



# 171

# CICS

*February 2000*

---

## **In this issue**

- 3 Duplicating SMP/E zones
  - 13 Further CICS Log Manager enhancements
  - 26 Controlling CICS resources from a batch program using the CICSplex SM API – part 2
  - 37 Monitoring CICS activity
  - 48 CICS news
- 

© Xephon plc 2000

# update

# **CICS Update**

---

## **Published by**

Xephon  
27-35 London Road  
Newbury  
Berkshire RG14 1JL  
England  
Telephone: 01635 38030  
From USA: 01144 1635 38030  
E-mail: [trevore@xephon.com](mailto:trevore@xephon.com)

## **North American office**

Xephon/QNA  
1301 West Highway 407, Suite 201-405  
Lewisville, TX 75077-2150  
USA  
Telephone: 940 455 7050

## **Contributions**

Articles published in *CICS Update* are paid for at the rate of £170 (\$250) per 1000 words and £90 (\$140) per 100 lines of code for original material. To find out more about contributing an article, without any obligation, please contact us at any of the addresses above and we will send you a copy of our *Notes for Contributors*.

## **CICS Update on-line**

Code from *CICS Update* can be downloaded from our Web site at <http://www.xephon.com/cicsupdate.html>; you will need the user-id shown on your address label.

## **Editor**

Trevor Eddolls

## **Disclaimer**

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

## **Subscriptions and back-issues**

A year's subscription to *CICS Update*, comprising twelve monthly issues, costs £175.00 in the UK; \$270.00 in the USA and Canada; £181.00 in Europe; £187.00 in Australasia and Japan; and £185.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the January 1994 issue, are available separately to subscribers for £16.00 (\$23.50) each including postage.

---

© Xephon plc 2000. 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. Copying permits are available from Xephon in the form of pressure-sensitive labels, for application to individual copies. A pack of 240 labels costs \$36 (£24), giving a cost per copy of 15 cents (10 pence). To order, contact Xephon at any of the addresses above.

*Printed in England.*

## Duplicating SMP/E zones

The use of SMP/E to control the maintenance levels of MVS (OS/390) software and subsystems such as CICS and DB2 is a relatively uncomplicated task. There is the Global zone, which contains the index to the Target and the Distribution (known as DLIB) zones; the Target zone, which contains the software maintenance information on the target libraries; and the DLIB zone, which contains the software maintenance information on the distribution libraries.

Software (FUNCTIONS) and maintenance (PTFs/APARs) are RECEIVED into the Global zone, APPLIED to the Target zone and the corresponding target libraries, and ACCEPTED to the DLIB zone and the corresponding distribution libraries. When new maintenance doesn't work out, the target libraries are restored to their prior level of maintenance by RESTOREing the maintenance applied to the Target zone from the DLIB zone. Copying the actual member from distribution libraries to the target libraries is carried out under SMP/E control.

The set-up of the Global, Target, and Distribution zones is straightforward – usually there is an install library (PDS) containing the JCL member for defining and initializing the SMP/E Global, Target, and DLIB zones' CSI (Consolidate Software Inventory, organized as VSAM KSDS) datasets. There's also the definition JCL for the actual target and distribution libraries for the software product. Also, there may well be JCL procedures and control statements set up already to RECEIVE, APPLY, and ACCEPT the software and maintenance. The systems programmer merely changes the JCL in these JCL members to follow the installation standards, and executes them as applicable.

Sounds pretty simple and straightforward, until one realizes that sometimes more than one Target zone is necessary because there is often more than one set of target libraries – sometimes known as run-time libraries – in use at an installation. For example, just for the CICS or DB2 subsystems, there could be as many as three sets of target libraries in use – and one set for SMP/E or systems programmer test, one set for the test regions/subsystems, and one set for the production

regions/subsystems. However, if the SMP/E environment remains the same one for each of the Global, Target, and DLIB zones, several problems may arise:

- APPLY of PTFs with the GROUP parameter (where specification of one or more PTFs may result in all other required PTFs being included in the APPLY process) can happen only once. After the first APPLY (most probably to the SMP/E or systems test target libraries), maintenance must be either copied to other target libraries using IEBCOPY, which is outside SMP/E control, or PTFs needing to be APPLIED to the other set(s) of target libraries must be specified in the APPLY control statements along with the REDO operand.
- A restore of PTFs can also happen once only to one set of libraries. However, in this case there is no workaround to allow SMP/E to do the work again – maintenance must be copied from the target libraries with the restored maintenance level to other sets of target libraries with IEBCOPY, again outside SMP/E control.

It would be extremely difficult to keep track of maintenance levels for different sets of target libraries if subsystems with multiple maintenance levels are required, in environments using products from Independent Software Vendors (ISVs).

In this case, it would be nice to have multiple Target zones' indexes in the Global zone relating to the same DLIB zone, with each Target zone corresponding to a specific set of target libraries. A set-up with multiple Target zones means that the systems programmer is allowed the luxury of:

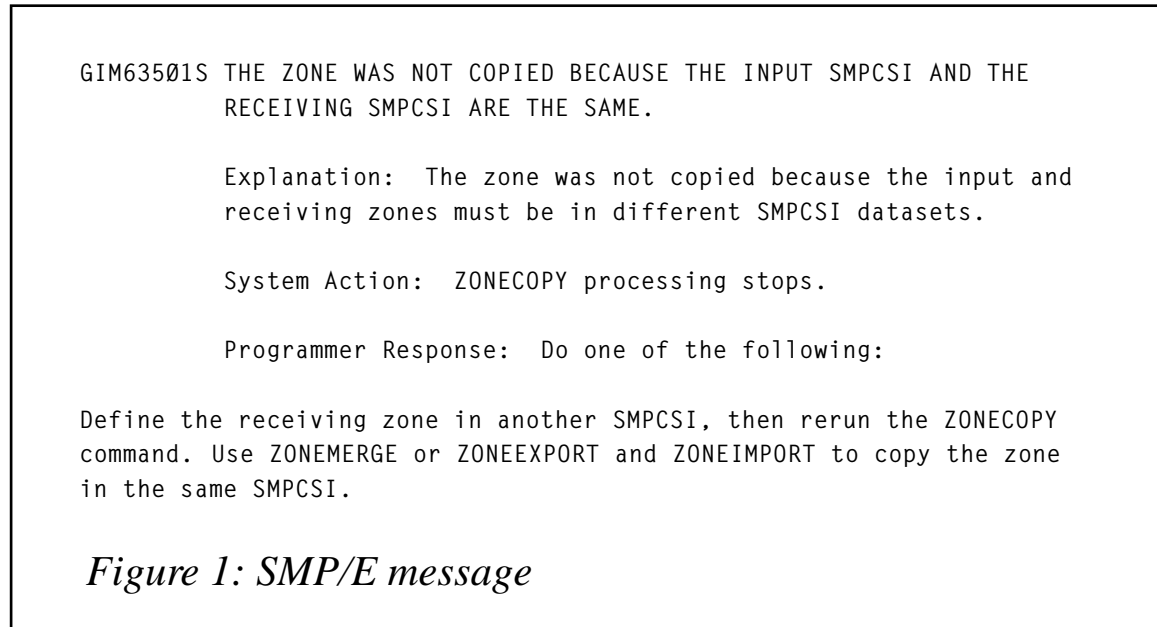
- Usage of the same set of APPLY control statements to apply maintenance to multiple Target zones/target libraries.
- Restoring maintenance for different sets of target libraries under the control of SMP/E.
- Having different target library sets at different maintenance levels.

The following is an example procedure for setting up multiple SMP/E Target zones using the ZONECOPY command and UCLIN statements.

According to *SMP/E R8.1 Reference Summary SX22-0006-11 2.22.1*, the input and receiving zones must be:

- Defined in the same Global zone
- Located in different CSI datasets.

If we run our ZONECOPY with the zones being defined in the same SMP/E CSI datasets, we will get the SMP/E message shown in Figure 1.



So the first action item is to define a new Target zone SMP/E CSI dataset and add its index entry to the Global zone. The LIST GLOBALZONE command is issued to display our zone indexes:

```
GIMSMP LVL 18.1.07 SMPLIST OUTPUT

GLOBAL ZONE ENTRIES

NAME

GLOBAL OPTIONS = CIC0410
ZONEINDEX = DZONE DLIB CICS.REL41.DZONE.CSI
           TZONE TARGET CICS.REL41.TZONE.CSI
SREL = C150
FMID = HCI4100 JCI4101 JCI4102 JCI4103 JCI4106 JCI4107
```

And the following IDCAMS JCL and control statement are submitted to create and initialize a new SMP/E CSI dataset for our new Target

zone. Instead of coding the VSAM parameters, we use the current Target zone CSI (CICS.REL41.TZONE.CSI) as a model for our new Target zone CSI (CICS.REL41.TZ9903.CSI):

```
//IDCAMs EXEC PGM=IDCAMs,REGION=4096K
//SMPEINIT DD DISP=SHR,DSN=SYS1.MACLIB(GIMZPOOL)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CL(NAME(CICS.REL41.TZ9903.CSI) MODEL(CICS.REL41.TZONE.CSI))
REPRO IFILE(SMPEINIT) ODS(CICS.REL41.TZ9903.CSI)
```

The results are shown in Figure 2.

```
IDCAMs SYSTEM SERVICES                                TIME: 08:12:04
05/28/99      PAGE      1

      DEFINE CL(NAME(CICS.REL41.TZ9903.CSI) MODEL(CICS.REL41.TZONE.CSI))

IDC0508I DATA ALLOCATION STATUS FOR VOLUME ODBS06 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME ODBS06 IS 0
IDC0512I NAME GENERATED-(D) CICS.REL41.TZ9903.CSI.DATA
IDC0512I NAME GENERATED-(I) CICS.REL41.TZ9903.CSI.INDEX
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

      REPRO IFILE(SMPEINIT) ODS(CICS.REL41.TZ9903.CSI)

IDC0005I NUMBER OF RECORDS PROCESSED WAS 1
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

*Figure 2: Using current Target zone CSI as a model*

Now that our new Target zone CSI dataset is ready, we execute the following SMP/E UCLIN statements against our Global zone to add the Zoneindex for this new Target zone:

```
//UCLIN EXEC PGM=GIMSMP,
//          REGION=4096K,
//*        TIME=120,
//          PARM='CSI=CICS.REL41.GZONE.CSI'
//SMPLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(GLOBAL ).
```

```

PAGE 0001 - NOW SET TO GLOBAL ZONE      DATE 05/28/99  TIME 08:14:13
GIMSMP LVL 18.1.07 SMPOUT  OUTPUT

GIM42401I  THE FOLLOWING PARAMETERS WERE SPECIFIED ON THE EXEC STATEMENT FOR GIMSMP:
'CSI=CICS.REL41.GZONE.CSI'.

      SET BDY(GLOBAL ).

GIM20501I  SET PROCESSING IS COMPLETE. THE HIGHEST RETURN CODE WAS 00.

      UCLIN .

      ADD GLOBALZONE ZONEINDEX((TZ9903,CICS.REL41.TZ9903.CSI,TARGET)) .

GIM25501I  ENTRY GLOBAL WAS UPDATED BY UCLIN.

      ENDUCL.

