174

CICS update

May 2000

In this issue

© Xephon plc 2000

CICS Update

Published by

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

North American office

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

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.

Buried treasure in the CICS TRACE TABLE

INTRODUCTION

Each new release of CICS will contain a collection of additional trace entries, introduced to provide diagnostic information on new components of the product added by the release. As CICS Transaction Server continues to evolve, so does the variety and content of the trace data associated with CICS.

There are a number of very useful CICS trace entries that can be used to provide a great deal of helpful information, provided you know what to look for within the (potentially) vast amount of trace data that CICS can generate. This article highlights a number of such trace entries, explaining when they were introduced, what they can offer you, and the trace options required to generate them.

A BRIEF BACKGROUND TO CICS TRACING

CICS trace is the primary tool for application and system debugging. It is a series of data entries, written in chronological order, that document the state of various system resources and activities. It shows the flow of activity of tasks on the CICS system, running under the quasi-reentrant (QR) TCB and other TCBs available for use within CICS Transaction Server. While originally intended for CICS system debugging, a subset of the trace entries can be useful for debugging application problems too, as will be explained later in the article.

CICS trace may be directed to the internal CICS trace table, which is a wraparound area of storage above the 16MB line in the CICS address space. This can then be viewed in both transaction and system dumps, to see the series of events leading up to the event that resulted in the dump being requested. The size of this trace table is controlled by the TRTABSZ system initialization parameter, and by the ‘Internal Trace Table Size’ option on the CETR main panel (for more details on CETR, see below).

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

In addition, trace entries can also be directed to the auxiliary trace destination, which is one or two CICS-managed BSAM datasets. These provide a larger repository for trace data than the internal CICS trace table, and can be used to record a longer period of system activity. The I/O activity needed to support auxiliary tracing means that this option has more impact upon CICS system performance than just using internal CICS tracing.

CICS also supports tracing to GTF. This technique can be useful when diagnosing problems that have a flow of events across several interconnected address spaces, such as between CICS and DBCTL.

CICS/ESA Version 3 introduced the CETR transaction to complement the restructuring of CICS trace control into a domain with that version of the product. CETR allows the user to dynamically control many aspects of CICS trace activity, including switching on and off internal and auxiliary tracing, setting the level of component tracing for the various functional areas within CICS, and using selective tracing for specific transactions and terminals. Such selective tracing can be very useful when debugging CICS applications – by providing detailed trace information for a specific program environment.

Note that the CICS internal trace table, auxiliary trace, and GTF trace destinations can wrap. CETR provides the means of controlling whether auxiliary tracing wraps by the 'Auxiliary Switch Status' option. This can specify NO, NEXT, and ALL. NO means that only the current BSAM dataset can be used. NEXT means that having filled the current dataset, CICS will switch to use the second dataset. ALL means that CICS will automatically switch between the two datasets as each one fills. ALL therefore allows for the overwriting of old trace data from earlier in the trace run.

CICS provides a batch utility program to format the auxiliary trace data. The name of the program is CICS release-specific – its suffix is the CICS release number. Therefore, in CICS/ESA 4.1.0 the program is DFHTU410; in CICS Transaction Server 1.3 (containing the CICS component CICS/ESA 5.3.0) it is DFHTU530. The utility program has a number of options that can be used to selectively filter specific trace entries when formatting the auxiliary trace data – these include

© 2000. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 410 0290.

filtering by task number, transaction identifier, trace entry number, etc. The most commonly used options when using the utility are to control the level of detail to be returned in the formatted trace. The options are ABBREV (for abbreviated), SHORT, and FULL. Abbreviated trace is the bare minimum of data for every trace entry, showing the flow of events through CICS and the commands issued by applications. It is useful when trying to establish a 'big picture' of what is happening at a given point during the CICS run. Short trace is similar to abbreviated trace, but also provides useful diagnostic information such as the time when the trace entry was issued, and time duration between this and the preceding trace entry. Full trace provides all this information, plus up to seven data items that can be used to trace information such as parameter lists, names of resources, control blocks, and response and reason codes.

Full trace is useful when a particular series of events has been isolated, and a problem needs to be analysed at a trace entry by trace entry level, seeing exactly what parameters (and their values) were being referenced at that point in time.

EIP LEVEL 1 AND LEVEL 2 TRACES

CICS trace is a powerful tool for debugging application program problems. By following the series of EXEC CICS requests issued from applications, programmers can track the flow of execution of a new application suite under development. In this way, CICS trace can be used in tandem with CEDF to diagnose and debug problems encountered during the application development cycle. One possible scenario would be to use CEDF when functionally verifying (ie unit testing) applications, and using CICS trace to analyse system testing of the programs, when multiple tasks are executing them in tandem.

Prior to CICS/ESA 4.1.0, the trace entries that were of most use in understanding the flow of EXEC CICS requests being issued by application programs were those issued on entry to, and exit from, the CICS EXEC Interface Program (DFHEIP). DFHEIP is the entry point into CICS from an application – it is branched to by the command level stub that is link-edited to the program, which is itself branched to as part of the translated EXEC CICS command being executed within the application. DFHEIP then vectors into one of a number of command level stubs for the various types of EXEC CICS request. Eventually, once CICS has processed the command, control will return to the application via DFHEIP once more.

DFHEIP has a pair of trace points at entry to and exit from the module – these have the trace point ID of AP 00E1, where AP denotes the Application Domain, the component of CICS that contains the remainder of unreconstructed code. (Starting with CICS/ESA 3.1.0, as areas of CICS have been rewritten into better designed, more encapsulated pieces of function, they have been restructured into their own domains. Examples include the storage manager domain, loader domain, program manager domain, etc.)

CICS/ESA 4.1.0 introduced another pair of DFHEIP trace entries. These are EI level 2 trace points, meaning that they are issued only if CICS has level 2 trace function set for the EI trace component, as controlled by the CETR transaction or SIT parameter. These new trace points are AP E160 and AP E161 (on entry and exit respectively). Note: the traditional AP 00E1 trace points are EI level 1 traces.

EI level 2 trace points have four data items associated with them. Item 1 is a concatenation of addresses and argument values, used by CICS when constructing the trace entry for the EXEC CICS command. Item 2 is the list of addresses of the command-level parameters built by the CICS translator for the EXEC CICS request. The final parameter is denoted by the high order bit (X'80000000') set on in the address. Item 3 is the address of this parameter list. Item 4 is the system EIB control block.

EI level 2 tracing is a very powerful tool for problem diagnosis of CICS applications. Where the AP 00E1 trace points show the flow of control for an application, and give a limited idea of the nature of the EXEC CICS requests, the EI level 2 trace points break down the actual requests themselves. In effect they are providing part of the data available under CEDF, but without the need to step through the application code screen by screen.

Application programmers can see exactly what values are being manipulated by their applications without having to step through the code on-line via CEDF. This can be useful in environments where, for example, CEDF usage cannot be tolerated for some reason, or in the case when a number of tasks are to be executed in a simulated production run.

An important point to note about EI level 2 trace interpretation is that the value of input and output fields passed on the request cannot be traced unless the length of the fields is also passed. An example of this is when tracing a file control request against a KSDS file. The value of the key field passed on the request (the ridfld) cannot be traced unless KEYLENGTH was specified on the EXEC CICS request. This is because the trace formatter has no idea how long the file's keylength is, and cannot arbitrarily display a certain number of bytes of data starting at the address where the key value is held. Another example of this is when tracing an EXEC CICS SEND MAP command with a FROM area, but no associated length value.

ABEND AEYD DIAGNOSTIC TRACE ENTRIES

CICS command protection was a function introduced in CICS/ESA 4.1.0. It is controlled by the SIT parameter CMDPROT. If set to YES, this instructs the CICS Exec Interface Program, DFHEIP, to perform verification checks on any output field addresses before allowing an EXEC CICS command to continue.

This verification is to prevent an application passing CICS an invalid address as the target for data returned from CICS to the application. An example of such an output field would be the one passed as the INTO area on an EXEC CICS READ command against a VSAM file. If CICS does not verify that such output field addresses are valid (ie the application has the right to update them) then it can unwittingly corrupt storage belonging to parts of the CICS address space that are not validly accessible by the application. Since CICS runs in storage protection key of 8 ('CICS key'), it has the ability to update storage in both storage protection keys 8 and 9 (key 9 is 'User key' storage). As such, there is the potential for an application to use CICS as a 'hired

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

gun' and update pages of storage that the application could not directly touch by itself. Command protection is therefore also sometimes referred to as hired gun checking. This is because it prevents CICS being used by the task issuing an EXEC CICS request to indirectly cause possible harm to storage belonging to other tasks in the system.

If DFHEIP detects that an EXEC CICS parameter for an output field addresses storage that is not available to the task that issued the request, it generates a program check. This is captured by the CICS System Recovery Program (DFHSRP). DFHSRP will issue a series of diagnostics such as exception trace entries, an abend AEYD, and a transaction dump.

```
ØØØ233 1 AP ØØE1 EIP   EXIT  ASSIGN                OK
ØØØ233 1 AP ØØE1 EIP   ENTRY HANDLE-ABEND

ØØØ233 1 AP ØØE1 EIP   EXIT  HANDLE-ABEND          OK

ØØØ233 1 AP ØØE1 EIP   ENTRY READ

ØØØ233 1 AP 1942 APLI  *EXC* Program-Check

ØØØ233 1 AP Ø79Ø SRP   *EXC* PROGRAM_CHECK

ØØØ233 1 AP Ø779 SRP   *EXC* ABEND_AEYD



AP Ø779 SRP  *EXC* - ABEND_AEYD PROGRAM(U392ØPRG) FUNCTION(Ø6Ø2)

                          PARAMETER_ADDRESS(ØØØØØØØØ)
```

Figure 1: Trace entries showing an AEYD abend

Figure 1 shows examples of the (edited) trace information seen when an invalid output parameter address is detected by CICS command protection checking. The series of abbreviated trace entries at the top shows that a CICS application has issued a number of EXEC CICS commands, culminating in an EXEC CICS READ to CICS file control. In this example, the address passed for the INTO area on the request is 00000000 (ie ‘low-core’, the start of the Prefix SaveArea or PSA, in the first page of every address space in MVS). A CICS task has no right to modify storage in the PSA (nor indeed does CICS for that matter) – it is used specifically by the hardware to represent processor information such as interrupt routine addresses for the operating system. As such, any attempt to store into this page would fail with an 0C4 protection exception anyway. However, since CICS command protection was activated, CICS was able to detect that the INTO field contained an invalid address and so prevent the attempt to store VSAM record data there before the protection exception 0C4 could occur.

When CICS detected the invalid output parameter, it forced a program check. This resulted in the exception traces AP 1942 and AP 0790 being issued. DFHSRP then determined that the program check was as a result of command protection validation, and so issued the further exception trace entry AP 0779. The full trace entry for this shows the address of the bad output area parameter as passed on the EXEC CICS READ command – in this example it is 00000000.

The example shows a bad output field address on a command that would, if left to completion, not have been able to be updated by CICS since it is in protected storage in low-core. However, the address could instead have been a validly accessible one within part of the CICS address space for a page of storage owned by another task. If command protection had not verified that such an address was out of bounds to the task asking for it to be updated by CICS, storage corruption of this area by CICS on behalf of the running task would have occurred. CICS would indeed have been a hired gun for the task.

Command protection can be used in tandem with EI level 2 trace to display all the output fields passed by an application, if further investigation of the program is required.

CICS/DB2 ADAPTOR TRACE ENTRIES

CICS Transaction Server Release 1.2 included a rewritten CICS/DB2

Adaptor. This is the code that interfaces between CICS and DB2 – it runs as a Task Related User Exit (TRUE) to the CICS external Resource Manager Interface (RMI).

Prior to CICS Transaction Server Release 1.2, the CICS/DB2 Adaptor was owned by the DB2 Development Laboratories in Santa Teresa, California, although the code itself was shipped as part of the CICS product delivery tape. When the Adaptor was transferred to CICS Development in Hursley, the opportunity was taken to restructure the code and improve a number of its components. Part of this restructure involved improvements to the tracing techniques used by the Adaptor modules. The old CICS/DB2 Adaptor had used user trace entries to record the flow of events through its code. The restructured Adaptor in CICS Transaction Server Release 1.2 issues new unique trace entries from the Application Domain (AP) trace point range. This is one example of the ways in which the new Adaptor code is more formally integrated into CICS Transaction Server than the old Adaptor in CICS/ESA 4.1.0.

If an exceptional condition occurred whilst running under the old-style Adaptor, a pair of user trace entries would be issued to document the fact. The end of data item 1 would contain *EXC* in both cases, to indicate that this was for an exceptional condition. Data item 2 in the first of the trace entries would contain the DB2 reason code for the failure, in its last fullword. Data item 2 in the second of the trace entries would contain the transaction identifier of the executing task in its first fullword.

With the restructure of the CICS/DB2 Adaptor, the opportunity to exploit conventional CICS tracing techniques allowed for the use of standard exception trace entries to be issued in exceptional conditions.

An example of such an exception condition is discussed below. A failure condition might occur as the result of an application issuing an EXEC SQL command.

The resulting AP 318E exception trace entry issued from the Adaptor has four data items.

Data item 1 is the CICS kernel error data for the failure. This includes the Abend code for the failure (AD2U) at offset 4. It also shows the name of the Adaptor program in control when the abend was issued – DFHD2EX1 at offset X'10'.

Data item 2 is the LOT control block, DFHD2LOT. The LOT (or Life Of Task) control block represents the unit of work that is executing the series of commands.

Data item 3 is the RCT entry being used by the request. This is mapped by the DSECT DFHD2RCT.

