



216

CICS

November 2003

In this issue

- 3 CICS TS 2.2 threadsafe usage and performance
- 12 Providing context-sensitive help in a CICS application
- 29 CICS file browse
- 53 December 2000 – November 2003 index
- 55 CICS news

update

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

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 December 1999 issue, are available separately to subscribers for £16.00 (\$24.00) each including postage.

***CICS Update* on-line**

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

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.

Contributions

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

© Xephon plc 2003. 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.

CICS TS 2.2 threadsafe usage and performance

In today's tough economy, when a lot of companies are looking for a way to cut operations cost, CICS TS 2.2 may present an opportunity to significantly reduce CPU overhead, when DB2 is used. This is achieved by making transactions 'threadsafe' and reducing TCB switches, from QR TCB and L8 TCB.

IBM documented up to 44% CPU reduction, processing 1 million rows, comparing CICS TS 1.3 with CICS TS 2.2 with threadsafe. Although the benefits of using threadsafe can be substantial, a lot of planning and preparation has to be done in order to take advantage of this feature; also, not every shop will be able to use it because of the restrictions associated with it.

WHAT IS THREADSAFE?

Threadsafe means that two copies of your programs can run at the same time on different TCBs without problems. Traditionally CICS has required only quasi-reentrancy and used only one QR TCB, and CICS programs were stopped only by EXEC CICS commands. Running two copies of the program means they must be fully reentrant. They must also be coded so that they do not update CICS shared memory without taking action to serialize the updates.

In an open transaction environment, programs that access shared resources must be aware that these resources can also be accessed by other user tasks running in an open TCB. Programs that use appropriate serialization techniques when accessing shared resources are described as threadsafe.

If CICS is connected to DB2 V6 or later, the task-related user exit is enabled with OPENAPI automatically for you when you start the CICS/DB2 adaptor. So, if the CICS program is threadsafe, it can run under its own L8 TCB – the same one that executes the DB2 commands.

The transaction starts on the QR TCB, but, when DB2 is invoked,

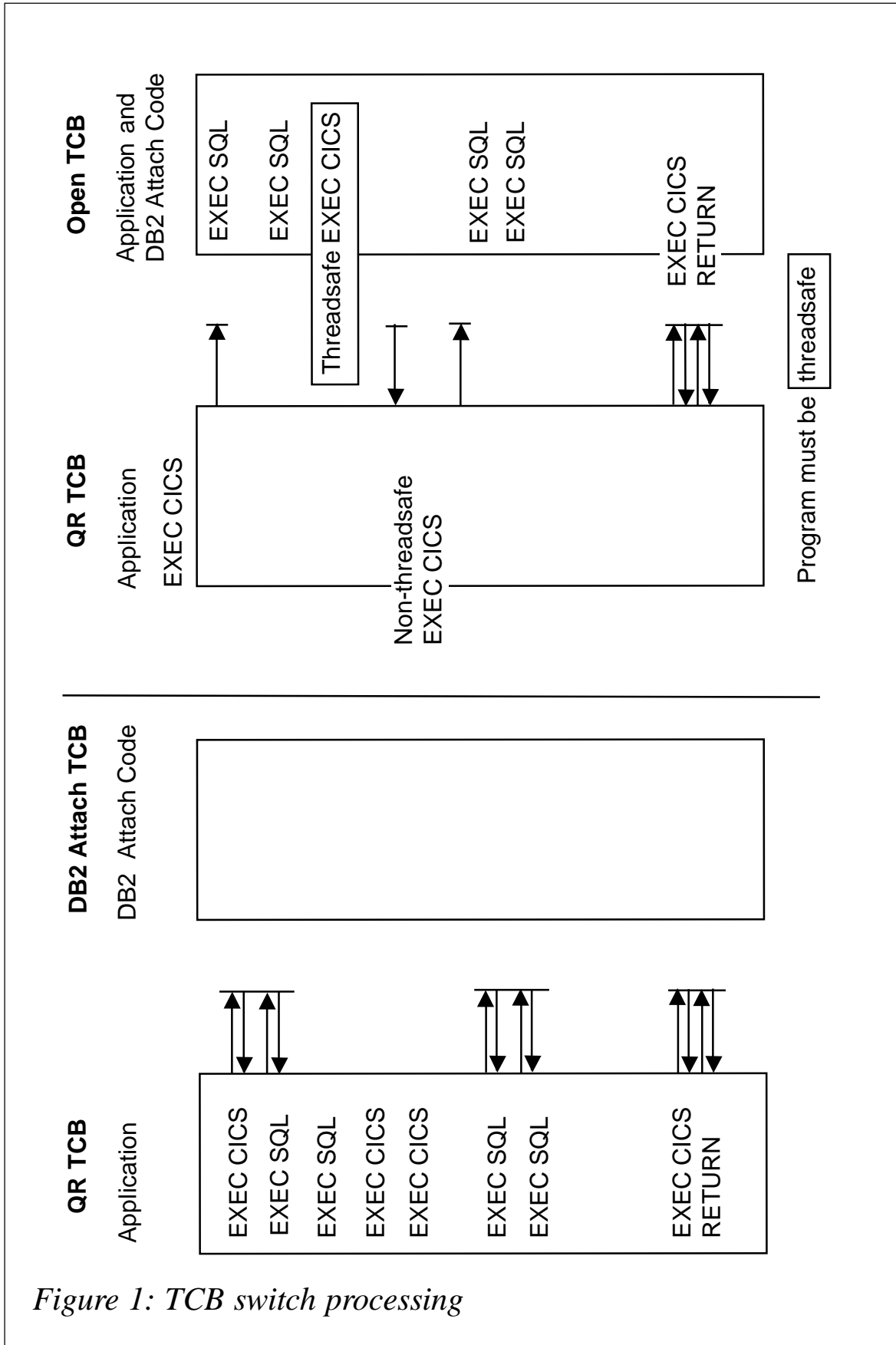


Figure 1: TCB switch processing

CICS switches control to an L8 TCB dedicated to this transaction. On return from DB2, if the application is threadsafe, there is no need to return to QR TCB. It continues to execute in the L8 TCB. Similarly, there is no need to switch TCBs to call CICS commands that are threadsafe. But, it is necessary to switch back to the QR TCB to perform some CICS commands and other functions, such as the syncpoint process. Usually, depending on the number of SQL calls and how they are interleaved with non-threadsafe EXEC CICS commands, the number of TCB switches will be reduced.

Figure 1 illustrates the TCB switch process, with and without threadsafe setting.

For most resources, such as files, DB2 tables, temporary storage queues, and transient data queues, CICS processing automatically ensures access in a threadsafe manner. But for resources that are accessed directly by user programs, like storage, the CICS CWA, global user exit global work areas, and storage acquired by EXEC CICS GETMAIN SHARED commands such as shared storage, it is the responsibility of the user program to ensure threadsafe processing. *CICS Application Programming Guide*, SC34-5702, should be reference for more details.

CONCURRENCY SETTING

CICS TS 2.2 has a new parameter under the RDO called CONCURRENCY, which tells CICS about the multi-use standards applying to the program. In particular, whether or not it can safely execute on multiple TCBs at the same time: this is the threadsafe property.

The CONCURRENCY(QUASIRENT) setting is a default for DB2 usage. Programs running with this setting will run under main CICS QR TCB and there is no performance improvement in using it.

Using the CONCURRENCY(THREADSAFE) setting, after DB2 activity, will let executing programs remain on the L8 TCB used

for DB2 access. It will not swap back to use the CICS QR TCB after DB2 activity, so giving a significant performance improvement for more DB2 activity.

Be aware that application programs will swap back to the QR TCB when an EXEC CICS command, which is not threadsafe, is issued. If your application does a lot of non-threadsafe EXEC CICS commands, your response time could actually double!

To take full advantage of this threadsafe performance improvement, it is recommended that you restructure your application programs to group DB2 activity together without any intervening EXEC CICS commands.

USER EXITS AND TCBS

The CONCURRENCY parameter does not only affect programs, but it also applies to TRUEs. When used with a TRUE, it means the TRUE will obey the threadsafe multi-activity rules and so can run on an L8 TCB. You need to be aware that the DB2 TRUE is enabled with CONCURRENCY(THREADSAFE), so that DB2 activity runs under an L8 TCB.

All CICS Global User Exits can now be invoked on any of the TCBS used by CICS. Since two instances of a GLUE program can now be running at the same time on different TCBS, all GLUE programs must be threadsafe. This means that they must be coded to support multiple access to shared resources. This presents a big challenge, since all vendor-supplied GLUE programs must be checked to see whether they are threadsafe.

Some customers have experienced problems in the past using threadsafe with Candle's exit for Omegamon and with XPEDITER. Check with the vendors whether there are any outstanding issues using threadsafe with their products.

There is also the area of GLUE parmlist changes. There are new TCB modes in DFHUEPAR.UEPGIND to show which TCB type is being used. Parmlist changes affect XFAINTU, XRSINDI, and XSNON.

THREADSAFE CODING

Whenever a GLUE program uses the GLUE workarea, access should be 'critical sectioned' to prevent multiple accesses. The most common problem is when one instance is updating a counter whilst another instance gets the previous number. A similar problem may occur when the exit program accesses some shared storage. The Assembler Compare-and-Swap (CS) instruction can be used to update fields or process ECBs in an atomic fashion. Use a DFHNQEDX FUNCTION(ENQUEUE) to delimit the start of the critical section and end with an equivalent DFHNQEDX FUNCTION(DEQUEUE).