GIM52402I  UCLIN PROCESSING IS COMPLETE FOR THE GLOBAL ZONE.

```

*Figure 3: Adding the Zoneindex*

```

UCLIN .
ADD GLOBALZONE ZONEINDEX((TZ9903,CICS.REL41.TZ9903.CSI,TARGET)) .
ENDUCL.

```

We should see the output shown in Figure 3. Doing another **SMP/E LIST GLOBALZONE**, we should now see the results shown below:

```

GIMSMP LVL 18.1.07 SMPLIST  OUTPUT

GLOBAL ZONE ENTRIES

NAME

GLOBAL  OPTIONS      = CIC0410

      ZONEINDEX = DZONE  DLIB   CICS.REL41.DZONE.CSI
              TZONE  TARGET CICS.REL41.TZONE.CSI
              TZ9903 TARGET CICS.REL41.TZ9903.CSI
      SREL      = C150
      FMID      = HCI4100 JCI4101 JCI4102 JCI4103 JCI4106 JCI4107

```

Our new Target zone is now ready for us to **ZONECOPY** against it. The following **SMP/E** control statement is executed:

```
//ZONECOPY EXEC PGM=GIMSMP,
//          REGION=4096K,
//*        TIME=120,
//          PARM='CSI=CICS.REL41.GZONE.CSI'
//SMPLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZ9903).
ZONECOPY (TZONE) INTO(TZ9903) RELATED(DZONE) OPTIONS(CIC0410) .
```

And the output shown in Figure 4 is received.

```
PAGE 0001 - NOW SET TO TARGET ZONE TZ9903 DATE 05/28/99 TIME 08:14:57
GIMSMP LVL 18.1.07 SMPDUT  OUTPUT

GIM42401I THE FOLLOWING PARAMETERS WERE SPECIFIED ON THE EXEC STATEMENT FOR GIMSMP:
'CSI=CICS.REL41.GZONE.CSI'.

      SET BDY(TZ9903).

GIM20501I SET PROCESSING IS COMPLETE. THE HIGHEST RETURN CODE WAS 00.

      ZONECOPY (TZONE) INTO(TZ9903) RELATED(DZONE) .

GIM20501I ZCOPY PROCESSING IS COMPLETE. THE HIGHEST RETURN CODE WAS 00.
```

*Figure 4: Result of ZONECOPY against new Target zone*

Two verifications can be executed to validate the correctness of our ZONECOPY:

- Execute the LIST NOACCEPT(DLIB zone name) command against both the current and the new Target zones. The list of non-accepted PTFs and functions should be identical.
- Execute the LISTNOAPPLY(current Target zone name) command against the new Target zone. We should receive the 'GIM20501I List processing is complete. The highest return code was 04' message, with the output shown in Figure 5.

With that, we know our new SMP/E Target zone is identical to our current one and ready to accept SMP/E processing, with the exception of DDDEFs.



```

PAGE 0002 - NOW SET TO TARGET ZONE TZ9903 DATE 05/28/99 TIME 08:17:01
GIMSMP LVL 18.1.07 SMPLIST OUTPUT

LIST SUMMARY REPORT FOR TZ9903

ENTRY-TYPE ENTRY-NAME STATUS

SYSMOD NOT FOUND

```

*Figure 5: Result from SMP/E control statement*

Yes, the DDDEFs – each SMP/E Targets zone would have a list of DDDEFs to allow SMP/E to allocate the specific target libraries associated with that Target zone. For example, the SDFHLOAD library associated with the TZONE Target zone would be CICS.REL41.SDFHLOAD while the same SDFHLOAD library associated with the TZ9903 Target zone would be CICS.REL41.TZ9903.SDFHLOAD. It would not do to apply maintenance meant for TZONE target library to TZ9903 library, so, unless each SMP/E Target zone has its specific JCLPROC for APPLY, the DDDEF entries need to be changed.

There are two ways to change DDDEF entries, using the ZONEEDIT command, or unloading the DDDEF entries to a sequential file to be altered and reloaded using the UCLIN command. The ZONEEDIT command is extremely helpful when we're dealing with OS/390 RES SMP/E Target zones, because, in this case, the name of the datasets usually remains the same, and ZONEEDIT control statements can be coded in such a manner as to look only for the existence of specific DASD volumes in the DDDEF entry to effect the change. However, in the case of a CICS subsystem, we needed to change the middle-level qualifier of the dataset name. When we issued the following ZONEEDIT command:

```

ZEDIT DDDEF.
CHANGE DATASET (CICS.REL41.*,CICS.REL41.TZ9903.*) .
ENDZEDIT.

```

the following problems were encountered:

- Because the distribution datasets had the same high-level qualifiers as the target datasets, their dataset names were changed also. Because there is only one set of distribution library datasets for the subsystem SMP/E environment, this is not acceptable.
- SMP/E PTS dataset name was changed. Because there is only one PTS for the subsystem SMP/E environment, this is also not acceptable.

In lieu of using the ZONEEDIT command, the second available method of SMP/E UNLOAD and UCLIN reload must be used. First, the DDDEF entries were unloaded:

```
//UNLOAD EXEC PGM=GIMSMP,
//          REGION=4096K,
//*        TIME=120,
//          PARM='CSI=CICS.REL41.GZONE.CSI'
```

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
VIEW      myid.TEMP.PDS(SMPDDDEF) - 01.01          CHARS 'CICS.REL41.SDFH'
Command ==> c'cics.rel41.sdfh' 'cics.rel41.tz9903.sdfh'      Scroll ==> CSR
000124          DATASET      (
==CHG>          CICS.REL41.TZ9903.SDFHAPD1
000126          )
000127          SHR
000128          .
000129 REP      DDDEF      ( SDFHAPD2 )
000130          DATASET      (
==CHG>          CICS.REL41.TZ9903.SDFHAPD2
000132          )
000133          SHR
000134          .
000135 REP      DDDEF      ( SDFHAUTH )
000136          DATASET      (
==CHG>          CICS.REL41.TZ9903.SDFHAUTH
000138          )
-----
CHARS 'CICS.REL41.SDFH' - changed to 'CICS.REL41.TZ9903.SDFH' 23 times on 23 |
line(s). |
-----
==CHG>          CICS.REL41.TZ9903.SDFHCLIB
```

*Figure 6: Using ISPF/PDF EDIT*

```
//SMPLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPPUNCH DD DISP=SHR,DSN=myid.TEMP.PDS(SMPDDDEF)
//SMPCNTL DD *
SET BDY(TZ9903 ).
UNLOAD DDDEF .
```

The UNLOAD having completed, ISPF/PDF EDIT was used to change only the target library entries using the command shown in Figure 6.

The SMP/E datasets of SMPMTS, SMPSCDS, and SMPSTS were also changed. These datasets, if they don't exist, must be copied from the same SMP/E datasets from the current Target zone using the DASD file copying utility of your choice. Take care that the dataset name of SMPPTS is not changed – again there being only one SMPPTS in the Global zone. In this case, the SMPPTS entry was manually restored after the ISPF/PDF EDIT change (see Figure 7).

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
VIEW      myid.TEMP.PDS(SMPDDDEF) - 01.01          CHARS 'CICS.REL41.SMP' c
Command ==> c 'cics.rel41.smp' 'cics.rel41.tz9903.smp'      Scroll ==> CSR
000268          DATASET          (
==CHG>          CICS.REL41.TZ9903.SMPMTS
000270          )
000271          SHR
000272          .
000273 REP          DDDEF          ( SMPPTS )
000274          DATASET          (
==CHG>          CICS.REL41.TZ9903.SMPPTS
000276          )
000277          SHR
000278          .
000279 REP          DDDEF          ( SMPSCDS )
000280          DATASET          (
==CHG>          CICS.REL41.TZ9903.SMPSCDS
000282          )
-----
CHARS 'CICS.REL41.SMP' - changed to 'CICS.REL41.TZ9903.SMP' 4 times on 4 |
line(s).                                                                    |
-----
==CHG>          CICS.REL41.TZ9903.SMPSTS
```

*Figure 7: Restoring SMPPTS*

Saving the EDITed DDDEF entries, SMP/E UCLIN is used to reload the DDDEF entries into the new Target zone:

```
//RELOAD EXEC PGM=GIMSMP,
//          REGION=4096K,
//*        TIME=120,
//          PARM='CSI=CICS.REL41.GZONE.CSI'
//SMPLIST DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SMPCNTL DD *
SET BDY(TZ9903 ).
//          DD DISP=SHR,DSN=myid.TEMP.PDS(SMPDDDEF)
```

And a partial SMP/E UCLIN report would look like Figure 8.

```
PAGE 0004 - NOW SET TO TARGET ZONE TZ9903   DATE 05/28/99   TIME 08:54:03
GIMSMP LVL 18.1.07 SMPDUT   OUTPUT

REP      DDDEF          ( SDFHAPD1 )
          DATASET      (
                        CICS.REL41.TZ9903.SDFHAPD1
                        )
          SHR
          .
GIM25501I  ENTRY SDFHAPD1 WAS UPDATED BY UCLIN.

