0.1 and WebSphere V5.0 – improvements for global transaction processing
 - 33 Newcopy in a CICSplex environment
 - 47 CICS news
-

© Xephon Inc 2004

update

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

Nicole Thomas
E-mail: nicole@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 2004. 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 message DFHLG0760 in problem determination

CICS systems may occasionally experience problems with the growth of the system log stream, DFHLOG. In such cases, message DFHLG0760 is useful in the problem determination process. This article describes how the message has been enhanced to further improve diagnosis of this sort of problem.

BACKGROUND TO CICS SYSTEM LOG MANAGEMENT

The CICS Log Manager component is built on the MVS System Logger subsystem to store and retrieve log data for task and system recovery purposes. This data is held on two separate log streams. The primary log stream is also referred to as journal name DFHLOG, and the secondary log stream as journal name DFHSHUNT.

DFHLOG is used to hold the log data written by tasks executing in the CICS system. These tasks are typically short-lived, meaning they do not persist for long periods of time within the CICS system. In particular, such tasks would not be expected to be present in a CICS system for the period between two successive activity keypoint operations. As such, their log data held on DFHLOG is very dynamic. Once tasks have completed successfully, CICS has no further need to retain the log data pertaining to the changes they made to recoverable resources in the system. This means that the log records written by tasks can be deleted from DFHLOG by CICS once they have completed and left the system. This process is known as log 'tail trimming' and is performed automatically by CICS as part of its housekeeping activities carried out during periodic activity keypoint operations.

Note: some tasks do remain within CICS for longer periods of time, however, and are known as 'long-running' tasks. They may represent low dispatching priority batch-style programs performing work against file or table resources. If the resources that they

modify are defined to be recoverable within the system, CICS will be logging their changes to DFHLOG. Considerable volumes of log data can become associated with a long-running task's Units Of Work (UOW). Amongst other things, a UOW manages the set of atomic changes made to recoverable resources by a program. UOWs are delimited by EXEC CICS SYNCPOINT commands, so, if long-running tasks do not issue periodic syncpoints, they will consist of single UOWs with long chains of log data associated with them.

Unlike DFHLOG, which holds data for tasks that are actively logging within the system, DFHSHUNT is used by CICS to store log data for those tasks that abend during particular stages of syncpoint operations and whose UOWs are shunted by CICS until they can be resolved. It is also used to store the data for those tasks that (for whatever reason) have failed to log in the period between two successive activity keypoints. CICS moves the log records for such inactive tasks from DFHLOG to DFHSHUNT as part of keypoint processing. This means that the space where this data used to reside on DFHLOG can be deleted by CICS later in the activity keypoint. The primary system log stream is therefore prevented from growing excessively large because of the log data for such inactive tasks.

Given the above, it can be seen that DFHSHUNT therefore holds log data for longer periods of time than DFHLOG, and is a far less dynamic repository for log data.

CICS MESSAGES DURING KEYPOINTS

CICS uses activity keypoints at appropriate times for managing the DFHLOG and DFHSHUNT log streams, by issuing IXGDELET calls to the MVS System Logger to delete those portions of the log stream data that are no longer required. It is the normal case for DFHLOG to be trimmed by each activity keypoint. Most tasks logging to it will complete quickly and leave CICS in the time interval between activity keypoints. It is also the normal case for DFHSHUNT not to be trimmed by activity keypoints. DFHSHUNT is the repository for log data written by UOWs that remain in a

CICS system for longer durations, and so the need for CICS to retain their log data will exist for a correspondingly longer period of time.

The following CSMT messages are written by CICS during keypoint operations. Their meanings are as follows:

- Message DFHRM0205 – this means that CICS has performed a successful activity keypoint.
- Message DFHLG0743 – this means that CICS has been able to trim DFHLOG or DFHSHUNT. Note: the appropriate log stream name is given as a text insert within the message.
- Message DFHLG0760 – this means that CICS has not been able to trim DFHLOG during the keypoint operation.

As originally implemented, DFHLG0760 gave the number of keypoints that had elapsed since the last keypoint that was successful in trimming the log stream. Note that message DFHLG0760 is issued only if DFHLOG is unable to be trimmed, not DFHSHUNT. This behaviour is different from that for message DFHLG0743, which will be issued when either the DFHLOG or DFHSHUNT log streams are trimmed by CICS. This is because DFHSHUNT is trimmed far less often than DFHLOG, and a DFHLG0760 message would therefore be seen on nearly every activity keypoint if CICS were to issue one for DFHSHUNT.

Given the above message semantics, a well-behaved typical CICS system would therefore be expected to issue the following diagnostics on each activity keypoint: a DFHRM0205 message, followed by a DFHLG0743 message for DFHLOG. Occasionally, an activity keypoint would also issue a DFHLG0743 message for DFHSHUNT when some long-duration UOW was able to complete and free up space on the secondary CICS system log stream. Every now and then, as long-running tasks were sometimes encountered by CICS during activity keypoint processing, trimming of DFHLOG would not be possible for a short period and so a keypoint would issue a DFHRM0205 message followed by a DFHLG0760 message. This would typically be a random

event, and not repeated for subsequent CICS activity keypoint operations.

Failure to trim DFHLOG means that the volume of log data held there will grow. The log stream definition for the CICS system logs should specify the (default) values of AUTODELETE(NO) and RETPD(0). This means that, unless CICS calls the MVS System Logger to trim the log streams, the data held there will be retained indefinitely. If successive DFHLG0760 messages are seen, this signals a problem to be investigated. It may be that a rogue program has entered a yielding invalid loop within CICS, issuing repetitive calls to update recoverable resources. CICS will not abend such a task with an AICA abend since the loop is yielding control to CICS.

Successive DFHLG0760 messages give the number of keypoints that have elapsed since the last successful trim of DFHLOG; as originally implemented, however, they did not provide any additional transactional information to assist in determining which task was preventing the system log from being trimmed.

DFHLG0760 MESSAGE ENHANCEMENTS

In order to assist with problem determination regarding the cause of successive DFHLG0760 messages (and the associated growth in size of the DFHLOG log data), IBM has enhanced the message to include the transaction identifier and task number of the task that owns the oldest data on the DFHLOG log stream. This information can therefore be used to investigate the particular task that is preventing CICS from trimming DFHLOG and so freeing up space on the log stream. Knowledge of the transaction identifier and its associated task number allows access to the transactional information for the task, as provided by the CEMT INQ TASK command, for example.

The message text and explanation for DFHLG0760 now read as follows:

EXPLANATION: The CICS log manager has been unable to trim the tail of MVS logger log stream 'lsn', during an activity keypoint.

CICS will attempt to trim the system log log streams during activity keypoint processing, to delete redundant data from the oldest end of the log stream (the tail).

Note that this message is only issued if the primary system log stream (DFHLOG) fails to be trimmed during a keypoint operation. It is not issued if a trim does not occur for the secondary system log stream (DFHSHUNT). This is since a trim of the primary system log stream is considerably more likely to occur per keypoint, and hence failures to trim the primary system log stream should be recognized, and investigated if required.

SYSTEM ACTION: *CICS continues processing.*

USER RESPONSE: *This is an informational message, but should be treated as a warning of a potential problem if the number of keypoints that have been unable to trim the primary system log stream continues to increase. Occasional keypoints that fail to trim DFHLOG are not unexpected events. However, if the message is issued by a number of sequential keypoints then this warrants further investigation.*

The keypoint operation was unable to trim the tail of DFHLOG because the oldest log records on the log stream belong to a unit of work that is still required. This may be a validly long-running UOW; alternatively, it may be part of a long-running task executing an application that generates log records but does not issue syncpoint requests regularly enough.

This may be a transient phenomenon because of an atypical long-running UOW. Review the number of keypoints that have been unable to trim the log. Check what message CICS issues for log stream 'lsn' at the next activity keypoint. Message DFHLG0743 indicates that the log stream is now successfully trimmed. Another DFHLG0760 message indicates that a long-running UOW still exists on the system. If the log stream still cannot be trimmed, use the CEMT INQUIRE UOW command to review the oldest UOWs on the system.

The transaction identifier and the task number of the task whose UOW relates to the oldest data on DFHLOG are also provided

in the DFHLG0760 message. The CEMT INQUIRE TASK command may be used to review them.

It is also worthwhile reviewing how often CICS is performing activity keypoints (as defined by the AKPFREQ system definition parameter).

SERVICE INFORMATION

These enhancements to message DFHLG0760 were shipped via APAR PQ74391 (PTF UQ80760) at CICS TS 1.3, and APAR PQ74396 (PTF UQ80644) at CICS TS 2.2. Note: DFHLG0760 message support was originally added to CICS TS 1.3 by APAR PQ34528 (PTF UQ44303); it is present at base code level in CICS TS 2.2.

EXAMPLE

The following example demonstrates the use of DFHLG0760 messages when investigating a problem with DFHLOG failing to be trimmed.

A CICS TS 2.2 system was executing a typical workload of short-lived transactions that performed recoverable work. This workload resulted in CICS logging the recoverable changes to the DFHLOG log stream, and so periodically the CICS system attached a CSKP task to perform an activity keypoint operation (the frequency of activity keypoints is specified by the AKPFREQ SIT parameter). These keypoints were successfully trimming the DFHLOG log stream in the majority of cases. Occasionally, a keypoint would be unable to trim DFHLOG because of a task in the system that happened to persist for the period between two successive activity keypoints; in other words, a long-running task. Such a task would prevent the keypoint operation from trimming DFHLOG if it were actively logging during the interval between keypoint operations.

The majority of activity keypoint operations caused CICS to issue DFHLG0743 messages, indicating that DFHLOG had been successfully trimmed. The occasional presence of a long-

running task led to several isolated keypoints being unable to trim DFHLOG, and so CICS issued a DFHLG0760 message instead. The details of the long-running task (transaction identifier and task number) were displayed in the DFHLG0760 message in each case. Since the long-running tasks completed soon afterwards, there were very few, if any, examples of two successive DFHLG0760 messages. As the long-running tasks completed and their log data was no longer required to be retained on the CICS primary system log stream, subsequent keypoints were able to trim DFHLOG once more and DFHLG0743 messages were resumed.

This behaviour did not by itself warrant any further attention. Provided that the DFHLOG log stream is trimmed in the majority of keypoint operations, the primary storage space for the log stream is being actively managed.

In the example (edited) messages below, it can be seen that a problem occurred at 19.33.17. The activity keypoint that took place then (message DFHRM0205) was not associated with a message DFHLG0743 for DFHLOG. Instead, message DFHLG0760 was issued. This was not an isolated incident, since the same situation occurred for subsequent activity keypoints. Eventually, the number of successive DFHLG0760 messages reached some threshold that warranted further problem determination (this might be five, as in this example). The offending task that was preventing DFHLOG from being trimmed was shown to be *transid: TEST, task number: 01310*, from the DFHLG0760 messages.