The bottom line is that access to the shared resource has to be serialized.

MIGRATION CONSIDERATIONS

Migration considerations are:

- 1 Check all vendor-supplied GLUEs for threadsafeness.
- 2 Recode your own GLUEs to obey threadsafe rules.
- 3 Ensure that access to shared resources is serialized.
- 4 Restructure your application programs to group DB2 activity together without any intervening EXEC CICS commands.
- 5 Establish installation standards for GLUE critical sectioning.
- 6 Use DFHNQEDX to establish critical sections.
- 7 Use CS instructions to update outside a critical section.
- 8 Use CS instructions to process ECBs.
- 9 Establish a good ENQ naming convention.

DETERMINING WHETHER A PROGRAM IS THREADSAFE

In order for the program to be threadsafe it has to be reentrant, which is achieved by using the RENT option (for LE programs) during the compile. If you are running Assembler programs, you

can test them for reentrancy by linking with the RENT option and then running in a CICS region with RENTPGM=PROTECT. SOC4 abends will result if non-reentrant programs attempt to modify themselves.

With CICS TS 2.2, IBM supplies a scan utility that can identify potential non-threadsafe programs. DFHEISUP works by scanning application load modules looking for occurrences of commands found in member DFHEIDTH.

DFHEISUP will report, for example, that a program issues an ADDRESS CWA command. Because the CWA is often used to maintain counters or address chains, a program addressing the CWA could be using it in a non-threadsafe manner. On the other hand, the program could also be using the CWA to check for operational flags, file DD names, or other uses that do not raise threadsafe issues. More worrying, DFHEISUP could report no hits on an application program, leading one to believe that the program was threadsafe, when the program was maintaining counters in a shared storage location whose address is passed in the incoming COMMAREA.

While DFHEISUP is helpful in the process of identifying threadsafe applications, the only way to ensure that an application is threadsafe is to have a competent programmer who is familiar with the application review it in its entirety.

CICS.SDFHSAMP(DFHEIDTH):

```
#####
# CICS LOAD MODULE SCANNER FILTER TABLE - THREADSAFE INHIBITORS      #
# This table identifies commands which "may" cause the program not to  #
# be threadsafe in that they allow accessibility to shared storage and #
# the application must have the necessary synchronization logic in     #
# place to guard against concurrent update.                            #
#####
#####
# The extract command gives addressibility to a global work area of a  #
# GLUE or TRUE.                                                         #
#####
EXTRACT EXIT GASET *
#####
# Getmain shared storage can be shared between CICS transactions.      #
#####
```



```

GETMAIN SHARED *
#####
# The CWA is shared between all CICS transactions.          #
#####
ADDRESS CWA *

```

Here is an example of adding the following non-threadsafe commands to this list:

```

ASKTIME
FORMATTIME
WRITE FILE
WRITEQ TD
WRITE JOURNALNAME
SYNCPPOINT
SYNCPPOINT ROLLBACK

```

Load module scan using supplied DFHEIDTH:

```

//*****
//* RUN THE LOAD MODULE SCANNER
//* DFHEIDTH CHECKS FOR 'SHARED ' COMMANDS IN THREADSAFE CHECKS
//*****
//*DFHSCAN EXEC PGM=DFHEI SUP, PARM=(' SUMMARY, DETAI LMODS' ), REGION=512M
//DFHSCAN EXEC PGM=DFHEI SUP, PARM=(' DETAI L, ALL' ), REGI ON=512M
//STEPLI B DD DSN= CI CS. TEST. LOADLI B, DI SP=SHR
//SYSPRI NT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//DFHFLTR DD DSN=CI CS. SDFHSAMP(DFHEI DTH), DI SP=SHR <-----
//*DFHFLTR DD DSN=CI CS. TEST. JCL(DFHEI DTH), DI SP=SHR
//* MODULE LI ST, TO BE CREATED DURING SUMMARY, DETAI L RUNS.
//DFHDTL DD DI SP=OLD, DSN=CI CS. SCANOUT
//*DFHDTL DD DSN=CI CS. SCANOUT, DI SP=(NEW, CATLG, DELETE),
//* DCB=(RECFM=FB, LRECL=80, BLKSI ZE=8000), SPACE=(CYL, (1, 1))
//* PDS CONTAINING LOADMODULES TO BE SCANNED
//DFHI N DD DI SP=SHR, DSN=CI CS. TEST. LOAD

```

CICS TS 2.2 ENHANCEMENT IN CICS/DB2 INTERFACE

DB2 users can benefit from the CICS/DB2 interface performance improvements – DB2 group attach and the RMI purge option. The DB2 group attach facility allows a CICS customer to exploit the DB2 facility whereby the name may be specified as a DB2 data sharing group instead of an explicit DB2 subsystem name.

Each group can contain one or more DB2 subsystems. This simplifies Application-Owning Region (AOR) cloning within a

sysplex, and provides a greater level of availability. The CICS/DB2 attachment facility is enhanced to exploit CICS Open Transaction Environment (OTE) functions. When CICS TS V2.2 is connected to DB2 V6 or later, the attachment facility will execute using CICS open TCBs and will utilize DB2 functions to move DB2 connections and threads between TCBs. CICS/DB2 applications coded to threadsafe standards, and defined to CICS as threadsafe, may benefit from reduced TCB switching. For threadsafe applications making heavy use of EXEC SQL, the reduced TCB switching provides a significant performance improvement. A new Resource Manager Interface (RMI) purge option is introduced in CICS TS V2.2. This allows the writer of TRUE to specify whether, before calling it, the RMI should defer purge and deactivate runaways.

OTE

The CICS/DB2 attachment facility now exploits the Open Transaction Environment (OTE) to enable the CICS/DB2 task-related user exit to invoke and return from DB2 without switching TCBs. To gain the performance benefits of the OTE, CICS must be connected to DB2 Version 6 or later, and use a threadsafe application program. The CICS OTE was introduced to enable applications to perform, under an open TCB, actions that are not permitted under the CICS quasi-reentrant (QR) TCB.

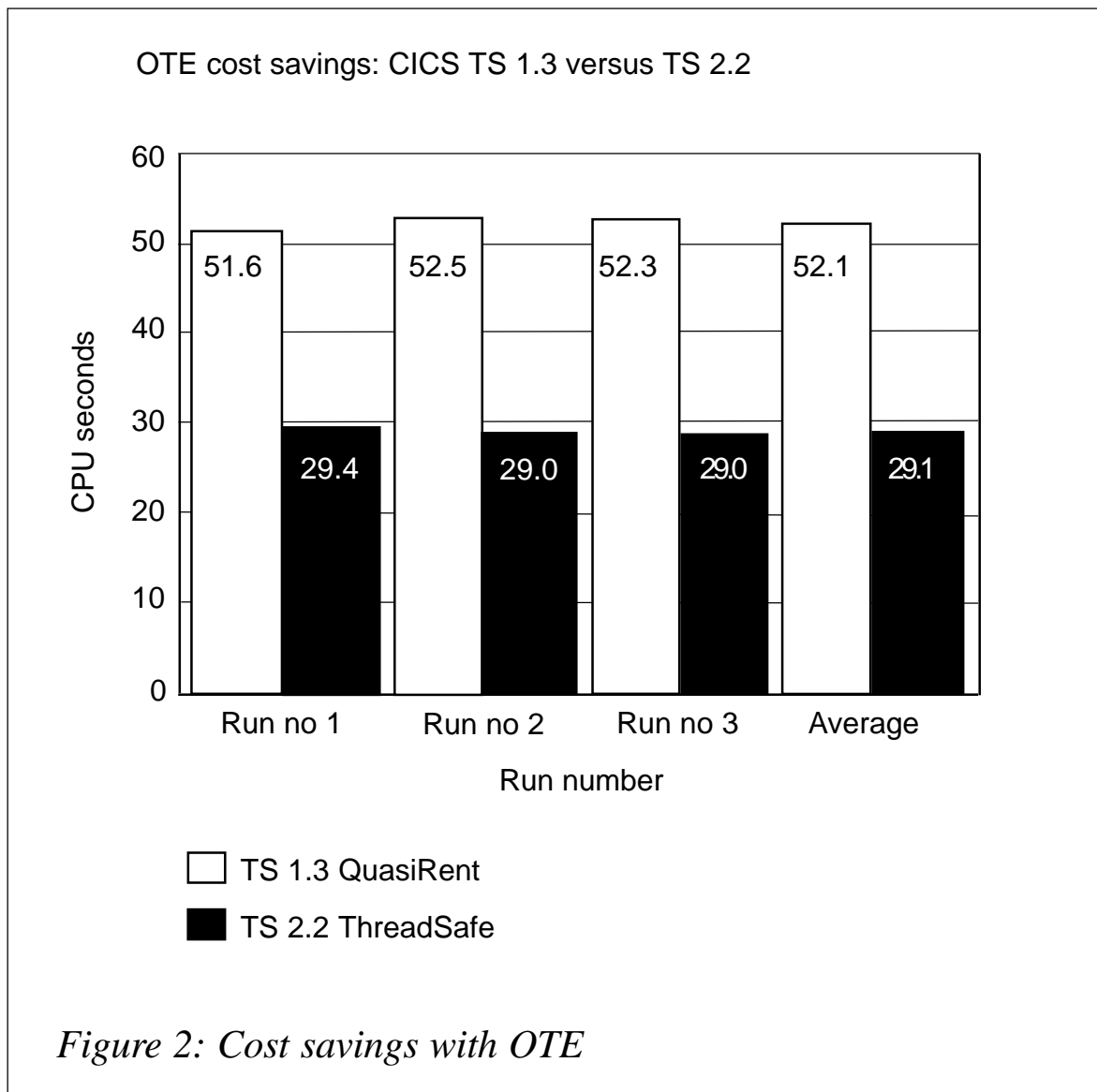
OTE was initially exploited only by Java applications that run in a Java Virtual Machine (JVM), enabling each JVM to run under its own TCB. With the new enhancements to OTE, the open TCB modes are extended to enable task-related user exits to exploit OTE and avoid the need to manage a private pool of TCBs. The CICS/DB2 adaptor is the first task-related user exit to exploit this OTE enhancement.