REP      DDDEF          ( SDFHAPD2 )
          DATASET      (
                        CICS.REL41.TZ9903.SDFHAPD2
                        )
          SHR
          .
GIM25501I  ENTRY SDFHAPD2 WAS UPDATED BY UCLIN.
```

*Figure 8: Partial SMP/E UCLIN report*

With that, the new Target zone is ready for you to APPLY new maintenance. The process for creation of a new Target zone described above is not difficult and, once it is established, the systems programmer will be able to control the maintenance levels for different sets of target libraries in the OS/390 environment with greater ease and accuracy.

---

*Chorng S (Jack) Hwang*  
*Principal*  
*HSA Systems (USA)*

© Xephon 2000

# Further CICS Log Manager enhancements

## INTRODUCTION

There have been several further APARs raised against the CICS Log Manager Domain that have enhanced the function of this component of CICS. This article describes the background to these alterations, and gives relevant APAR and PTF numbers so that readers can opt to install this service if they so wish. It also explains a common user misconception with the CICS Log Manager, when considering long-running tasks and the DFHSHUNT log stream.

## CICS TRANSACTION SERVER

This article makes reference to CICS Transaction Server for OS/390. The CICS Transaction Server is a member of the OS/390 family of MVS-based software servers. IBM has integrated CICS with a set of other supporting software, offering a single product in their place.

The CICS component (or element) of CICS Transaction Server Release 1 has a release number of 0510. The CICS component of CICS Transaction Server Release 2 has a release number of 0520. There is no separately orderable product such as 'CICS/ESA 5.1.0', however – it is the CICS component of CICS Transaction Server Release 1.

IBM has recently made generally available the latest release in the CICS Transaction Server series – CICS Transaction Server Release 3. The CICS component of CICS Transaction Server Release 3 has a release number of 0530. Unless otherwise indicated, the enhancements described in this article will be present in the base code level of CICS Transaction Server Release 3, and no additional service need be applied to that release to exploit the particular piece of new function.

## THE CICS LOG MANAGER DOMAIN

A detailed overview of the CICS Log Manager Domain is available in

my previous article *The CICS Log Manager*, which appeared in *CICS Update*, Issue 151, June 1998.

APAR PQ11032 / PTFS UQ14441 AND UQ14442

There are many examples of batch utilities, such as DFHJUP, which have been developed to print, copy, or delete data held in sequential datasets. Prior to CICS Transaction Server, journal data was stored in a number of such BSAM-managed sequential datasets. The CICS/ESA Release 4.1.0 Journal Control Program (DFHJCP) managed the writing of this data for both CICS system logs and user journals. Once the data had been written, programs such as DFHJUP could then be run to manipulate it.

With the advent of CICS Transaction Server, journal data is now held in log streams. The MVS System Logger subsystem provides a set of macros to allow programs an API for reading, writing, and deleting log data (amongst other functions). However, given the large number of user-generated and vendored batch utilities that already exist, there needed to be a way for existing programs to run unchanged when being directed at journal data in log streams rather than in sequential datasets.

This facility was provided by the SUBSYS parameter specified on the DD statement for the log stream in the JCL of the batch program. SUBSYS allows the specification of a number of mandatory and optional sub-parameters that define the batch job's usage of the log stream to the MVS Subsystem Interface (SSI).

By means of this approach at the JCL level, the batch programs themselves can be left unchanged. DFHJUP, for example, knows nothing of log streams itself, and still issues sequential access method calls to read log data from what it believes to be a sequential journal dataset. The SSI intercepts these requests and invokes an exit routine (as named on the SUBSYS statement); this exit routine then issues the appropriate MVS System Logger macro call to process the log stream. Upon completion of this request, the exit routine returns control via the SSI back to the batch program (DFHJUP in this example). DFHJUP continues processing as if it had just completed invoking a sequential access method function.

Log records written by CICS to the CICS system log streams, or to user journals or forward recovery logs, have a certain format and certain characteristics. CICS therefore provides an exit routine that is designed to be used against such records. The exit routine consists of two load modules. MVS exit requirements require these to reside in the MVS Linklist – as such, they are installed into the CICS SDFHLINK library. The SUBSYS parameter specifies the name of one of these exit routine’s load modules – this routine then loads and executes code in the other load module during the course of its work. The original naming convention for these exits was as follows. CICS Transaction Server Release 1 shipped with DFHLG510 (the ‘front-end’ exit routine load module, as named on the SUBSYS statement), and DFHGT510 (the ‘back-end’ exit routine load module, invoked by DFHLG510). CICS Transaction Server Release 2 shipped with DFHLG520 and DFHGT520. The naming convention was therefore release-specific, and resolved from the release number of the CICS component of the CICS Transaction Server release.

A problem with this approach was that JCL would therefore require alterations when running a batch program against log data produced by either a CICS Transaction Server Release 1 or Release 2 system. The SUBSYS sub-parameter for the exit name would need to be either DFHLG510 or DFHLG520, depending on the level of CICS that had generated the log data. In addition, it is possible that log stream data may be generated by several CICS systems at different release levels. A scenario would be a VSAM file’s forward recovery journal, where after-image log records are merged together from a number of different CICS systems that accessed the same file. If this is the case, which is the correct exit name to specify on the JCL?

The design of the exit routines ensured they were downwardly compatible, so a batch job specifying DFHLG520 on its SUBSYS parameter would be able to intercept and process log records generated from a CICS Transaction Server Release 1 system. However, this still left the problem of maintaining the JCL correctly, and having to manually change the name of the exit routine whenever a new CICS release was implemented.

To resolve this, APAR PQ11032/PTFs UQ14441 and UQ14442 defined two new exit routine load modules – DFHLGCNV and DFHGTCNV – in CICS Transaction Server Releases 1 and 2 respectively. These effectively replaced the older, release number-based names. The recommendation is to use DFHLGCNV as the exit name on the job's SUBSYS parameter. By having a generic naming convention, there is no need to modify batch job JCL when implementing multiple CICS releases, which helps simplify batch processing of CICS log data.

Since these exit routine load modules are loaded from SDFHLINK in the MVS Linklist, the highest CICS release on any given MVS image should be referenced first in the MVS Linklist concatenation. This lets batch programs always run with the highest available version of the exit routines. A given CICS release's exit routines are backwardly compatible with lower CICS releases' log records.

Users should note that CICS Transaction Server Release 3 only ships DFHLGCNV and DFHGTCNV – there is no concept of DFHLG530 nor DFHGT530 at this CICS release.

Below is an example of some (edited) JCL showing a SUBSYS entry for a log stream:

```
//SYSUT1 DD DSN=TEST.CICSA.ANDYSLOG,  
//          DCB=BLKSIZE=32760,  
//          SUBSYS=(LOGR,DFHLGCNV,'TO=(1999/360,14:13)',DELETE)
```

The LOGR parameter defines that requests against this DD statement's dataset (ie log stream) are to be carried out by the LOGR subsystem. DFHLGCNV can be seen as the name of the exit routine to be driven by the SSI when a request against the dataset is to be intercepted.

Full details about the various options that can be specified for the SUBSYS parameter are documented in the *CICS Operations and Utilities Guide*, in the chapter headed *Using batch jobs to read log streams*.

APAR PQ26852

Having defined log streams for use with CICS Transaction Server,



users have to manipulate the log data written to their journals (mapped onto the underlying MVS log streams).

As stated above, IBM provides the DFHJUP utility to perform various functions against journals in CICS Transaction Server. Using DFHJUP, users can print from their log streams, copy their log data to another dataset or log stream, and delete data from log streams. DFHJUP provides a powerful series of filtering choices that can be specified on the OPTION and CONTROL statements of its SYSIN data. These enable the user to be very selective in what log data is and is not manipulated by DFHJUP.

A typical requirement for batch processing of a journal is for a utility such as DFHJUP to copy some or all of the log data on a journal to another medium (for example an archive dataset), and then delete the copied portion of the log data from the journal. The delete step could be carried out by using a program such as IEFBR14, since it is a jobstep's termination that drives the DELETE option on the JCL's SUBSYS parameter.

Such a copy and delete caretaking operation may need to be run on a daily basis. Accordingly, there are several factors to consider when writing the batch job to process this piece of work. Firstly, it is important that only log data that has been successfully copied to the other media is subsequently deleted. In other words, if the copy step were to fail for some reason, or if new data were added to the log while the copy step was still running, any uncopied data on the log at the end of the copy stage should not then be deleted by the delete phase. Secondly, it is important to try to minimize the programming intervention required before submitting the job, by making its job control statements and optional parameters as generic as possible.

When designing the JCL to perform such an operation, the recommendation is to use two job steps – the first to perform the copy operation and the second to perform the delete processing. The delete step is made dependent upon the copy step completing successfully. This is standard conditional JCL programming.

In addition to this, the SUBSYS parameter is recommended to specify a 'TO=' value that does not specify a Julian date. In the example JCL

shown below, a TO= value of 10am (10:00) is specified:

```
//COPYJNL EXEC PGM=DFHJUP
//STEPLIB DD DSN=CTS120.CICS520.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1 DD DSN=TEST.CICSA.ANDYSLOG,
//          DCB=BLKSIZE=32760,
//          SUBSYS=(LOGR,DFHLGCV,'TO=(,10:00)',COMPAT41)
//SYSUT4 DD DSN=TEST.COPY1,DISP=(NEW,CATLG),
//          UNIT=SYSDA,VOL=SER=USRPAC,
//          SPACE=(TRK,(3,1))
//SYSIN DD *
OPTION COPY
END
//CHKCOPY IF (COPYJNL.RC = 0) THEN
//IEFBR14 EXEC PGM=IEFBR14
//LOGSTRM DD DSN=TEST.CICSA.ANDYSLOG,
//SUBSYS=(LOGR,DFHLGCV,'TO=(,10:00)',DELETE)
//CHKCOPY ENDIF
/*
```

Note that the same TO= value is specified on both jobstep's SUBSYS parameters. This means that the limiting position in the log data being copied is the same as that used when subsequently deleting the log data. This is important because MVS log streams allow multiple jobs to manipulate them concurrently. In other words, a CICS system could be writing new log records to a journal mapped to a log stream while batch utilities such as DFHJUP were reading log data from the same log stream.

Note that if the TO= were omitted on the SUBSYS parameter, and so allowed to default to the current end of the log stream, it would be possible for uncopied log data to be deleted by the second job step. This is because the calculation of the limiting point for the delete operation would be based on the state of the log stream at the time the second job step connected to it. If another subsystem, such as CICS, were to have written data to the log stream while the copy was taking place, this data would then be included in the range of log data to be deleted.

Not specifying a Julian date value on the TO= option of the SUBSYS parameter means that the MVS System Logger will default to the date of the day the job is being run (ie today's date). This means the job does not require amendment prior to each submission, so simplifying the

use of the job and reducing the need for manual alteration of the JCL before each submission.

The recommendation is to submit the copy and delete job reasonably soon after the time specified on the TO= option. If a customer site chose to archive and delete all log data on a particular log stream from the previous day, then submitting DFHJUP with these options shortly after 10am each day would let it copy and delete all those log records that were written to the journal between 10am of the previous day and 10am of the current day. If company policy was to archive and delete log data weekly, the same JCL could be used without amendment. The range of log record dates selected is always from the oldest existing record on the log stream up until a record matches the TO= date/time specification.

Users should note that a choice of a TO= time value close to midnight is not recommended. This is because there is the potential for the copy step to take long enough to span the midnight boundary. In that case, the delete step that followed would run after midnight, and so default to a Julian date for the TO= option that was one higher than the date used for the copy step. This would give the potential for uncopied data to be deleted.

PQ26852 updated CICS documentation (the *CICS Operations and Utilities Guide*) to clarify the advice for copying and deleting log stream data in an atomic manner, as described above.

APAR PQ28115 / PTFs UQ34294 AND UQ34295

OS/390 Release 3 introduced new optional parameters that may be specified when defining a log stream to the MVS System Logger – RETPD and AUTODELETE. RETPD allows the specification of a retention period (in days) for log data before it is physically deletable from a log stream. AUTODELETE (YES or NO) specifies whether log data may be automatically physically deleted by the MVS System Logger, or whether an explicit IXGDELET macro call is required to action the delete. The default values are AUTODELETE(NO) and RETPD(0).

The two parameters are intended to be used in tandem to help

automation of certain log stream management issues. The four combinations (and their resulting actions) are described in more detail below:

- AUTODELETE(NO) and RETPD(0) – the default.

When an IXGDELET macro call is issued, the MVS system Logger logically deletes the log data up to the point specified on the IXGDELET. The data will be physically deleted at the time of the next MVS System Logger offload processing. This combination should be used for the CICS system log streams DFHLOG and DFHSHUNT.

- AUTODELETE(NO) and RETPD(*n*) – where *n* is a number of days greater than zero.

This results in log data remaining on the log stream for *n* days, whether an IXGDELET macro call has been issued against it or not. Any such data that has been deleted by an IXGDELET call will be marked inactive. DFHJUP will record whether a given block of log data was retrieved from the active or inactive portions of the log stream when printing out log data.

- AUTODELETE(YES) and RETPD(0).

Periodically, the MVS System Logger will check its inventory data and physically delete any offload datasets with log data older than the retention period. In this case the RETPD value is zero, indicating that the data can be physically deleted as soon as the MVS System Logger chooses to do so.

- AUTODELETE(YES) and RETPD(*n*).

In this case, the physical deletion of the log data can only occur once *n* days have elapsed since it was written.

The use of the two new parameters can be summarized as follows. AUTODELETE(NO) means that physical deletion of log data can occur only after an explicit IXGDELET has been issued against it, and after any retention period (as specified by RETPD) has expired. AUTODELETE(YES) means that physical deletion can occur either

when an IXGDELET has been issued or when any retention period has elapsed.

Physical deletion may not actually occur at the exact moment after a retention period has expired; it will take place at a point after this when the MVS System Logger next performs housekeeping work for the log stream's offload datasets.

A customer had developed a CICS Global User Exit program to run in the XLGSTRM exit point within the CICS Log Manager Domain. This exit may be used to influence the processing of a subsequent call from CICS to define a log stream to the MVS System Logger, if a request has been made against a log stream that does not currently exist. CICS will define the log stream based upon a number of variable parameters that can be modified via the use of the XLGSTRM exit.

The customer wanted to specify RETPD on the log stream definition. However, CICS Transaction Server did not recognize the correct MVS System Logger parameter list for this optional value. To rectify this, APAR PQ28115 shipped PTFs UQ34294 and UQ34295 against CICS Transaction Server Releases 2 and 3 respectively. This updated the parameter list version used by CICS to allow RETPD to be recognized on the subsequent log stream definition.

#### APAR PQ27786 / PTF UQ35145

It is possible that a job can issue a request to read or delete a particular block of log data, which may then fail because that block is not present on the log stream. There are several reasons why this situation may occur. The most likely is that an earlier request to delete the log stream data has been processed already, and so a request to then retrieve the log data would be expected to fail with such a 'not found' condition. Alternatively, an incorrect log block identifier may have been passed on the request to the MVS System Logger, specifying a non-existent block of log data. Finally, there is the rare possibility of an internal logic error in CICS or another subsystem resulting in a requested log block not being located properly.

If such a failure occurs on an MVS System Logger request, the

response code is set to an error (00000008) and the reason code to IXGRSNCODENOBLOCK (00000804).

Depending on the type of application that is using the MVS System Logger to access log stream data, such '804' conditions may or may not be expected. For example, the log stream may be being used as a sequential scratch pad, with records being written and deleted by the application, depending on the logic of the program. The program may well issue read requests against the log stream and expect that in some cases an 804 will be returned. It would then handle this in an analogous manner to a CICS command-level application program that issued an EXEC CICS READ against a VSAM KSDS file and received NOTFND as the response from CICS, indicating that no record was present on the dataset with a key matching the Ridfld value passed on the request.

In the case of the CICS system log, however, it is never expected that an 804 condition be returned on a request made by CICS against the DFHLOG or DFHSHUNT log streams. CICS is designed to manage the state of the system log itself, issuing delete requests against the underlying log streams as and when it decides that recovery information at the oldest end of the system log streams is no longer required. Such action is referred to as 'trimming the log tail'. As CICS manages the deletion of its own system-critical recovery information on the system log, non-CICS jobs such as DFHJUP or IEFBR14 should not be run against the CICS system log streams with the DELETE option specified on their SUBSYS parameter. Neither should the CICS system log streams be defined with the AUTODELETE(YES) parameter specified. This would allow the MVS System Logger to automatically delete data on the log streams when a specified retention period had been exceeded.

If (for whatever reason) an 804 condition were to occur when CICS was referencing its system log streams, certain diagnostic actions would be taken. CICS would issue a DFHLG0772 message and initiate a system dump. Depending on the circumstances of the failure, various other messages would be issued and action taken by CICS to quiesce the system and terminate itself in as controlled a manner as possible. CICS has to do this since data integrity has been compromised,

because it can no longer rely on its system log to provide consistent information on the state of its recoverable resources.

In order to diagnose the cause of the 804 condition, IBM requires the diagnostics taken when the failure occurs. From a CICS perspective, the dump from the DFHLG0772 message gives useful information on the state of the CICS resources at the time of the failure, and DFHJUP prints of the DFHLOG and DFHSHUNT log streams provide a snapshot of the state of the log data itself when the failure occurred. However, MVS System Logger diagnosis requires information from the System Logger address space. To provide this, MVS System Logger APAR OW27057 (PTFs UW57210, UW57211 and UW57212) was raised to provide a dump containing the necessary diagnostic information for the MVS Logger address space and Coupling Facility structure information (if applicable) if an 804 condition occurred.

Subsystems that exploit the MVS System Logger (such as CICS) have to specify whether or not they require this additional diagnostic information when issuing MVS System Logger requests. In other words, if they do not expect to receive 804 conditions, then the information would be beneficial for problem diagnosis purposes. If they are designed to expect such conditions returned from the MVS System Logger, then such diagnostics would not be relevant to them. As stated above, CICS never expects such a response and so APAR PQ27786 / PTF UQ35145 shipped this enablement function at CICS Transaction Server Release 3.

#### LONG-RUNNING TASKS AND THE DFHSHUNT LOG STREAM

There is some confusion in the user community about the nature and role of the DFHSHUNT CICS system log stream. This confusion is often related to the concept of ‘shunting’ a CICS Unit Of Work (UOW).

In CICS Transaction Server, the Log Manager Domain stores recoverable system activity on the CICS system log. This is a single entity in concept, but is actually composed of two log streams – DFHLOG and DFHSHUNT. DFHLOG is used for all normal system logging activity. It is intended to store log data for those tasks with

relatively short-lived UOWs. As and when these tasks commit their changes with an EXEC CICS SYNCPOINT (or implicit syncpoint at task detach time), CICS knows that their log data is no longer required for task or system backout purposes, and so is able then to trim the log stream at a later point to delete this portion of the log data. Such trimming is automatically and transparently handled by CICS, as part of activity keypoint processing.

DFHSHUNT was implemented as a repository for log data for those UOWs that (for various reasons) failed at a critical point in their processing and required 'shunting' by the CICS Recovery Manager Domain. This concept was introduced with CICS Transaction Server Release 1 – it provided a means to place the resolution of the UOW 'on ice' until the problem that led to its failure could be resolved. CICS could then unshunt the UOW and allow it to complete. DFHSHUNT was provided because any such shunted UOWs would most likely be retained by CICS for a considerable period of time, when compared to the normal throughput of non-shunted work. If their log data were left on the DFHLOG log stream, it would therefore prevent trimming of the log portion that contained this data. This would lead to inefficient use of the log stream's structure storage on the Coupling Facility.

When CICS Transaction Server was developed, it was noticed that a performance and structure usage optimization could be achieved by also moving the log data of long-running tasks from DFHLOG to DFHSHUNT. CICS deems a task to be long-running if its UOW does not generate any log data written to the DFHLOG system log stream within the duration of a complete keypoint interval (ie two successive keypoint operations occur and the UOW has not logged in all that time). If it detects a UOW in such a condition, CICS moves the log data for that task from DFHLOG to DFHSHUNT. This allows the portion of DFHLOG's log stream that contained this data to be freed when DFHLOG is next trimmed by CICS.

The confusion occurs if users examine the state of UOWs relating to such long-running tasks. Using the CEMT INQ UOW command, information about UOWs on the system can be returned. Since the log data for these tasks has been moved to DFHSHUNT, it is sometimes



supposed that the UOWs themselves ought to be in a shunted state. However, this is not the case. Shunting is a specific action taken by CICS in response to a certain type of failure during task processing. Movement of log data from DFHLOG to DFHSHUNT is a CICS Log Manager optimization that does indeed occur when shunting UOWs, but is much more likely to be seen because CICS is handling long-running tasks' log data. In my experience, more UOWs fall into the long-running category than fail at a critical juncture and so are shunted/unshunted by CICS.

That is, UOWs typically appear with a waitstate of ACTIVE when viewed via CEMT I UOW, even if their log data has been moved to DFHSHUNT owing to the optimization described above.

## SUMMARY AND CONCLUSIONS

I hope that this article has helped explain the background to these various enhancements to the CICS Logger Domain and associated utilities, why they were provided, and the points to consider when using them and interpreting the output they produce. Readers who wish to discuss the material in this article further may contact me via e-mail, at [andy\\_wright@uk.ibm.com](mailto:andy_wright@uk.ibm.com). CICS is a registered trademark of International Business Machines corporation.

---

*Andy Wright*  
*CICS Change Team Programmer*  
*IBM (UK)*

© IBM 2000

---

Why not share your expertise and earn money at the same time? *CICS Update* is looking for technical articles and hints and tips about improving CICS performance, as well as CICS utilities that experienced CICS users have written to make their life, or the lives of their users, easier.

Article can be sent or e-mailed to Trevor Eddolls at any of the addresses on page 2. Why not call now for a free copy of our *Notes for Contributors*.

## Controlling CICS resources from a batch program using the CICSplex SM API – part 2

*This month we conclude the code for a batch COBOL II program that uses the CICSplex SM API to control resources.*

```
120-EDIT-CONTEXT-SCOPE.
  MOVE 'N' TO INVALID-SCOPE-SW.
  EVALUATE CPSM-CONTEXT-SA
  WHEN 'CICSPLXP'      *** our production regions ***
    IF CPSM-SCOPE-SA EQUAL 'CICSPLXP' OR 'CICSP1' OR
      'CICSP4' OR 'ACICPA1' OR 'ACICPA4' OR
      'ACICPZ1A' OR 'ACICPZ4A' OR 'ACICICPA' OR
      'ACICICPC' OR 'ACICICPD' OR 'ACICICPE' OR
      'ACICICPF' OR 'ACICICPG' OR 'ACICPODM' OR
      'CICSPTOR' OR 'CICSPAOR'
      CONTINUE
    ELSE
      MOVE 'Y' TO INVALID-SCOPE-SW
    END-IF

  WHEN 'CICSPLXM'      *** our demo (end user test) regions ***
    IF CPSM-SCOPE-SA EQUAL 'CICSPLXM' OR 'CICSM1' OR
      'CICSM4' OR 'CICSTA1' OR 'CICSTA4' OR
      'CICSTB' OR 'CICSTD' OR 'CICSTF' OR
      'CICSDBTA' OR 'CICSMODM' OR 'CICSMTOR' OR
      'CICSMAOR'
      CONTINUE
    ELSE
      MOVE 'Y' TO INVALID-SCOPE-SW
    END-IF

  WHEN 'CICSPLXD'      *** our development regions ***
    IF CPSM-SCOPE-SA EQUAL 'CICSPLXD' OR 'CICSD1' OR
      'CICSD4' OR 'CICSDVLP' OR 'CICSDVLX' OR
      'CICSDVL1'
      CONTINUE
    ELSE
      MOVE 'Y' TO INVALID-SCOPE-SW
    END-IF

  WHEN 'CICSPLXT'      *** our Tech Services' test regions ***
    IF CPSM-SCOPE-SA EQUAL 'CICSPLXT' OR 'CICSTT' OR
      'CICSTD' OR 'CICSTGT' OR 'CICSTIT' OR
      'CICSTCT' OR 'CICSTHT' OR 'CICSTGD' OR
      'CICSTID' OR 'CICSTCD' OR 'CICSTHD' OR
      'CICSTC' OR 'CICSTI' OR 'CICSTTOR' OR
```

```

        'CICSTAOR'
        CONTINUE
    ELSE
        MOVE 'Y' TO INVALID-SCOPE-SW
    END-IF

WHEN OTHER
    MOVE INVALID-CONTEXT-MSG TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    MOVE CRS-COMMAND-RECORD TO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1 TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1
    MOVE 1501 TO RETURN-CODE-SA

END-EVALUATE.

IF INVALID-SCOPE-SW = 'Y'
    MOVE INVALID-SCOPE-MSG TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    MOVE CRS-COMMAND-RECORD TO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1 TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1
    MOVE 1502 TO RETURN-CODE-SA
END-IF.

130-CONNECT-TO-CPSM.
MOVE ZERO TO CPSM-RESP-SA
        CPSM-REAS-SA.
EXEC CPSM CONNECT
        CONTEXT(CPSM-CONTEXT-SA)
        SCOPE(CPSM-SCOPE-SA)
        VERSION('0130')
        THREAD(CPSM-THREAD-SA)
        RESPONSE(CPSM-RESP-SA)
        REASON(CPSM-REAS-SA)
END-EXEC.