Edited CSMT messages:

```
DFHRM0205 19:31:45 An activity keypoint has been successfully taken.  
DFHLG0743 19:31:45 Tail of log stream ANDY.PROD1LOG.DFHLOG deleted at  
block id X'000000000027A11D'.
```

```
DFHRM0205 19:32:13 An activity keypoint has been successfully taken.  
DFHLG0743 19:32:13 Tail of log stream ANDY.PROD1LOG.DFHLOG deleted at  
block id X'000000000047B3C4'.
```

```
DFHRM0205 19:33:17 An activity keypoint has been successfully taken.  
DFHLG0760 19:33:17 Log stream ANDY.PROD1LOG.DFHLOG not trimmed by
```

keypoint processing. Number
of keypoints since last trim occurred: 1. History
point held by transaction: TEST, task number: 01310.

>>>> 3 more activity keypoints occurred, each unable to trim DFHLOG <<<<

DFHRM0205 19:37:38 An activity keypoint has been successfully taken.
DFHLG0760 19:37:38 Log stream ANDY.PROD1LOG.DFHLOG not trimmed by
keypoint processing. Number
of keypoints since last trim occurred: 5. History
point held by transaction: TEST, task number: 01310.

DFHDU02031 19:37:52 A transaction dump was taken for dumpcode: ATCH

DFHAC2236 19:37:52 Transaction TEST abend ATCH in program ANDYPGM term
V999. Updates to local
recoverable resources will be backed out.

DFHRM0205 19:39:00 An activity keypoint has been successfully taken.
DFHLG0743 19:39:00 Tail of log stream ANDY.PROD1LOG.DFHLOG deleted at
block id X'000000000117C37F'.

DFHRM0205 19:39:35 An activity keypoint has been successfully taken.
DFHLG0743 19:39:35 Tail of log stream ANDY.PROD1LOG.DFHLOG deleted at
block id X'000000000147C614'.

CEMT INQ TASK was then used to analyse this task, and the decision was taken to purge it from the CICS system. This action was taken, as seen from the DFHAC2236 message (at time 19.37.52). Once the TEST task had been successfully purged, and its recoverable changes backed out by CICS, its log data on the DFHLOG log stream was no longer required and so CICS was able to free up the space on the log stream at the next activity keypoint. This successful trim of DFHLOG resulted in the resumption of DFHLG0743 messages (time 19.39.00).

This was a straightforward example taken from a deliberately looping task. In reality, however, problem determination with CEMT INQ TASK may be used in conjunction with a short period of auxiliary trace recording. This could reveal, before the decision was taken on whether to issue a purge against it or not, the exact nature of the recoverable work being performed by the offending task. Perhaps the task in question represents a new program that has recently been put into production, and which has a logic error that leads to an invalid (yielding) loop within it. Alternatively,

it may be a truly long-running task that could be expected to complete by itself in the near future.

When printing the auxiliary trace entries for analysis, the trace formatting program could be set to return only trace data pertinent to this particular task, by use of the TASKID= option in the SYSIN statement for the job. This makes trace interpretation much easier, by suppressing those trace entries recorded by other tasks irrelevant to the problem determination.

CONCLUSION

I hope this article has helped shed some light on the use of the enhanced DFHLG0760 messages in assisting with the problem determination of the growth of DFHLOG log streams.

Andy Wright (andy_wright@uk.ibm.com)
CICS Change Team
IBM (UK)

© IBM 2004

TCP/IP printers under CICS: using a TSO netstat command from a CICS transaction

Printers with an IP address can be managed by the TSO **netstat** command. For CICS, it is possible during PLT post-initialization to create a dynamic TSO environment that will be available until CICS stops. Locally, routines SRTRTSOE and SRTREFTS provide these functions. CICS must have access to the libraries containing routines IKJTSOEV and IKJEFTSR. CICS must access library SYSEXEC for its EXECs and standard exit SYSTSPRT.

Other requirements depend on the functions carried out – thus the use of **netstat** requires one profile OMVS (UID) for the userid with the CICS started task.

Transaction S£GT (SARRGT) provides terminal management,

particularly of printers. It makes it possible for CICS to obtain internal information about CICS as well as have an external connection to TCP/IP using the routine SARRVTAM.

The message in the CICS start-up log (TSO is available under CICS) looks like this:

```
STC16635 +W: SRTRTSOE after IKJTSOEV call. RC 0000. TSOEV RC: 0000.  
11.14.29 STC16635 +W: SRTRTSOE after IKJTSOEV call. TSOEV RS 0000.  
TSOEV AB:
```

There is no more logging on and off to CICS and TSO or switching from an application to ICCF to submit TSO commands such as **netstat** for CICS printers under TCP/IP. It permits CICS users to transfer files that would normally require them to be logged in under TSO. It reduces the number of TSO users because they can now transfer files under CICS.

SARRGT

```
CBL XOPTS(ANSI 85, SP) NOTEST SSRANGE  
  IDENTIFICATION DIVISION.  
  PROGRAM-ID.    SARRGT.  
  
  ENVIRONMENT DIVISION.  
  DATA DIVISION.  
  WORKING-STORAGE SECTION.  
  *---  
  01  W-NOMPROG          PIC X(8) VALUE 'SARRGT ' .  
  *-- Retour d'EXEC.  
  01  W-RESP             PIC 9(4).  
  01  W-RESP2            PIC 9(4).  
  01  W-MESS-ANO         PIC X(40) VALUE SPACE.  
  *---  
  LINKAGE SECTION.  
  *-- EXCI communication.  
  01  DFHCOMMAREA.  
      03  L-REQUEST      PIC X(8).  
      03  L-RESP         PIC 9(4).  
      03  L-RESP2        PIC 9(4).  
      03  L-LUNAME       PIC X(8).  
      03  L-TERMI D      PIC X(4).  
      03  L-TERMST       PIC S9(8) BINARY.  
      03  L-SERVST       PIC S9(8) BINARY.  
      03  L-TERMTY       PIC S9(8) BINARY.  
      03  L-TASKI D      PIC S9(8) BINARY.  
      03  L-TRANI D      PIC X(4).
```

```

      03 L-USERID          PIC X(8).
      03 L-USERNM         PIC X(20).
* ---
* ---
PROCEDURE DIVISION.
* ---
TRAIT-REQUEST.
  EVALUATE EIBCALEN
  WHEN NOT 76
    STRING W-NOMPROG ' : Commarea length error.'
    DELIMITED BY SIZE INTO W-MESS-ANO
    EXEC CICS WRITEQ TD
      QUEUE('CSDL')
      FROM(W-MESS-ANO)
      RESP(W-RESP)
    END-EXEC
  WHEN OTHER
    EVALUATE L-REQUEST
    WHEN 'INQUIRE'
* -- Request by LUNAME
      EXEC CICS INQUIRE
        NETNAME(L-LUNAME)
        DEVICE(L-TERMTY)
        RESP(W-RESP)
        RESP2(W-RESP2)
        TASKID(L-TASKID)
        TERMINAL(L-TERMI D)
        TERMSTATUS(L-TERMST)
        TRANSACTION(L-TRANID)
        SERVSTATUS(L-SERVST)
        USERID(L-USERID)
        USERNAME(L-USERNM)
      END-EXEC
    WHEN 'ACQUIRE'
* -- Acquire du terminal.
      EXEC CICS SET TERMINAL(L-TERMI D)
        ACQUIRED
        RESP(W-RESP)
        RESP2(W-RESP2)
      END-EXEC
    WHEN 'RELEASE'
* -- Release du terminal.
      EXEC CICS SET TERMINAL(L-TERMI D)
        RELEASED
        RESP(W-RESP)
        RESP2(W-RESP2)
      END-EXEC
    WHEN 'INSERV'
* -- In service du terminal.
      EXEC CICS SET TERMINAL(L-TERMI D)

```

```

        INSERVICE
        RESP(W-RESP)
        RESP2(W-RESP2)
    END-EXEC
    WHEN 'OUSERV'
*-- Outservice du terminal.
        EXEC CICS SET TERMINAL(L-TERMID)
        OUTSERVICE
        RESP(W-RESP)
        RESP2(W-RESP2)
    END-EXEC
    WHEN 'PURGE'
*-- Purge task.
        EXEC CICS SET TASK(L-TASKID)
        PURGE
        RESP(W-RESP)
        RESP2(W-RESP2)
    END-EXEC
    WHEN 'FORCE'
*-- Forcepurge task.
        EXEC CICS SET TASK(L-TASKID)
        FORCEPURGE
        RESP(W-RESP)
        RESP2(W-RESP2)
    END-EXEC
    WHEN OTHER
        MOVE 99 TO W-RESP
    END-EVALUATE
    END-EVALUATE
    MOVE W-RESP TO L-RESP
    MOVE W-RESP2 TO L-RESP2
    GOBACK.

```

An example of a REXX CLIST that manages TCP/IP printers:

```

/* REXX -- exec identifier comment -- */

/* Controls & initialization. */
PARSE SOURCE . . @xec . /* Get exec name as parm. */
SIGNAL ON SYNTAX /* Traps syntax check */
PARSE ARG @lunm @stad @debug @spill /* Get arguments ASIS. */
IF @lunm = "." THEN call explain /* Active explain feature */
IF @debug ^= '' /* Debugging options */
THEN DO
    @debug = SUBSTR(@debug, 1, 1) /* trunc. */
    IF VERIFY(@debug, ACEFILNOR) = 0 /* valid option? */
    THEN TRACE VALUE(@debug) /* y -> set. */
END
IF @spill ^= '' /* Catch-up parm. */
THEN SAY "W: Unexpected input '@spill'" will not be processed."