CICS TS V2.2 has three separate pools of open TCBs:

- L8 – TCBs allocated for non-Java programs accessing a resource manager through a task-related user exit enabled with OPENAPI option. Used by the CICS/DB2 attachment.

Note: the OPENAPI option specifies that the task-related user exit program is using non-CICS APIs. If OPENAPI is omitted, CICS will assume that the task-related user exit is either using only the CICS API, or that it performs its own TCB switch to invoke non-CICS services.

- H8 – TCBs allocated by hot-pool HPJ-compiled Java programs.
- J8 – TCBs allocated for the execution of a JVM program (Java programs that require a JVM).



CPU SAVING WITH THREADSAFE

IBM's Redbook *DB2 for z/OS and OS/390 Version 7 Selected Performance Topics* indicates that CPU saving with threadsafe may be as high as 44%, depending on environment.

Figure 2 compares the most efficient environments on TS 1.3, CONCURRENCY(QuasiRent) and TS 2.2 CONCURRENCY(THREADSAFE). These measurements were done with CICS TS 2.2 and CICS TS 1.3 using DB2 7.1.

The delta between CICS TS 1.3 and CICS TS 2.2 of step 2 then gives us the cost savings between CICS TS 1.3 and CICS TS 2.2 running a transaction which fetches 1 million rows.

The CPU cost saving is $(1 - 29.1/52.1) * 100 = 44\%$.

IBM states that the number of instructions that were given to switch TCBs to execute an SQL, which would be a round trip, was "Approximately 1.5K instructions per SQL". That is the amount you would save if your program is threadsafe.

CONCLUSION

In the absolute best case, you can save up to 44% of CPU overhead by using threadsafe. More typical CPU saving reported by customers is 10-25%. Ensuring that a program is threadsafe is difficult and may be a time-consuming effort. Threadsafe implementation requires planning and research effort, but at the same time may bring very fruitful results, depending on installation.

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

© Xephon 2003

Providing context-sensitive help in a CICS application

When we started enhancing an existing CICS application, one of

the key requirements was to provide a context-sensitive help to the users. Instead of a standard help being displayed for each of the screens, we wanted to display field-level help if the user places the cursor in any of the fields and presses PF1.

One option is to let each of the individual programs handle this. As the system was quite large, it made sense to come up with a generic solution.

The key steps involved in building this generic help solution are:

- 1 Creation of BMS maps for each of the fields.
- 2 Creation of a help control table that has the mapping between the fields and the associated help screens.
- 3 Creation of a common help screen identification routine.
- 4 Modification to the existing programs so that the common help screen identification routine is invoked when PF1 is pressed.

HELP CONTROL TABLE

The help control table has the following fields:

- Source map ID.
- Type of help (S implies screen-level/F implies field-level).
- Field name (SCREEN is used to signify screen-level help).
- Field's row position – applicable for fields only.
- Field's starting column position – applicable for fields only.
- Field's ending column position – applicable for fields only.
- Help map ID.

This control table needs to be updated whenever required – including the following conditions:

- 1 When a field's position in the map gets changed.
- 2 When a new field is added to a screen.

- 3 When field-level help is to be added/removed.
- 4 When a new screen is included in the application.

A simple ISPF-based interface was developed to add/modify/delete rows in this table.

For most of the input fields, to display the help involved describing the field and providing the validation rules associated with it. Hence the same help screen had to be displayed for a given field irrespective of the screen from which the help is requested. The field name column included in the table provides us with an option to query the table based on this column to ensure this and to get the list of exceptions to this rule.

MODIFYING EXISTING PROGRAMS

The existing programs are modified in such a way that instead of displaying the screen-level help, the help screen identification routine is invoked to determine the map to be displayed. The inputs to the routine are:

- Source map ID.
- Cursor position (CPOS from CICS).

A major factor that needs to be taken into consideration is that the existing application was designed in such a way that each program contained the logic to handle a screen and the associated help screen.

In addition to displaying the help screen based on the context, the programs were enhanced so as to emulate fully-fledged help navigation similar to that of ISPF panels. The actions taken, which depend on the AID keys pressed on the help screen, were as follows:

- F1 – Help on help.
- F3 – Return to the main screen.
- F5 – Provide screen-level help (applicable for field-level help only).

HELP SCREEN IDENTIFICATION ROUTINE

The pseudo-logic of this common help display routine is as follows:

- Get the row and column value based on the cursor position.
- Get the map ID for the screen-level help by reading the help control table record that meets *all* the following conditions (an application-level help can be returned if the screen-level help is also not available):
 - Source map ID = input map ID.
 - Type of help = S (implying screen level).
- Get the help map ID for the field from which the user has pressed the PF1 key, read the help control table record that meets *all* the following conditions:
 - Source map ID = input map ID.
 - Type of help = F (implying field level).
 - Field's starting row position = cursor row position.
 - Field's starting column position \leq cursor column position.
 - Field's ending column position \geq cursor column position.
- If a record meeting these conditions is not found, it implies either that the cursor is not placed in a field or that the field from which the user pressed PF1 doesn't have a mapping help screen. Hence the screen-level help needs to be displayed.
- Return to the calling program the map ID of the help screen to be displayed – the field or screen level – depending on the information obtained from the help control table.

CONCLUSION

Using a control table and an independent routine to do the mapping enabled us to achieve the objective of including a

context-sensitive help to an existing application with minimal intrusion. Once this process was established, the field-level helps were included on a gradual basis without having to change the source program every time. I believe that this solution is equally applicable when designing a new CICS system.

C Sasirekha
System Software Group
Tata Consultancy Services (India)

© Xephon 2003

CICS Performance Monitor for z/OS

Since the early days of CICS on S/390, performance tools have played an important part in online problem determination and offline trend analysis. Several vendors have produced such performance tools over the years.

IBM has recently introduced several CICS systems management tools. The latest addition to this family of products is CICS Performance Monitor for z/OS (CICS PM).

The remainder of this article will describe what CICS PM provides, examine the underlying CICSplex SM technology utilized by CICS PM, outline the main function provided by its Graphical User Interface (GUI), and discuss the facilities available for managing the performance of CICS TS via an Internet browser. A detailed article on the usage of the CICS PM workstation client and browser interface will be provided in a follow-on article.

WHAT IS CICS PERFORMANCE MONITOR?

CICS PM is a real-time performance monitoring, management, and troubleshooting tool for CICS TS 1.3 and CICS TS 2.2. The product, initially released in December 2002, is undergoing rapid enhancement to become a fully-capable performance management tool for CICS.

In its first release, CICS PM provides the ability to simply define, activate/deactivate, and monitor thresholds in multiple CICS systems. Thresholds can be set on any attribute of a CICS object in the CICSplex. In addition, IBM has made extensive enhancements to the CICS TS base in terms of data collection facilities and functionality. This enables you to determine what's happening in your CICS systems now. The next version will tell you what happened in the past (ie history).

CICS PM V1.2, available in 2003, will provide further enhancements to the product's data capture capabilities, access to task detail history records, and various enhancements to the CICS PM workstation client and browser-based viewsets, which provide display and navigation details of the extensive amounts of data which CICS PM can access. With CICS PM V1.2, you will be able to determine what is happening with your system at a specific time interval, ie what happened two hours ago. Note that CICS PM does not intend to provide access to offline historical data that is primarily used for long-term trending and report generation. CICS PM's complementary product, CICS PA, provides such facilities.