Data item 4 is the CSUB control block, DFHD2CSB. The CSUB (or CICS subtask) control block represents the TCB that is being used to process the particular request to DB2. The CICS/DB2 Adaptor cannot use the QR TCB to process requests in DB2 since this would prevent sub-dispatching of other tasks by the CICS Dispatcher under this TCB while a request was being executed within DB2 itself. Hence a series of TCBs are used to subtask the DB2 requests issued by CICS applications. This is a standard technique used by other RMI TRUEs such as the DBCTL Adaptor.

The CSUB control block contains a very useful feature, which can be analysed from the trace entry. An imbedded 'trace table' within the control block shows the last (decimal) ten operations carried out on that subtask. These begin at offset X'1E0', and each entry is X'10' bytes long. They are delimited by the eyecatchers '>>Trace Start >>' and '<<Trace End <<'.

Each X'10' byte entry contains a fullword hexadecimal counter, a fullword character string to identify the type of request, and then sections for the DB2 response and reason codes from the request. Note that APAR PQ30799 / PTFs UQ36758 and UQ36759 have enhanced the CSUB trace table to include the task number of the application. This is stored in packed decimal format in the second, third, and fourth bytes of each entry's first fullword, leaving the first byte of the entry for the monatomically increasing hexadecimal counter. Since there are only ten slots within the imbedded table, there was no compelling reason for the counter to use a fullword of storage – a single byte is more than sufficient to ensure a uniquely increasing range throughout

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

the ten entries. Note: PTFs UQ36758 and UQ36759 enhanced CICS Transaction Server Release 1.2 and Release 1.3 respectively.

The DFHD2CSB is an internal trace table that can be seen at the end of the control block. Note: the example discussed predates the application of APAR PQ30799, so that the task numbers are not stored within the first fullword of each entry, and the counter field is still a fullword value. To locate the most recent entry into the trace table, locate the entry with the highest counter value. In our example it is the entry with a counter of 0000000D, at offset X'200' into the control block.

The possible character strings at offset 4 into the entries, and their meanings, are given below:

- ABRT – abort request
- API – SQL or IFI request
- COMM – commit request
- CTHD – create thread request
- ERRH – error handler request
- IDEN – identify request
- PREP – prepare request
- PSGN – partial signon request
- SIGN – full sign-on request
- SYNC – single phase request
- TERM – terminate thread request.

In our example trace entry, error handling has been driven as the result of a previous authorization failure (frb reason code 00F30034).

Understanding this internal trace table allows you to track the most recent requests to have been executed under this subtask, and so aid in analysis of throughput for CICS/DB2 requests, and help when dealing with problem determination issues.

Note, in addition to being traced for exception trace entries as described above, the CSUB will also be traced on normal trace entries from the new CICS/DB2 Adaptor if the RI trace component is set to 1-2. RI is the trace component for the CICS RMI (Resource Manager Interface). This can be useful when examining situations that do not result in exceptional conditions occurring.

TRACE ENTRY SUPPRESSION

Most events in life involve a trade-off of some kind, and CICS trace recording is no exception.

In order to help reduce performance overheads, some users will run their production CICS systems with limited or no CICS trace recording active. This is advantageous to them until an unexpected situation occurs that requires trace data for its problem resolution.

Another reason why CICS trace entries may not be recorded is suppression by one of a number of different monitoring packages.

While CICS does generate unsolicited exception trace entries when unexpected (exceptional) conditions occur, which are recorded regardless of trace suppression, these in themselves may not be sufficient to totally isolate and resolve a particular problem.

When examining the data in CICS trace (either from the internal trace table or from auxiliary trace) it is important to be aware of the scope of trace suppression on the CICS system.

FURTHER READING

The CICS bibliography provides a detailed overview of every trace entry issued by CICS. However, the particular manual that lists the trace entries varies depending upon what release or version of the CICS product is being used.

For CICS/ESA 4.1.0, the trace entries are documented in the *Diagnosis Handbook.*

For CICS Transaction Server Release 1.1 and Release 1.2, they are documented in the *User's Handbook*. For CICS Transaction Server

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

Release 1.3, the volume of trace entries has grown sufficiently to warrant a complete manual; this is entitled *CICS Trace Entries*.

SUMMARY AND CONCLUSIONS

I hope that this article has helped explain the background to a number of useful features of the CICS trace table. Readers who wish to discuss the material in this article further may contact me via e-mail, at andy_wright@uk.ibm.com. CICS is a registered trademark of International Business Machines Corporation.

Andy Wright
CICS Change Team
IBM (UK)
© IBM 2000

Tuning AOR/FOR Traffic in CICS/TS 1.2

INTRODUCTION

CICS Version 5 (in CICS/Transaction Server Version 1) provides a significant enhancement over CICS 4.1 in the area of ISC and MRO connection and session terminal names. Version 5 allows the systems programmer to quickly analyse and tune the ‘width’ and the ‘breadth’ of the connections for the optimal traffic size. This eliminates some transaction queueing and also dramatically reduces the number of GETMAINs and FREEMAINs during function shipping.

BACKGROUND

Prior to CICS 4.1.0, entering a one- or two-character prefix on the session definitions for MRO connections was required. This prefix would be used as the first one or two bytes of the TCTTE name. In other words, it was the prefix of the name of the ‘terminal’ used for that session between the two CICS regions in question. CICS would come up with the rest of the name by using the following simple algorithm: start with the prefix and count up by 1 (in decimal). If the prefix was ‘A’ then the terminals could run from ‘A1’ to a maximum of ‘A999’.

If the prefix was ‘AA’ then they could run from ‘AA1’ to ‘AA99’ (the names are all four bytes; blanks are padded if necessary). For example, assume you had an FOR connected to an AOR with the following session definitions: SENDCount=4 , RECEIVECount=4, SENDPfx=F, and RECEIVEPfx=A. When looking at the TCTs in your FOR using your monitor or in the summary report from the SMF statistics (what everyone still calls ‘shutdown stats’), you would find eight terminals involved in the communication from this FOR to that AOR – A1 through A4 (Receives) and F1 through F4 (Sends).

There was only one problem with this refinement. The prefix picked could cause CICS to create an MRO ‘terminal’ name that was a duplicate of a real 3270 device. Installation, of course, would not occur and a problem would be created. Therefore you needed yet another naming convention to keep track of, as well as possibly having to add code to your terminal autoinstall exit in order to avoid such collisions.

With CICS 4.1.0, picking a send and receive prefix was no longer required. You could let the system default to a SENDPfx of ‘>’ and a RECEIVEPfx of ‘<’. The terminal naming algorithm also changed. It became bizarre.

The terminal names currently employed by CICS connections when you allow the default prefixes ‘>’ for SEND and ‘<’ for RECEIVE start with ‘>AAA’, then ‘>AAB’, then ‘>AAC’, etc, until the number of SEND sessions are used up. If you had a SENDCount of 4, then the first SEND session would be ‘>AAA’ and the last SEND session would be ‘>AAD’. The first RECEIVE session would be ‘<AAE’ followed by ‘<AAF’ until you finished the RECEIVEs. If you finished them and ended with ‘<AAK’, then the next connection installed would have as its first SEND session the terminal ‘>AAL’ and so on. The possibility of a duplicate terminal name is greatly reduced. Determining which terminals go where in a busy FOR that is connected to many AORs is difficult because the terminals are all listed alphabetically on the summary report, corresponding to the order in which the connections were installed. You must therefore determine the order in which the connections were installed in your region in order to identify to which AOR the terminals belong on the summary report.

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

DISCUSSION

IBM has enhanced the CICS SMF statistics with Transaction Server. Figure 1 shows the left-hand side of the terminal section of the summary report.

```
CICS 5.2.Ø Statistics Utility Program
Summary Report
_____________________________________________________
Terminal
_________
  Term          Terminal  Acc   Conn  …
  ID            LUname    Type  Meth   ID        …
_____________________________________________________

  <AAX           ISC      CONV  MRO    XAØ7
  <AAY           ISC      CONV  MRO    XAØ7
  <AAZ           ISC      CONV  MRO    XAØ7
  <AAØ           ISC      CONV  MRO    XAØ7
```

Figure 1: Terminal section (left-hand side) report

The new Connection ID column is a much-needed improvement. If your connection names have anything to do with your region names, and they probably do, you can easily determine which terminals go with which regions from this column.

This section of the report shows that the traffic flows on only a few terminals – because the Send sessions are not important to the FOR. Virtually all the traffic occurs on the Receive sessions. Similarly, all the traffic in the AOR occurs on the Send links. The reason is the SEND sessions are used for traffic originating in the local region. The RECEIVE sessions are used for traffic originating in the *other* region. (FORs don’t initiate work for AORs. It’s the other way around.)

The report also shows that the terminal traffic is not load-balanced. The links are bi-directional, which means a transaction will use the same session for multiple requests. The terminals are searched sequentially and the first one free in the list becomes the one used for this transaction. If the region is not very busy, you may find only the first terminal being used for everything.

It is important to have enough sessions defined. If your regions are very busy and you do not have enough sessions defined, you will see significant traffic on all the terminals for that connection and you will experience queueing for allocating sessions. These queueing delays can be avoided simply by increasing the RECEIVECount and SENDCount on the SESSION definition. You must, of course, select a count that is not so large as to let one AOR flood the FOR with too many requests during a peak period. (A count of 20 is adequate for busy regions.)

You should also balance the number of SEND AOR sessions with the number of RECEIVE FOR sessions. Having them out of sync is confusing. We standardized on 20 and 5 – an AOR has 20 SENDs and 5 RECEIVEs and an FOR has 20 RECEIVEs and 5 SENDs.

```
…
Connection name                          CONN1
…
Peak outstanding allocates                 100
Total number of allocates                10000
Average number of queued allocates         <<< THESE 2 LINES WILL
Failed link allocates                      250
Failed allocates due to sessions in use    <<<  BE NON-ZERO
….
```

Figure 2: Connection section report

Figure 2 describes the effect of an insufficient number of sessions. It is also taken from the summary report but appears in the Connection section. The two lines indicated will be non-zero, indicating transactions are waiting for a session so they can issue a request to the other region. This is not good.

Examine Figure 3, which is taken from the right-hand side of the terminal section of the summary report.

The input messages and output messages show the volume of function ships this region (the FOR) is doing with the region listed in the

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
 Storage    Input     Output   Xmission     Avg TIOA     Avg logged on
  Viols   Messages  Messages   Errors     Storage        Time

    0      1558597  1546723        0          5912
    0       169530   168089        0          5912
    0         5339     5225        0          5912
    0          131      123        0          5912
    0            0        0        0          5912
    0            0        0        0          5912
```

Figure 3: Terminal section (right-hand side) report

CONNID column off to the left in Figure 1 (the AOR). The number of messages on each successive terminal for the connection should fall to 0 (before all terminals are occupied) as in Figure 3, indicating that no transactions were queued up waiting for a session.

The 'Average TIOA Storage' gives you the average size that was required for each request. This is the key piece of data required for tuning the traffic. In the example in Figure 3, the average storage size required to ship the request to the other region was 5,912 bytes. If the session was defined to provide 8KB of storage to start with, this 8KB chunk would be GETMAINed and the data placed into it. If, however, the session was defined to provide a 2KB chunk of storage, a small problem would arise. The 2KB chunk would be obtained, the 5,192 bytes of data would not fit, the 2KB chunk would be FREEMAINed, a 5,912 byte chunk would be GETMAINed, and so forth.

Having enough storage at the outset avoids the unnecessary GETMAIN/FREEMAIN pair in the middle. Set the size to be larger than the Average TIOA Storage value in the report – larger by adding 24 bytes for CICS use and "round the total up to a multiple of 64 bytes… (to) ensure a good use of operating system pages" – (from the *Performance Guide*).

The RDO parameter for setting the initial storage size for MRO function ships is on the SESSION definition. It is IOAREALEN. The default value is 0. This forces CICS to interrogate the data length, and according to the *Performance Guide*, "…get a storage size exactly the size of the outgoing message, plus 24 bytes for CICS requirements". The *Guide* notes that if no value is specified, CICS will use a size of 4KB. (It seems that CICS adds the aforementioned 24 bytes to the 4KB and actually uses 4,120 bytes. This is a popular value seen in the summary reports.)

Two questions arise: are these IOAREALENs re-used, and where are they? They apparently are re-used and are above the line, according to research. Since they are reused, designating a large value for IOAREALEN is not very wasteful. The number of IOAREALENs matches the number of sessions defined for the connection and is not related to transaction volume. If your session count is a reasonable number, the storage consumed should be relatively small, and you can easily calculate the amount. For example, if you have 20 sessions and each of these has an IOAREALEN of 4KB and you want to increase it to 8KB, you will use an additional (20 * 4KB) or 80KB. This is an inconsequential amount of storage.

The average message transmitted on the MRO link in Figure 3 was 5,912. A quick check of our resource definition showed a 4,096 value coded. This was not quite large enough so the value was subsequently changed to 8KB.

The location of the IOARELENs is actually the SMTP subpool. This subpool is above the line and defined as that which "…holds line and terminal I/O areas" (*Performance Guide*). This can be seen readily in the summary report in the storage manager statistics section.

RESULTS

The summary report will show you the results of your handiwork. After you have increased the IOAREALEN to be larger than the Average TIOA Storage, here is where you look to see what happened (see Figure 4).

This busy FOR dropped the number of GETMAINs and FREEMAINs for the SMTP subpool by two-thirds – over 2 million. It's always nice to get measurable results. (Transaction volume and mix were roughly equivalent for the two weeks.)

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
STORAGE MANAGER STATISTICS
------------------------------------------------------------------
Subpool   Location  Access               Getmain          Freemain
Name                                     Requests         Requests
------------------------------------------------------------------
Before:
SMTP      ECDSA     CICS                 3,171,53Ø        3,171,525

After:    (a week later after raising the IOARELEN to 8KB)
SMTP      ECDSA     CICS                   914,744          914,744
```

Figure 4: Statistics

SUMMARY

Firstly, run the summary report for a typical day for your FOR(s) and AORs connected to them with the SYSIN parameters SELECT TYPE=(TERMINAL,STORAGE) and SUMMARY.