```

```

IF @lunm = "" /* Required. */
THEN EXIT 99
IF @stad = "" /* Required. */
THEN EXIT 98
@stad = C2X(@stad)
IF LENGTH(@stad) ^= 8 /* FW required. */
THEN EXIT 97
IF DATATYPE(@stad, 'X') ^= 1
THEN EXIT 96
IF @stad = 0 /* Must be set */
THEN EXIT 95
/* Process: Netstat query on luname */
@? = OUTTRAP("@out.", , "NOCONCAT")
"NETSTAT TELNET (LUN "@lunm
IF RC ^= 0
THEN RETURN RC /* Netstat ?? */
SELECT
WHEN @out.0 = 6
THEN NOP
WHEN @out.0 = 8
/* Put line 7 at designated address. */
THEN @? = STORAGE(@stad, 79, @out. 7)
OTHERWISE DO @ix = 1 BY 1 TO @out.0
SAY VALUE("@out. "@ix)
END
END /* Select. */
RETURN @out.0

/* Basic Syntax error routine. */
syntax:
SAY "E: Syntax Check "RC" in "@xec" from line "SIGL"."
SAY "E: Source: ' "SOURCELINE(SIGL)'" . "
SAY "E: Message: ' "ERRORTXT(RC)'" . "
PARSE SOURCE @env
SAY "E: Environment: ' "@env"' . "
EXIT RC

/* Explain process. */
explain:
"NEWSTACK"
QUEUE @xec" explain; TEST : Ok. ; M. A. J LE 05/12/2002. "
QUEUE " "
QUEUE "OBJECT : Netstat query of a telnet connexion. "
QUEUE " Output is of 6 lines for an unknown connexion (RC 6). "
QUEUE " Output is of 8 lines for an known connexion RC(8). "
QUEUE " If so the seventh line data is returned (as 79 bytes)"
QUEUE " at given address. "
QUEUE "W: Netstat queries the TN3270 session. That half-session"

```

```

QUEUE " may remain even if CICS has its terminal in REL state."
QUEUE " "
QUEUE "HISTORY : "
QUEUE "2002.339 : Tested. "
QUEUE "2002.338 : Ready for testing. "
QUEUE " "
QUEUE "VARIABLES : "
QUEUE "@lunm : LUNAME selection."
QUEUE "@stad : Storage address where to return trapped output."
QUEUE " "
QUEUE "EXECUTION DEPENDENCIES : "
QUEUE "Restriction : OMVS UID needed."
QUEUE " Parameter : "
QUEUE " @lunm : (R)."
QUEUE " @stad : (R)."
QUEUE "Environment : TSO; IKJEFTSR compatible."
QUEUE "Top level execution : Yes for test purpose."
QUEUE " Known call(s) : None."
QUEUE "Low level execution : Normal."
QUEUE " Known caller(s) : PGM = SARTGT."
QUEUE "Noticeable features : CICS driven exec under SRTREFTS!"
QUEUE "" /* EOF. */
"subcom ispeexec"
IF RC = 0 /* Available. */
THEN DO
    "alloc reuse file(xplain) unit(vio) space (1,1) tracks recfm(fb)",
        "lrecl(80) mgmtclas(temp) dsorg(ps)"
    "execio * diskw xplain (finis)"
    "DELSTACK"
    ADDRESS ISPEXEC
    "LINIT DATAID(@XPLAIN) DDNAME(XPLAIN)"
    IF RC > 0
    THEN DO
        "SETMSG MSG("zerrmsg")"
        EXIT 9
    END
    "BROWSE DATAID("@XPLAIN") PANEL(MDBW$)"
    IF RC > 0
    THEN DO
        "SETMSG MSG("zerrmsg")"
        EXIT 9
    END
    "LMFREE DATAID("@XPLAIN)"
    IF RC > 0
    THEN DO
        "SETMSG MSG("zerrmsg")"
        EXIT 9
    END
    ADDRESS TSO "free fi(xplain)"

```



```

END
ELSE DO WHILE QUEUED() > 0
    PARSE PULL @?
    SAY @?
    "DELSTACK"
END

/* REXX -- exec identifier comment -- */

/* Controls & initialization. */
PARSE SOURCE . . @xec . /* Get exec name as parm. */
SIGNAL ON SYNTAX /* Traps syntax check */
PARSE ARG @lunm @applid @debug @spill /* Get arguments ASIS. */
IF @lunm = "." THEN call explain /* Active explain feature */
IF @debug ^= '' /* Debugging options */
THEN DO
    @debug = SUBSTR(@debug, 1, 1) /* trunc. */
    IF VERIFY(@debug, ACEFILNOR) = 0 /* valid option? */
    THEN TRACE VALUE(@debug) /* y -> set. */
END
IF @spill ^= '' /* Catch-up parm. */
THEN SAY "W: Unexpected input '@spill' will not be processed."

IF @lunm = "" /* Required. */
THEN EXIT 99

/* Process: Netstat query on luname */
@? = OUTTRAP("@out. ", , "NOCONCAT")
"NETSTAT TELNET (LUN "@lunm
IF RC ^= 0
THEN RETURN RC /* Netstat ?? */
SELECT
WHEN @out.0 = 6 /* No connection. */
THEN NOP
WHEN @out.0 = 8 /* Connection; ck' applid. */
THEN DO
    IF @applid ^= ''
    THEN DO
        @? = SUBSTR(@out.7, 69, 8)
        IF @? ^= @applid & @? ^= ' '
        THEN @out.0 = 10 /* Already in session. */
    END
END
END
OTHERWISE DO @ix = 1 BY 1 TO @out.0
    SAY VALUE("@out."@ix)
END
END /* Select. */
RETURN @out.0

/* Basic Syntax error routine. */

```

```

syntax:
  SAY "E: Syntax Check "RC" in "@xec" from line "SIGL"."
  SAY "E: Source: ' "SOURCELINE(SIGL)"' ."
  SAY "E: Message: ' "ERRORTXT(RC)"' ."
  PARSE SOURCE @env
  SAY "E: Environment: ' "@env"' ."
EXIT RC

/* Explain process.                               */
explain:
  "NEWSTACK"
  QUEUE @xec" explain; TEST : Ok. ; M.A.J LE 19/02/2003."
  QUEUE " "
  QUEUE "OBJECT : Netstat query of a telnet connexion."
  QUEUE " Output is of 6 lines for an unknown connexion (RC 6)."
  QUEUE " Output is of 8 lines for an known connexion RC(8)."
  QUEUE " If @plid is received a check is made:"
  QUEUE " -RC stays to 8 if equal or no session;"
  QUEUE " -RC is set to 10 if not equal;"
  QUEUE "W: Netstat queries the TN3270 session. That half-session"
  QUEUE " may remain even if CICS has its terminal in REL state."
  QUEUE " "
  QUEUE "HISTORY : "
  QUEUE "2003.50 : New. "
  QUEUE " "
  QUEUE "VARIABLES : "
  QUEUE "@lunm : LUNAME selection. "
  QUEUE "@plid : CICS applid to check. "
  QUEUE " "
  QUEUE "EXECUTION DEPENDENCIES : "
  QUEUE "Restriction : OMVS UID needed. "
  QUEUE " Parameter : "
  QUEUE " @lunm : (R). "
  QUEUE " @plid : (0). "
  QUEUE "Environment : TS0; IKJEFTSR compatible. "
  QUEUE "Top level execution : Yes for test purpose. "
  QUEUE " Known call(s) : None. "
  QUEUE "Low level execution : Normal. "
  QUEUE " Known caller(s) : PGM = SARRVTAM. "
  QUEUE "Noticeable features : CICS driven exec under SRTREFTS|"
  QUEUE "" /* EOF. */
  "subcom ispexec"
  IF RC = 0 /* Available. */
  THEN DO
    "alloc reuse file(xplain) unit(vio) space (1,1) tracks recfm(f b)",
    "lrecl(80) mgmtclas(temp) dsorg(ps)"
    "execio * diskw xplain (finis"
  ADDRESS ISPEXEC
  "LMI NIT DATAID(@XPLAIN) DDNAME(XPLAIN)"

```

```

IF RC > 0
THEN DO
  "SETMSG MSG("zerrmsg")"
  EXIT 9
END
"BROWSE DATAID("@XPLAIN") PANEL(MDBW$)"
IF RC > 0
THEN DO
  "SETMSG MSG("zerrmsg")"
  EXIT 9
END
"LMFREE DATAID("@XPLAIN)"
IF RC > 0
THEN DO
  "SETMSG MSG("zerrmsg")"
  EXIT 9
END
ADDRESS TSO                               /* Swap ev.          */
"free fi (xplai n)"
END
ELSE DO WHILE QUEUED() > 0
  PARSE PULL @?
  SAY @?
END
"DELSTACK"

```

Claude Dunand
Systems Programmer (France)

© Xephon 2004

CICS Transaction Gateway V5.0.1 and WebSphere V5.0 – improvements for global transaction processing

CICS Transaction Gateway (CTG) is IBM's strategic connector for connection to CICS, supporting J2EE for connectivity between the WebSphere software platform for e-business and CICS applications. CTG is a set of client and server software components that allow a Java application to invoke services in a CICS region.

WebSphere Application Server Enterprise Edition V5.0 on

distributed platforms offers new advanced transactional capabilities by providing 'dynamic application adapter support' and 'last participant support'. These enhancements offer the ability to build and deploy rich, open, standards-based application adapters for CICS Transaction Server and provides automated coordination for transactions that include two-phase commit resources and a single one-phase commit resource.

The latest release of CTG V5.0.1 takes advantage of the last participant support in WebSphere Application Server (WAS) V5.0, which allows WAS to include CTG connections in a global transaction and commit the CICS resources pragmatically in the middle of the two-phase commit sequence.

Another very important enhancement from a design point of view is Remote Gateway support on WebSphere Application Server for z/OS. This new feature removes the restriction of running CICS on the same LPAR as WAS when CTG V5.0.1 and WAS 5.0 for z/OS are installed.

Let's take a closer look at CTG and how you can benefit from running the latest version of CTG V5.0.1 with WebSphere Application Server V5.0 on distributed platforms and on z/OS.

CICS TRANSACTION GATEWAY V5 HIGHLIGHTS

Highlights include:

- Available on multiple platforms, including AIX, Windows NT, Windows 2000, Solaris, HP-UX, OS/390, Linux/390.
- Delivers best-of-breed performance and scalability by using memory-based access to optimize connection to CICS applications.
- Provides implementation of the J2EE connector architecture on all supported platforms, including OS/390 – this supports two-phase commit with WebSphere for z/OS.
- Offers a comprehensive security-rich environment for optimal authentication and authorization. CTG External Security

Interface (ESI) enables an application to verify a userid/password combination before it is flowed into CICS. This interface is supported only by a distributed version of the CTG.

- Allows you to choose application server programming models for deployment across most major execution platforms.
- CTG client daemon provides the CICS client/server connectivity using the same technology as previously provided by the CICS universal client.
- On z/OS, the External CICS Interface (EXCI) is used in place of the client daemon, and provides access to COMMAREA-based CICS programs.
- LINKable CICS applications, designed with a COMMAREA interface, are accessed via ECI requests using the CTG's ECIRrequest class. All implementations of the CTG support ECIRrequest.
- 3270-based CICS applications may be accessed via the CTG's External Presentation Interface (EPI) classes. Only the distributed versions of the CTG support EPI.
- CTG on distributed platforms can use a network connection to access the host CICS system via SNA or TCP62 protocol. SNA support is available only for Windows and AIX distributions of the CTG.
- For ECI requests only, CTG client daemon provides TCP/IP connections to CICS TS for z/OS V2.2.

Figure 1 illustrates data flow, using the CICS Transaction Gateway.

CICS TRANSACTION GATEWAY ON OS/390 VERSUS DISTRIBUTED

The primary differences in the ECI support offered when the CTG is running on OS/390 are as follows:

- 1 When using asynchronous calls, specific reply solicitation calls are not supported.