IF CPSM-RESP-SA = EYUVALUE(OK)
    MOVE 'Y' TO CPSM-CONNECT-SW
    MOVE CONN-SUCCESS-MSG-SA TO ML1-MESSAGE
    MOVE CPSM-CONTEXT-SA TO ML1-CONTEXT
    MOVE CPSM-SCOPE-SA TO ML1-SCOPE
    MOVE 'N/A' TO ML1-ACTION
        ML1-RESOURCE
    MOVE ML1-MESSAGE-LINE1 TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 3
ELSE
    MOVE CONN-FAILED-MSG-SA TO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1 TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2

```

```

PERFORM 750-CONVERT-ERR-TO-TEXT
MOVE CPSM-CONTEXT-SA      TO EL2-CONTEXT
MOVE CPSM-SCOPE-SA        TO EL2-SCOPE
MOVE 'CONNECT'            TO EL2-ACTION
MOVE SPACES                TO EL2-RESOURCE
MOVE EL2-ERROR-LINE2      TO PRINT-LINE-OUT
WRITE PRINT-LINE-OUT AFTER 1
* DON'T ABORT IF THE CMAS IS UNAVAILABLE - ASSUME REGIONS
* ARE ALSO UNAVAILABLE
IF CPSM-RESP-SA NOT EQUAL EYUVALUE(NOTAVAILABLE)
    MOVE 3000              TO RETURN-CODE-SA
END-IF
END-IF.

140-DISCONNECT-FROM-CPSM.
MOVE ZERO TO CPSM-RESP-SA
      CPSM-REAS-SA.

EXEC CPSM DISCONNECT
      THREAD(CPSM-THREAD-SA)
      RESPONSE(CPSM-RESP-SA)
      REASON(CPSM-REAS-SA)
END-EXEC.

IF CPSM-RESP-SA = EYUVALUE(OK)
    MOVE DISCONN-SUCCESS-MSG-SA TO ML1-MESSAGE
    MOVE CPSM-CONTEXT-SA        TO ML1-CONTEXT
    MOVE CPSM-SCOPE-SA          TO ML1-SCOPE
    MOVE 'N/A'                  TO ML1-ACTION
                                ML1-RESOURCE
    MOVE ML1-MESSAGE-LINE1      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
ELSE
    MOVE DISCONN-FAILED-MSG-SA TO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    PERFORM 750-CONVERT-ERR-TO-TEXT
    MOVE CPSM-CONTEXT-SA      TO EL2-CONTEXT
    MOVE CPSM-SCOPE-SA        TO EL2-SCOPE
    MOVE 'DISCONNECT'         TO EL2-ACTION
    MOVE SPACES                TO EL2-RESOURCE
    MOVE EL2-ERROR-LINE2      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1
* DON'T ABORT IF THE REGION OR CICSPLEX IS UNAVAILABLE.
IF CPSM-RESP-SA NOT EQUAL EYUVALUE(NOTAVAILABLE) AND
    RETURN-CODE-SA < 5
    MOVE 3001                  TO RETURN-CODE-SA
END-IF
END-IF.

```

200-PROCESS-ACTIONS.  
PERFORM 210-EDIT-ACTION.

IF RETURN-CODE-SA < 5  
\* READ THE FIRST RESOURCE RECORD FOR THE ACTION  
PERFORM 800-READ-COMMAND-FILE

IF COMMAND-EOF-SW = 'Y'  
MOVE NO-RESOURCE-MSG TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 2  
MOVE SPACES TO ML1-MESSAGE  
MOVE '\* END OF FILE \*' TO ML1-RESOURCE  
MOVE CPSM-CONTEXT-SA TO ML1-CONTEXT  
MOVE CPSM-SCOPE-SA TO ML1-SCOPE  
MOVE CRS-ACTION-SA TO ML1-ACTION  
MOVE ML1-MESSAGE-LINE1 TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 1  
MOVE 1504 TO RETURN-CODE-SA  
END-IF

PERFORM 300-PROCESS-RESOURCES  
UNTIL NEW-CONTEXT-FOUND OR  
NEW-ACTION-FOUND OR  
RETURN-CODE-SA > 4 OR  
COMMAND-EOF-SW EQUAL 'Y'  
END-IF.

210-EDIT-ACTION.

IF CRS-ACTION-LIT NOT EQUAL 'ACTION='  
MOVE NO-ACTION-MSG TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 2  
MOVE CRS-COMMAND-RECORD TO EL1-ERROR-MSG  
MOVE EL1-ERROR-LINE1 TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 1  
MOVE 1503 TO RETURN-CODE-SA  
ELSE  
\* RESET THE NEW-ACTION SWITCH FOR CONTROL BREAK PROCESSING  
MOVE 'N' TO NEW-ACTION-FOUND-SW  
IF CRS-ACTION EQUAL 'CLOSE FILE' OR  
'ENABLE FILE' OR 'OPEN FILE' OR  
'DISABLE TRAN' OR 'ENABLE TRAN' OR  
'DISABLE PROG' OR 'DISCARD PROG' OR  
'ENABLE PROG' OR 'NEWCOPY PROG'  
MOVE CRS-ACTION TO CRS-ACTION-SA  
ELSE  
MOVE INVALID-ACTION-MSG TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 2  
MOVE CRS-COMMAND-RECORD TO EL1-ERROR-MSG  
MOVE EL1-ERROR-LINE1 TO PRINT-LINE-OUT  
WRITE PRINT-LINE-OUT AFTER 1

```

        MOVE 1503 TO RETURN-CODE-SA
    END-IF
END-IF.

```

300-PROCESS-RESOURCES.

```

*   ONLY PROCESS RESOURCES FOR ACTIVE REGIONS. IF A REGION OR
*   CMAS WAS DOWN AT THE TIME THE CONNECT WAS PERFORMED, THE
*   CONNECT WOULD HAVE RECEIVED A "NOTAVAILABLE" CONDITION. THE
*   CPSM-CONNECT STMT BELOW CAUSES THE PROGRAM TO ESSENTIALLY
*   BYPASS ALL RECORDS FOR THE GIVEN CONTEXT & SCOPE, SO IT CAN
*   PROCESS RESOURCES FOR OTHER REGIONS (IF SUCH RECORDS EXIST
*   ON ISEQL01).

```

```

PERFORM 305-EDIT-RESOURCE.
IF RETURN-CODE-SA < 5
    ADD 1 TO RES-OR-GRPS-PROC-AA
    IF CPSM-CONNECT-SUCCEEDED
        PERFORM 310-CHECK-FOR-RES-GRP
    END-IF
END-IF.

```

```

IF RETURN-CODE-SA < 5
    PERFORM 800-READ-COMMAND-FILE
    IF CRS-CONTEXT-LIT = 'CONTEXT='
        SET NEW-CONTEXT-FOUND TO TRUE
    END-IF
    IF CRS-ACTION-LIT = 'ACTION='
        SET NEW-ACTION-FOUND TO TRUE
    END-IF
END-IF.

```

305-EDIT-RESOURCE.

```

*   THE ONLY EDITING TO DO IS TO MAKE SURE A RESOURCE HAS
*   BEEN SPECIFIED FOR A GIVEN ACTION AND THAT IT STARTS IN
*   COLUMN 1.
IF CRS-CONTEXT-LIT = 'CONTEXT=' OR
CRS-CONTEXT-LIT(1:1) = SPACE OR
CRS-ACTION-LIT = 'ACTION='
    MOVE NO-RESOURCE-MSG          TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    MOVE SPACES                   TO ML1-MESSAGE
    MOVE 'MISSING/NOT COL 1'      TO ML1-RESOURCE
    MOVE CPSM-CONTEXT-SA          TO ML1-CONTEXT
    MOVE CPSM-SCOPE-SA           TO ML1-SCOPE
    MOVE CRS-ACTION-SA            TO ML1-ACTION
    MOVE ML1-MESSAGE-LINE1       TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1
    MOVE 1504                     TO RETURN-CODE-SA
END-IF.

```

```

310-CHECK-FOR-RES-GRP.
  IF CRS-GROUP-IND = '@'
    PERFORM 810-READ-GROUP-FILE
      UNTIL (GRS-GROUP-IND = '@' AND
             GRS-GROUP-NAME = CRS-GROUP-NAME) OR
             GROUP-EOF-SW = 'Y'
    IF GROUP-EOF-SW = 'N'
      * GET THE FIRST RESOURCE FOR THE GROUP
        PERFORM 810-READ-GROUP-FILE

      * PROCESS UNTIL A NEW GROUP IS READ OR EOF REACHED
        MOVE ZERO TO RESOURCES-IN-GROUP-AA
        PERFORM 320-PROCESS-GROUP-RECORDS
          UNTIL GRS-GROUP-IND = '@' OR
                GROUP-EOF-SW = 'Y' OR
                RETURN-CODE-SA > 4

        IF RESOURCES-IN-GROUP-AA = ZERO
          MOVE NO-RES-FOR-GRP-MSG-SA TO EL1-ERROR-MSG
          MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
          WRITE PRINT-LINE-OUT AFTER 2
          MOVE SPACES                TO EL1-ERROR-MSG
          STRING '  CONTEXT: ' CPSM-CONTEXT-SA
                '  SCOPE: '    CPSM-SCOPE-SA
                '  ACTION: '   CRS-ACTION-SA
                '  GROUP: '   '@' CRS-GROUP-NAME
                DELIMITED BY SIZE INTO EL1-ERROR-MSG
          MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
          WRITE PRINT-LINE-OUT AFTER 1
          MOVE 2200                 TO RETURN-CODE-SA
        END-IF

      ELSE
        MOVE GROUP-NOT-FOUND-MSG-SA TO EL1-ERROR-MSG
        MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 2

        MOVE USE-GENERIC-SCOPE-MSG-SA TO EL1-ERROR-MSG
        MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 1

        MOVE SPACES                TO EL1-ERROR-MSG
        STRING '  CONTEXT: ' CPSM-CONTEXT-SA
              '  SCOPE: '    CPSM-SCOPE-SA
              '  ACTION: '   CRS-ACTION-SA
              '  GROUP: '   '@' CRS-GROUP-NAME
              DELIMITED BY SIZE INTO EL1-ERROR-MSG
        MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 1
        MOVE 2200                 TO RETURN-CODE-SA
      END-IF
    END-IF
  END-IF

```

```

        END-IF

ELSE
    MOVE CRS-RSRC-NAME TO TEMP-RSRC-SA
    PERFORM 330-PROCESS-RESOURCE
END-IF.

320-PROCESS-GROUP-RECORDS.
ADD 1          TO RESOURCES-IN-GROUP-AA.
MOVE GRS-GROUP-RSRC TO TEMP-RSRC-SA.
PERFORM 330-PROCESS-RESOURCE.
IF RETURN-CODE-SA < 5
    PERFORM 810-READ-GROUP-FILE
END-IF.