Secondly, increase or decrease the number of sessions with the SENDCount and RECEIVECount parms to eliminate queueing or to dump that extra 100 sessions you never use. Having the same value in the sessions from the AOR to the FORs as from the FOR to AORs makes the most sense and also seems safest. (It certainly will avoid some session management and negotiating between regions if one region is sending 20 requests out to a region that will accept only 10 at a time.)

Thirdly, look in the Terminal section of your reports for the Average TIOA Storage used for each region. Obviously, the FORs will have the most activity, so let them drive the process. Select a number that will meet on-going needs.

Fourthly, make your change to the IOAREALENs and run the report again. Look in the STORAGE MANAGER STATISTICS section. Do a find on 'SMTP' to determine whether it made a significant difference. You might be surprised!

Paul C Gordon
Assistant Vice President
Bank of America (USA)

© Xephon 2000

NEWCOPY of programs in an MRO environment – revisited

John Hall has sent in some new source members for his article NEWCOPY of programs in an MRO environment, *which was published in* CICS Update, *Issue 169, December 1999. Please note that MRONCOP2 is unchanged. Amendments to other programs are denoted by ‘01’ to the right of the amended line. The old code on our Web site has been updated to reflect the change.*

MROIDSYS

```
*ASM XOPTS(SP)
         TITLE ‘MROIDSYS  -  FIND NUMBER OF ATTACHED AORS’
         LCLC  &REL
&REL     SETC  ‘1.Ø’
         DFHREGS
COMMREG  EQU   4
SYSREG   EQU   6
         DFHEISTG
ACQUIRED EQU   C’A’
RELEASED EQU   C’R’
CONNECT  DS    F
PROTOCOL DS    F                                                     Ø1
CONACC   DS    F
STATUS   DS    F
SAVE14   DS    F
MROIDSYS CSECT
         B     START
         DC    C’MROIDSYS ‘
         DC    C’R: &REL  ‘
         DC    C’&SYSDATE ‘
         DC    C’&SYSTIME ‘
*----------------------------------*
*        Retrieve any COMMAREA                                         *
*----------------------------------*
START    DS    ØH
         OC    EIBCALEN,EIBCALEN       is there a COMMAREA ?
         BZ    RETURN                  no
         L     COMMREG,DFHEICAP        address of COMMAREA
         USING COMMDSCT,COMMREG
         BAL   R14,GETSYS
RETURN   DS    ØH
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
         EXEC  CICS RETURN
*-----------------------------------*
*        Subroutines                                          *
**                                                                   **
**---------GETSYS---------------------**
*        Get SYSIDs of all connected AORs                             *
*-----------------------------------*
GETSYS   EQU   *
         ST    R14,SAVE14
         LA    R2,1
**       get the SYSID of the region we are in                       **
         EXEC  CICS ASSIGN SYSID(SYSIDS)
         LA    SYSREG,SYSIDS+L'SYSIDS
         XC    CONACC,CONACC
         EXEC  CICS HANDLE CONDITION END(CONEND)
       EXEC  CICS INQUIRE CONNECTION START
CONLOOP  EXEC  CICS INQUIRE CONNECTION(CONNECT)                        +
               ACCESSMETHOD(CONACC)                                    +
               PROTOCOL(PROTOCOL)                                      +
               CONNSTATUS(STATUS)                                      +
               NEXT
*                                                                    Ø1
         CLC   PROTOCOL,DFHVALUE(EXCI)              do not include   Ø1
         BE    CONLOOP                              EXCI connections Ø1
         LA    R2,1(,R2)
         MVC   Ø(4,SYSREG),CONNECT                  save SYSID
         MVC   5(4,SYSREG),CONACC                   save access type Ø1
         MVI   4(SYSREG),ACQUIRED                   set system indic
         CLC   STATUS,DFHVALUE(ACQUIRED)            system available ?
         BE    CONLOOP1
         MVI   4(SYSREG),RELEASED                   it seems not
CONLOOP1 DS    ØH
         LA    SYSREG,L'SYSIDS(,SYSREG)
         B     CONLOOP
CONEND   EXEC  CICS INQUIRE CONNECTION END
         ST    R2,SYSCNT
         L     R14,SAVE14
         BR    R14
*-----------------------------------*
*        Record maps                                                  *
*-----------------------------------*
COMMDSCT DSECT
SYSCNT   DS    F
SYSIDS   DS    CL9                                                   Ø1
         END
```

MRONCOP1

```
*ASM XOPTS(SP)
```

```
         TITLE ‘MRONCOP1  -  MRO NEWCOPY RETURN MESSAGES’
         LCLC  &REL
&REL     SETC  ‘1.Ø’
         DFHREGS
COMMREG  EQU   5
SYSREG   EQU   6
         DFHEISTG
ACQUIRED EQU   C’A’
RELEASED EQU   C’R’
************************
SYSCNT   DS    F
SYSIDS   DS    5ØCL9             max number of SYSIDS = (49 + 1)    Ø1
************************
SAVE14   DS    F
CRES     DS    F
LENF     DS    H
ITNUM    DS    H
************************
TQNAME   DS    CL8
TSREC    DS   ØCL(12+MSSGL)
TSSYSID  DS    CL4
TSPGMID  DS    CL8
TSMSG    DS    CL(MSSGL)
************************
CMAREA   DS   ØCL12
CMPGMID  DS    CL8
CMSCLINE DS    FL4
************************
STRTREC  DS   ØCL12
STRTSYS  DS    CL4
STRTTSQ  DS    CL8
STRTRECL EQU   *-STRTREC
************************
MSGLINE  DS    CL35
         COPY  DFHAID                         AID key definitions
         COPY  DFHBMSCA                       BMS attribute definitions
*----------------------------------*
*        Screen Map                                                   *
*----------------------------------*
         COPY  NEWCOPY
NEXTSYS  EQU   (LINEØØ5L-LINEØØ4L)            length of detail line
MSSGL    EQU   L’MSGARØ4O
MAXSYS   EQU   14              maximum number of SYSID lines on screen
MRONCOP1 CSECT
         B     START
         DC    C’MRONCOP1 ‘
         DC    C’R: &REL  ‘
         DC    C’&SYSDATE ‘
         DC    C’&SYSTIME ‘
```

```
*----------------------------------*
*        Program flow                                                *
*----------------------------------*
START    DS    ØH
         MVC   TQNAME(4),EIBTRMID              set up TS queue name
         MVC   TQNAME+4(4),=C’REMQ’
         MVC   STRTTSQ(8),TQNAME                  QNAME passed to AOR
*----------------------------------*
*        AIDs : ENTER  refresh screen                                *
*               PF3    return to first screen                        *
*               PF4    return                                        *
*               PF8    scroll forward                                *
*----------------------------------*
         EXEC  CICS HANDLE AID                                        +
                               PF3 (GOBACK)                           +
                               PF4 (RETURN)                           +
                               PF8                                    +
                             ENTER                                    +
                            ANYKEY (INVKEY)
*
         EXEC  CICS IGNORE CONDITION MAPFAIL
         EXEC  CICS RECEIVE MAP(‘NEWCOPY’) MAPSET(‘NEWCOPY’) ASIS
*----------------------------------*
*        If returning from screen send, we have COMMAREA             *
*----------------------------------*
         OC    EIBCALEN,EIBCALEN                  is there a COMMAREA ?
         BZ    NOCOMM                             no - first time thru
         L     COMMREG,DFHEICAP                   address of COMMAREA
         MVC   CMAREA(L’CMAREA),Ø(COMMREG)        restore COMMAREA
NOCOMM   DS    ØH
         BAL   R14,GETSYS                         fill in system-ids
         OC    CMPGMID,CMPGMID                    have we a program-id ?
         BZ    GETPGMID                           no
         MVC   CMDINPO(L’CMDINPO),CMPGMID         restore screen prog-id
         MVI   CMDINPA,DFHBMPRO                 protect program-id field
         BAL   R14,READTSQ                        read message queues
         B     SENDMAP
GETPGMID DS    ØH
         CLI   EIBAID,DFHPF8                      are we scrolling ?
         BE    SENDMAP                            go to next screen
         OC    CMDINPI,CMDINPI                    PGMID entered ?
         BZ    NOINPUT                            tell them if not
         MVI   CMDINPA,DFHBMPRO                   protect program id
         MVC   CMPGMID,CMDINPI                    and store it
         BAL   R14,STRTALL                        start NEWCOPY tasks
         XC    CMSCLINE,CMSCLINE                  reset scroll to zero
*----------------------------------*
*        Send map and return here                                    *
*----------------------------------*
```

```
SENDMAP  DS    ØH
         EXEC  CICS SEND MAP(‘NEWCOPY’) MAPSET(‘NEWCOPY’)              +
               FROM(NEWCOPYS)                                          +
               LENGTH(=AL2(NEWCOPYL))
*
         EXEC  CICS RETURN TRANSID(EIBTRNID)                           +
               COMMAREA(CMAREA) LENGTH(12)
*----------------------------------*
*        Error conditions                                             *
*----------------------------------*
NOINPUT  DS    ØH
         MVI   MSGLINE,X’4Ø’
         MVC   MSGLINE+1(L’MSGLINE-1),MSGLINE
         MVC   MSGLINE(L’NOPROG),NOPROG
         B     GOBACK
*
INVKEY   DS    ØH
         MVI   MSGLINE,X’4Ø’
         MVC   MSGLINE+1(L’MSGLINE-1),MSGLINE
         MVC   MSGLINE(L’IKMSG),IKMSG
         B     GOBACK
*----------------------------------*
*        Go back to initial screen with optional message              *
*----------------------------------*
GOBACK   DS    ØH
         EXEC  CICS START TRANSID(‘NCOP’)                              +
               FROM(MSGLINE) LENGTH(35)                                +
               TERMID(EIBTRMID)
*
         EXEC  CICS RETURN
*----------------------------------*
*        Return                                                       *
*----------------------------------*
RETURN   DS    ØH
         EXEC  CICS SEND CONTROL ERASE FREEKB .                      Ø1
         EXEC  CICS RETURN
*----------------------------------*
*        SUBROUTINES                                                  *
*----------------------------------*
**                                                                   **
**------------ GETSYS -----------------**
*        Find attached systems                                        *
*----------------------------------*
GETSYS   DS    ØH
         ST    R14,SAVE14
         LA    RØ,(4+5Ø*5)
         STH   RØ,LENF
         EXEC  CICS LINK PROGRAM(‘MROIDSYS’)                           +
               COMMAREA(SYSCNT)                                        +
               LENGTH(LENF)
         L     R2,SYSCNT                       number of MRO systems
         L     R14,CMSCLINE                    current start line
         CLI   EIBAID,DFHPF8                   are we scrolling ?
         BNE   GETSYS1
         LA    R14,MAXSYS(R14)                 add 1 pageful
         ST    R14,CMSCLINE                    save start line
         SR    R2,R14                   is there a next page to go to?
         BP    GETSYS1                         yes - so go scroll
         L     R2,SYSCNT                       get number of systems
         XC    CMSCLINE,CMSCLINE            reset scroll amount to zero
GETSYS1  DS    ØH
         L     R14,CMSCLINE                    get current start line
         MH    R14,=Y(L’SYSIDS)                get disp into SYSIDS
         LA    SYSREG,SYSIDS(R14)
         LA    R7,SYSIDØ4O                  point to map start field
         CH    R2,=Y(MAXSYS)                   do not exceed screen
         BNH   SYSLOOP                         limits
         LA    R2,MAXSYS
SYSLOOP  DS    ØH
         MVC   Ø(4,R7),Ø(SYSREG)               complete SYSID
         MVI   TSMSG,X’4Ø’                     clear message area
         MVC   TSMSG+1(MSSGL-1),TSMSG
         CLI   4(SYSREG),RELEASED           is this system available?
         BNE   STARTASK
         MVC   TSMSG(L’NOTAVBL),NOTAVBL     no - send unavailable msg
STARTASK DS    ØH
         MVC   (MSGARØ4O-SYSIDØ4O)(MSSGL,R7),TSMSG      display message
         LA    SYSREG,L’SYSIDS(,SYSREG)        increment SYSID
         LA    R7,NEXTSYS(,R7)                 increment map pointer
         BCT   R2,SYSLOOP                      loop until last AOR
         L     R14,SAVE14
         BR    R14
**                                                                   **
**------------ STRTALL ----------------**
*        Start tasks to refresh program in all regions                *
*----------------------------------*
STRTALL  DS    ØH
         ST    R14,SAVE14
         L     R2,SYSCNT                       get number of systems
         LA    SYSREG,SYSIDS                   get first SYSID
         LA    R5,1                            set up counter reg
         LA    R7,SYSIDØ4O                     point to map start field
         MVC   STRTSYS,SYSIDS
STLOOP   DS    ØH
         XC    TSREC,TSREC                     clear TS rec
         CLI   4(SYSREG),RELEASED            is this system available?
         BE    STRTEND                         no further action
         MVC   TSMSG(L’REFMSG),REFMSG          set up default message
         MVC   TSSYSID(4),Ø(SYSREG)            move in this SYSID
         MVC   TSPGMID(8),CMPGMID              move in program name
**    Start transaction                                              **
         EXEC  CICS START TRANSID(‘NCO2’)                              +
               INTERVAL(2)                                             +
               FROM(STRTREC)                                           +
               LENGTH(=Y(STRTRECL))                                    +
               SYSID(Ø(SYSREG))                                        +
               RESP(CRES)
*                                                                    Ø1
         CLC   CRES,DFHRESP(NORMAL)                                  Ø1
         BE    WRITETS                                               Ø1
         MVC   TSMSG(L’FAILMSG),FAILMSG        set up failed message Ø1
WRITETS  DS    ØH                                                    Ø1
**    Delete TS queue                                                **
         EXEC  CICS DELETEQ TS QUEUE(TQNAME)                           +
               SYSID(Ø(SYSREG))                                        +
               RESP(CRES)
**    Write new TS queue                                             **
         EXEC  CICS WRITEQ TS QUEUE(TQNAME) FROM(TSREC)                +
               LENGTH(7Ø)                                              +
               SYSID(Ø(SYSREG))                                        +
               RESP(CRES)
STRTEND  DS    ØH
         LA    R5,1(,R5)                       increment count
         LA    SYSREG,L’SYSIDS(,SYSREG)        next SYSID
         LA    R7,NEXTSYS(,R7)                 next message line
         BCT   R2,STLOOP                       process next record
         L     R14,SAVE14
         BR    R14
**                                                                   **
**------------ READTSQ -----------------**
*        Read TS queues and send messages to screen                   *
*----------------------------------*
READTSQ  DS    ØH
         ST    R14,SAVE14
         LA    RØ,1
         STH   RØ,ITNUM                        get first item
         L     R2,SYSCNT                       get number of systems
         XR    R14,R14
         CLI   EIBAID,DFHPF8                   are we scrolling ?
         BNE   NOSCRL
         L     R14,CMSCLINE                    get current start line
         SR    R2,R14
         MH    R14,=Y(L’SYSIDS)                get disp into SYSIDS
NOSCRL   DS    ØH
         LA    SYSREG,SYSIDS(R14)
         LA    R7,SYSIDØ4O                     point to map start field
         CH    R2,=Y(MAXSYS)                   do not exceed screen
         BNH   READLOOP                          limits
         LA    R2,MAXSYS
READLOOP DS    ØH
         CLI   4(SYSREG),RELEASED           is this system available?
         BNE   STRTREAD
         MVC   TSMSG(L’NOTAVBL),NOTAVBL        send unavailable msg
         B     READEND                         no further action
STRTREAD DS    ØH
         XC    TSREC,TSREC                     clear TS rec
         EXEC  CICS READQ TS QUEUE(TQNAME)                             +
               ITEM(ITNUM)                                             +
               INTO(TSREC)                                             +
               SYSID(Ø(SYSREG))                                        +
               RESP(CRES)
READEND  DS    ØH
         MVC   (MSGARØ4O-SYSIDØ4O)(MSSGL,R7),TSMSG    display message
         LA    SYSREG,L’SYSIDS(,SYSREG)          next SYSID
         LA    R7,NEXTSYS(,R7)                   next message line
         BCT   R2,READLOOP                       process next record
         L     R14,SAVE14
         BR    R14
*----------------------------------*
*        CONSTANTS                                                    *
*----------------------------------*
NOPROG   DC    C’Enter PROGRAM to NEWCOPY’
IKMSG    DC    C’Function not available  ‘
NOTAVBL  DC    C’ — SYSTEM NOT AVAILABLE    — ‘
REFMSG   DC    C’ — NEWCOPY COMMAND ISSUED  — ‘
FAILMSG  DC    C’ — ISSUE NEWCOPY FAILED    — ‘                    Ø1
         END
```