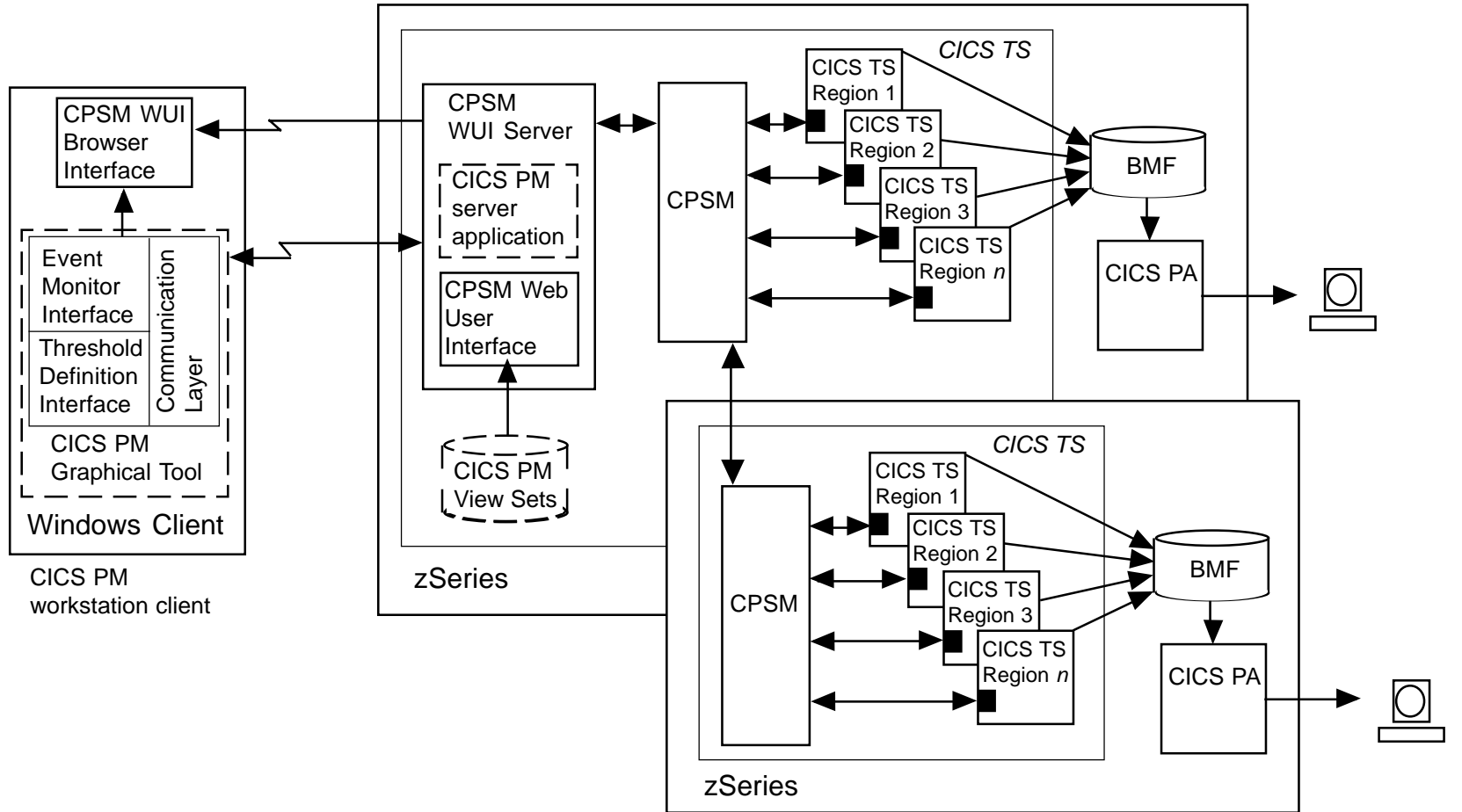
CICSPLEX SYSTEM MANAGER RECAP

CICS PM is built on the facilities provided by CICSplex System Manager (CICSplex SM or CPSM), a component of CICS Transaction Server. As such, it provides the same unique abilities to view and manage CICS resources across the CICSplex in a single system image from a single point of control.

CICSplex SM provides systems management for various levels of CICS TS, along with CICS NT and CICS OS/2. It provides the following interfaces to its systems management function:

- TSO End User Interface (EUI).
- Web User Interface (WUI) via the user's Internet browser of choice.
- Application Programming Interface (API), accessible from CICS, z/OS batch, TSO, and NetView in a variety of languages

Figure 1: CICSplex SM



CICS PM components are shown within dashed lines

such as Assembler, COBOL, C, and REXX.

Single system image management is available for the following range of tasks:

- Business Application Services – a super CICS Resource Definition Online (RDO) interface that allows you to define and install CICS resources.
- Operations – a super CEMT for CICS operational control.
- Monitoring – access to cached monitoring and statistical data.
- Threshold analysis – a sophisticated facility that allows you to specify thresholds against any CICS resource attribute. You can combine multiple thresholds using Boolean algebra to be timer activated, along with various aggregation techniques, etc.
- Automation – both internally via the definition of actions to be taken against the CICS systems, and externally via event notification to an automation product such as NetView (ie for subsequent resolution through the CICSplex SM API interface).
- Dynamic Workload Management – the ability to route various types of work requests to CICS regions in order to optimize throughput or achieve response time goal criteria.
- Integration into the Tivoli suite of systems management products (such as Tivoli Business System Manager).

As you can see, CICSplex SM provides many facilities for systems management. CICS PM utilizes the WUI technology, single system image, operations, and thresholding capabilities (in a simplified form). In order to exploit the facilities of CICS PM, you must install CICSplex SM and have it executing on your systems. However, this does not mean that you have to configure all of CICSplex SM's facilities in order to use it – you need only to establish the components that CICS PM exploits.

In order to permit users to rapidly exploit the capabilities of CICS

PM, IBM provides a simplified set-up procedure. CICS PM uses familiar CICS terminology in its interfaces and publications rather than CICSplex SM terminology. In addition, CICS PM provides a single publication focused on non-CICSplex SM users. Reference to the CICSplex SM publications is not required in order to use the product.

Here are two examples of facilities that simplify the installation procedure:

- The batch procedure, CMZBATCH, simplifies the population of the CICSplex SM data repository.
- The WUI server viewset auto import facility eliminates the need to sign on to the WUI server to manually import viewsets for end user usage.

The overall architecture is displayed in Figure 1. CICS PM components are shown within dashed lines.

BASE CICS TS ENHANCEMENTS

In order to provide the data facilities of CICS PM, IBM made enhancements to base CICS TS to allow users to gather statistics and browse domain manager subpools, and identify the current program executing for a CICS task. For example, additions to CICS monitoring provide application naming support, a new CMF resource class provides file by task support, new statistics for MVS Workload Manager, and various other additions to the CICS SPI and statistics (support for these features was also added to CICS PA via PTF).

APPLICATION NAMING SUPPORT

Application naming support provides the ability to tag CMF data with a unique token, which allows more granular interpretation of the data, and provides a common key on which to aggregate this data. This token takes the form of a 'transaction id' and 'program name'. Utilization of this token allows more granular identification of a transaction, or to relate individual transactions into a single application name.

Although these fields are referred to as tranid and program, they can be any unique set of characters for identifying any application. The support uses standard user Event Monitoring Points (EMPs) via EXEC CICS MONITOR ENTRYNAME() POINT() DATA1() DATA2(). However, unlike other user data added via EMPs, the application naming data is preserved across performance record output(s).

Two 'special' EMPs are defined that can be used by user or ISV applications in any combination.

Support is also provided for CICS PA Release 2 for usage in report forms, list, and summary reports.

CICSPLEX SM ENHANCEMENT

IBM made several changes to support the collection and incorporation of new data into base tables. In addition, they

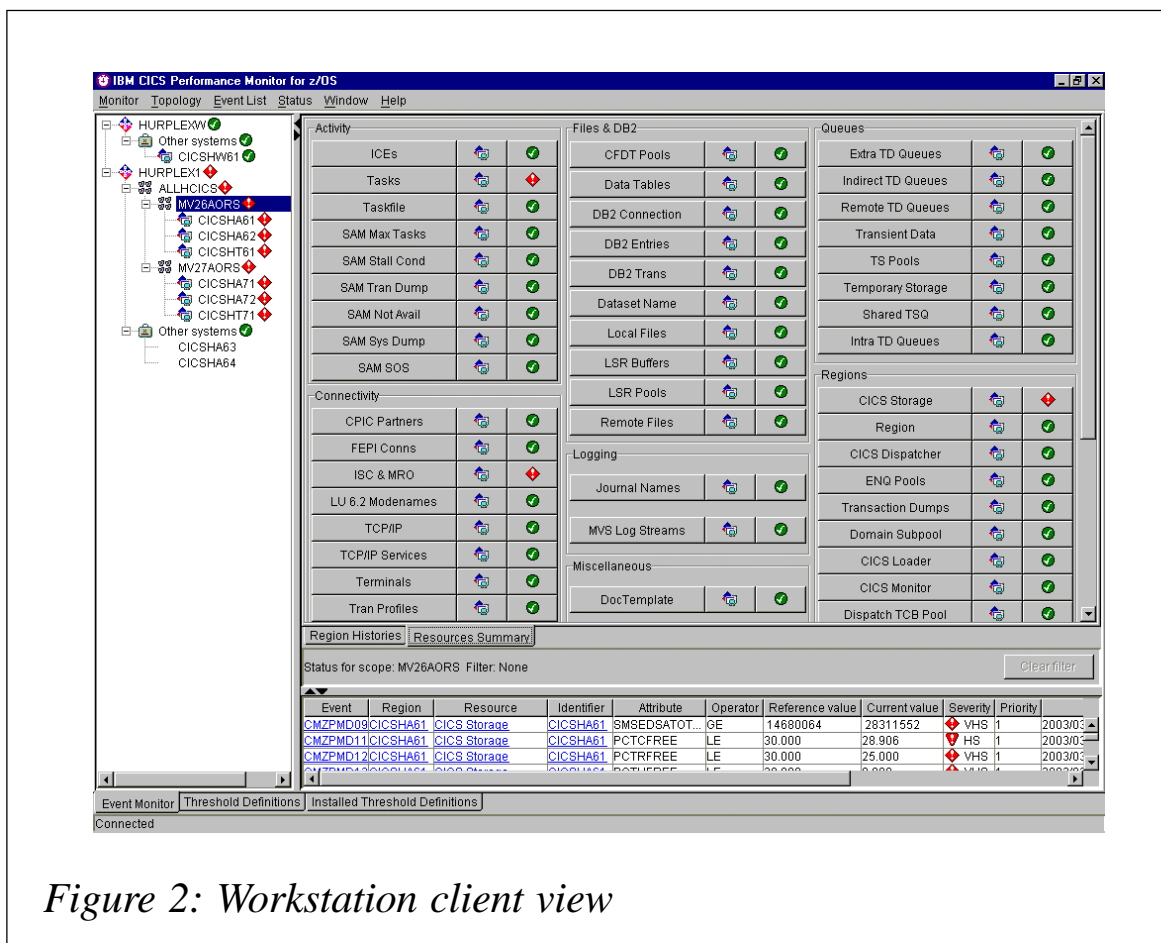


Figure 2: Workstation client view

added functional support for the event monitor GUI by providing a documented interface utilizing URL encoding for the following:

- Single sign-on
- GET data
- GETDEF data.

Single sign-on is the ability to sign on (using userid and password) to the CICS PM client and then to launch instances of a browser to obtain detailed information without additional sign-on requests. This is implemented using RACF passticket support. When a browser instance is about to be launched, a RACF passticket is obtained from the server and then presented to the server with the target URL.

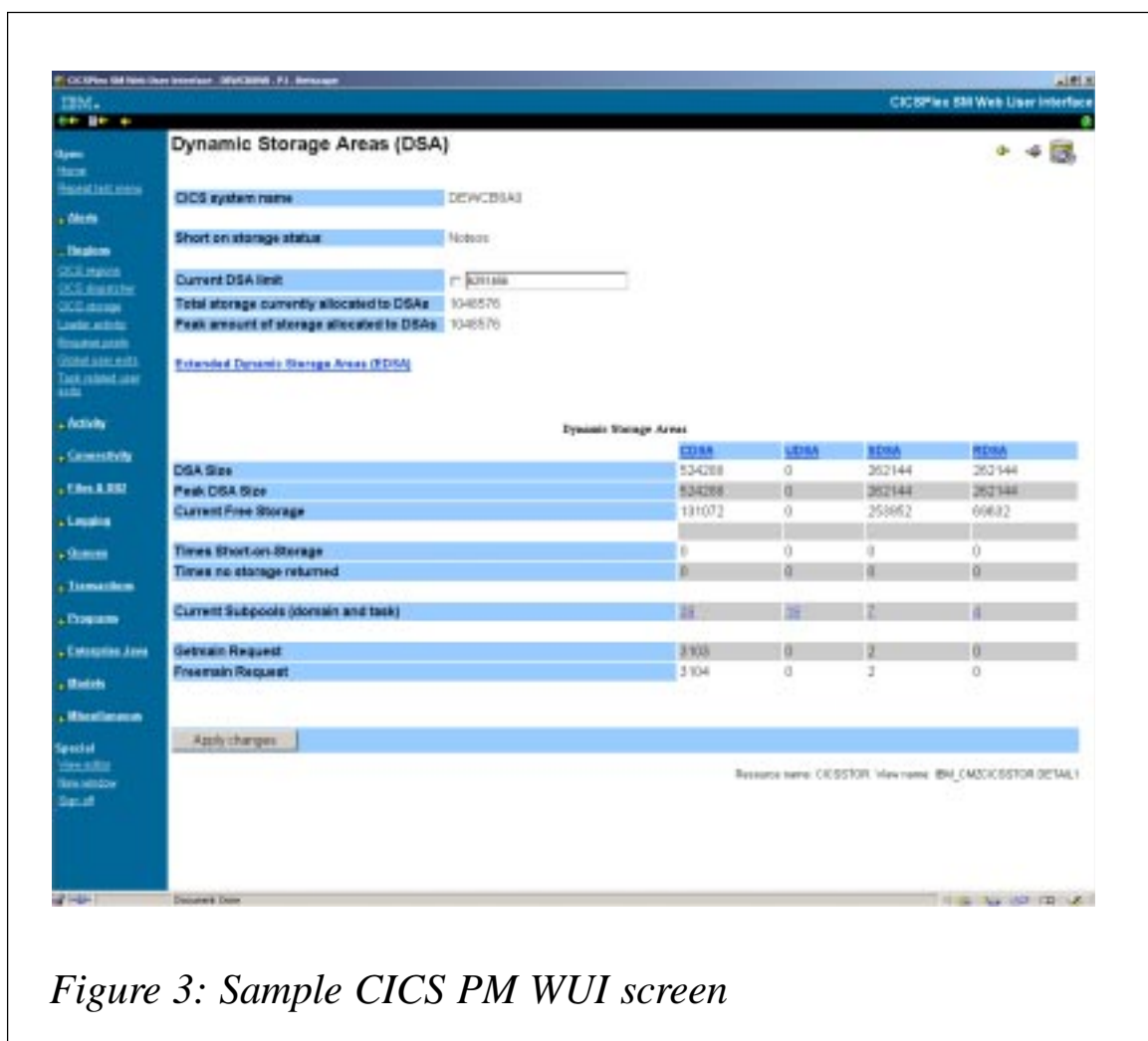


Figure 3: Sample CICS PM WUI screen

GET data encoding provides a response to data requests, which is in a form more appropriate to programmatic access; basically an &-delimited set of keywords and keyvalues, rather than HTML. GETDEF data provides a similar programmatic interface to object meta data (ie data about the object classes).

IBM also introduced asynch GET support to allow concurrent access to multiple classes of data. Additional layout form types were also provided for the WUI such as matrix layout support.

CICS PM WORKSTATION CLIENT

The CICS PM workstation client is a Java swing application that relates scopes – CICS regions, resource classes, and events. The workstation client also provides facilities for the definition/modification of thresholds, and can launch the WUI to view detailed information relative to the event being investigated.

Context and scope are automatically assigned by cursor positioning on the topology pane, and WUI launch can also be to the resourceclass or the resourceclass and instance (eg PROGRAM(*) or PROGRAM(ABC*)). When navigating from the client to the WUI, no additional sign-on of the user is required because the client/WUI utilizes RACF passticket support.

A Java Runtime Environment (JRE) is required on the client workstation, and the client code is installable on Windows using the Microsoft Software Installer (MSI) in either attended or unattended modes (Windows NT4 service pack 6; Windows 2000 and XP are supported). The main workstation client view is shown in Figure 2.

A supporting WUI server client application is also provided to process client requests.

Sample threshold definitions are provided as part of the installation package.

CICS PM WEB USER INTERFACE VIEWSETS

So far, we have concentrated on threshold event management.

However, customers spend a great deal of time looking at detailed information about the current state of their CICS systems. It is not surprising, then, that CICS PM provides access to an extensive array of information regarding CICS systems and their resources. This information is displayed to the end user when he or she launches his or her Internet browser to the WUI server. A sample CICS PM WUI screen is shown in Figure 3.

Examples of the types of data accessible by CICS PM

The CICS PM View sets provide real-time access to:

- All CICS system performance data.
- Resource-related performance data.
- Task-level performance data collected by the CICS monitoring facility (CMF), including file resource usage by task.
- Allow operational changes to key system and resource parameters:
 - Maxtask limit, Dynamic Storage Area (DSA) limits.
 - Dispatcher settings, purging or force purging tasks.
 - Delete temporary storage queues, opening/closing files.
 - Discard unwanted CICS resource definitions.
- System:
 - CICS regions.
 - CICS storage, dispatcher, loader,
- Resources:
 - Programs, files, connections,
 - Logstreams and journals, DB2 entries,
 - Temporary storage queues, transient data queues,
- Tasks:

- Task identification information,
- Task status, suspend information,
- Unit-of-work analysis including enqueues, dataset name failures,
- CICS system-level viewsets:
 - CICS region.
 - CICS storage (DSAs and subpool usage).
 - Transaction manager.
 - CICS dispatcher.
 - Loader.
 - Temporary storage, including shared temporary storage pools.
 - Transient data.
 - DB2 connection.
 - Recovery manager, including unit-of-work analysis.
 - Enqueue pools.
 - JVMpool.
 - User exits – global and task related.
- CICS Application resource level viewsets.
 - Transactions.
 - Transaction classes.
 - Programs.
 - Files, including data tables, dataset names, and local shared resource (LSR) pools.
 - Connections.
 - Logstreams and journals.

- Temporary storage queues.
- Transient data queues.
- Terminals.
- DB2 entries.
- System and transaction dumps.
- Enterprise Java resources.

Other viewsets:

- Access to the performance information about all the active tasks in the CICS systems:
 - Task identification information,
 - Task status, suspend information,
 - Unit-of-work (UOW) analysis.
 - CPU usage,
 - Storage usage - task, program, shared.
 - Program, file, temporary storage requests, and others
....
- Extensive range of task level views containing all CMF performance data – including file resource usage by individual task.
- Outstanding events together with threshold definitions.