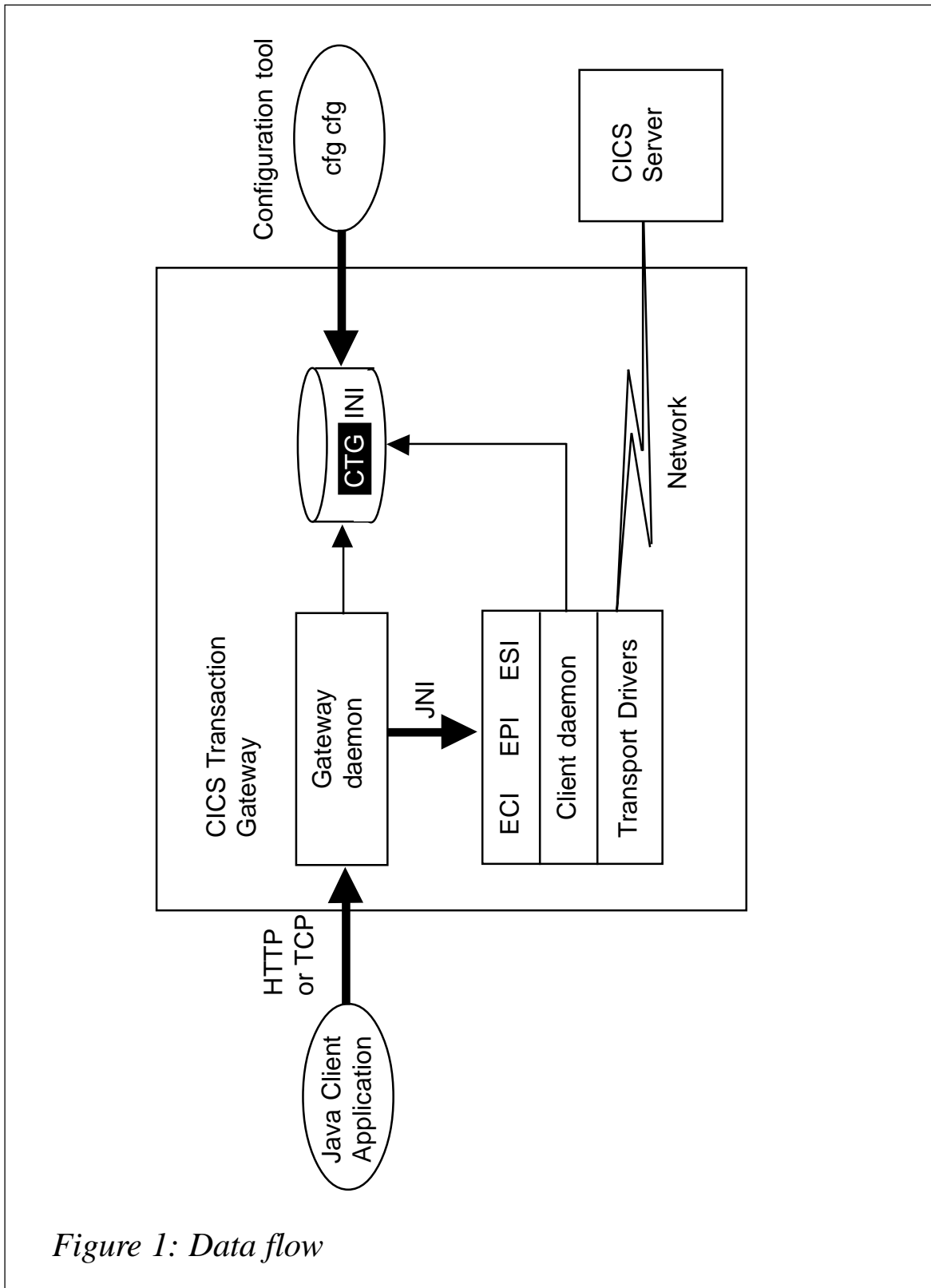


Figure 1: Data flow

- 2 The userid and password passed on ECI requests are verified within the CTG with RACF; afterwards the verified

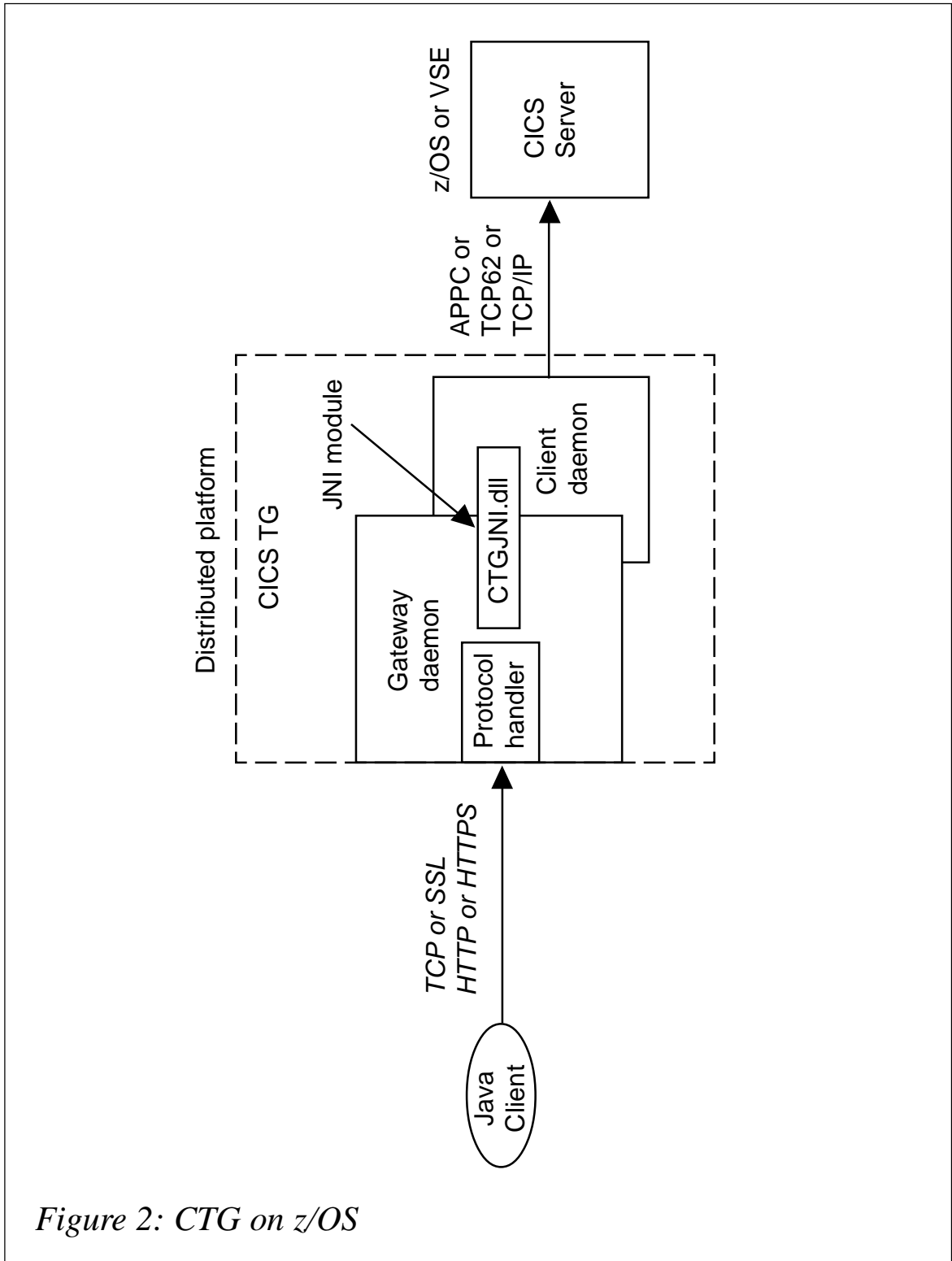


Figure 2: CTG on z/OS

userid is passed on to CICS. With CTG on distributed platforms, both the userid and password are passed to CICS in each ECI request.

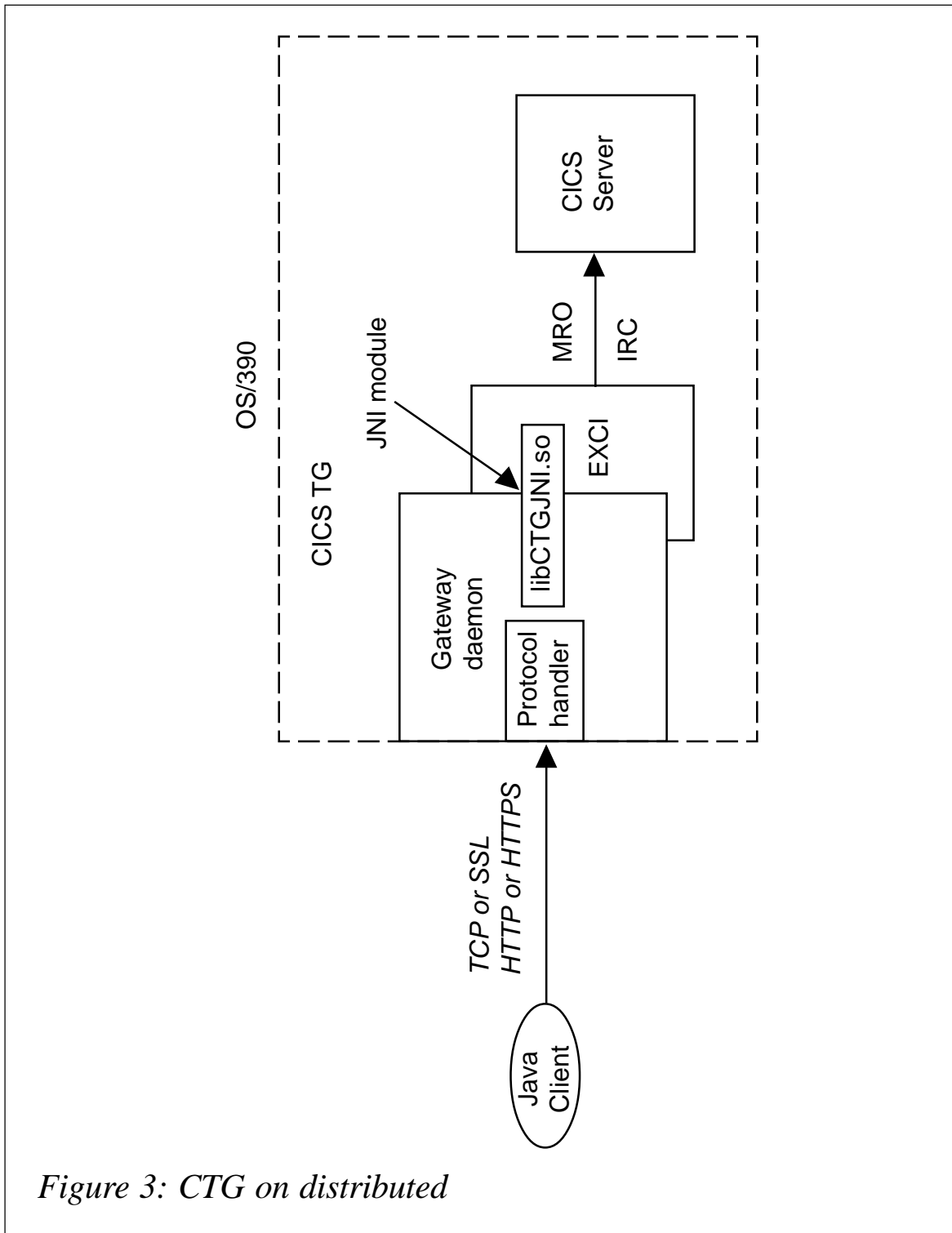


Figure 3: CTG on distributed

- 3 The ECIRRequest method listSystems() does not return the list of usable servers, since any CICS region within the Parallel Sysplex can be reached.