MRONCOPY

```
*ASM XOPTS(SP)
         TITLE ‘MRONCOPY  -  MRO NEWCOPY INITIAL SCREEN SEND’
         LCLC  &REL
&REL     SETC  ‘1.Ø’
         DFHREGS
SYSREG   EQU   6
         DFHEISTG
ACQUIRED EQU   C’A’
RELEASED EQU   C’R’
************************
SYSCNT   DS    F
SYSIDS   DS    5ØCL9             max number of SYSIDS = (49 + 1)    Ø1
************************
SAVE14   DS    F
CRES     DS    F
```

```
LENF     DS    H
TSMSG    DS    CL(MSSGL)
CMPGMID  DS    CL8
QNAME    DS    CL8
         COPY  DFHAID                 AID key definitions
         COPY  DFHBMSCA               BMS attribute definitions
*----------------------------------*
*        Screen Map                                                   *
*----------------------------------*
         COPY  NEWCOPY
NEXTSYS  EQU   (LINEØØ5L-LINEØØ4L)    length of detail line
MSSGL    EQU   L’MSGARØ4O
MAXSYS   EQU   14               maximum number of SYSID lines on screen
MRONCOPY CSECT
         B     START
         DC    C’MRONCOPY ‘
         DC    C’R: &REL  ‘
         DC    C’&SYSDATE ‘
         DC    C’&SYSTIME ‘
*----------------------------------*
*        Initialization                                               *
*        Set up screen area                                           *
*----------------------------------*
START    DS    ØH
         MVC   QNAME(4),EIBTRMID
         MVC   QNAME+4(4),=C’NCPY’
         EXEC  CICS READQ TS QUEUE(QNAME) ITEM(1)                      +
               INTO(CMPGMID)                                           +
               RESP(CRES)
         BAL   R14,GETSYS
*----------------------------------*
*        RETRIEVE and display any messages                            *
*----------------------------------*
         LA    RØ,L’MSGLINEO
         STH   RØ,LENF
         EXEC  CICS RETRIEVE INTO(MSGLINEO) LENGTH(LENF)               +
               RESP(CRES)
*----------------------------------*
*        Delete any old TS queue and                                  *
*        SEND ERASE on first screen                                   *
*----------------------------------*
SEND1ST  DS    ØH
         EXEC  CICS DELETEQ QUEUE(QNAME)                               +
               RESP(CRES)
         EXEC  CICS SEND MAP(‘NEWCOPY’) MAPSET(‘NEWCOPY’)              +
               WAIT                                                    +
               ERASE                                                   +
               ALARM
*----------------------------------*
```

```
*        RETURN                                                      *
*----------------------------------*
RETURN   EXEC  CICS RETURN TRANSID(NEXTRAN)
*----------------------------------*
*        Subroutines                                                 *
*----------------------------------*
**                                                                  **
**----------- GETSYS -----------------**
*        Find attached systems                                       *
*----------------------------------*
GETSYS   DS    ØH
         ST    R14,SAVE14
         LA    RØ,(4+5Ø*5)
         STH   RØ,LENF
         EXEC  CICS LINK PROGRAM('MROIDSYS')                           +
               COMMAREA(SYSCNT)                                        +
               LENGTH(LENF)
         L     R2,SYSCNT                       number of MRO systems
         LA    SYSREG,SYSIDS                   point to first SYSID
         LA    R7,SYSIDØ4O                     point to map start field
         CH    R2,=Y(MAXSYS)                   do not exceed screen
         BNH   SYSLOOP                         limits
         LA    R2,MAXSYS
SYSLOOP  DS    ØH
         MVC   Ø(4,R7),Ø(SYSREG)               complete SYSID
         MVI   TSMSG,X'4Ø'                     clear message area
         MVC   TSMSG+1(MSSGL-1),TSMSG
         LA    R15,ACMETHTB                  point to access meths   Ø1
         LA    R14,ACTABNO                   & get no. of entries    Ø1
GETSYS1Ø DS    ØH                                                    Ø1
         CLC   5(4,SYSREG),8(R15)            ACCMETH found ?         Ø1
         BE    GETSYS2Ø                      yes, complete desc      Ø1
         LA    R15,ACMLENF(,R15)             point to next entry     Ø1
         BCT   R14,GETSYS1Ø                  go back for more        Ø1
GETSYS2Ø DS    ØH                                                    Ø1
         MVC   (ACTYPØ4O-SYSIDØ4O)(5,R7),Ø(R15)   display access methØ1
         CLI   4(SYSREG),RELEASED            is this system available?
         BNE   STARTASK
         MVC   TSMSG(L'NOTAVBL),NOTAVBL      no - send unavailable msg
STARTASK DS    ØH
         MVC   (MSGARØ4O-SYSIDØ4O)(MSSGL,R7),TSMSG     display message
         LA    SYSREG,L'SYSIDS(,SYSREG)        increment SYSID
         LA    R7,NEXTSYS(,R7)                 increment map pointer
         BCT   R2,SYSLOOP                      loop until last AOR
         L     R14,SAVE14
         BR    R14
*----------------------------------*
*        CONSTANTS                                                   *
*----------------------------------*
```

```
NEXTRAN  DC    C’NC01'
NOTAVBL  DC    C’ — SYSTEM NOT AVAILABLE    — ‘
ACMETHTB DS    ØH                                                    Ø1
         DC    CL8’VTAM    ‘,FL4’6Ø’                                 Ø1
ACMLENF  EQU   *-ACMETHTB                    length of table entry   Ø1
         DC    CL8’IRC     ‘,FL4’121'                                Ø1
         DC    CL8’INDIRECT’,FL4’122'                                Ø1
         DC    CL8’XM      ‘,FL4’123'                                Ø1
         DC    CL8’XCF     ‘,FL4’665'                                Ø1
         DC    CL8'        ‘,FL4’Ø’                                  Ø1
ACTABNO  EQU   (*-ACMETHTB)/ACMLENF          num of table entries    Ø1
         END
```

John Hall
CICS Systems Programmer
Cooperative Insurance Society (UK)

© J D Hall 2000

CICS/TS 1.3 NEWCOPY facility for DOCTEMPLATES – part 2

This month we conclude the code for creating document templates defined to CICS in a DOCTEMPLATE resource definition.

PROGRAM XEPNTEM4

```
*******************************************
* MODULE NAME      XEPNTEM4.COB
* DOES DISCARD ETC
* INVOKED BY LINK
*****************************************
 IDENTIFICATION DIVISION.
 PROGRAM-ID. XEPNTEM4.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 Ø1 XEP-ATTRIBUTES.
    Ø3  XEP-DOC-1 PIC X(13) VALUE IS 'TEMPLATENAME('.
    Ø3  XEP-DOC-2 PIC X(Ø7) VALUE IS 'DDNAME('.
    Ø3  XEP-DOC-3 PIC X(11) VALUE IS 'APPENDCRLF('.
    Ø3  XEP-DOC-4 PIC X(11) VALUE IS 'MEMBERNAME('.
    Ø3  XEP-DOC-5 PIC X(12) VALUE IS 'DESCRIPTION('.
    Ø3  XEP-DOC-6 PIC X(Ø5) VALUE IS 'TYPE('.
 Ø1 XEP-build.
    Ø2 XEP-ATTRIBUTES-RESULT PIC X(161) value spaces.
    Ø2 filler redefines XEP-attributes-result.
       Ø3 toChar pic x occurs 161 times
                 indexed by toIndex.
    Ø2 XEP-newstring pic x(58).
    Ø2 filler redefines XEP-newstring.
       Ø3 fromChar pic x occurs 58 times
                 indexed by fromIndex, lastIndex.
    Ø2 XEP-attrib-len   pic s9(4) comp value Ø.
 LINKAGE SECTION.
 Ø1 DFHCOMMAREA.
    Ø3  LS-DOC-NM PIC x(8).
    Ø3  LS-DOC-TN PIC X(48).
    Ø3  LS-DOC-CR PIC X(3).
    Ø3  LS-DOC-DD PIC X(8).
    Ø3  LS-DOC-MN PIC X(8).
    Ø3  LS-DOC-DS PIC X(58).
    Ø3  LS-DOC-TY PIC X(Ø6).
    Ø3  LS-RESP   PIC S9(8) COMP.

 PROCEDURE DIVISION.

 AA-MAIN SECTION.

     set toIndex to 1
     MOVE XEP-DOC-1 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-tn TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE XEP-DOC-2 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-DD TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE XEP-DOC-3 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-CR TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE XEP-DOC-4 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-MN TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE XEP-DOC-5 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-DS TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE XEP-DOC-6 TO XEP-NEWSTRING
     PERFORM APPEND
     MOVE LS-DOC-TY TO XEP-NEWSTRING
```

```
     PERFORM APPEND
     perform varying XEP-attrib-len from 161 by -1
             until toChar(XEP-attrib-len) = ")"
     end-perform
* It doesn’t matter if discard fails, it just means that the
* doctemplate was not previously installed
     EXEC CICS DISCARD DOCTEMPLATE(LS-DOC-NM)
               NOHANDLE
     END-EXEC
     EXEC CICS CREATE DOCTEMPLATE(LS-DOC-NM)
               ATTRIBUTES(XEP-ATTRIBUTES-RESULT)
               ATTRLEN(XEP-ATTRIB-LEN)
               RESP(LS-RESP)
     END-EXEC
     EXEC CICS RETURN END-EXEC.
 append.
     perform varying lastIndex from 58 by -1 until
             fromChar(lastIndex) not equal space
             or
             lastIndex = 1
     end-perform
     perform varying fromIndex from 1 by 1 until
             fromIndex > lastIndex
             or
             toIndex >= 161
             move fromChar(fromIndex) to toChar(toINdex)
             set toIndex up by 1
     end-perform
         if (toIndex > 1)
            set toIndex down by 1
            if (toChar(toIndex) not = "(" )
               set toIndex up by 1
               move ") " to toChar(toIndex)
               set toIndex up by 2
            else
               set toIndex up by 1
               end-if
         end-if.
 AA999-EXIT.
     EXIT.
                    STOP RUN.
```