Of course, all of this data is available via the CICSplex SM API.

Performance overheads

The performance costs of using CICS PM vary with utilization of function.

Existence cost

The existence cost of CICS PM is the same as for CICSplex SM. This is essentially heartbeat processing in the CICS region

agents, which provide basic data about the corresponding CICS address space. This cost is extremely low.

Client and browser connect

When the client connects to the WUI server, the client requests topology data from the server (CPLEXDEF, CSYSDEF, and CSYSGROUP records). EVENT and EVENTDTL records are also retrieved to populate views with active events.

Threshold definition management

For definition manipulation, CICSplex SM API requests are generated to make the appropriate changes on the CPSM data repository (EYUDREP).

For install and discard, the server program issues commands as outlined above, which affect EYUDREP and the CICSplex SM runtime (via RTAACTV).

Active threshold execution

Agents in the CICS regions sample the requested data at the requested frequency. Obviously the more data sampled, and the more frequent the sampling rate, the more CPU cycles are expended. The CMAS evaluates the thresholds to determine whether they are true or false. If true then an event is raised; if false then a resolution is raised if the event was previously true.

Browser request for data

The request for data at the browser results in an architected URL being driven to the WUI server. WUI server transforms request for data to CICSplex SM API request. CMAS processes request by identifying CICS regions to request data from. It then drives requests to the agents in those regions to gather the information at the CMAS(s) concerned, who return that data to the requesting CMAS. The WUI server then combines the data with the viewset definitions, building HTML for presentation back at the browser.

SUMMARY

CICS PM helps you improve CICS system availability while reducing system management costs. Its workstation client allows you to detect early performance problems and provides easy-to-use, intuitive, cross-system navigation for fast problem determination.

The browser interface enables you to access all CICS systems and resource information, and make changes to key system and resource parameters.

CICS PM is built on standard APIs and proven CICS TS system management technologies, and provides out-of-the-box installation and set-up, with comprehensive support for the latest CICS TS functions.

REFERENCES

CICS Performance Monitor for z/OS, 5655-146. For more information, visit www.ibm.com/cics/products/tools.html.

Redbook: *IBM Tools: CICS Performance Monitor V1,1* 6972.

GLOSSARY

Context – a name given to a set of logically related CICS systems. For example, a TESTPLEX or PRODUCTIONPLEX.

Scope – a name given to a logical subset of CICS systems within a Context. For example, TORSONQA, PAYRAORS.

Dr Paul Johnson

CICS Transaction Server Systems Management Planning/Development

IBM Hursley (UK)

© IBM 2003

CICS file browse

I support the mainframe interfaces for an application that uses a Web-based program to integrate and enhance the functions of several mainframe legacy applications. This application utilizes MQSeries for messaging and the 3270 Bridge to 'screen scrape' parts of the legacy application. In order to resolve problems relating to the mainframe interfaces, it was necessary to log every message coming and going to and from the Web application and each message to and from the 3270 Bridge. Every time an MQ put or get is done, a header containing time, date, and source information is appended to the MQ message area and the entire record area is written to an ESDS and used as a message log file. This ESDS file contains variable records that can vary from 109 bytes to 30,029 bytes. Because part of the application utilizes 'screen scraping' the record can contain virtually any combination of data. We required the ability to browse the file under CICS in order to analyse problems within the interfaces.

PROGRAM OVERVIEW

I wrote the program described in the following text to allow programmers and technical users to be able to browse the application error log file. This program allows most of the functionality of the PDF browse and uses similar keys and commands in order to make it easy to use. This program was written to read an ESDS dataset. I believe that it could be easily modified to read a KSDS in the same manner.

This program allows the user to scroll up, down, left and right by page, by an amount entered in the command field, or by maximum. TOP and BOTTOM are also recognized, and scroll to the beginning and end of the file respectively. There are also commands that relate to a specific data area peculiar to my application's log file and can be used as an example that can be modified to locate other fixed areas in a record. The program allows the data to be displayed in hex format and a FIND command is also provided.

Normal display mode is shown below:

```

                                Aces Error Log Browse
Start Key: 00000000                                Di sp: 00001
End Key: 00019B71

Command ==>
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----
TC1B    20030428 09:16:17:89>> @TC1G09 WHSRI 01.....06                ....
TC1B    20030428 09:16:33:18>> @TC1P09 WHSRI 010000010005146106          0....
TC1A    20030428 13:24:00:83>> @TC1G13 WHSAI 00000100050734...         ....
TC2     20030428 13:24:01:73>> @11813  CIH .....
TC2     20030428 13:24:01:73>> @11813MDMD ..... MQCI CS
TC2     20030428 13:24:04:23>> @11813  CIH ..... CSQCBDCO.....
TC2     20030428 13:24:04:23>> @11813MDMD ..... MQCI CS
TC2     20030428 13:24:04:23>> @11813  CIH .....
TC2     20030428 13:24:04:23>> @11813MDMD ..... MQCI CS
TC2     20030428 13:24:08:13>> @11813  CIH ..... CSQCBDCO.....
TC2     20030428 13:24:08:13>> @11813MDMD ..... MQCI CS
TC2     20030428 13:24:08:13>> @11813  CIH .....
TC2     20030428 13:24:08:13>> @11813MDMD ..... MQCI CS
TC2     20030428 13:24:12:79>> @11813  CIH ..... CSQCBDCO.....
TC2     20030428 13:24:12:79>> @11813MDMD ..... MQCI CS
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----
PF2 = HEX, PF5 = RFIND, PF7 = UP, PF8 = DOWN, PF10 = LEFT, PF11 = RIGHT
                                Press CLEAR to Exit
```

The only unprotected field is the command line. Several commands can be entered and some have acceptable abbreviations. The parsing routines in the 0600 series paragraphs determine which action to take based on what was entered on the command line. The user interface also uses PF keys alone or in combination with a command argument. The start key and end key fields show the starting and ending RBA of the records displayed. The disp field is the horizontal byte offset of the data being displayed. Two ruler lines are displayed, one each above and below the data display. The lower ruler line is overwritten by error messages from the program, and the last line contains an area for informational messages. The normal display mode shows 79 bytes of the current 15 records with any non-displayable characters replaced by a period. In hex mode, the display changes to show five records in 15 lines. The 15 lines consist of a display representation of the record followed by two lines showing the hex representation of the data.

The hex display mode is shown below:

Aces Error Log Browse

Start Key: 00000000

Di sp: 00001

End Key: 00017632

Command ==>

```
-----1-----2-----3-----4-----5-----6-----7-----
TC1B    20030428 09:16:17:89>> @TC1G09 WHSRI 01.....06      ....
ECFC4444FFFFFFFF4FF7FF7FF7FF6647ECFCFF4ECEDCFF000000000000000FF4444444444440000
3312000020030428009A16A17A89EE0C33170906829901000000000000000060000000000000000
TC1B    20030428 09:16:33:18>> @TC1P09 WHSRI 010000010005146106      0...
ECFC4444FFFFFFFF4FF7FF7FF7FF6647ECFDFF4ECEDCFFFFFFFFFFFFFFFFF444444444444F000
3312000020030428009A16A33A18EE0C3317090682990100000100051461060000000000000000
TC1A    20030428 13:24:00:83>> @TC1G13 WHSAI 00000100050734..      ....
ECFC4444FFFFFFFF4FF7FF7FF7FF6647ECFCFF4ECECC44FFFFFFFFFFFFFFFF00444444444440000
3311000020030428013A24A00A83EE0C3317130682190000000100050734000000000000000000
TC2     20030428 13:24:01:73>> @11813  CIH .....
ECF44444FFFFFFFF4FF7FF7FF7FF6647FFFFF44CCC40000000B00000000444444400000000000
3320000020030428013A24A01A73EE0C1181300398000020004000000000000000000000000
TC2     20030428 13:24:01:73>> @11813MDMD .....MQCICS
ECF44444FFFFFFFF4FF7FF7FF7FF6647FFFFFD CDC440000000000000000000000000000000000
3320000020030428013A24A01A73EE0C1181344400000100000008FFFFFF00000031000048393200
-----1-----2-----3-----4-----5-----6-----7-----
PF2 = HEX, PF5 = RFIND, PF7 = UP, PF8 = DOWN, PF10 = LEFT, PF11 = RIGHT
Press CLEAR to Exit
```

PROGRAM LOGIC

If the COMMAREA contains data from the last execution, a RECEIVE MAP is done and the command line is checked for data. If data has been entered, the command is validated and parsed. The command can be entered anywhere on the line so the data entered must be left justified by the program. Many of the commands can have an abbreviation such as F or FIND, which both activate the find process. Some of the commands have arguments. For example a FIND command can specify a start column and/or the prev operand. The parsing routines must take these factors into account and set flags that control processing.

The main process paragraph (0200-PROCESS) evaluates the AID key and performs the appropriate routine. If *Enter* was pressed, each of the mode flags that can be applied to *Enter* are checked and the appropriate action taken. If none of the flags are

set, the logic just drops out the end of the paragraph and the file is re-read using the same keys and the same data is displayed. The PF2 key causes the hex mode to be toggled. PF5 causes the find process to be initiated again. The PF7 key will scroll backward by amount, maximum, or, if no command was entered, one screen of data. PF8 has the same effect as PF7, except that it will scroll forward. The PF10 and PF11 keys scroll the data left and right respectively, by amount, maximum, or one screen full of data.