EPI and ESI interfaces are not available with the CICS Transaction Gateway for OS/390.

When running WAS for z/OS, CTG local protocol is used, which directly invokes the underlying transport mechanism using the Java Native Interface (JNI) module libCTGJNI.so – see Figure 2.

WAS for z/OS is an XA-compliant resource and it uses RRS for two-phase commit. WAS on distributed cannot use RRS and can carry out only one-phase commit.

Figures 2 and 3 illustrate the differences between CTG on z/OS and distributed.

CICS TRANSACTION GATEWAY V5.0.1

CTG V5.0.1 is required for interoperation with IBM WebSphere Application Server for z/OS V5. CTG V5.0.1 was electronically generally available (eGA) on 1 August 2003. CICS Universal Client Version 5.0.1 was also eGA on the same date.

The following functional enhancements are delivered in CTG 5.0.1:

- Remote Gateway support on IBM WebSphere Application Server for z/OS.
- Full accessibility, includes discontinuation of EPIBeans, VB/VBscript samples, and some terminal servlet function.
- Control of SSL cipher suite (enforcement of 128 bit SSL).
- IBM AIX V5.2 and IBM z/OS V1.4 support.
- Updated Java 1.3.1 service refresh.
- J2EE ECIInteractionSpec methods setTPNName() and setTranName().
- New EPI exit CICS_EPIStartTranExtendedExit (with Term Index).
- Memory mapped tracing for client daemon.
- EPI/terminal recovery for CICS server outages.

In addition to these enhancements, the following APARs have been addressed:

- Improved performance of EPI flows (null stripping).
- Five re-tries of re-tryable failed EXCI allocates.
- Non-swappable CICS TG address space.
- Improved performance of compression exits.

WEBSHERE AND CICS ECI RESOURCE ADAPTER

WebSphere for z/OS communicates with CICS through one resource adapter per J2EE connector architecture. This resource adapter is written specifically for the Enterprise system it supports, but the same resource adapter can be plugged in to any application server.

Figure 4 shows how WebSphere connects to CICS from a session bean, using the J2EE connector architecture.

The CICS ECI resource adapter allows WebSphere to take much tighter control in how connections to CICS are made. It allows WebSphere to manage the connections by using the system contracts of the resource adapter.

CICSECI resource adapter is required specifically for WebSphere

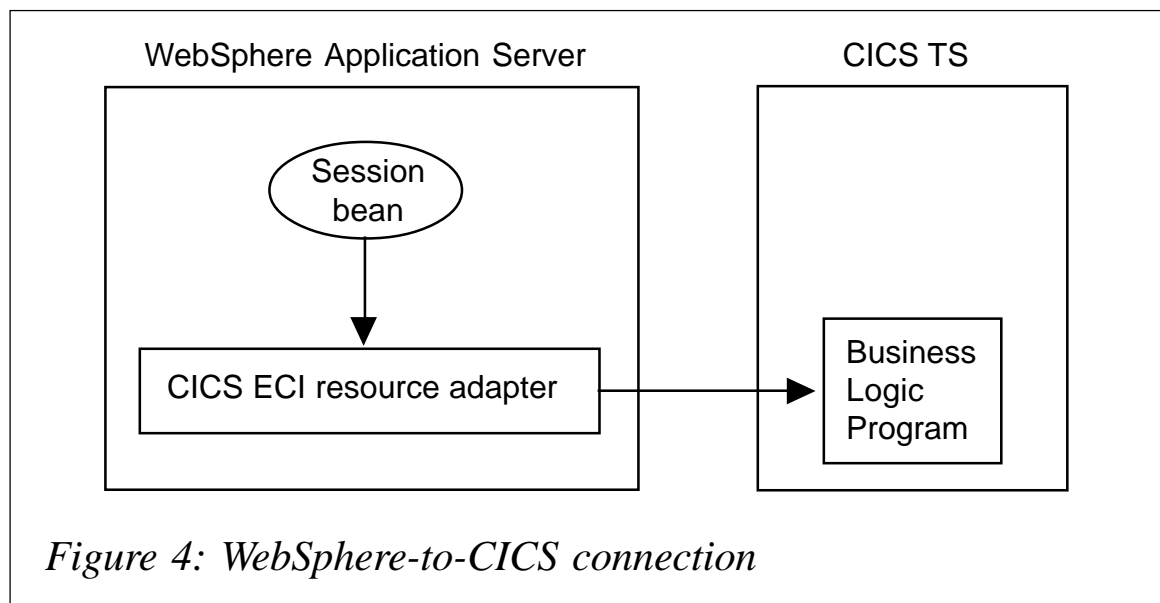


Figure 4: WebSphere-to-CICS connection

on z/OS. This is because WebSphere on z/OS does not implement a fully J2EE connector architecture-compliant environment. The good news is, because WebSphere on z/OS is not constrained to the J2EE connection architecture specification, it can use RRS to provide two-phase commit, which is not available on distributed platforms.

CICS ECI resource adapter allows WebSphere to perform the following:

- Provide two-phase commit processing using RRS.
- Flow security credentials based on deployment information specified in the Enterprise application. The userid you flow to CICS can come from:
 - the Enterprise application
 - the userid of the J2EE server running the enterprise application
 - the userid of the caller of the enterprise application
 - a userid mapped to an EJB security role
- Provide sysplex-wide pooling of connections.

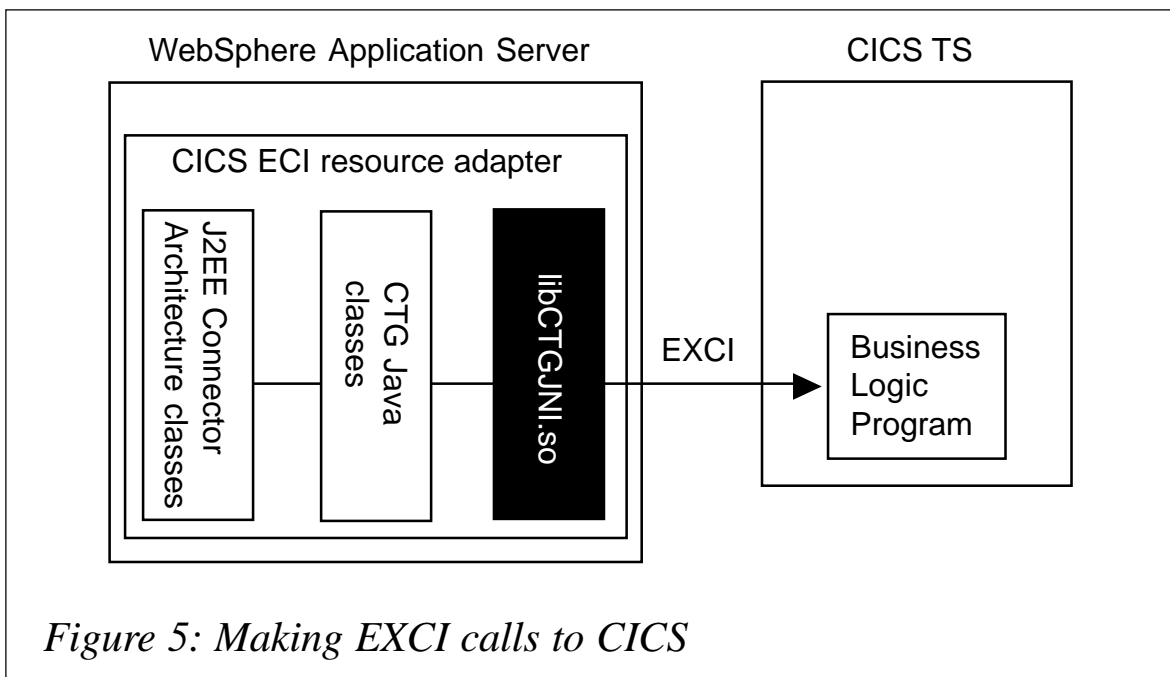


Figure 5: Making EXCI calls to CICS

WHY DO WE NEED CICS TRANSACTION GATEWAY?

CICS Transaction Gateway ships the CICS ECI resource adapter. The CICS ECI resource adapter internally uses the CICS Transaction Gateway in local mode to make the physical connection to CICS. It uses the same set of files as usual – ctgserver.jar to provide the Java interface and libCTGJNI.so to make EXCI calls to CICS, as illustrated in Figure 5.

GLOBAL TRANSACTIONS

Now, let's take a look at how resource adapters are used to coordinate a unit of work. In order to get a better understanding, let's review first what a global transaction is.

A global transaction is a single transaction that performs two-phase commit, allowing applications to coordinate multiple resources, such as Java Connector Architecture (JCA) connectors and relational databases.

In a managed environment, the application server uses a transaction manager to coordinate the transaction. The application server will provide the following functions:

- Inform the transaction manager when a transaction begins.
- Perform the work of the transaction.
- Tell the transaction manager to commit the transaction.

The transaction manager uses the XA resource interface of the resource adapters to coordinate the two-phase commit process across multiple resource managers.

Global transactions are also referred to in the J2EE connector architecture specification as JTA transactions, and they are supported by the resource adapter implementing the `javax.transaction.xa.XAResource` interface.

Two-phase commit works as follows:

- 1 In phase one, the transaction manager asks all resource

managers to prepare to commit their work. If a resource manager can commit its work, it replies affirmatively, and hardens its recoverable data to permanent storage. A negative reply reports an inability to commit for any reason.

- 2 In phase two, the transaction manager directs all resource managers either to commit or roll back work done on behalf of the global transaction, based on the replies from phase one.

WHAT IS AN XA RESOURCE?

XA resource is one that supports the two-phase commit protocol.

WebSphere for z/OS V5 provides XA resource support, but WebSphere on distributed platform does not provide this support, which limits it to one-phase commit.