HTML TEMPLATES

TEMPLATE XEPNTEM

```
<!doctype html public "-//IEFT//DTD HTML 3.2//EN">
 <html>
 <head>
 <SCRIPT LANGUAGE-"JavaScript">
 // dfhsetcursor function
 // sets focus to the first input field
 function dfhsetcursor(n)
   {for (var i=0;i<document.XEPNTEM.elements.length;i++)
     {if (document.XEPNTEM.elements[i].name == n)
         {document.XEPNTEM.elements[i].focus();
          document.XEPNTEM.DFH_CURSOR.value=n;
          break}}}
 // dfhinqcursor function
 function dfhinqcursor(n) {
    document.XEPNTEM.DFH_CURSOR.value=n}
 // checkInput function
 // called when the user submits the form
 function checkInput(checkIt) {
   if (anyChar(checkIt.docnm))
      checkIt.docnm.value = prompt("Enter doctemplate");
      else return true;
      return false;
  }
 // anyChar function
 // checks that entererd data is alpha or extra characters
 function anyChar(tObj) {
   var extraChars=". -,1234567890/"
   if (tObj.value.length == 0) return true;
   for(var i=0;i<tObj.value.length; i++){
      var ch = tObj.value.charAt(i);
      ch = ch.toUpperCase();
      search = extraChars.indexOf(ch);
      if (search == -1 && (ch < 'A' || ch > 'Z' ))
      return true;
   }
   return false;
 }
 </SCRIPT>
<title> CICS/TS 1.3 Newcopy facility</title>
</head>
<body  bgcolor="#90C5B0"
onLoad="dfhsetcursor('docnm')"
link="#0000FF" vlink="#800080" alink="#FF0000">
<BASE href="http://&hostv;">
<form name="XEPNTEM"
 onSubmit="return checkInput(document.XEPNTEM)"
 action="/CICS/CWBA/XEPNTEM2"
 method="post">
<input type="hidden" name="DFH_CURSOR" value="docnm">
<CENTER>
<h1>Newcopy Facility </H1>
```

```
<h2>CICS/TS 1.3 Doctemplate </H2>
<p>
<h2 align=center>Please enter a doctemplate name  </h2>
<p>
<table colspan=2 bgcolor="#B7D9CB" border="2">
<CENTER>
<tr>
<td colspan=2 nowrap> Doctemplate  </td>
<TD colspan=2 nowrap>
<input type="text" name="docnm" size="11" maxlength="8" value="&docnm;"
  onFocus="dfhinqcursor('docnm');this.select()"
  title="enter the doc template name">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Action     </td>
<TD colspan=2 nowrap>
<input type="radio" name="act" value="Create" CHECKED
  onFocus="dfhinqcursor('act')" title="select an action">Create
</TD>
</tr>
</table>
<p>
<input type="submit" name="submit" value="Process request"
onFocus="dfhinqcursor('rdupd')" title="press this button to process
request">
</CENTER>
</form>
</body>
</html>
```

TEMPLATE XEPNTEM2

```
<!doctype html public "-//IEFT//DTD HTML 3.2//EN">
 <html>
 <head>
 <SCRIPT LANGUAGE-"JavaScript">
 // dfhsetcursor function
 function dfhsetcursor(n)
   {for (var i=0;i<document.XEPNTEM2.elements.length;i++)
     {if (document.XEPNTEM2.elements[i].name == n)
         {document.XEPNTEM2.elements[i].focus();
          document.XEPNTEM2.DFH_CURSOR.value=n;
          break}}}
 // dfhinqcursor function
 function dfhinqcursor(n) {
    document.XEPNTEM2.DFH_CURSOR.value=n}
 // checkData function
 // called when the user submits the form
 function checkData(fObj) {
   if (anyChar(fObj.docnm))
      fObj.docnm.value = prompt("Enter doctemplate:");
      else return true;
      return false;
  }
 // anyChar function
 // checks that entererd data is alpha or extra characters
 function anyChar(tObj) {
   var extraChars=". -,1234567890/"
   if (tObj.value.length == 0) return true;
   for(var i=0;i<tObj.value.length; i++){
      var ch = tObj.value.charAt(i);
      ch = ch.toUpperCase();
      search = extraChars.indexOf(ch);
      if (search == -1 && (ch < 'A' || ch > 'Z' ))
      return true;
   }
   return false;
 }
 </SCRIPT>
<title> CICS/TS 1.3 Newcopy utility</title>
</head>
<body  bgcolor="#90C5B0"
onLoad="dfhsetcursor('docnm')"
link="#0000FF" vlink="#800080" alink="#FF0000">
<BASE href="http://&hostv;">
<form name="XEPNTEM2"
 onSubmit="return checkData(document.XEPNTEM2)"
 action="/CICS/CWBA/XEPNTEM3"
 method="post">
<input type="hidden" name="DFH_CURSOR" value="docnm">
<CENTER>
<h1 ALIGN=CENTER>Newcopy utility </H1>
<h2 ALIGN=CENTER>CICS/TS 1.3 Doctemplate </H2>
<p>
<h2 align=center>Please enter a doctemplate name  </h2>
<p>
<CENTER>
<table colspan=2 bgcolor="#B7D9CB" border="2">
<tr>
<td colspan=2 nowrap> Doctemplate  </td>
<TD colspan=2 nowrap>
<input type="text" name="docnm" size="11" maxlength="8" value="&docnm;"
  onFocus="dfhinqcursor('docnm');this.select()"
  title="enter the doc template name">
</TD>
</tr>
<tr>
```

```
<td colspan=2 nowrap> Template name</td>
<TD colspan=2 nowrap>
<input type="text" name="doctn" size="11" maxlength="48" value="&doctn;"
  onFocus="dfhinqcursor('doctn');this.select()"
  title="enter 48 char template name">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Append CRLF  </td>
<TD colspan=2 nowrap>
<select name="doccr" size="1">
<option VALUE="YES" SELECTED>YES</option>
<option VALUE="NO">NO</option>
</TD>
</tr>
<tr>
<td colspan=2 nowrap> DD Name      </td>
<TD colspan=2 nowrap>
<input type="text" name="docdd" size="11" maxlength="8" value="&docdd;"
  onFocus="dfhinqcursor('docdd');this.select()"
  title="enter PDS name">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Member name  </td>
<TD colspan=2 nowrap>
<input type="text" name="docmn" size="11" maxlength="8" value="&docmn;"
  onFocus="dfhinqcursor('docmn');this.select()"
  title="enter PDS member name">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Description  </td>
<TD colspan=2 nowrap>
<input type="text" name="docds" size="20" maxlength="58" value="&docds;"
  onFocus="dfhinqcursor('docds');this.select()"
  title="enter description">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Type         </td>
<TD colspan=2 nowrap>
<select name="docty" size="1">
<option VALUE="EBCDIC" SELECTED>EBCDIC</option>
<option VALUE="BINARY">BINARY</option>
</TD>
</tr>
</table>
<p>
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
<input type="submit" name="submit" value="Process request"
onFocus="dfhinqcursor('rdupd')" title="press this button to process
request">
</CENTER>
</form>
</body>
</html>
```

TEMPLATE XEPNTEM3

```
<!doctype html public "-//IEFT//DTD HTML 3.2//EN">
 <html>
 <head>
 <SCRIPT LANGUAGE-"JavaScript">
 // dfhsetcursor function
 function dfhsetcursor(n)
   {for (var i=0;i<document.XEPNTEM3.elements.length;i++)
     {if (document.XEPNTEM3.elements[i].name == n)
         {document.XEPNTEM3.elements[i].focus();
          document.XEPNTEM3.DFH_CURSOR.value=n;
          break}}}
 // dfhinqcursor function
 function dfhinqcursor(n) {
    document.XEPNTEM3.DFH_CURSOR.value=n}
 // checkIt function
 // called when the user submits the form
 function checkIt(fObj) {
   if (anyChar(fObj.docnm))
      fObj.docnm.value = prompt("Enter doctemplate");
      else return true;
      return false;
  }
 // anyChar function
 // checks that entererd data is alpha or extra characters
 function anyChar(tObj) {
   var extraChars=". -,1234567890/"
   if (tObj.value.length == 0) return true;
   for(var i=0;i<tObj.value.length; i++){
      var ch = tObj.value.charAt(i);
      ch = ch.toUpperCase();
      search = extraChars.indexOf(ch);
      if (search == -1 && (ch < 'A' || ch > 'Z' ))
      return true;
   }
   return false;
 }
 </SCRIPT>
<title> CICS/TS 1.3 Newcopy utility</title>
</head>
```

```
<body  bgcolor="#90C5B0"
onLoad="dfhsetcursor('docnm')"
link="#0000FF" vlink="#800080" alink="#FF0000">
<BASE href="http://&hostv;">
<form name="XEPNTEM3"
 onSubmit="return checkIt(document.XEPNTEM3)"
 action="/CICS/CWBA/XEPNTEM2"
 method="post">
<input type="hidden" name="DFH_CURSOR" value="docnm">
<CENTER>
<h1 ALIGN=CENTER>Newcopy Utility </H1>
<h2 ALIGN=CENTER>CICS/TS 1.3 DOCTEMPLATE </H2>
<p>
<h2 align=center>Please enter a doctemplate name  </h2>
<p>
<CENTER>
<table colspan=2 bgcolor="#B7D9CB" border="2">
<tr>
<td colspan=2 nowrap> Doctemplate  </td>
<TD colspan=2 nowrap>
<input type="text" name="docnm" size="11" maxlength="8" value="&docnm;"
  onFocus="dfhinqcursor('docnm');this.select()"
  title="enter the doc template name">
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Response     </td>
<TD colspan=2 nowrap>&resp;
</TD>
</tr>
<tr>
<td colspan=2 nowrap> Action     </td>
<TD colspan=2 nowrap>
<input type="radio" name="act" value="Create" CHECKED
  onFocus="dfhinqcursor('act')" title="select an action">Create
</TD>
</tr>
</table>
<p>
<input type="submit" name="submit" value="Process request"
onFocus="dfhinqcursor('rdupd')" title="press this button to process
request">
</CENTER>
</form>
</body>
</html>
```

David Clancy
Circle Computer Group (UK)

© Circle Computer Group 2000

Getting rid of null-use resources

A common management problem in most CICS systems is the sheer volume of obsolete CSD resources that remain in the system year after year because no-one dares to remove them. Not only do they increase (cold) start times and complicate CSD management, but they needlessly swell migration inventories for Year 2000 conversion or new CICS releases. In order to avoid removing resources that are used only occasionally (eg once a month), a systematic approach is required. The following system allows us to regularly list ‘null-use’ CSD resources, and the period for which they have not been used. We can then confidently approach application teams with hard evidence in order to have the null-use resources removed from the CSD file (and load libraries).

The key to the system is a VSAM KSDS (NULLFILE), which has the following format:

```
Key :
Resource Type      (FILE or TRAN etc)              PIC X(4).
Filler             (always LOW-VALUE)              PIC X.
Resource Name      (right justified with blanks)   PIC X(8).

Data :
Count  (times record was updated)                  PIC S9(8) COMP.
Filler                                             PIC X.
First Date  (first date resource was not used)     PIC X(10)
Filler                                             PIC X.
Last Date (last date resource was not used)        PIC X(10).
```

The FILLER fields are not strictly necessary and are there only to allow easy scanning using VSAM file browsing tools such as FileAid. NULLFILE is updated by a nightly batch job (STATJOB) which scans CICS statistics for null-use resources. How these statistics are obtained will vary from site to site. We record our CICS statistics by writing the DFHSTUP statistics reports to a GDG for each CICS system. To get the null-use statistics, we first execute REXX procedure STATEXTR, which is invoked from TSO/ISPF batch as follows (eg):

```
    ISPSTART CMD (%STATEXTR CICS.A*.DFHSTUP TRAN)
```

This will extract null-use TRANSACTION resources for all AORs

© 2000. Xephon UK telephone 01635 33848, fax 01635 38345. USA telephone (303) 410 9344, fax (303) 410 0290.

(assuming we have the naming convention CICS.A*nn*.DFHSTUP for AORS).

STEP ONE

STATEXTR first invokes the ISPF LMDINIT service to create a list of all DFHSTUP report DSNs for the relevant GDG template and their creation dates. The dates are in the format yyyy/mm/dd. Since LMDINIT will return a list sorted by name and creation date, the earliest date will be the first entry and the latest date the last entry, irrespective of the number of CICS systems, provided that all the generations are created concurrently. The earliest and latest creation dates and the total number of generations are saved to a parameter file (PARMFILE). Then each DSN in the list is ALLOCated in turn to the DFHSTUP DDname and control is passed to COBOL program STATPRG1. STATPRG1 scans DFHSTUP for the selected null-use resource(s) and writes these to a work file (WORK1).

Unfortunately, the scan logic relies on hard-coded displacements and contents of various fields and headers within the DFHSTUP report and must be revised for a new CICS release. All such fields are coded in WORKING STORAGE tables for ease of maintenance. An edit macro could have reduced reliance on hard-coded displacements, but this would have drastically increased run times. To reduce the size of NULLFILE (and run times), exception tables have been coded so that non-application null-use resources can be omitted (eg programs DFH*, transactions C*, etc).

STEP TWO

After STATEXTR has successfully executed, WORK1 is sorted by resource type and resource name to work file SORTFILE.