If any command or PF key causes a search to be done (eg PF5, MAX, TOP, etc), the file is positioned as needed. If a find is being done, the file is read and an attempt is made to match the data entered in the find string. If a column was specified, an attempt is made to match only data beginning in that column for the length of the string. If no column is entered, the records are read and examined for the string by incrementing the offset one byte at a time. If PF5 was pressed, the find starts at the record and offset where it left off. If the column was entered, the next record is read. If the column is not entered, the offset is incremented by 1 and the record is examined to find the string.

After the AID key has been processed, the file is read either forwards or backwards, depending on the direction that has been set. If no direction was set, the direction stored in the COMMAREA from the last execution is used.

As each record is read, all bytes are moved one at a time from the current displacement within the record to the screen area. Each byte is checked to be sure it is a displayable character and, if not, it is replaced by a period (full stop). If hex mode is on, the byte is also translated into the display representation of its hex value. This logic loops until end of file, or the screen contains 15 records in normal mode or five records in hex mode.

At this point, the correct records have been read and the display area of the map has been built. Before the map is sent, the program calculates the correct ruler numbers to be displayed. The ruler is used in conjunction with the disp field at the top of the map. For example, if the disp is 123 the ruler would begin with -

--+----3--.... If any error was detected, the lower ruler line is overlaid with the error message highlighted.

DSOU0902 SOURCE CODE

```
IDENTIFICATION DIVISION.
PROGRAM-ID.      DSOU0902.
AUTHOR.          HCABLE.
DATE-WRITTEN.    FEBRUARY 2003.
DATE-COMPILED.

*****
** This program browses the Data Standardization error log file.*
** It will scroll forward and back through the file and left   *
** right through the records. Most of the functions of the TSO *
** browse (find, scroll by amount, hex) are included.          *
**                                                             *
** The COMMAREA contains start and end keys and the current   *
** state. No attempt is made to save the current records. If  *
** the operation is a horizontal scroll, the records are       *
** read and the correct section displayed.                    *
**                                                             *
** Valid commands are:                                       *
** -----*
** F or FIND - string, column, PREV (used with Enter)        *
** M or MAX - left, right, up, down (used with PF7, PF8, PF10,*
**             PF11)                                          *
** ## (of characters or lines) - left, right, up, down (used *
**             with PF7, PF8, PF10, PF11). Valid values are 1 *
**             to 999999.                                     *
** TOP - go to top of file (used with enter)                  *
** BOT or BOTTOM - go to the bottom of the file (used with   *
**             enter)                                         *
** T or TIME - match on time or a portion of time (used with *
**             enter). The time value can be entered with or  *
**             without colons and any part (HH or HHMM or     *
**             HHMMSS...). The program assumes that each of the *
**             time divisions is filled in (ie 08 must be     *
**             entered for 8 am.)                             *
** TASK - match on task number (used with enter). The task   *
**             number can be entered with or without leading  *
**             zeroes.                                         *
** HEX or HEX ON or HEX OFF - HEX by itself will toggle the  *
**             current mode (the same as PF2). HEX ON will turn *
**             HEX on if it is off or will leave it on if it is *
**             already on. HEX OFF works the opposite of HEX ON.*
**                                                             *
** PF key usage:                                             *
** -----*
** PF2 - switch hex on/off                                   *
```

```

**      PF5 - repeat the last find command      *
**      PF7 - scroll the file backward          *
**      PF8 - scroll the file forward           *
**      PF10 - scroll the records left          *
**      PF11 - scroll the records right         *

```

```

*****
*****

```

MAINTENANCE LOG

```

*
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.      IBM-S390.
OBJECT-COMPUTER.     IBM-S390.
*

```

```

DATA DIVISION.
WORKING-STORAGE SECTION.

```

```

01  WS-MISC.
    03  WS-RESP-CODE          PIC S9(8)  COMP VALUE ZERO.
    03  WS-RESP-CODE-DISPLAY PIC S9(8).
    03  WS-PROCESS-FLAG      PIC X    VALUE 'N'.
        88  WS-88-PROCESS-ERROR VALUE 'Y'.
    03  WS-ERRLOG-FLAG       PIC X    VALUE 'N'.
        88  WS-88-ERRLOG-EOF    VALUE 'Y'.
    03  WS-ERRLOG-KEY        PIC 9(8)  COMP.
    03  WS-LOG-RECORD        PIC X(32000).
    03  WS-LOG-RECORD-LENGTH PIC 9(4)  COMP.
    03  WS-READ-COUNT        PIC 99   VALUE ZERO.
    03  WS-SUB                PIC 99   VALUE ZERO.
    03  WS-CHAR-SUB          PIC 99   VALUE ZERO.
    03  WS-CMMND-SUB         PIC 99   VALUE ZERO.
    03  WS-HOLD-LINE         PIC X(79).
    03  FILLER REDEFINES WS-HOLD-LINE.
        05  WS-LOG-BYTE      PIC X    OCCURS 79 TIMES.
    03  WS-DISPLAY-LINE     PIC X(79).
    03  FILLER REDEFINES WS-DISPLAY-LINE.
        05  WS-DISPLAY-CHAR  PIC X    OCCURS 79 TIMES.
    03  WS-HEX1-LINE        PIC X(79).
    03  FILLER REDEFINES WS-HEX1-LINE.
        05  WS-HEX1-CHAR     PIC X    OCCURS 79 TIMES.
    03  WS-HEX2-LINE        PIC X(79).
    03  FILLER REDEFINES WS-HEX2-LINE.
        05  WS-HEX2-CHAR     PIC X    OCCURS 79 TIMES.
    03  WS-RULER-DISPLAY.
        05  FILLER           PIC X(25) VALUE
            '-----1-----2-----+'.
        05  FILLER           PIC X(25) VALUE
            '----3-----4-----5'.
        05  FILLER           PIC X(25) VALUE
            '-----6-----7-----+'.
        05  FILLER           PIC X(25) VALUE

```

```

      '----8-----+-----9-----+-----0'.
03 WS-ERROR-DI SPLAY          PIC X(79) VALUE SPACES.
03 WS-ERROR-FLAG              PIC X      VALUE 'N' .
      88 WS-88-DI SPLAY-ERROR          VALUE 'Y' .
03 WS-QUOTI ENT              PIC S9(5) COMP.
03 WS-REMAI NDER             PIC S9(5) COMP.
03 WS-RULER-START            PIC 999    VALUE ZERO.
03 WS-RULER-START-LENGTH    PIC 999    VALUE ZERO.
03 WS-RULER-END-LENGTH      PIC 999    VALUE ZERO.
03 WS-RULER-OFFSET          PIC 999    VALUE ZERO.
03 WS-SCROLL-FLAG           PIC X      VALUE 'N' .
      88 WS-88-SCROLLING              VALUE 'Y' .
03 WS-CHAR-TEST              PIC X.
      88 SPECIAL-CHAR              VALUES ARE
          '\', '~', '!', '@', '#', '$', '%', '^',
          '&', '*', '(', ')', '-', '_', '=', '+',
          '[', '{', ']', '}', '\\', '|', ';', ':',
          ',', '<', '>', '/', '?', '"', QUOTE.
03 WS-COMMAND-LI NE.
      05 WS-COMMAND-CHAR      PIC X OCCURS 50 TIMES.
03 WS-HOLD-COMMAND-LI NE    PIC X(50).
03 WS-CMMND-START           PIC 99     VALUE ZERO.
03 WS-CMMND-END             PIC 99     VALUE ZERO.
03 WS-CMMND-LENGTH         PIC 99     VALUE ZERO.
03 WS-AI D                  PIC X      VALUE ' ' .
      88 WS-88-SCROLL-KEY      VALUES ARE
          '7', '8', ':', '#'.
03 WS-SCROLL-TYPE          PIC X      VALUE ' ' .
      88 WS-88-SCROLL-MAX      VALUE 'M' .
      88 WS-88-SCROLL-TOP      VALUE 'T' .
      88 WS-88-SCROLL-BOT      VALUE 'B' .
      88 WS-88-SCROLL-TIME     VALUE 'X' .
      88 WS-88-SCROLL-TASK     VALUE 'K' .
      88 WS-88-SCROLL-AMT      VALUE 'A' .
      88 WS-88-SCROLL-FI ND    VALUE 'F' .
03 WS-SEARCH-TIME          PIC X(8)   VALUE '00:00:00' .
03 WS-SEARCH-TIME-NUMERI C  PIC 9(6)  VALUE ZEROES.
03 WS-SEARCH-TIME-START    PIC 9      VALUE ZERO.
03 WS-SEARCH-TIME-LENGTH  PIC 99     VALUE ZERO.
03 WS-TIME-SUB             PIC 99     VALUE ZERO.
03 WS-SEARCH-TASK          PIC 9(14)  VALUE ZEROES.
03 WS-TASK-OFFSET          PIC 99     VALUE ZERO.
03 WS-SEARCH-TASK-LENGTH  PIC 99     VALUE ZERO.
03 WS-SCROLL-AMT           PIC 9(6)   VALUE ZEROES.
03 WS-AMT-OFFSET          PIC 99     VALUE ZEROES.
03 WS-DI SP-CALC           PIC 9(5)   VALUE ZEROES.
03 WS-FI ND-START         PIC 99     VALUE ZEROES.
03 WS-FI ND-SUB           PIC 99     VALUE ZEROES.
03 WS-QUOTE-FLAG          PIC X      VALUE 'N' .
      88 WS-88-QUOTED-STRI NG    VALUE 'Y' .