The coordination of multiple XA resources in a global transaction using two-phase commit is more expensive in terms of system resources than the coordination of a single resource. XA requires additional calls and logging to ensure distributed data integrity. The use of XA resources in J2EE applications is becoming a more and more common occurrence.

It is possible for a resource adapter that does not implement the XA resource interface to participate in a global transaction using last resource optimization (also known as last-agent optimization). This 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, followed by the call to commit for two-phase commit resources.

WHY DO WE NEED 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 is provided by WAS V5.0 on distributed platforms only. On the mainframe, RRS is used for two-phase commit. WAS on distributed can't use RRS and can do only one-phase commit.

Coordinating access to one-phase commit and two-phase commit capable resources within the same transaction introduces an increased risk of an heuristic (ie not necessarily the desired) outcome to the transaction. That is, the transaction manager cannot be sure that all resources were completed in the same direction (either committed or rolled back). For this reason, to enable an application to coordinate access to one-phase commit and two-phase commit capable resources within the same transaction, you configure the application to accept the increased risk of an heuristic outcome.

Applications must be configured to accept the additional risk of such mixed outcome hazards. The transaction service can be configured to write an extra log entry prior to committing the one-phase commit resource. This would ensure proper heuristic reporting during recovery.

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, followed by the call to commit for two-phase commit resources.

LAST PARTICIPANT SUPPORT IN WEBSHERE APPLICATION SERVER V5.0

Last participant support is an important new feature for applications that need to access multiple resource managers when one of those resource managers has a transaction-support of LocalTransaction, like CTG V5 ECI connector. The LocalTransaction resource is enlisted in the transaction by the container and, at transaction commit, the two-phase commit resources are prepared first using the two-phase commit protocol; 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.

Last participant support provides automated coordination for a transaction that includes two-phase commit resources and a single one-phase commit resource. This support eliminates hand-coding in this scenario and allows you to include one-phase commit resources, common for many legacy and package applications, in real transactions.

Last participant support allows you to access exactly one single-phase commit resource manager in a transaction that also involves one or more two-phase resource managers. There is an implicit risk of mixed outcome if the one-phase resource manager fails. XA resources are rolled back but the outcome of the one-phase resource is unknown.

Last participant support can be set in WAS by using the administrative console page: click **Applications/Applications/application_name/Last Participant Support Extension**.

Note: last participant support is not intended to be a performance optimization.

CONFIGURING WAS TO USE A REMOTE INSTANCE OF CTG

Applications executing in WebSphere V4 on OS/390 have a restriction on the CTG implementation. Using the J2EE Connection APIs, one must configure CTG on the same LPAR as the WebSphere J2EE Server and must also place at least one CICS region on that LPAR.

This restriction was removed with WebSphere V5 and CICS TG V5.0.1! Now there is the option of retaining the 'all in one place' configuration or using a 'remote' instance of CTG on either another LPAR or on a distributed platform. Please note that when CTG for z/OS is used, CICS and CTG must reside in the same LPAR.

This is a very important enhancement for customers who would

like to run WebSphere only on a dedicated processor that is not licensed for CICS, achieving considerable saving on the processor licence cost. Until now, for connectivity between WAS for z/OS and CICS TS, this was only possible using CICS TS 2.2 EJB server, using JCICS API, where CICS could reside on the other LPAR.

CONCLUSION

Customers now have more flexibility with CTG placement, where fewer restrictions are placed.

With Remote Gateway support we now have an option of running CICS TS and CTG on the legacy LPAR (using CTG on OS/390 configuration), allowing separation of WebSphere from CICS, if desired.

WebSphere Application Server V5.0 on distributed platforms, using CTG V5.0.1 for OS/390, can now create an effect of two-phase commit, by using last participant support, which provides automated coordination for a transaction that includes two-phase commit resources and a single one-phase commit resource.

These enhancements help CICS Transaction Gateway V5.0.1 and WebSphere Application Server V5 significantly reduce the complexity of interacting with back-end CICS systems, while providing more enriched functionality.

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

© Xephon 2004

```

*                               Function ....: PHASEIN           *
*                               (PHASEIN/NEWCOPY)                *
*                                                                    *
*                               Program .....: CCWT*            *
*                               (PGMNAME/AABB*)                 *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*                                                                    *
*-----*

```

Pf1 ==> Help, Pf3/Clear End work, Enter ==> Select

It's better not to insert in the program field name anything too generic, such as A*, because if there are too many programs that start with A, the table in the program that contains the result may not be large enough .

The result of first map is:

```

*****
*                               Utility for a NEWCOPY of programs in a CICSplex           *
*-----*
*                                                                    *
*                                                                    *
*                               Context .....: CXPLPROD         *
*                                                                    *
*                               Scope .....: ALLAORS             *
*                                                                    *
*                               Function ....: PHASEIN           *
*                                                                    *
*                               Program .....: CCWT*            *
*                                                                    *
*                               Number/Entry:           2        *
*                                                                    *
*                                                                    *
*                                                                    *
*-----*

```

PHASEIN Successfully Executed
 Pf3 ==> End Session, Other Pf Keys to Return Main Menu'

The PHASEIN command is successfully executed and two programs CCWT* have been found.

You need to define in the RDO environment the CPLX transaction associated with the CPSMMENU program in the PCT.

CPSMMENU and CPSMNEWC are COBOL programs, and, like the MPSCPSM mapset, are defined in the PPT.

To compile and assemble the CPSMMENU program you must use the DFHEITVL procedure. For the CPSMNEWC program you need to use the EYUEITVL procedure, using the appropriate parameters.

Compile, linkedit parameter, and considerations about these arguments are on page 71 of the *CICSplex SM Application Programming Guide*.

CPSMNEWC

```

IDENTIFICATION DIVISION.
PROGRAM-ID. CPSMNEWC.
*****
* MODULE NAME = CPSMNEWC CALLED BY CXPSMENU VIA LINK          *
* CICS version TS V1.3 or later and CICSplex SM V140          *
*****

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
Ø1 W-ERRORS-DESC      PIC X(4096) VALUE SPACES.
Ø1 W-L-ERRORS-DESC   PIC S9(8) USAGE BINARY VALUE ZERO.
Ø1 W-TAB-RESOURCE.
    Ø5 W-RESOURCE     PIC X(8) OCCURS 5 TIMES.
Ø1 W-NUMPAR          PIC S9(4) USAGE BINARY.
Ø1 W-IND             PIC S9(4) USAGE BINARY.
Ø1 W-RESPONSE       PIC S9(8) USAGE BINARY.
Ø1 W-REASON         PIC S9(8) USAGE BINARY.
Ø1 W-THREAD         PIC S9(8) USAGE BINARY.
Ø1 W-RESULT-TOK     PIC S9(8) USAGE BINARY.
Ø1 W-RECCNT        PIC S9(8) USAGE BINARY.
Ø1 W-CRITERIA       PIC X(80) VALUE SPACES.
Ø1 W-CRITERIALEN    PIC S9(8) USAGE BINARY.
Ø1 W-INTO-OBJECTLEN PIC S9(8) USAGE BINARY.
Ø1 W-FILTER-TOKEN   PIC S9(8) USAGE BINARY.
Ø1 W-RESULT         PIC S9(8) USAGE BINARY.
Ø1 W-RESULT2        PIC S9(8) USAGE BINARY.
Ø1 W-COUNT          PIC S9(8) USAGE BINARY.
Ø1 W-COUNT2         PIC S9(8) USAGE BINARY.
Ø1 PICZZZ9A         PIC ZZZ9.
Ø1 PICZZZ9B         PIC ZZZ9.
Ø1 PICZZZZZZZ9     PIC ZZZZZZZ9.
Ø1 W-MSG-A.
    Ø5 FILLER       PIC X(Ø1) VALUE ' '.

```

```

Ø5  W-MSG-ACT    PIC  X(Ø8) VALUE SPACES.
Ø5  FILLER      PIC  X(Ø1) VALUE ' '.
Ø5  W-MSG-ACTOK PIC  X(7Ø) VALUE SPACES.
COPY EYULØØ17.
*****
LINKAGE SECTION.
*****
Ø1  DFHCOMMAREA PIC  X(2ØØ).
Ø1  W-COMMAREA  REDEFINES DFHCOMMAREA.
Ø3  W-CICS      PIC  X(Ø8).
Ø3  W-ORA       PIC  X(Ø8).
Ø3  W-DATA      PIC  X(Ø8).
Ø3  W-CONTEXT   PIC  X(Ø8).
Ø3  W-SCOPE     PIC  X(Ø8).
Ø3  W-ACTION.
Ø5  W-ACT       PIC  X(Ø3).
Ø5  FILLER      PIC  X(Ø5).
Ø3  W-PROGRAM   PIC  X(Ø8).
Ø3  W-COUNTO    PIC  X(Ø8).
Ø3  W-CICLO     PIC  X(Ø1).
Ø3  W-APPLID    PIC  X(Ø8).
Ø3  W-USRID     PIC  X(Ø8).
Ø3  W-ESI TO    PIC  X(Ø2).
Ø3  W-MSG.
Ø5  W-MSG1      PIC  X(4Ø).
Ø5  FILLER      PIC  X(Ø1) VALUE '-'.
Ø5  W-MSG2      PIC  X(39).
*****
PROCEDURE DIVISION.
*****
*   Move parameters into corresponding variables
EVALUATE W-ACT
  WHEN 'NEW' MOVE 'NEWCOPY ' TO W-ACTION
  WHEN 'PHA' MOVE 'PHASEIN ' TO W-ACTION
  WHEN OTHER
    MOVE 'KO' TO W-ESI TO
    MOVE 'Action not available, possible: Newcopy/Phasein'
      TO W-MSG
    GO TO ENDIT
END-EVALUATE.
PERFORM CONNECT-CPSM.
PERFORM MAKE-FILTER
PERFORM READ-RESULT
PERFORM SET-PROGRAM
GO TO ENDIT.
* -----
CONNECT-CPSM.
* -----*
*   OBTAIN A CPSM API CONNECTION. *
*   THE API WILL RETURN A TOKEN IDENTIFYING THE THREAD IN *