STEP THREE

COBOL program STATPRG2 is then invoked to update NULLFILE. STATPRG2 reads PARMFILE to get the total number of generations and the first and last creation dates (FRSTDATE and LASTDATE). It then reads SORTFILE. If the number of entries for a particular

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

resource type and resource name is equal to the total number of generations, we know that this resource has not been used in any of the reported CICS systems from FRSTDATE to LASTDATE. It is therefore recorded to NULLFILE.

If the record already exists, only the LASTDATE and count fields are updated – the FRSTDATE field is not changed.

The logic above assumes that the DFHSTUP reports for all CICS are created at the same time.

When STATPRG2 has finished processing SORTFILE, it updates the LASTDATE field of the NULLFILE control record (key = LOW-VALUES). This field is updated if, and only if, the new LASTDATE is greater than the current control record LASTDATE. The FRSTDATE field of the control record is updated (only) by the very first run of STATPRG2. STATJOB need not be run nightly, but must be run often enough to ensure contiguity of the GDG generations – ie the earliest current GDG date should not be later than the control record LASTDATE. If there is such a gap, it is possible that some 'used' resources may be inadvertently recorded as not used. We eliminate the chance of such a gap by running the job nightly.

If you obtain your shutdown statistics using a PLTSD program, you could skip Steps one and two and instead LINK to, or CALL, a suitably modified version of STATPRG2 to update NULLFILE.

THE NULL-USE REPORT

The null-use report is created by COBOL program STATPRG3 (input parameter = resource type or '*'). STATPRG3 first reads the control record to get FRSTDATE and LASTDATE. It then scans NULLFILE for the selected resource type. There are three cases to consider:

- If the resource LASTDATE is less than the control LASTDATE, we know that the resource has been used and we delete the record from NULLFILE (with a suitable message).
- If the LASTDATEs are equal, we know that the resource has not been used (at least since the resource's FRSTDATE) and we write the resource type, name, and FRSTDATE to the report.

- If the resource LASTDATE is greater than the control record LASTDATE, this is a logic error, and we generate a suitable error message. The file is probably damaged (control record out of sync due to a failure) and should be recovered from a back-up.

The usefulness of the null-use report increases over time. It should be run at least once a month. Resources not used for extended periods will remain in NULLFILE. The FRSTDATE fields will allow us to identify intermittently used resources – STATPRG3 could be further refined with the addition of a FRSTDATE input parameter.