330-PROCESS-RESOURCE.
MOVE SPACES TO CPSM-CRITERIA-SA.
MOVE 'N'    TO NODATA-SW.
EVALUATE CRS-ACTION-SA
WHEN 'CLOSE FILE'
    STRING 'FILE=' TEMP-RSRC-SA '.'
        DELIMITED BY SIZE
        INTO CPSM-CRITERIA-SA
    MOVE 14          TO CPSM-CRITLEN-SA
    MOVE 'LOCFILE'   TO CPSM-OBJNAME-SA
    MOVE 'CLOSE'     TO CPSM-ACTION-SA
    MOVE 'BUSY(WAIT)..' TO CPSM-PARM-SA
    MOVE 11          TO CPSM-PARMLEN-SA
    MOVE 'Y'         TO PARM-REQUIRED-SW
    PERFORM 340-EXEC-CPSM

WHEN 'ENABLE FILE'
    STRING 'FILE=' TEMP-RSRC-SA '.'
        DELIMITED BY SIZE
        INTO CPSM-CRITERIA-SA
    MOVE 14          TO CPSM-CRITLEN-SA
    MOVE 'LOCFILE'   TO CPSM-OBJNAME-SA
    MOVE 'ENABLE'    TO CPSM-ACTION-SA
    MOVE 'N'         TO PARM-REQUIRED-SW
    PERFORM 340-EXEC-CPSM

WHEN 'OPEN FILE'
    STRING 'FILE=' TEMP-RSRC-SA '.'
        DELIMITED BY SIZE
        INTO CPSM-CRITERIA-SA
    MOVE 14          TO CPSM-CRITLEN-SA
    MOVE 'LOCFILE'   TO CPSM-OBJNAME-SA
    MOVE 'OPEN'      TO CPSM-ACTION-SA
    MOVE 'N'         TO PARM-REQUIRED-SW
    PERFORM 340-EXEC-CPSM

```



```

WHEN 'DISABLE TRAN'
  STRING 'TRANID=' TEMP-RSRC-SA '.'
    DELIMITED BY SIZE
                                INTO CPSM-CRITERIA-SA
MOVE 16                          TO CPSM-CRITLEN-SA
MOVE 'LOCTRAN'                   TO CPSM-OBJNAME-SA
MOVE 'DISABLE'                   TO CPSM-ACTION-SA
MOVE 'N'                         TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

MOVE 'REMTRAN'                   TO CPSM-OBJNAME-SA
PERFORM 340-EXEC-CPSM

```

```

WHEN 'ENABLE TRAN'
  STRING 'TRANID=' TEMP-RSRC-SA '.'
    DELIMITED BY SIZE
                                INTO CPSM-CRITERIA-SA
MOVE 16                          TO CPSM-CRITLEN-SA
MOVE 'LOCTRAN'                   TO CPSM-OBJNAME-SA
MOVE 'ENABLE'                    TO CPSM-ACTION-SA
MOVE 'N'                         TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

MOVE 'REMTRAN'                   TO CPSM-OBJNAME-SA
PERFORM 340-EXEC-CPSM

```

```

WHEN 'DISABLE PROG'
  STRING 'PROGRAM=' TEMP-RSRC-SA '.'
    DELIMITED BY SIZE
                                INTO CPSM-CRITERIA-SA
MOVE 17                          TO CPSM-CRITLEN-SA
MOVE 'PROGRAM'                   TO CPSM-OBJNAME-SA
MOVE 'DISABLE'                   TO CPSM-ACTION-SA
MOVE 'N'                         TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

WHEN 'DISCARD PROG'
  STRING 'PROGRAM=' TEMP-RSRC-SA '.'
    DELIMITED BY SIZE
                                INTO CPSM-CRITERIA-SA
MOVE 17                          TO CPSM-CRITLEN-SA
MOVE 'PROGRAM'                   TO CPSM-OBJNAME-SA
MOVE 'DISCARD'                   TO CPSM-ACTION-SA
MOVE 'N'                         TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

WHEN 'ENABLE PROG'
  STRING 'PROGRAM=' TEMP-RSRC-SA '.'
    DELIMITED BY SIZE

```

```

                                INTO CPSM-CRITERIA-SA
MOVE 17                        TO CPSM-CRITLEN-SA
MOVE 'PROGRAM'                 TO CPSM-OBJNAME-SA
MOVE 'ENABLE'                  TO CPSM-ACTION-SA
MOVE 'N'                       TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

WHEN 'NEWCOPY PROG'
  STRING 'PROGRAM=' TEMP-RSRC-SA '.'
  DELIMITED BY SIZE
                                INTO CPSM-CRITERIA-SA
MOVE 17                        TO CPSM-CRITLEN-SA
MOVE 'PROGRAM'                 TO CPSM-OBJNAME-SA
MOVE 'NEWCOPY'                 TO CPSM-ACTION-SA
MOVE 'N'                       TO PARM-REQUIRED-SW
PERFORM 340-EXEC-CPSM

```

```

WHEN OTHER
  MOVE INVALID-ACTION-MSG      TO PRINT-LINE-OUT
  WRITE PRINT-LINE-OUT AFTER 2
  MOVE SPACES                   TO EL2-RESPONSE
                                EL2-REASON
  MOVE CPSM-CONTEXT-SA         TO EL2-CONTEXT
  MOVE CPSM-SCOPE-SA           TO EL2-SCOPE
  MOVE CRS-ACTION-SA           TO EL2-ACTION
  MOVE CPSM-CRITERIA-SA        TO EL2-RESOURCE
  MOVE EL2-ERROR-LINE2         TO PRINT-LINE-OUT
  WRITE PRINT-LINE-OUT AFTER 1
  MOVE 1503                     TO RETURN-CODE-SA

```

END-EVALUATE.

340-EXEC-CPSM.

```

*   DON'T PERFORM THE ACTION DURING A 'TEST' RUN, WHERE THE
*   FORMAT OF ISEQL01 IS SIMPLY BEING CHECKED.
  IF TEST-SW = 'N'
    PERFORM 350-CONTINUE-WITH-EXEC
  END-IF.

```

350-CONTINUE-WITH-EXEC.

```

MOVE ZERO TO CPSM-RESP-SA
          CPSM-REAS-SA
          CPSM-RESULT-SA.

```

```

IF PARM-REQUIRED-SW = 'Y'
  EXEC CPSM PERFORM OBJECT(CPSM-OBJNAME-SA)
                                ACTION      (CPSM-ACTION-SA)
                                CRITERIA    (CPSM-CRITERIA-SA)
                                LENGTH      (CPSM-CRITLEN-SA)
                                PARM        (CPSM-PARM-SA)

```

```

                PARMLEN      (CPSM-PARMLEN-SA)
                RESULT      (CPSM-RESULT-SA)
                THREAD      (CPSM-THREAD-SA)
                RESPONSE     (CPSM-RESP-SA)
                REASON       (CPSM-REAS-SA)
        END-EXEC
ELSE
        EXEC CPSM PERFORM OBJECT(CPSM-OBJNAME-SA)
                ACTION      (CPSM-ACTION-SA)
                CRITERIA    (CPSM-CRITERIA-SA)
                LENGTH      (CPSM-CRITLEN-SA)
                RESULT      (CPSM-RESULT-SA)
                THREAD      (CPSM-THREAD-SA)
                RESPONSE     (CPSM-RESP-SA)
                REASON       (CPSM-REAS-SA)
        END-EXEC
END-IF.

EVALUATE CPSM-RESP-SA

WHEN EYUVALUE(OK)
        MOVE SPACES                TO ML1-MESSAGE
        STRING PROCESS-SUCCESS-MSG-SA CPSM-OBJNAME-SA
                ')....' DELIMITED BY SIZE INTO ML1-MESSAGE
        MOVE CPSM-CONTEXT-SA        TO ML1-CONTEXT
        MOVE CPSM-SCOPE-SA          TO ML1-SCOPE
        MOVE CPSM-ACTION-SA         TO ML1-ACTION
        MOVE CPSM-CRITERIA-SA       TO ML1-RESOURCE
        MOVE ML1-MESSAGE-LINE1      TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 2

WHEN EYUVALUE(NODATA)
        MOVE SPACES                TO EL1-ERROR-MSG
        STRING RSRC-NOT-FOUND-MSG-SA CPSM-OBJNAME-SA
                ')....' DELIMITED BY SIZE INTO EL1-ERROR-MSG
        MOVE EL1-ERROR-LINE1        TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 2
        PERFORM 750-CONVERT-ERR-TO-TEXT
        MOVE CPSM-CONTEXT-SA        TO EL2-CONTEXT
        MOVE CPSM-SCOPE-SA          TO EL2-SCOPE
        MOVE CPSM-ACTION-SA         TO EL2-ACTION
        MOVE CPSM-CRITERIA-SA       TO EL2-RESOURCE
        MOVE EL2-ERROR-LINE2        TO PRINT-LINE-OUT
        WRITE PRINT-LINE-OUT AFTER 1
*       SET THE RETURN CODE TO 4 FOR MISSING RESOURCES. FOR
*       TRANSACTIONS, THE RESOURCE MUST NOT BE DEFINED AS LOCAL
*       OR REMOTE.
        IF CPSM-OBJNAME-SA = 'LOCTRAN'
                MOVE 'Y' TO NODATA-SW
        ELSE

```

```

        IF CPSM-OBJNAME-SA = 'REMTRAN'
            IF NODATA-SW = 'Y'
                MOVE 4 TO RETURN-CODE-SA
            END-IF
        ELSE
            MOVE 4 TO RETURN-CODE-SA
        END-IF
    END-IF

WHEN EYUVALUE(NOTAVAILABLE)
    MOVE SPACES                TO EL1-ERROR-MSG
    STRING SCOPE-UNAVAIL-MSG-SA CPSM-OBJNAME-SA
        ')....' DELIMITED BY SIZE INTO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    PERFORM 750-CONVERT-ERR-TO-TEXT
    MOVE CPSM-CONTEXT-SA      TO EL2-CONTEXT
    MOVE CPSM-SCOPE-SA        TO EL2-SCOPE
    MOVE CPSM-ACTION-SA       TO EL2-ACTION
    MOVE CPSM-CRITERIA-SA     TO EL2-RESOURCE
    MOVE EL2-ERROR-LINE2     TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1

WHEN OTHER
    MOVE SPACES                TO EL1-ERROR-MSG
    STRING PROCESS-FAILED-MSG-SA CPSM-OBJNAME-SA
        ')....' DELIMITED BY SIZE INTO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    PERFORM 750-CONVERT-ERR-TO-TEXT
    MOVE CPSM-CONTEXT-SA      TO EL2-CONTEXT
    MOVE CPSM-SCOPE-SA        TO EL2-SCOPE
    MOVE CPSM-ACTION-SA       TO EL2-ACTION
    MOVE CPSM-CRITERIA-SA     TO EL2-RESOURCE
    MOVE EL2-ERROR-LINE2     TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 1
    MOVE CHECK-LOG-MSG-SA     TO EL1-ERROR-MSG
    MOVE EL1-ERROR-LINE1      TO PRINT-LINE-OUT
    WRITE PRINT-LINE-OUT AFTER 2
    MOVE 3002                 TO RETURN-CODE-SA

END-EVALUATE.

750-CONVERT-ERR-TO-TEXT.
    COMPUTE SUB-SS = CPSM-RESP-SA - RESP-FIRST-NUM-SA + 1.
    IF SUB-SS < 1 OR SUB-SS > RESP-NUM-ENTRIES-SA
        MOVE CPSM-RESP-SA      TO EYUDA-NUM-SA
        MOVE EYUDA-CHAR-SA     TO EL2-RESPONSE
    ELSE
        MOVE RESPONSE-TEXT-TB(SUB-SS) TO EL2-RESPONSE

```

```

END-IF.