```

```

*      VARIABLE W-THREAD.                                     *
*-----*
EXEC CPSM CONNECT
      CONTEXT(W-CONTEXT)
      SCOPE(W-SCOPE)
      VERSION(' Ø14Ø' )
      USER(W-USERID)
      THREAD(W-THREAD)
      RESPONSE(W-RESPONSE)
      REASON(W-REASON)

END-EXEC.
IF W-RESPONSE NOT = EYUVALUE(OK) GO TO NO-CONNECT
END-IF.
*-----*

MAKE-FILTER.
*-----*
*      CREATE A FILTER.                                     *
*      CREATE A FILTER WHICH WILL MATCH ONLY THOSE PROGRAMS WITH *
*      A LANGUAGE OF ASSEMBLER.                             *
*      THE FILTER WILL BE USED IN A SUBSEQUENT MARK COMMAND. *
*-----*
STRING 'PROGRAM=' W-PROGRAM
      ' ' DELIMITED BY SPACE
      INTO W-CRITERIA
MOVE 18 TO W-CRITERIALEN.
EXEC CPSM SPECIFY FILTER(W-FILTER-TOKEN)
      CRITERIA(W-CRITERIA)
      LENGTH(W-CRITERIALEN)
      OBJECT('PROGRAM ')
      THREAD(W-THREAD)
      RESPONSE(W-RESPONSE)
      REASON(W-REASON)

END-EXEC.
IF W-RESPONSE NOT = EYUVALUE(OK) GO TO NO-FILTER
END-IF.
*-----*

READ-RESULT.
*-----*
*      GET THE PROGRAM RESOURCE TABLE.                     *
*      CREATE A RESULT SET CONTAINING ENTRIES FOR ALL PROGRAMS *
*      WITH NAMES NOT BEGINNING DFH, EYU OR IBM.           *
*      THE NUMBER OF ENTRIES MEETING THE CRITERIA IS RETURNED IN *
*      VARIABLE W-RECCNT.                                   *
*-----*
MOVE Ø TO W-RESULT.
EXEC CPSM GET OBJECT('PROGRAM ')
      COUNT(W-COUNT)
      FILTER(W-FILTER-TOKEN)
      RESULT(W-RESULT)
      THREAD(W-THREAD)

```

```

                RESPONSE(W-RESPONSE)
                REASON(W-REASON)

        END-EXEC.
        IF W-RESPONSE NOT = EYUVALUE(OK) GO TO NO-GET
        END-IF.
        MOVE W-COUNT TO PICZZZZZZ9.
        MOVE PICZZZZZZ9 TO W-COUNT0.

* -----
SET-PROGRAM.
* -----*
*   RETRIEVE INFORMATION ABOUT EACH PROGRAM.           *
*   FETCH EACH ENTRY AND USE TPARSE TO OBTAIN EACH ATTRIBUTE. *
*   DISPLAY DETAILS OF EACH PROGRAM TO THE USER.      *
* -----*

        IF W-ACT = 'PHA'
                EXEC CPSM PERFORM SET ACTION(' PHASEIN' )
                                RESULT(W-RESULT)
                                ALL
                                THREAD(W-THREAD)
                                RESPONSE(W-RESPONSE)
                                REASON(W-REASON)

                END-EXEC
        ELSE
                EXEC CPSM PERFORM SET ACTION(' NEWCOPY' )
                                RESULT(W-RESULT)
                                ALL
                                THREAD(W-THREAD)
                                RESPONSE(W-RESPONSE)
                                REASON(W-REASON)

                END-EXEC.
        IF W-RESPONSE NOT = EYUVALUE(OK)
                PERFORM NO-PERFORM
                PERFORM READ-FEEDBACK
                GO TO ENDIT
        ELSE
                MOVE W-ACTION                TO W-MSG-ACT
                MOVE 'Successfully Executed ' TO W-MSG-ACTOK
                MOVE W-MSG-A                TO W-MSG
                MOVE 'OK'                   TO W-ESI TO
        END-IF.

* -----*
READ-FEEDBACK.
* -----*

        MOVE 4096 TO W-L-ERRORS-DESC.
        MOVE 2    TO W-COUNT.
        EXEC CPSM FEEDBACK INTO(W-ERRORS-DESC)
                                LENGTH(W-L-ERRORS-DESC)
                                COUNT(W-COUNT)
                                FIRST
                                RESULT(W-RESULT)

```

```

                                THREAD(W-THREAD)
                                RESPONSE(W-RESPONSE)
                                REASON(W-REASON)

END-EXEC.
IF W-RESPONSE NOT = EYUVALUE(OK)
    GO TO NO-FEEDBACK
ELSE
    MOVE W-ERRORS-DESC TO W-MSG
END-IF.
*-----*
*   PROCESSING FOR API FAILURES.   *
*-----*
NO-FEEDBACK.
    MOVE 'Error feedback.' TO W-MSG1.
    MOVE 'KO'                TO W-ESIT0.
    GO TO SCRNL0G.
NO-CONNECT.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error connecting to API.' TO W-MSG1.
    GO TO SCRNL0G.
NO-PERFORM.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error performing set API.' TO W-MSG1.
    PERFORM SCRNL0G.
NO-FILTER.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error creating filter.' TO W-MSG1.
    GO TO SCRNL0G.
NO-GET.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'No program found      ' TO W-MSG1.
    GO TO SCRNL0G.
NO-MARK.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error marking result set.' TO W-MSG1.
    GO TO SCRNL0G.
NO-COPY.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error copying result set.' TO W-MSG1.
    GO TO SCRNL0G.
NO-DELETE.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error deleting from result set.' TO W-MSG1.
    GO TO SCRNL0G.
NO-FETCH.
    MOVE 'KO'                TO W-ESIT0.
    MOVE 'Error fetchng result set.' TO W-MSG1.
    GO TO SCRNL0G.
NO-LOCATE.
    MOVE 'KO'                TO W-ESIT0.

```

```

MOVE 'Error locating top of result set.' TO W-MSG1.
GO TO SCRNLG.
NO-TRANSLATE.
MOVE 'KO' TO W-ESITO.
MOVE 'Error translating attribute.' TO W-MSG1.
GO TO SCRNLG.
SCRNLG.
MOVE W-RESPONSE TO PICZZZ9A.
MOVE W-REASON TO PICZZZ9B.
STRING 'RESPONSE=' DELIMITED BY SIZE
PICZZZ9A DELIMITED BY SIZE
' REASON=' DELIMITED BY SIZE
PICZZZ9B DELIMITED BY SIZE
INTO W-MSG2.

ENDIT.
*-----*
* TERMINATE API CONNECTION. *
*-----*
EXEC CPSM TERMINATE RESPONSE(W-RESPONSE) REASON(W-REASON)
END-EXEC.
EXEC CICS RETURN END-EXEC.
END-PROGRAM.

```

CPSMMENU

```

IDENTIFICATION DIVISION.
PROGRAM-ID. CPSMMENU.
DATE-WRITTEN JULY 2003.
*****
* THE TRANSID PROGRAM IS 'CPLX' AND THE PROGRAM IS FOR *
* CICSplex SM V140 AND CICS TS V1.3 OR LATER *
*****
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 PROGRAM-NAME PIC X(08) VALUE 'CPSMMENU'.
77 TRANSID-NAME PIC X(04) VALUE 'CPLX'.
77 IN-NETNAME PIC X(08) VALUE SPACES.
77 UNDERSCORE PIC X(08) VALUE ALL '_'.
*****
*--> USER'S MESSAGES
*****
77 MSG-PFKO PIC X(58) VALUE
' Pf Key not Valid '.
77 MSG-FORMKO PIC X(58) VALUE
' Please select an action (Max. 8 Chars) '.
01 MSG-FORMKO2.
03 FILLER PIC X(51) VALUE
' Please select a program name or a partial name with'.

```



```

Ø3 FILLER PIC X(20) VALUE
' a wildcard ' .
Ø1 END-MESSAGE.
Ø3 FILLER PIC X(58) VALUE
' CPLX Program terminated - Goodbye and thank you ' .
Ø3 FILLER PIC X(04) VALUE SPACES.
Ø1 OK-MESSAGE.
Ø3 FILLER PIC X(58) VALUE
' Newcopy successfully executed ' .
Ø3 FILLER PIC X(08) VALUE SPACES.
Ø1 KO-MESSAGE.
Ø3 FILLER PIC X(58) VALUE
' CPLX Program abnormally terminated, contact your system pr' .
Ø3 FILLER PIC X(08) VALUE ' ogrammer' .
Ø1 PAUSE-MESSAGE.
Ø3 FILLER PIC X(46) VALUE
' Request Forwarded - Please Wait for the Answer' .
Ø3 FILLER PIC X(16) VALUE
' ...Working... ' .
Ø1 ABS-TIME PIC S9(15) COMP-3 VALUE +0.
Ø1 ORA-OGGI .
Ø3 HH-OGGI PIC 9(02) VALUE 0.
Ø3 MN-OGGI PIC 9(02) VALUE 0.
Ø3 SS-OGGI PIC 9(02) VALUE 0.
Ø1 DATA-OGGI .
Ø3 AA-OGGI PIC 9(02) VALUE 0.
Ø3 MM-OGGI PIC 9(02) VALUE 0.
Ø3 GG-OGGI PIC 9(02) VALUE 0.
Ø1 REC-TERM.
Ø3 IN-TRAN PIC X(04) VALUE SPACES.
Ø3 IN-SEP PIC X(01) VALUE SPACES.
Ø3 IN-RISO PIC X(04) VALUE SPACES.
Ø3 FILLER PIC X(11) VALUE SPACES.
Ø1 MSG-WTO.
Ø3 FILLER PIC X(10) VALUE '-CPSMMENU ' .
Ø3 FILLER PIC X(11) VALUE ' Newcopy ' .
Ø3 WTO-CMD PIC X(08) VALUE ' Command: ' .
Ø3 FILLER PIC X(01) VALUE ' ' .
Ø3 WTO-PGM PIC X(08) VALUE SPACES.
Ø3 WTO-ESIT O PIC X(12) VALUE ' Executed ' .
Ø3 FILLER PIC X(03) VALUE ' by ' .
Ø3 WTO-USER PIC X(08) VALUE SPACES.
Ø3 FILLER PIC X(03) VALUE ' - ' .
Ø3 DATA-WTO.
Ø5 WTO-GG PIC X(02) VALUE SPACES.
Ø5 FILLER PIC X(01) VALUE '/' .
Ø5 WTO-MM PIC X(02) VALUE SPACES.
Ø5 FILLER PIC X(01) VALUE '/' .
Ø5 WTO-AA PIC X(02) VALUE SPACES.
Ø3 FILLER PIC X(03) VALUE ' - ' .

```

```

Ø3  ORA-WTO.
    Ø5  WTO-HH  PIC X(Ø2) VALUE SPACES.
    Ø5  FILLER  PIC X(Ø1) VALUE ':'.
    Ø5  WTO-MN  PIC X(Ø2) VALUE SPACES.
    Ø5  FILLER  PIC X(Ø1) VALUE ':'.
    Ø5  WTO-SS  PIC X(Ø2) VALUE SPACES.
Ø3  FILLER      PIC X(Ø7) VALUE ' *** '.
Ø1  LENG-WTO    PIC S9(Ø4) COMP VALUE +1ØØ.
Ø1  LENG-COMM   PIC S9(Ø4) COMP VALUE +2ØØ.
Ø1  W-COMMAREA.
    Ø3  W-CICS   PIC X(Ø8).
    Ø3  W-ORA    PIC X(Ø8).
    Ø3  W-DATA   PIC X(Ø8).
    Ø3  W-CONTEXT PIC X(Ø8).
    Ø3  W-SCOPE  PIC X(Ø8).
    Ø3  W-ACTION PIC X(Ø8).
    Ø3  W-PROGRAM PIC X(Ø8).
    Ø3  W-COUNT  PIC X(Ø8).
    Ø3  W-CICLO  PIC X(Ø1).
    Ø3  W-APPLID.
        Ø5  W-APPL  PIC X(Ø3).
        Ø5  FILLER  PIC X(Ø5).
    Ø3  W-USRID  PIC X(Ø8).
    Ø3  W-ESI TO  PIC X(Ø2).
    Ø3  W-MSG    PIC X(8Ø).
    Ø3  FILLER   PIC X(45).