STATEXTR

```
/* REXX */
/*********************************************************************/
/*  STATEXTR   Get statistics for selected null-use resources        */
/*             in DFHSTUP report.                                    */
/*                                                                   */
/*  Parms : MyDsn - Base GDG Name for DFHSTUP report.                */
/*          ResParm - FILE, PROG, TRAN (blank = *)                   */
/*                                                                   */
/*  Calls : STATPRG1          (COBOL prog)                           */
/*                                                                   */
/*********************************************************************/
 TRACE off;
 ADDRESS ISPEXEC;
 "CONTROL ERRORS RETURN";
 ARG Parms;
 ZERRLM = ''; Listid = '';
 LDsn = ''; NoGens = Ø; GenList. = ''; GenDate. = '';
 FrstDate = ''; LastDate = ''; Msg = '';
 PARSE VAR Parms MyDsn ResID;
 x = CheckInputOptions();
 /* Get a list of all members of the GDG (GenList) */
 /* and their creation dates (GenDate) */
 x = GetGDGMembers();
 FrstDate = STRIP(GenDate.1);
 LastDate = STRIP(GenDate.NoGens);
 /* Now process each member of the GDG */
 DO i = 1 TO NoGens              /* for all av generations */
    x = ProcessGDGMember(GenList.i,GenDate.i)
 END;

 /* Record date, no. gens parms in PARMFILE */
 /* (input parms for STATPRG2)              */
 IF (NoGens > 999999) | (NoGens < 1) THEN ,
    Error(2Ø,'NoGens parm ('NoGens') is invalid');
 Line. = '';
 GenLen = 6;                             /* Length of NoGens parm */
 Line.1 = NoGens;
 DO WHILE LENGTH(Line.1) < GenLen
    Line.1 = 'Ø'Line.1;             /* Left-justify with 'Ø' */
 END;
 Line.1 = FrstDate' 'LastDate' 'Line.1;
 Line.1 = STRIP(Line.1);
 IF LENGTH(Line.1) <> ,
    (LENGTH(FrstDate) + 1 + LENGTH(LastDate) + 1 + GenLen) THEN ,
    Error(2Ø,'Length error for STATPRG2 parms');
 ADDRESS TSO "EXECIO 1 DISKW PARMFILE (FINIS STEM Line.";
 SAY 'STATPRG2 Input Parms = ' Line.1;
 /* Successful completion message */
 SAY;
 SAY 'DFHSTUP extract completed.';
 EXIT Ø;

/*******************************************************************/
GetGDGMembers:
/*******************************************************************/
 "LMDINIT LISTID(Listid) LEVEL("MyDsn")";
 IF rc <> THEN ,
    Error(rc,'LMDINIT error for 'MyDsn);
 DO FOREVER
    "LMDLIST LISTID(&Listid) DATASET(LDsn) STATS(YES)"
    IF rc > 8 THEN ,
       Error(rc,'LMDLIST error for 'MyDsn)
    IF rc > THEN LEAVE
    IF STRIP(ZDLCDATE) <> '' THEN DO    /* If not GDG base */
       NoGens = NoGens + 1
       GenList.NoGens = LDsn
       GenDate.NoGens = ZDLCDATE
    END
 END;
 "LMDFREE DATAID(&Listid)";
 IF Nogens = THEN ,
    Error(2Ø,'No generations found for 'MyDsn);
 RETURN Ø;

/**********************************************************************/
/* Process the GDG member using the STATPRG1 utility program          */
/**********************************************************************/
ProcessGDGMember:
 ARG MyDsn,CrDate;
 SAY;
 SAY MyDsn' Created 'CrDate;
 ZERRLM = '';
 ADDRESS TSO;
 x = MSG('OFF');
 "FREE F(DFHSTUP)";
 x = MSG('ON');
 "ALLOC F(DFHSTUP) DA('"MyDsn"') SHR";
 IF rc <> THEN Error(rc, 'ALLOC error for 'MyDsn);
 ADDRESS ISPEXEC;
 "SELECT PGM(STATPRG1) PARM("ResID")";
 IF rc <> THEN ,
    Error(rc,'CALL to STATPRG1 failed.');
Ø116 Ø

 RETURN Ø;
 /*****************************************************************/
 CheckInputOptions:
 /*****************************************************************/
 IF MyDsn = '' THEN ,
    Error(2Ø,'No input file name ||');
 SELECT
    WHEN ResID = '*'     THEN NOP
    WHEN ResID = ''      THEN ResID = '*'
    WHEN ResID = 'FILE'  THEN NOP
    WHEN ResID = 'FILES' THEN ResID = 'FILE'
    WHEN ResID = 'TRAN'  THEN NOP
    WHEN ResID = 'TRANS' THEN ResID = 'TRAN'
    WHEN ResID = 'PROG'  THEN NOP
    WHEN ResID = 'PROGS' THEN ResID = 'PROG'
    OTHERWISE DO
       Msg = 'Invalid resource option - must be '
       Msg = Msg||'FILE, TRAN, PROG, (blank) or ''*'''
       x = Error(8,Msg)
    END
 END;
 RETURN Ø;

 /*****************************************************************/
 Error:
 /*****************************************************************/
 ARG Myrc,Msg;
 ADDRESS ISPEXEC;
 SAY Msg ' rc=' MyRc;
 SAY ZERRLM;
 ADDRESS TSO;
 x = MSG('OFF');
 "FREE F(DFHSTUP)";
 x = MSG('ON');
 ADDRESS ISPEXEC;
 "LMDFREE DATAID(&Listid)";
 ZERRLM = '';
 EXIT Myrc;
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

STATPRG1

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.    STATPRG1.
******************************************************************
*
*FUNCTION: FIND NULL-USE CSD RESOURCES         (CICS 4.1)
*          (INVOKED BY REXX STATEXTR)
*
*INPUT PARMS  : RESOURCE TYPE 'FILE' 'TRAN' 'PROG' OR '*'
*
*INPUT FILE   : DFHSTUP GDG MEMBER (DD NAME = DFHSTUP)
*               (STATEXTR MUST ALLOCATE THIS TO DSN)
*
*OUTPUT FILE  : WORK1 (DDNAME = WORK1)
*
*CALLS/LINKS  : (NONE)
*
*NOTES:
* THIS PROGRAM ANALYSES CICS 4.1 DFHSTUP REPORTS. IT MUST BE
* REVISED WHEN MIGRATING TO CICS TS.
*
******************************************************************

 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT WORK1 ASSIGN TO SYSØØ2-DA-339Ø-S-WORK1
     FILE STATUS IS W-WORK1-STATUS
     ORGANIZATION IS SEQUENTIAL ACCESS IS SEQUENTIAL.
     SELECT DFHSTUP ASSIGN TO SYSØØ3-DA-339Ø-S-DFHSTUP
     FILE STATUS IS W-DFHSTUP-STATUS
     ORGANIZATION IS SEQUENTIAL ACCESS IS SEQUENTIAL.
 DATA DIVISION.
 FILE SECTION.
 FD WORK1
     RECORD CONTAINS 2Ø CHARACTERS
     BLOCK  CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD.
 Ø1 F-WORK1-BUFFER.
    Ø2 F-WORK1-RESOURCE-ID                      PIC X(4).
    Ø2 FILLER                                   PIC X(1).
    Ø2 F-WORK1-RESOURCE-NAME                    PIC X(8).
    Ø2 FILLER                                   PIC X(7).
 FD DFHSTUP
     RECORD CONTAINS 133 CHARACTERS
     BLOCK  CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD.
 Ø1 F-DFHSTUP-BUFFER.
```

```
    02 F-DFHSTUP-PRINT-CTL-CHAR                 PIC X.
    02 F-DFHSTUP-FIRST-CHARS                    PIC X(8).
       88 SW-IGNORE-RECORD VALUE '________'
                                 '  ______'
                                 '  Trans '
                                 '  ID    '
                                 'PROGRAMS'
                                 '  Progra'
                                 '  Name  '
                                 '  File  '
                                 '  Name  '
                                 '        '
                                 'Summary '.
    02 FILLER                                   PIC X(124).

*************************
 WORKING-STORAGE SECTION.
*************************
 01  C-CONSTANTS.
     02 C-EYECATCHER     VALUE '*START OF WORKING STORAGE*'
                                                PIC X(26).
     02 C-PROGRAM-ID     VALUE 'STATPRG1'       PIC X(8).
     02 C-VERSION        VALUE '01.00'          PIC X(5).
     02 C-TOTALS         VALUE '*TOTALS*'       PIC X(8).
     02 C-EXCEPTION-ENTRIES VALUE +10           PIC S9(4) COMP.
**** THIS IS WHERE THE RESOURCE NAME STARTS IN THE DFHSTUP REPT
     02 C-BEGIN-COL         VALUE +4            PIC S9(4) COMP.
     02 C-MAX-RESOURCES     VALUE +3            PIC S9(4) COMP.
 01  C-RESOURCE-VALUES.
     02 C-FILE-VALUES.
        03 C-FILE-ID        VALUE 'FILE'        PIC X(4).
        03 C-FILE-INDEX             VALUE +1    PIC S9(4) COMP.
        03 C-FILE-LENGTH            VALUE +8    PIC S9(4) COMP.
        03 C-FILE-BEGIN-NULL-COL VALUE +12      PIC S9(4) COMP.
        03 C-FILE-END-NULL-COL   VALUE +73      PIC S9(4) COMP.
        03 C-FILE-BEGIN   PIC X(25) VALUE
          'FILES - Requests'.
        03 C-FILE-BEGIN-LENGTH   VALUE +16      PIC S9(4) COMP.
     02 C-PROG-VALUES.
        03 C-PROG-ID        VALUE 'PROG'        PIC X(4).
        03 C-PROG-INDEX             VALUE +2    PIC S9(4) COMP.
        03 C-PROG-LENGTH            VALUE +8    PIC S9(4) COMP.
        03 C-PROG-BEGIN-NULL-COL VALUE +19      PIC S9(4) COMP.
        03 C-PROG-END-NULL-COL   VALUE +46      PIC S9(4) COMP.
        03 C-PROG-BEGIN   PIC X(25) VALUE
          'PROGRAMS '.
        03 C-PROG-BEGIN-LENGTH   VALUE +9       PIC S9(4) COMP.
     02 C-TRAN-VALUES.
        03 C-TRAN-ID        VALUE 'TRAN'        PIC X(4).
        03 C-TRAN-INDEX             VALUE +3    PIC S9(4) COMP.
        03 C-TRAN-LENGTH            VALUE +4    PIC S9(4) COMP.
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
       Ø3 C-TRAN-BEGIN-NULL-COL VALUE +59      PIC S9(4) COMP.
       Ø3 C-TRAN-END-NULL-COL   VALUE +64      PIC S9(4) COMP.
       Ø3 C-TRAN-BEGIN   PIC X(25) VALUE
         'TRANSACTION STATISTICS'.
       Ø3 C-TRAN-BEGIN-LENGTH   VALUE +22      PIC S9(4) COMP.
Ø1  C-RESOURCE-VALUES-TABLE REDEFINES C-RESOURCE-VALUES.
    Ø2 FILLER OCCURS 3.
       Ø3 C-RESOURCE-ID                        PIC X(4).
       Ø3 C-RESOURCE-INDEX                     PIC S9(4) COMP.
       Ø3 C-RESOURCE-LENGTH                    PIC S9(4) COMP.
       Ø3 C-RESOURCE-BEGIN-NULL-COL            PIC S9(4) COMP.
       Ø3 C-RESOURCE-END-NULL-COL              PIC S9(4) COMP.
       Ø3 C-RESOURCE-BEGIN                     PIC X(25).
       Ø3 C-RESOURCE-BEGIN-LENGTH              PIC S9(4) COMP.
Ø1  C-EXCEPTION-TABLE-VALUES.
    Ø2 C-FILE-EXCEPTION-VALUES.
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
    Ø2 C-PROG-EXCEPTION-VALUES.
       Ø3 FILLER                 VALUE 'DFH*'  PIC X(8).
       Ø3 FILLER                 VALUE 'CEE*'  PIC X(8).
       Ø3 FILLER                 VALUE 'CSQ*'  PIC X(8).
       Ø3 FILLER                 VALUE 'DSN*'  PIC X(8).
       Ø3 FILLER                 VALUE 'EDC*'  PIC X(8).
       Ø3 FILLER                 VALUE 'IBM*'  PIC X(8).
       Ø3 FILLER                 VALUE 'IGZ*'  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
    Ø2 C-TRAN-EXCEPTION-VALUES.
       Ø3 FILLER                 VALUE 'C*'    PIC X(8).
       Ø3 FILLER                 VALUE 'DSNC'  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
       Ø3 FILLER                 VALUE SPACES  PIC X(8).
Ø1  C-EXCEPTION-TABLE REDEFINES
    C-EXCEPTION-TABLE-VALUES.
    Ø2 C-EXCEPTIONS OCCURS 3.
```

```
       Ø3 C-EXCEPTION      OCCURS 1Ø           PIC X(8).
Ø1  W-SWITCHES.
    Ø2 W-EOF-SWITCH         VALUE SPACE        PIC X.
       88 SW-EOF            VALUE 'E'.
    Ø2 W-LAST-FILE-SWITCH   VALUE SPACE        PIC X.
       88 SW-LAST-FILE      VALUE 'L'.
    Ø2 W-LAST-TRAN-SWITCH   VALUE SPACE        PIC X.
       88 SW-LAST-TRAN      VALUE 'T'.
    Ø2 W-LAST-PROG-SWITCH   VALUE SPACE        PIC X.
       88 SW-LAST-PROG      VALUE 'L'.
    Ø2 W-START-FILE-SWITCH  VALUE SPACE        PIC X.
       88 SW-START-FILE     VALUE 'L'.
    Ø2 W-START-PROG-SWITCH  VALUE SPACE        PIC X.
       88 SW-START-PROG     VALUE 'T'.
    Ø2 W-START-TRAN-SWITCH  VALUE SPACE        PIC X.
       88 SW-START-TRAN     VALUE 'T'.
    Ø2 W-WORK1-FILE-SWITCH  VALUE SPACE        PIC X.
       88 SW-WORK1-FILE-OPEN    VALUE 'O'.
       88 SW-WORK1-FILE-NOT-OPEN VALUE SPACE.
    Ø2 W-DFHSTUP-FILE-SWITCH VALUE SPACE       PIC X.
       88 SW-DFHSTUP-FILE-OPEN  VALUE 'O'.
       88 SW-DFHSTUP-FILE-NOT-OPEN VALUE SPACE.
    Ø2 W-RESOURCE-INDEX     VALUE +Ø           PIC S9(4) COMP.
       88 SW-FILE-INDEX     VALUE +1.
       88 SW-PROG-INDEX     VALUE +2.
       88 SW-TRAN-INDEX     VALUE +3.
    Ø2 W-EXCEPTION-SWITCH   VALUE SPACE        PIC X.
       88 SW-EXCEPTION      VALUE 'E'.
       88 SW-NOT-EXCEPTION  VALUE SPACE.
    Ø2 W-RESOURCE-PARM      VALUE SPACES       PIC X(4).
       88 SW-FILE           VALUE 'FILE'.
       88 SW-PROG           VALUE 'PROG'.
       88 SW-TRAN           VALUE 'TRAN'.
       88 SW-ALL            VALUE '*   ' ' * ' '  *'.
       88 SW-VALID-RESOURCE VALUE 'TRAN' 'PROG' 'FILE' '*'.
Ø1  W-WORK-FIELDS.
    Ø2 W-RETURN-CODE-PIC    VALUE Ø            PIC 9(6).
    Ø2 W-WORK-BUFFER                           PIC X(1ØØ).
    Ø2 W-FILE-OPERATION                        PIC X(12).
    Ø2 W-COMMAND-LENGTH                        PIC S9(8) COMP.
    Ø2 W-COMMAND                               PIC X(4Ø).
    Ø2 W-WORK1-STATUS                          PIC 9(2).
    Ø2 W-DFHSTUP-STATUS                        PIC 9(2).
    Ø2 W-EXCEPTION                             PIC X(8).
    Ø2 I                                       PIC S9(4) COMP.
    Ø2 J                                       PIC S9(4) COMP.
    Ø2 K                                       PIC S9(4) COMP.
    Ø2 L                                       PIC S9(4) COMP.
    Ø2 P                                       PIC S9(4) COMP.
    Ø2 W-COMMAND-PTR                           PIC S9(4) COMP.
    Ø2 W-WORK-PTR                              PIC S9(4) COMP.
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
     Ø2 W-WORK-LENGTH                           PIC S9(4) COMP.
     Ø2 W-RESOURCE-NAME                         PIC X(8).
     Ø2 W-LABEL                                 PIC X(4).
     Ø2 W-WORK-PIC                              PIC 9(6).
     Ø2 W-MSG                                   PIC X(6Ø).
     Ø2 W-WORK8                                 PIC X(8).

******************************************************************
 LINKAGE SECTION.
******************************************************************
 Ø1  L-PARM-FIELDS.
     Ø2 L-PARM-LENGTH                           PIC S9(4) COMP.
     Ø2 L-PARMS                                 PIC X(1ØØ).
******************************************************************
 PROCEDURE DIVISION USING L-PARM-FIELDS.
******************************************************************
*******************
 ØØØØ-MAIN SECTION.
*******************
     PERFORM P-INITIALISE.
     PERFORM P-PROCESS.
     PERFORM P-CLEANUP.
     MOVE W-RETURN-CODE-PIC TO RETURN-CODE.
 ØØØØ-RETURN.
     GOBACK.
 ØØØØ-EXIT.
     EXIT.
******************************************************************
 P-PROCESS.
******************************************************************
     PERFORM UNTIL SW-EOF
        READ DFHSTUP
           AT END SET SW-EOF TO TRUE
        END-READ
        IF NOT SW-EOF THEN
           PERFORM P-PROCESS-RECORD
           IF SW-LAST-FILE AND SW-LAST-PROG AND SW-LAST-TRAN
              SET SW-EOF TO TRUE
           END-IF
        END-IF
     END-PERFORM.
******************************************************************
 P-PROCESS-RECORD.
******************************************************************
**** LOOK FOR A 'START RESOURCE' DELIMITER     *******************
     PERFORM VARYING P FROM 1 BY 1 UNTIL P > C-MAX-RESOURCES
        MOVE C-RESOURCE-BEGIN-LENGTH(P) TO W-WORK-LENGTH
        IF F-DFHSTUP-BUFFER(2:W-WORK-LENGTH) =
           C-RESOURCE-BEGIN(P)
           PERFORM P-SET-START-SWITCH
           MOVE +999 TO P
        END-IF
     END-PERFORM.

**** IF NOT 'START RESOURCE'  *******************
     IF P NOT = +999 THEN
        EVALUATE TRUE
**** LOOK FOR AN 'END RESOURCE' DELIMITER *******************
           WHEN F-DFHSTUP-BUFFER(4:LENGTH OF C-TOTALS) =
              C-TOTALS
              PERFORM P-SET-LAST-SWITCH
           WHEN SW-IGNORE-RECORD
              CONTINUE
           WHEN OTHER
**** ELSE LOOK FOR 'NULL USE' *******************
              PERFORM P-SELECT-RESOURCE
        END-EVALUATE
     END-IF.

******************************************************************
 P-SELECT-RESOURCE.
******************************************************************
     EVALUATE TRUE
        WHEN SW-FILE AND SW-START-FILE AND NOT SW-LAST-FILE
           PERFORM P-PROCESS-RESOURCE
        WHEN SW-PROG AND SW-START-PROG AND NOT SW-LAST-PROG
           PERFORM P-PROCESS-RESOURCE
        WHEN SW-TRAN AND SW-START-TRAN AND NOT SW-LAST-TRAN
           PERFORM P-PROCESS-RESOURCE
        WHEN SW-ALL AND SW-START-FILE AND NOT SW-LAST-FILE
           PERFORM P-PROCESS-RESOURCE
        WHEN SW-ALL AND SW-START-PROG AND NOT SW-LAST-PROG
           PERFORM P-PROCESS-RESOURCE
        WHEN SW-ALL AND SW-START-TRAN AND NOT SW-LAST-TRAN
           PERFORM P-PROCESS-RESOURCE
        WHEN OTHER
           CONTINUE
     END-EVALUATE.

******************************************************************
 P-SET-LAST-SWITCH.
******************************************************************
     EVALUATE TRUE
        WHEN SW-START-FILE SET SW-LAST-FILE TO TRUE
        WHEN SW-START-PROG SET SW-LAST-PROG TO TRUE
        WHEN SW-START-TRAN SET SW-LAST-TRAN TO TRUE
        WHEN OTHER         CONTINUE
     END-EVALUATE.

******************************************************************
 P-SET-START-SWITCH.
******************************************************************
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
     EVALUATE TRUE
        WHEN C-RESOURCE-ID(P) = 'FILE' AND NOT SW-LAST-FILE
           SET SW-START-FILE TO TRUE
           SET SW-FILE-INDEX TO TRUE
        WHEN C-RESOURCE-ID(P) = 'PROG' AND NOT SW-LAST-PROG
           SET SW-START-PROG TO TRUE
           SET SW-PROG-INDEX TO TRUE
        WHEN C-RESOURCE-ID(P) = 'TRAN' AND NOT SW-LAST-TRAN
           SET SW-START-TRAN TO TRUE
           SET SW-TRAN-INDEX TO TRUE
        WHEN OTHER
           CONTINUE
     END-EVALUATE.

******************************************************************
 P-PROCESS-RESOURCE.
******************************************************************
     MOVE W-RESOURCE-INDEX TO I.
     PERFORM P-CHECK-EXCEPTION.
     IF SW-NOT-EXCEPTION THEN
        PERFORM P-CHECK-NULLUSE
     END-IF.

******************************************************************
 P-CHECK-EXCEPTION.
******************************************************************
     SET SW-NOT-EXCEPTION TO TRUE.
     MOVE C-RESOURCE-LENGTH(I)                TO W-WORK-LENGTH.
     MOVE SPACES                              TO W-RESOURCE-NAME.
     MOVE F-DFHSTUP-BUFFER(C-BEGIN-COL:W-WORK-LENGTH)
                                              TO W-RESOURCE-NAME.
     PERFORM VARYING J FROM 1 BY 1 UNTIL J > C-EXCEPTION-ENTRIES
        OR SW-EXCEPTION
           MOVE C-EXCEPTION(I J) TO W-EXCEPTION
           IF W-EXCEPTION(1:1) = '*' THEN
              MOVE +2Ø TO W-RETURN-CODE-PIC
              MOVE 'LOGIC ERROR FOR EXCEPTION ENTRY' TO W-MSG
              PERFORM P-ERROR
              GO TO ØØØØ-RETURN
           END-IF
           MOVE SPACES           TO W-WORK8
           MOVE W-RESOURCE-NAME  TO W-WORK8(1:W-WORK-LENGTH)
           PERFORM VARYING K FROM 1 BY 1 UNTIL K > W-WORK-LENGTH
              OR W-EXCEPTION = SPACES
              EVALUATE TRUE
                 WHEN W-EXCEPTION(K:1) = '*'
                    COMPUTE L = W-WORK-LENGTH - K + 1
                    MOVE SPACES TO W-EXCEPTION(K:L)
                    MOVE SPACES TO W-WORK8(K:L)
                    MOVE +999 TO K
                 WHEN W-EXCEPTION(K:1) = SPACE
```

```
                    MOVE +999 TO K
                 WHEN W-WORK8(K:1) = SPACE
                    MOVE +999 TO K
                 WHEN OTHER CONTINUE
              END-EVALUATE
           END-PERFORM
        IF W-EXCEPTION = SPACES THEN
           MOVE +999 TO J
        ELSE
           IF W-WORK8 = W-EXCEPTION THEN
              SET SW-EXCEPTION TO TRUE
           END-IF
        END-IF
     END-PERFORM.

******************************************************************
 P-CHECK-NULLUSE.
******************************************************************
**** IF 'NULL-USE' LINE FOUND, RECORD IT TO WORK1 FILE
     COMPUTE W-WORK-LENGTH = C-RESOURCE-END-NULL-COL(I) -
                             C-RESOURCE-BEGIN-NULL-COL(I) + +1.
     MOVE C-RESOURCE-BEGIN-NULL-COL(I) TO W-WORK-PTR.
     MOVE SPACES TO W-WORK-BUFFER.
     MOVE F-DFHSTUP-BUFFER(W-WORK-PTR:W-WORK-LENGTH)
                                       TO W-WORK-BUFFER.
**** IF ONLY BLANKS OR 'Ø', THEN THIS IS A 'NULL-USE' RECORD
     INSPECT W-WORK-BUFFER REPLACING ALL 'Ø' BY SPACE.
     IF W-WORK-BUFFER = SPACES THEN
        MOVE SPACES TO F-WORK1-BUFFER
        MOVE C-RESOURCE-ID(I) TO F-WORK1-RESOURCE-ID
        MOVE W-RESOURCE-NAME TO F-WORK1-RESOURCE-NAME
        WRITE F-WORK1-BUFFER
        IF W-WORK1-STATUS NOT = 'ØØ' THEN
           MOVE +2Ø TO W-RETURN-CODE-PIC
           MOVE 'WRITE WORK1' TO W-FILE-OPERATION
           PERFORM P-FILE-ERROR
           GO TO ØØØØ-RETURN
        END-IF
     END-IF.

******************************************************************
 P-INITIALIZE.
******************************************************************
**** CHECK FOR VALID RESOURCE TYPE - '*' IS DEFAULT
     IF L-PARM-LENGTH < +1 THEN
        SET SW-ALL TO TRUE
     ELSE
        MOVE L-PARMS(1:L-PARM-LENGTH) TO W-RESOURCE-PARM
        PERFORM P-CHECK-RESOURCE-TYPE
     END-IF.
```