COMPUTE SUB-SS = CPSM-REAS-SA - REAS-FIRST-NUM-SA + 1.
IF SUB-SS < 1 OR SUB-SS > REAS-NUM-ENTRIES-SA
    MOVE CPSM-REAS-SA          TO EYUDA-NUM-SA
    MOVE EYUDA-CHAR-SA        TO EL2-REASON
ELSE
    MOVE REASON-TEXT-TB(SUB-SS) TO EL2-REASON
END-IF.

800-READ-COMMAND-FILE.
READ COMMAND-FILE-IN INTO CRS-COMMAND-RECORD
    AT END MOVE 'Y' TO COMMAND-EOF-SW.
IF COMMAND-EOF-SW = 'N'
    ADD 1 TO COMMAND-RECORDS-READ-AA
END-IF.

810-READ-GROUP-FILE.
READ GROUP-FILE-IN INTO GRS-GROUP-RECORD
    AT END MOVE 'Y' TO GROUP-EOF-SW.
IF GROUP-EOF-SW = 'N'
    ADD 1 TO GROUP-RECORDS-READ-AA
END-IF.

```

---

*Randy Schulting*  
*Technical Services Analyst*  
*CUNA Mutual Group (USA)*

© Xephon 2000

---

## Monitoring CICS activity

This article describes how to keep an on-line and historical view of all the system and application activities in the CICS session, which are written to an application and/or CICS transient data. This process is done almost automatically.

We have developed a rather simple method to keep a close track on these activities, to archive those activities onto a monthly tape while CICS is running, ensuring that each month has its own tape, and to ensure data continuity when CICS is terminated or cancelled.

The solution described here is valid for any installation that uses CICS/ESA with JES2 and SDSF. It has been tested on CICS Versions 3.3 and 4.1 with MVS and OS/390.

## APPLICATION AND SYSTEM QUEUES

To start from the DCT, we have a TD queue whose name is well known to all the application programmers. This name is stored in the CICS CWA. The application programmer can retrieve the name from the CWA and use this queue to write application notes about things that are worth taking care of but are not critical enough to be written to the system console. In the test environment this queue may be used as a debugging tool.

We use the DCT indirection mechanism to point from the queue logical name to the extra-partition queue, and from the extra-partition queue to the SDSI. The DCT source code is listed below:

```
DFHDCT TYPE=SDSCI, DSCNAME=DCPMJNA, RECSIZE=142, BLKSIZE=146,
BUFNO=1,RECFORM=VARUNB, TYPEFLE=OUTPUT
.....
DFHDCT TYPE=EXTRA, DESTID=DJNA, DSCNAME=DCPMJNA, OPEN=INITIAL
.....
DFHDCT TYPE=INDIRECT, DESTID=DCPM, INDDEST=DJNA
.....

DFHDCT TYPE=SDSCI, DSCNAME=DCPMJNF, RECSIZE=142, BLKSIZE=146,
BUFNO=1,RECFORM=VARUNB, TYPEFLE=OUTPUT
.....
DFHDCT TYPE=EXTRA, DESTID=DJNF, DSCNAME=DCPMJNF, OPEN=INITIAL
.....
DFHDCT TYPE=INDIRECT, DESTID=SKØ3, INDDEST=DJNA
```

In the table above there are two ‘application’ queues – one is for the general application queue, and the other is for the ATM sub-system, which is monitored by a different group of people.

For the CICS system programmer, all CICS system queues (full description of which might be found in the CICS resource definition macro) have been split into two groups:

- The CICS system queues that deal with terminals and VTAM-CICS activity.
- The rest of the CICS system queues.

This separation has been done because the queues that hold ‘network’ information tend to be very big, and it is more convenient to see the messages sent to the ‘non-network’ queue.

It is possible, of course, to point from many indirect queues to one extra-partition queue, although in this method data from different queues mix in the extra-partition queue.

On the whole, in our shop, we keep four different SDSCI. They are for:

- The main application log.
- The 'special' application log.
- General CICS queue.
- The CICS queues that deal with terminals, VTAM-CICS transportation, etc.

It is an installation decision whether and how to split the CICS queue into different DDnames. One might consider splitting the 'application logs' by the degree of severity. Another method is to keep all the CICS built-in queues, which deal with the RDO, in a single separate DDname to keep an eye on the RDO activity. Anyhow, the mechanism described here can be tailored easily.

In our CICS job we have four DDnames, one for each SDSCI.

The declarations of those DD cards are listed below:

```
//DCPMJNT DD SYSOUT=M,DCB=(DSORG=PS,RECFM=V,BLKSIZE=660),FREE=CLOSE
//DCPMJNL DD SYSOUT=M,DCB=(DSORG=PS,RECFM=V,BLKSIZE=660),FREE=CLOSE
//DCPMJNA DD SYSOUT=M,DCB=(DSORG=PS,RECFM=V,BLKSIZE=730)
//DCPMJNF DD SYSOUT=M,DCB=(DSORG=PS,RECFM=V,BLKSIZE=730)
```

The 'FREE=CLOSE' parameter means that whenever the external queue, which points to the SDSCI, is closed in CICS, the DDname will be freed. The BLKSIZE parameter should be chosen according to the installation, but one should not use too big a block size, because this will cause a delay in writing some of the records to JES.

#### DETECTING THE QUEUE CONTENTS WHILE CICS IS RUNNING

The output of the queues is written to JES2, so it is possible, while CICS is running, for the application programmers to see all the records that were written to the transient data. After locating the CICS job in JES, using the SDSF display command '?' will show all the DDnames.

An example is shown below:

DCPMJNA	SS50CICP	139	MMU	M	3,257	389,474	1				
DCPMJNF	SS50CICP	138	MMU	M	1,352	154,574	1				
DCPMJNL	SS50CICP	137	MMU	M	1,446	136,370	1				
DCPMJNT	SS50CICP	136	MMU	M	28,490	2M	1				
DCPMJNA	SS50CICP	135	MMU	M LOCAL	2,740	319,299	1	3	12	LINE	
DCPMJNF	SS50CICP	134	MMU	M LOCAL	806	115,948	1	3	11	LINE	
DCPMJNL	SS50CICP	133	MMU	M LOCAL	787	68,010	1	3	10	LINE	
DCPMJNT	SS50CICP	132	MMU	M LOCAL	13,203	1M	1	3	9	LINE	
DCPMJNA	SS50CICP	131	MMU	M LOCAL	4,628	567,519	1	3	8	LINE	
DCPMJNF	SS50CICP	130	MMU	M LOCAL	782	115,090	1	3	7	LINE	
DCPMJNL	SS50CICP	129	MMU	M LOCAL	942	83,043	1	3	6	LINE	
DCPMJNT	SS50CICP	128	MMU	M LOCAL	37,998	3M	1	3	5	LINE	
DCPMJNF	SS50CICP	125	MMU	M LOCAL	417	61,972	1	3	3	LINE	
DCPMJNA	SS50CICP	124	MMU	M LOCAL	2,503	281,485	1	3	4	LINE	
DCPMJNL	SS50CICP	123	MMU	M LOCAL	1,353	130,076	1	3	2	LINE	
DCPMJNT	SS50CICP	122	MMU	M LOCAL	19,131	1M	1	3	1	LINE	
GDMT	SS50CICP	110	MMU	A	0						1
DFHCXRF	SS50CICP	106	MMU	T	68	7,816	1				

The default sort of the DDnames is by DSID. Using this sort, the currently active DDname will always display on the top of the list, so if the application programmer wishes to see the current notes of his/her application he/she will select 'S' near the DCPMJNA on the top of the list.

Application DDname can be easily located. The programmer can now see all the messages that were issued by the application. The system programmer can look at a different DDname to see what is going on in CICS.

## SWITCHING TO A NEW GENERATION

In order to switch to a new generation, four operations have to be performed inside CICS:

- All the queues (the extra-partition queues) must be closed. The syntax of the command is:

```
CEMT SET TDQ(DJN*) CLOSE
```

- All the DDnames must be de-allocated using SVC99. The syntax of the command is:

```
ADYN FREE DDname(DCPMJN*) UNALLOC
```



This step is not necessary if 'FREE=CLOSE' is declared.

- The new DDname must be allocated again using:

```
ADYN ALLOC DDname(DCPMJN*) SYSOUT(M)
```

- Lastly, the queue should be opened again using:

```
CEMT SET TDQ(DJN*) open
```

The \* stands for all the suffixes of the queues and DDnames.

Between the queue closing and re-opening it is impossible to write to it, but since this is a very short period, at non-business hours, it is not a problem.

Right after the DDname is allocated again, a new generation is created, and a batch program may be used to copy the older generation to a monthly tape.

## AUTOMATING THE SWITCHING PROCESS

The sequence of four operations, described above, can be carried out in several ways:

- Using a product (in our case Control-O from BMC/New Dimension) to log into the CICS and use the transaction ADYN. This solution works for us in our production environment. The transaction ADYN comes with the CICS installation in the group DFH\$UTIL.

The source of the Control-O program is attached to this article, but it is useless for CICS installations that do not have Control/O. This source program can be used as an example of how to use TDYN Transaction after signing on to CICS.

- Using the transaction TDYN and the program it starts to de-allocate and allocate the DD cards. TDYN has the same functionality as ADYN, but it is not a conversational transaction and, as such, does not require any terminal to be logged on. This is a general solution, which will hold for any CICS-MVS installation. Its disadvantage is that there is no control over the results of the allocation/de-allocation.

TPADYN3 is the Assembler program that the TDYN transaction uses. The source of TPADYN3 and the routine it uses, JHTRT, are part of this article. Please note that with slight modifications this program may be useful for other tasks in CICS system programming.

- It is also possible to use the CICS external interface (EXCI) to issue the DFH99 program. The advantage of this solution is that there is full control and guarantee over the results of the allocation/de-allocation. The disadvantage is that EXCI is rather complicated.

A batch-to-CICS interface can be found in the February 1999 issue of *CICS Update*, Issue 159, pages 25-31, *Exploiting EXCI to manage CICS files from batch*.

In each of the methods selected, first de-allocation and then allocation must be done. If the parameter 'FREE=CLOSE' is defined in the CICS job JCL, then de-allocation is simple, using CEMT to close the external queue. Closing the queue will flush all the records to JES and the queue will be de-allocated. If 'FREE=CLOSE' is not defined, then after closing the queue it must be de-allocated by using ADYN or TDYN, which eventually issues the MVS SVC99.

The process of de-allocation and allocation must be done, of course, for all the queues. After the de-allocation and allocation of the queue, a new DDname will appear in JES2 and all the activities of the CICS system and applications will be written into the new DDname.

If the process of de-allocation and allocation is done daily at 5am, then each DDname represents the CICS activities for the day. This information will be kept on JES until CICS is recycled.

## KEEPING AND MAINTAINING THE ACTIVITIES

After the switching (de-allocation and allocation) has taken place, the data should be collected from JES and stored on a monthly tape. The scheduler (in our case Control-M) does this immediately after the de-allocation and allocation program (which is done inside CICS). It starts the following job:

```
//SS01DCPP JOB (SS04,626,99),SDSFBAT,NOTIFY=S004
```

```

//*****
/* SDSF batch job to release CICS massive printing *
/* this job must run on a system where CICS is running *
/* the first two steps simply invoke from batch program SDSF *
/* which allows the user to imitate the on-line operations *
/* from batch. Note that when this program is running the second *
/* DDname is what we want to copy to the tape, since the *
/* "log switching" has already been done *
//*****
//EXEC PGM=SDSF,PARM='++24,131'
//ISFOUT DD SYSOUT=T
//ISFIN DD *
PREFIX SS50*
DA
FIND SS50CICP
++?
SORT DSID D
FIND DCPMJNA
DOWN 1
FIND DCPMJNA
++S
PRINT ODSN 'SYS0.DCPMJNA' * MOD
PRINT
PRINT CLOSE
//EXEC PGM=SDSF,PARM='++24,131'
//ISFOUT DD SYSOUT=T
//ISFIN DD *
PREFIX SS50*
DA
FIND SS50CICP
++?
SORT DSID D
FIND DCPMJNF
DOWN 1
FIND DCPMJNF
++S
PRINT ODSN 'SYS0.DCPMJNF' * MOD
PRINT
PRINT CLOSE
/* .----- .
/* |
/* | STEP #1 :
/* | CICS/LOG DATE CHECK
/* | POSSIBLE RC:
/* | 10...NEW DAY
/* | 20...NEW MONTH
/* | 20...NEW YEAR
/* | 99...SAME DAY
/* |
/* | .----- .

```

```

/*
//CHECKDT EXEC PGM=IKJEFT01,DYNAMNBR=30,TIME=1440
//SYSTSPRT DD SYSOUT=T
//SYSPRINT DD SYSOUT=M
//SYSPROC DD DSN=SYS2.CLIST,DISP=SHR
//ISPPLIB DD DSN=SYS1.ISP.SISPPENU,DISP=SHR
//ISPMLIB DD DSN=SYS1.ISP.SISPMENU,DISP=SHR
//ISPSLIB DD DSN=SYS1.ISP.SISPSLIB,DISP=SHR
//ISPTLIB DD DSN=SYS1.ISP.SISPTENU,DISP=SHR
//ISPPROF DD UNIT=VIO,SPACE=(TRK,(9,1,4)),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//ISPTABL DD UNIT=VIO,SPACE=(TRK,(9,1,4)),
//          DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//ISPLIST DD DSN=&&LIST,UNIT=VIO,SPACE=(TRK,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLOG DD DSN=&&LOG,UNIT=VIO,SPACE=(TRK,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//TMOND DD DSN=SYSO.CICSPROD.DATEBU,DISP=SHR
//SYSTSIN DD DSN=PMTF.PARMLIB(CBKDATE),DISP=SHR
/*
/* -----
/* | STEP #2 : |
/* | IF IT IS THE SAME DATE, |
/* | THE BACK-UP WAS RERUN - |
/* | COPY TO THE SAME MONTH |
/* | -----
/*NODUMP EXEC PGM=IEFBR14,
/* COND=(99,NE,CHECKDT)
//DUMSDAY1 EXEC PGM=IEBGENER,
// COND=(99,NE,CHECKDT)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNA,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNA(0),DISP=MOD
//SYSIN DD DUMMY
/*
//DUMSDAY2 EXEC PGM=IEBGENER,
// COND=(99,NE,CHECKDT)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNF,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNF(0),DISP=MOD
//SYSIN DD DUMMY
/*
/* -----
/* | STEP #3 : |
/* | IF IT IS A NEW DAY AT |
/* | THE SAME MONTH |
/* | WRITE INTO EXISTING TAPE |
/* | -----

```

```

//* -----
//DUMPMOD1 EXEC PGM=IEBGENER,
//      COND=(10,NE,CHECKDT)
//SYSPRINT DD  SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNA,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNA(0),DISP=MOD
//SYSIN  DD  DUMMY
/*
//DUMPMOD2 EXEC PGM=IEBGENER,
//      COND=(10,NE,CHECKDT)
//SYSPRINT DD  SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNF,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNF(0),DISP=MOD
//SYSIN  DD  DUMMY
/*
/*
/* -----
/* | STEP #3 : |
/* | TAKE THE DUMPED FILE AND |
/* | COPY IT IN DISP=NEW TO |
/* | NEW CUMM. MONTH TAPE (DETAIL) |
/* | _____ |
/*
//COPYNEW1 EXEC PGM=IEBGENER,COND=(20,NE,CHECKDT)
//SYSPRINT DD  SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNA,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNA(+1),DISP=(NEW,CATLG),
//      DCB=(MODEL,RECFM=VB,LRECL=125,BLKSIZE=1340,
//      DSORG=PS),UNIT=TP90 ,VOLUME=(,,50)
//SYSIN  DD  DUMMY
/*
/*
//COPYNEW2 EXEC PGM=IEBGENER,COND=(20,NE,CHECKDT)
//SYSPRINT DD  SYSOUT=*
//SYSUT1 DD DSN=SYSO.DCPMJNF,DISP=SHR
//SYSUT2 DD DSN=SYSO.CICSLOGT.DCPMJNF(+1),DISP=(NEW,CATLG),
//      DCB=(MODEL,RECFM=VB,LRECL=125,BLKSIZE=1340,
//      DSORG=PS),UNIT=TP90 ,VOLUME=(,,50)
//SYSIN  DD  DUMMY
/*
/*
//RECYCL1 EXEC PGM=IEFBR14
//FILE DD DSN=SYSO.DCPMJNA,DISP=(OLD,DELETE,DELETE)
//      DD DSN=SYSO.DCPMJNF,DISP=(OLD,DELETE,DELETE)
//RECYCL2 EXEC PGM=IEFBR14
//FILE DD DSN=SYSO.DCPMJNA,DISP=(NEW,CATLG),
//      DCB=(MODEL,RECFM=VB,LRECL=125,BLKSIZE=1340,DSORG=PS),
//      UNIT=3390,VOL=SER=TEST03,SPACE=(TRK,(5,1))
//      DD DSN=SYSO.DCPMJNF,DISP=(NEW,CATLG),
//      DCB=(MODEL,RECFM=VB,LRECL=125,BLKSIZE=1340,DSORG=PS),

```

```
//          UNIT=3390,VOL=SER=TEST03,SPACE=(TRK,(5,1))
```

The job above contains a call to SYS2.CLIST(CHKDATE); the CLIST source is attached below. What SYS2.CLIST(CHKDATE) is doing is rather simple. It compares today's date with the day in the file SYSO.CICSPROD.DATEBU. It returns RC20 for a new year, 10 for a new month, and 99 for the same month.

The execution day is copied to file SYSO.CICSPROD.DATEBU at the end of the CLIST.

The file SYSO.CICSPROD.DATEBU should be unique at an installation.

The program was changed and it is now Y2K-enabled.

```
/* REXX */
TRACE N
STAT=MSG(ON)
CURDATE = DATE('E')
CURDAY  = SUBSTR(CURDATE,1,2)
CURMON  = SUBSTR(CURDATE,4,2)
CURYER  = SUBSTR(CURDATE,7,2)
READ:
  "EXECIO 1 DISKRU TMOND"
  PULL TMOND
  SAY 'Today is.....' CURDATE
  SAY 'Last run was at.....' TMOND
  TMDATE = TMOND
  TMDAY  = SUBSTR(TMDATE,1,2)
  TMMON  = SUBSTR(TMDATE,4,2)
  TMYER  = SUBSTR(TMDATE,7,2)
NEWYEAR:
  IF CURYER <> TMYER THEN DO CC = 20
                          TMDATE = CURDATE
                          PUSH TMDATE
                          "EXECIO 1 DISKW TMOND"
                          SIGNAL EOF
  END
NEWMON:
  IF CURMON <> TMMON THEN DO CC = 20
                          TMDATE = CURDATE
                          PUSH TMDATE
                          "EXECIO 1 DISKW TMOND"
                          SIGNAL EOF
  END
NEWDAY:
```

```

IF CURDAY <> TMDAY THEN DO CC = 10
                        TMDATE = CURDATE
                        PUSH TMDATE
                        "EXECIO 1 DISKW TMOND"
                        SIGNAL EOF
                        END
SAMEDAY:
  IF CURDAY = TMDAY THEN CC = 99
EOF:
  "EXECIO 0 DISKW TMOND (FINIS"
  EXIT CC

```

## READING THE LOG

Usually application programmers like to know what is going on now in the currently running CICS, but sometimes they discover a problem in an application a week or even a month later.

The method described here enables the application/system programmer to easily browse the CICS system log, as long as the data he/she wishes to read is not older than the number of generations we keep for the tape. Since each tape contains one month, it is recommended to keep at least six generations – enabling a half-year history log.

The job to retrieve the data is fairly simple and it is listed here:

```

//Ss004DMP JOB (SS01,A1,20),URIC,MSGCLASS=T,NOTIFY=S004
//IEBGENER EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=SYSO.CICSLOGT.DCPMJNA(0),DISP=SHR
//*SYSUT1 DD DSN=SYSO.CICSLOGT.DCPMJNF(0),DISP=SHR
//SYSUT2 DD DSN=SYSP.LOGCHK2,DISP=SHR
//SYSIN DD DUMMY

```

All that the CICS system programmer has to do is to fill in the right generation in the DDname SYSUT1. If we are in April and after a cold start to CICS we wish to know who has deleted that transaction from the RDO back in January, we need to insert ‘-3’ inside the brackets.

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

---

*Uri Cohen*  
*CICS Systems Programmer (Israel)*

© Xephon 2000

---

# CICS news

---

Sybase has announced its OpenDoor enterprise portal, which integrates existing business applications and databases.

OpenDoor includes built-in support for portal systems management, mainframe integration with CICS and IMS, XML support for major data sources, integrated internationalization, and single sign-on security.

It allows personalized integration to real-time business events, ERP applications, and native connectivity for major relational databases.

There's built-in support for load balancing, clustering, and RAS facilities.

For further information contact:  
Sybase, 6475 Christie Avenue, Emeryville,  
CA 94608-9967, USA.  
Tel: (510) 922 3500.  
Sybase (UK), Sybase Court, Crown Lane,  
Maidenhead, Berks, SL6 8QX, UK.  
Tel: (01628) 597100.  
<http://www.sybase.com>.

\* \* \*

IBM has announced CICS VSAM Recovery VSE/ESA (CICSVR) Version 1.2, supporting both CICS Transaction Server for VSE and CICS/VSE Version 2.3. It's for automating VSAM recovery procedures.

Features include an Archive Utility, which creates a report with summary information on all the VSAM datasets detected on each journal and makes copies of system logs or user journals.

There's also a forward recovery function, which helps recreate one or more VSAM datasets from back-up copies when a

recovery is required. The forward recovery doesn't apply to VSAM ESDS inserts that have been marked for deletion by CICS.

Also new is a backout function, which removes partially completed transactions from one or more VSAM datasets when CICS on-line backout has failed.

For further information contact your local IBM representative.

URL: <http://www.ibm.com>.

\* \* \*

IBM has announced that the Web User Interface component of CICSplex System Manager in CICS Transaction Server for OS/390 Release 3 is now available. This interface to CICSplex SM for use with a Web browser will be included in new shipments.

The CICSplex System Manager Web User Interface is provided as an alternative to the existing 3270 interface and is intended for use by functions such as Help Desk and operations. This interface is not intended for administrator configuration of CICSplex SM. It provides a Single System Image (SSI), suitable for managing CICS resources from a single point of control.

In addition, IBM has announced enhancements to CICS Transaction Server for OS/390 Release 3, including Java Hot Pooling and CICS Web Interface API enhancements.

The CICS Web Interface has been enhanced with the introduction of new API commands to facilitate the processing of HTML forms.

For further information contact your local IBM representative.

URL: <http://www.ibm.com>.



**xephon**