```

COPY MPSCPSM.

COPY DFHBMSCA.

COPY DFHAID.

LINKAGE SECTION.

```

Ø1  DFHCOMMAREA      PIC X(2ØØ).

```

* P R O C E D U R E D I V I S I O N

PROCEDURE DIVISION.

MAIN.

```

EXEC CICS IGNORE CONDITION ERROR END-EXEC.
MOVE LOW-VALUE TO MPSØ10 MPSØ20.
PERFORM ASSIGN-CICS THRU EX-ASSIGN-CICS
IF EIBCALEN = Ø
PERFORM SEND-MAP1 THRU EX-SEND-MAP1.
MOVE DFHCOMMAREA TO W-COMMAREA.
IF W-CICLO = '1'
PERFORM RECEIVE-MAP1 THRU EX-RECEIVE-MAP1
PERFORM KEY-MANAGEMENT THRU EX-KEY-MANAGEMENT
PERFORM WORK-PROCESS THRU EX-WORK-PROCESS
MOVE 'OK' TO W-ESI TO
MOVE OK-MESSAGE TO W-MSG

```

```

        PERFORM WRITE-QUE          THRU EX-WRITE-QUE
        PERFORM SEND-MAP2         THRU EX-SEND-MAP2.
    IF W-CICLO = ' 2'
        PERFORM RECEIVE-MAP2     THRU EX-RECEIVE-MAP2.
    IF W-CICLO = ' 3'
        PERFORM RECEIVE-HLP      THRU EX-RECEIVE-HLP.
    MOVE 'Error cycle on RETURN TRANSID' TO
        PAUSE-MESSAGE.
    PERFORM END-PGM              THRU EX-END-PGM.
LABEL-END.
    EXIT.
*****
*      END TASK
*****
END-PGM.
    EXEC CICS
            SEND          FROM          (PAUSE-MESSAGE)
                        LENGTH        (62)
                        ERASE
    END-EXEC.
    EXEC CICS
            RETURN
    END-EXEC.
    GOBACK.
EX-END-PGM.
    EXIT.
*****
** KEY MANAGEMENT AND WORK *****
*****
KEY-MANAGEMENT.
    IF EIBAID = DFHPF1
        PERFORM SEND-MAP-HLP     THRU EX-SEND-MAP-HLP.
    IF EIBAID = DFHPF3          OR
        DFHCLEAR
        MOVE END-MESSAGE        TO PAUSE-MESSAGE
        PERFORM END-PGM         THRU EX-END-PGM.
    IF EIBAID NOT = DFHENTER
        MOVE MSG-PFKO           TO MP1MSGO
        PERFORM SEND-MAP1      THRU EX-SEND-MAP1.
*--> FOR DFHENTER WE PROCEED ON WORKING
EX-KEY-MANAGEMENT.
    EXIT.
WORK-PROCESS.
    PERFORM CTRL-FIELDS         THRU EX-CTRL-FIELDS.
    MOVE MP1CTXI                TO W-CONTEXT.
    MOVE MP1SCOI                TO W-SCOPE.
    MOVE MP1FUNI                TO W-ACTION.
    MOVE MP1PGMI                TO W-PROGRAM.
    INSPECT W-PROGRAM           REPLACING ALL '_' BY SPACES.
    EXEC CICS LINK PROGRAM(' CPSMNEWC' )

```

```

                COMMAREA (W-COMMAREA)
                LENGTH   (LENG-COMM)
END-EXEC.
IF W-ESIT0 NOT EQUAL 'OK'
    MOVE W-MSG TO MP1MSG0
    PERFORM SEND-MAP1 THRU EX-SEND-MAP1
ELSE
    MOVE W-ACTION TO MP2FUN0
    MOVE W-MSG TO MP2MSG0
    MOVE W-PROGRAM TO MP2PGM0
    MOVE W-COUNT TO MP2ENT0.
EX-WORK-PROCESS.
EXIT.
CTRL-FIELDS.
IF MP1FUNI EQUAL SPACES OR LOW-VALUE OR
(MP1FUNI NOT = 'NEWCOPY ' AND NOT = 'PHASEIN ')
    MOVE MSG-FORMK0 TO MP1MSG0
    MOVE -1 TO MP1FUNL
    PERFORM SEND-MAP1-C THRU EX-SEND-MAP1-C
ELSE
    NEXT SENTENCE.
IF MP1PGMI = SPACES OR LOW-VALUE OR UNDERSCORE
    MOVE MSG-FORMK02 TO MP1MSG0
    MOVE -1 TO MP1PGML
    PERFORM SEND-MAP1-C THRU EX-SEND-MAP1-C.
EX-CTRL-FIELDS.
EXIT.
* RETURN TO TRANSID ITSELF
RETURN-TRANSID.
EXEC CICS
        RETURN TRANSID (TRANSID-NAME)
        COMMAREA (W-COMMAREA)
        LENGTH (LENG-COMM)
END-EXEC.
EX-RETURN-TRANSID.
EXIT.
* MAP FUNCTIONS
SEND-MAP1.
EXEC CICS
        SEND MAP ('MPS01')
        MAPSET ('MPSCPSM')
        FREEKB
        FRSET
        ERASE
        ALARM
END-EXEC.
MOVE '1' TO W-CICLO.
PERFORM RETURN-TRANSID THRU EX-RETURN-TRANSID.
EX-SEND-MAP1.
EXIT.

```

```

SEND-MAP2.
EXEC CICS
      SEND      MAP          ('MPSØ2')
              MAPSET        ('MPSCPSM')
              FREEKB
              FRSET
              ERASE
              ALARM

      END-EXEC.
      MOVE '2' TO W-CICLO.
      PERFORM RETURN-TRANSID THRU EX-RETURN-TRANSID.
EX-SEND-MAP2.
EXIT.
SEND-MAP-HLP.
EXEC CICS
      SEND      MAP          ('MPSHLP')
              MAPSET        ('MPSCPSM')
              FREEKB
              FRSET
              ERASE
              ALARM

      END-EXEC.
      MOVE '3' TO W-CICLO.
      PERFORM RETURN-TRANSID THRU EX-RETURN-TRANSID.
EX-SEND-MAP-HLP.
EXIT.
SEND-MAP1-C.
EXEC CICS
      SEND      MAP          ('MPSØ1')
              MAPSET        ('MPSCPSM')
              CURSOR
              FREEKB
              FRSET
              ERASE
              ALARM

      END-EXEC.
      MOVE '1' TO W-CICLO.
      PERFORM RETURN-TRANSID THRU EX-RETURN-TRANSID.
EX-SEND-MAP1-C.
EXIT.
RECEIVE-MAP1.
EXEC CICS
      RECEIVE   MAP          ('MPSØ1')
              MAPSET        ('MPSCPSM')

      END-EXEC.
EX-RECEIVE-MAP1.
EXIT.
RECEIVE-MAP2.
EXEC CICS
      RECEIVE   MAP          ('MPSØ2')

```

```

                                MAPSET      (' MPSCPSM' )
END-EXEC.
IF EIBAID = DFHPF3              OR
    DFHCLEAR
    MOVE      END-MESSAGE      TO      PAUSE-MESSAGE
    PERFORM  END-PGM           THRU  EX-END-PGM.
    PERFORM  SEND-MAP1        THRU  EX-SEND-MAP1.
EX-RECEIVE-MAP2.
EXIT.
RECEIVE-HLP.
EXEC CICS
    RECEIVE  MAP              (' MPSHLP' )
           MAPSET            (' MPSCPSM' )
END-EXEC.

```

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

Marco Busichella
Systems Programmer (Italy)

© Xephon 2004

Microsoft has released the beta version of its Host Integration Server (HIS) 2004. The new version adds support for XML, Microsoft's .NET Framework, and Visual Studio .NET development tools. This is good news for CICS sites that want to let PC users access applications.

Through its .NET Framework and Visual Studio .NET, Microsoft has included a new Transaction Integrator (TI) that links Windows-based computers to mainframe and mid-range applications. The new TI Designer lets developers wrap existing applications like CICS or IMS and breaks them down into .NET client components. Windows programs can make calls as clients to mainframes as servers, and host-initiated processing allows mainframes and AS/400s to act as clients to Windows servers.

HIS 2004 accesses CICS etc using ODBC, COM-based OLE DB, or .NET Framework-enabled Data Providers for DB2, as well as an OLE DB Provider for AS/400 and mainframe file systems. In addition, HIS 2004 offers a two-phase commit distributed transactions over TCP/IP to DB2 running on most popular computing platforms, plus a new data access tool allowing administrators and developers to easily create and manage connections to DB2 and host file systems.

With security being a big issue for Microsoft platforms, the company has revamped the support for Secure Sockets Layer (SSL) and Transport Layer Security (TLS).

The technology allows for a connection for SNA applications across a routable IP network by supporting HPR/IP (High-Performance Routing over IP).

For further information contact your local Microsoft representative.
URL: <http://www.microsoft.com>.

Cape Clear Software has announced Cape Clear Data Interchange, which presents a Web services-based approach to integrating data with enterprise applications.

The product provides a visual environment for transforming diverse data sources, such as text files, spreadsheets, and ZIP files, into XML Schema. A run-time capability securely routes data back to the appropriate back-end application.

A series of wizards analyse new data and suggest appropriate XML Schema representations and mappings, to reduce complexity of describing and integrating new data formats. Pre-built support is included for a variety of data formats, including CICS, CORBA, Excel, EDI, Java, .Net, and WSDL/SOAP. Support for proprietary or non-standard formats can be added by data analysts.

The product also supports Business Process Execution Language (BPEL) for sequential workflows and composite Web services. Through this support, users can chain sets of Web services to deliver a single, composite service.

Cape Clear Data Interchange requires the Cape Clear Business Integration Suite, which includes the tools to design, develop, integrate, deploy, and manage Web Services. It is available immediately on IBM AIX, Linux, Microsoft Windows, and Sun Solaris.

For further information contact:
Cape Clear Software, 880 Winter Street,
Suite 300, Waltham, MA 02451, USA.
Tel: (781) 622 2258.
URL: <http://www.capeclear.com/products/di/index.shtml>.

* * *