© 2000. Reproduction prohibited. Please inform Xephon of any infringement.

```
**** OPEN WORK1 & DFHSTUP FILES
     PERFORM P-OPEN-FILES.
     EVALUATE TRUE
        WHEN SW-FILE
           SET SW-LAST-TRAN TO TRUE
           SET SW-LAST-PROG TO TRUE
        WHEN SW-TRAN
           SET SW-LAST-FILE TO TRUE
           SET SW-LAST-PROG TO TRUE
        WHEN SW-PROG
           SET SW-LAST-FILE TO TRUE
           SET SW-LAST-TRAN TO TRUE
        WHEN OTHER
           CONTINUE
     END-EVALUATE.

******************************************************************
 P-OPEN-FILES.
******************************************************************
     OPEN EXTEND WORK1.
     IF W-WORK1-STATUS NOT = 'ØØ' THEN
        MOVE +2Ø TO W-RETURN-CODE-PIC
        MOVE 'OPEN WORK1' TO W-FILE-OPERATION
        PERFORM P-FILE-ERROR
        GO TO ØØØØ-RETURN
     END-IF.
     SET SW-WORK1-FILE-OPEN TO TRUE.
     OPEN INPUT DFHSTUP.
     IF W-DFHSTUP-STATUS NOT = 'ØØ' THEN
        MOVE +2Ø TO W-RETURN-CODE-PIC
        MOVE 'OPEN DFHSTUP' TO W-FILE-OPERATION
        MOVE W-DFHSTUP-STATUS TO W-WORK1-STATUS
        PERFORM P-FILE-ERROR
        GO TO ØØØØ-RETURN
     END-IF.
     SET SW-DFHSTUP-FILE-OPEN TO TRUE.

******************************************************************
 P-CHECK-RESOURCE-TYPE.
******************************************************************
     EVALUATE TRUE
        WHEN W-RESOURCE-PARM = SPACES OR LOW-VALUES
           SET SW-ALL TO TRUE
        WHEN SW-FILE CONTINUE
        WHEN SW-PROG CONTINUE
        WHEN SW-TRAN CONTINUE
        WHEN SW-ALL  CONTINUE
        WHEN OTHER
           MOVE +2Ø TO W-RETURN-CODE-PIC
           STRING
              W-RESOURCE-PARM ' IS AN INVALID RESOURCE TYPE.'
              ' MUST BE ''FILE'', ''PROG'', ''TRAN'' OR ''*'''
              DELIMITED BY SIZE INTO W-MSG
           END-STRING
           PERFORM P-ERROR
           GO TO ØØØØ-RETURN
     END-EVALUATE.
     DISPLAY ' '.

******************************************************************
 P-CLEANUP.
******************************************************************
     IF SW-WORK1-FILE-OPEN THEN
        CLOSE WORK1
        SET SW-WORK1-FILE-NOT-OPEN TO TRUE
     END-IF.
     IF SW-DFHSTUP-FILE-OPEN THEN
        CLOSE DFHSTUP
        SET SW-DFHSTUP-FILE-NOT-OPEN TO TRUE
     END-IF.

******************************************************************
 P-FILE-ERROR.
******************************************************************
     STRING
        W-FILE-OPERATION                      DELIMITED BY '  '
        ' FILE FAILED, STATUS CODE='          DELIMITED BY SIZE
         W-WORK1-STATUS                       DELIMITED BY SIZE
         INTO W-MSG
     END-STRING.
     PERFORM P-ERROR.

******************************************************************
 P-ERROR.
******************************************************************
     DISPLAY W-MSG.
     DISPLAY 'RC=' W-RETURN-CODE-PIC.
     PERFORM P-CLEANUP.
     MOVE W-RETURN-CODE-PIC TO RETURN-CODE.
```

STATPRG2

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.    STATPRG2.
******************************************************************
* RECORD NULLUSE CICS RESOURCES TO PERMANENT 'NULLUSE' FILE
******************************************************************
* FUNCTION : RECORD NULLUSE CICS RESOURCES TO PERMANENT
* 'NULLUSE' FILE
*INPUT FILES : INPUT PARMS - FRSTDATE, LASTDATE, NOGENS
*            : SORTED WORK FILE CONTAINING NULLUSE RESOURCES
```

```
*OUTPUT FILE : NULLFILE
*PARMS       : NO. OF GDG GENERATIONS IN INPUT FILE
*              EARLIEST DATE OF GDG GENERATIONS
*              LATEST DATE OF GDG GENERATIONS
*
*IF THE NUMBER OF DUPLICATE RECORDS IN THE INPUT FILE EQUALS THE
*NUMBER OF GENERATIONS, THEN THE RELEVANT RESOURCE HAS NOT BEEN
*USED FROM START DATE TO LAST DATE.
******************************************************************
 ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
     SELECT SORTFILE ASSIGN TO SYSØØ2-DA-339Ø-S-SORTFILE
     FILE STATUS IS W-SORTFILE-STATUS
     ORGANIZATION IS SEQUENTIAL ACCESS IS SEQUENTIAL.
     SELECT PARMFILE ASSIGN TO SYSØØ3-DA-339Ø-S-PARMFILE
     FILE STATUS IS W-PARMFILE-STATUS
     ORGANIZATION IS SEQUENTIAL ACCESS IS SEQUENTIAL.
     SELECT NULLFILE ASSIGN TO NULLFILE
     RECORD KEY IS F-NULLFILE-KEY
     FILE STATUS IS W-NULLFILE-STATUS W-VSAM-CODE
     ORGANIZATION IS INDEXED ACCESS IS DYNAMIC.

 DATA DIVISION.
 FILE SECTION.
 FD SORTFILE
     RECORD CONTAINS 2Ø CHARACTERS
     BLOCK  CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD.
 Ø1 F-SORTFILE-READ-BUFFER.
    Ø2 F-SORTFILE-RESOURCE-ID                   PIC X(4).
       88 SW-PROG             VALUE 'PROG'.
       88 SW-FILE             VALUE 'FILE'.
       88 SW-TRAN             VALUE 'TRAN'.
       88 SW-VALID-RESOURCE   VALUE 'TRAN' 'PROG' 'FILE'.
    Ø2 FILLER                                   PIC X(1).
    Ø2 F-SORTFILE-RESOURCE-NAME                 PIC X(8).
    Ø2 FILLER                                   PIC X(7).
 FD PARMFILE
     RECORD CONTAINS 28 CHARACTERS
     BLOCK  CONTAINS Ø RECORDS
     RECORDING MODE IS F
     LABEL RECORDS ARE STANDARD.
 Ø1 F-PARMFILE-READ-BUFFER.
    Ø2 F-PARMFILE-START-DATE                    PIC X(1Ø).
    Ø2 FILLER                                   PIC X.
    Ø2 F-PARMFILE-LAST-DATE                     PIC X(1Ø).
    Ø2 FILLER                                   PIC X.
    Ø2 F-PARMFILE-NO-GENS                       PIC X(6).
* NULLFILE FD
```

```
     COPY NULLFILE.
*
*************************
 WORKING-STORAGE SECTION.
*************************
 Ø1  C-CONSTANTS.
     Ø2 C-EYECATCHER     VALUE '*START OF WORKING STORAGE*'
                                                PIC X(26).
     Ø2 C-PROGRAM-ID     VALUE 'STATPRG2'       PIC X(8).
     Ø2 C-VERSION        VALUE 'Ø1.ØØ'          PIC X(5).
 Ø1  W-SWITCHES.
     Ø2 W-SORTFILE-OPEN-SWITCH     VALUE SPACE  PIC X.
        88 SW-SORTFILE-OPEN        VALUE 'O'.
        88 SW-SORTFILE-CLOSED      VALUE SPACE.
     Ø2 W-PARMFILE-OPEN-SWITCH     VALUE SPACE  PIC X.
        88 SW-PARMFILE-OPEN        VALUE 'O'.
        88 SW-PARMFILE-CLOSED      VALUE SPACE.
     Ø2 W-NULLFILE-OPEN-SWITCH     VALUE SPACE  PIC X.
        88 SW-NULLFILE-OPEN-IO     VALUE 'U'.
        88 SW-NULLFILE-OPEN-OUTPUT VALUE 'O'.
        88 SW-NULLFILE-OPEN        VALUE 'U' 'O'.
        88 SW-NULLFILE-CLOSED      VALUE SPACE.
 Ø1  W-WORK-FIELDS.
     Ø2 W-SORTFILE-STATUS    VALUE 'ØØ'         PIC X(2).
     Ø2 W-PARMFILE-STATUS    VALUE 'ØØ'         PIC X(2).
     Ø2 W-WORK-PIC           VALUE Ø            PIC 9(8).
     Ø2 W-VSAM-CODE.
        Ø3 W-VSAM-RETURN-CODE       VALUE ØØ    PIC 9(2).
        Ø3 W-VSAM-COMPONENT-CODE    VALUE Ø     PIC 9(1).
        Ø3 W-VSAM-REASON-CODE       VALUE ØØØ   PIC 9(3).
     Ø2 W-NULLFILE-STATUS    VALUE 'ØØ'         PIC X(2).
     Ø2 W-RECORD-COUNT       VALUE +Ø           PIC S9(8) COMP.
     Ø2 W-RETURN-CODE-SAVE                      PIC S9(8) COMP.
     Ø2 W-RETURN-CODE-PIC    VALUE Ø            PIC 9(6).
     Ø2 I                                       PIC S9(8) COMP.
     Ø2 W-ERROR-MSG                             PIC X(4Ø).
     Ø2 W-NO-GENS-PIC        VALUE Ø            PIC 999999.
     Ø2 W-PREVIOUS-READ-BUFFER VALUE SPACES     PIC X(2Ø).
     Ø2 W-RESOURCE-COUNT     VALUE +Ø           PIC S9(4) COMP.
     Ø2 W-NULLUSE-FILE       VALUE +Ø           PIC S9(8) COMP.
     Ø2 W-NULLUSE-PROG       VALUE +Ø           PIC S9(8) COMP.
     Ø2 W-NULLUSE-TRAN       VALUE +Ø           PIC S9(8) COMP.
     Ø2 W-NULLUSE-NEW                           PIC S9(8) COMP.
     Ø2 W-NULLUSE-OLD                           PIC S9(8) COMP.
     Ø2 W-NO-GENS            VALUE Ø            PIC S9(4) COMP.
```

Editor’s note: this article will be continued in the next issue.

David Roth
CICS Consultant (Germany)

© Xephon 2000

CICS news

CICS users can benefit from the latest announcement from Software AG and Microsoft. The two companies have announced plans to uprate Microsoft's Host Integration Server 2000 Platform with Software AG's enterprise integration products. The resulting software will provide the means to extend COM Transaction Integrator (COMTI) to 3270 I/O-based CICS applications. The partnership also includes support for Software AG's Natural developer tools and Adabas database.

For further information contact:
Software AG, Charter Court, 74-78 Victoria Street, St Albans, AL1 3XH, UK.
Tel: (01727) 844455.
Microsoft, Microsoft Place, Winnersh Triangle, Wokingham, Berks, RG11 5TP, UK.
Tel: (01734) 270001.

Software AG, 11190 Sunrise Valley Drive, Reston, VA 22091, USA.
Tel: (703) 860 5050.
Microsoft, 1 Microsoft Way, Redmond, WA 98052-6399, USA.
Tel: (425) 882 8080.

URL: http://www.softwareag.com
URL: http://www.microsoft.com.

* * *

There's good news for CICS users running on VSE operating systems. IBM, pointing out that VSE users are demanding greater interoperability with other other servers, has announced plans to create VSE e-business connectors, promising easy access to VSE resources from other systems.

The forthcoming Version 2.5, says the vendor, will include server code that runs on VSE itself plus associated JavaBeans and servlets that run on Java-capable clients. Supported clients include systems running IBM's WebSphere Application Server.

Hence the OS will be the platform on which CICS Web Support and the CICS 3270 Bridge function will become generally available.

Among the other planned enhancements are an increase in dynamic classes, VSAM exploitation of IXFP/SnapShot, support for Enterprise Storage Server (ESS) FlashCopy, and a VSAM hashing algorithm for faster access to large VSAM LSR buffer pools. Also, Fast Service Upgrade (FSU) from VSE/ESA V2R4 will be provided.

Meanwhile, Version 2.5 will no longer support LANRES/VSE, Distributed Workstation Facility (DWF), OCR/MICR devices, VisualLift runtime environment, and the distributed centrally-managed remote unattended VSE environment.

For further information contact your local IBM representative.
http://www.software.ibm.com/data/cics.

* * *