```

	88	WS-88-UNQUOTED-STRING	VALUE 'N'.
	Ø3	WS-TALLY	PIC S9(4) COMP VALUE ZEROES.
	Ø3	WS-UNSTRING-AREA	PIC X(5Ø).
	Ø3	WS-COUNT	PIC 999.
	Ø3	WS-POINTER	PIC 999.
	Ø3	WS-DELIMITER	PIC X VALUE SPACE.
	Ø3	WS-OFFSET	PIC 99 VALUE ZEROES.
	Ø3	WS-HEX-ZERO	PIC X VALUE X'ØØ'.
	Ø3	WS-STRING-FLAG	PIC X VALUE 'N'.
	88	WS-88-STRING-FOUND	VALUE 'Y'.
	Ø3	WS-FIND-DISP-MAX	PIC 9(5) VALUE ZEROES.
Ø1		WS-HEX-CONVERSION-AREA.	
	Ø3	WS-HOLD-RBA	PIC X(4).
	Ø3	WS-DISPLAY-RBA	PIC X(8).
	Ø3	WS-HOLD-BYTE	PIC X.
	Ø3	WS-DISPLAY-BYTE	PIC XX.
	Ø3	WS-CSUB1	PIC S9(4) COMP.
	Ø3	WS-CSUB2	PIC S9(4) COMP.
	Ø3	WS-HEX.	
Ø341ØØ	Ø5	WS-LEFT	PIC X.
Ø342ØØ	Ø5	WS-RIGHT	PIC X.
Ø343ØØ	Ø3	WS-CHAR2 REDEFINES WS-HEX	PIC S9(3) COMP-3.
Ø344ØØ	Ø3	WS-BIN REDEFINES WS-HEX	PIC S9(4) COMP.
Ø345ØØ			
Ø346ØØ	Ø3	WS-CHAR3	PIC 9(3).
		COPY DFHAID.	
		*COPY DSMØ9Ø2.	
		* DSMØ9Ø2 MAP AREA.	
Ø1		WS-DSMØ9Ø2-MAP.	
	Ø3	WS-MAP-TIOA-PREFIX	PIC X(12).
		*	
	Ø3	WS-MAP-STRBA-L	PIC S9(4) COMP.
	Ø3	WS-MAP-STRBA-A	PIC X.
	Ø3	WS-MAP-STRBA-C	PIC X.
	Ø3	WS-MAP-STRBA-P	PIC X.
	Ø3	WS-MAP-STRBA-H	PIC X.
	Ø3	WS-MAP-STRBA-V	PIC X.
	Ø3	WS-MAP-STRBA	PIC X(8).
		*	
	Ø3	WS-MAP-DISP-L	PIC S9(4) COMP.
	Ø3	WS-MAP-DISP-A	PIC X.
	Ø3	WS-MAP-DISP-C	PIC X.
	Ø3	WS-MAP-DISP-P	PIC X.
	Ø3	WS-MAP-DISP-H	PIC X.
	Ø3	WS-MAP-DISP-V	PIC X.
	Ø3	WS-MAP-DISP	PIC X(5).
		*	
	Ø3	WS-MAP-ENDRBA-L	PIC S9(4) COMP.
	Ø3	WS-MAP-ENDRBA-A	PIC X.
	Ø3	WS-MAP-ENDRBA-C	PIC X.

	Ø3	WS-MAP-ENDRBA-P	PIC X.	
	Ø3	WS-MAP-ENDRBA-H	PIC X.	
	Ø3	WS-MAP-ENDRBA-V	PIC X.	
	Ø3	WS-MAP-ENDRBA	PIC X(8).	
*				
	Ø3	WS-MAP-CMMND-L	PIC S9(4)	COMP.
	Ø3	WS-MAP-CMMND-A	PIC X.	
	Ø3	WS-MAP-CMMND-C	PIC X.	
	Ø3	WS-MAP-CMMND-P	PIC X.	
	Ø3	WS-MAP-CMMND-H	PIC X.	
	Ø3	WS-MAP-CMMND-V	PIC X.	
	Ø3	WS-MAP-CMMND	PIC X(50).	
*				
	Ø3	WS-MAP-RULER1-L	PIC S9(4)	COMP.
	Ø3	WS-MAP-RULER1-A	PIC X.	
	Ø3	WS-MAP-RULER1-C	PIC X.	
	Ø3	WS-MAP-RULER1-P	PIC X.	
	Ø3	WS-MAP-RULER1-H	PIC X.	
	Ø3	WS-MAP-RULER1-V	PIC X.	
	Ø3	WS-MAP-RULER1	PIC X(79).	
*				
	Ø3	WS-MAP-LINES OCCURS 15 TIMES.		
	Ø5	WS-MAP-LINE-L	PIC S9(4)	COMP.
	Ø5	WS-MAP-LINE-A	PIC X.	
	Ø5	WS-MAP-LINE-C	PIC X.	
	Ø5	WS-MAP-LINE-P	PIC X.	
	Ø5	WS-MAP-LINE-H	PIC X.	
	Ø5	WS-MAP-LINE-V	PIC X.	
	Ø5	WS-MAP-LINE	PIC X(79).	
*				
	Ø3	WS-MAP-RULER2-L	PIC S9(4)	COMP.
	Ø3	WS-MAP-RULER2-A	PIC X.	
	Ø3	WS-MAP-RULER2-C	PIC X.	
	Ø3	WS-MAP-RULER2-P	PIC X.	
	Ø3	WS-MAP-RULER2-H	PIC X.	
	Ø3	WS-MAP-RULER2-V	PIC X.	
	Ø3	WS-MAP-RULER2	PIC X(79).	
*				
	Ø3	WS-MAP-MESSAGE-L	PIC S9(4)	COMP.
	Ø3	WS-MAP-MESSAGE-A	PIC X.	
	Ø3	WS-MAP-MESSAGE-C	PIC X.	
	Ø3	WS-MAP-MESSAGE-P	PIC X.	
	Ø3	WS-MAP-MESSAGE-H	PIC X.	
	Ø3	WS-MAP-MESSAGE-V	PIC X.	
	Ø3	WS-MAP-MESSAGE	PIC X(50).	
	Ø1	WS-COMMAREA.		
	Ø3	WS-START-KEY	PIC 9(8)	COMP.
	Ø3	WS-START-KEY-X REDEFINES WS-START-KEY.		
	Ø5	FILLER	PIC X(4).	

```

Ø3 WS-END-KEY PIC 9(8) COMP.
Ø3 WS-END-KEY-X REDEFINES WS-END-KEY.
Ø5 FILLER PIC X(4).
Ø3 WS-SCROLL-DIRECTION PIC X(7).
88 WS-88-SCROLL-FORWARD VALUE 'FORWARD'.
88 WS-88-SCROLL-BACK VALUE 'BACK'.
Ø3 WS-DISP PIC 9(5).
Ø3 WS-HEX-MODE PIC XXX.
88 WS-88-HEX-ON VALUE 'ON'.
88 WS-88-HEX-OFF VALUE 'OFF'.
Ø3 WS-FIND-STRING PIC X(50).
Ø3 WS-FIND-LENGTH PIC 99 VALUE ZEROES.
Ø3 WS-FIND-COLUMN PIC 9(5) VALUE ZEROES.
Ø3 WS-FIND-DIRECTION PIC X VALUE SPACE.
88 WS-88-FIND-NEXT VALUE 'N'.
88 WS-88-FIND-PREV VALUE 'P'.
*
LINKAGE SECTION.
Ø1 DFHCOMMAREA PIC X(81).
*
EJECT
PROCEDURE DIVISION.
ØØØØ-MAIN.
PERFORM Ø1ØØ-INITIALIZE THRU
Ø199-INITIALIZE-EXIT.
IF WS-88-PROCESS-ERROR
CONTINUE
ELSE
PERFORM Ø2ØØ-PROCESS THRU
Ø299-PROCESS-EXIT
END-IF.
PERFORM Ø8ØØ-SEND-MAP THRU
Ø8ØØ-SEND-MAP-EXIT.
PERFORM Ø9ØØ-RETURN THRU
Ø9ØØ-RETURN-EXIT.
Ø1ØØ-INITIALIZE.
MOVE EIBAID TO WS-AID.
* Use CLEAR key to exit.
IF WS-AID = DFHCLEAR
EXEC CICS
RETURN
END-EXEC
END-IF.
IF EIBCALEN = Ø
* initial entry
MOVE ZEROES TO WS-START-KEY
WS-END-KEY
MOVE 1 TO WS-DISP
SET WS-88-HEX-OFF TO TRUE
MOVE 'N' TO WS-SCROLL-FLAG

```

```

        SET WS-88-SCROLL-FORWARD TO TRUE
ELSE
    MOVE DFHCOMMAREA          TO WS-COMMAREA
    PERFORM Ø81Ø-RECEIVE-MAP THRU
        Ø81Ø-RECEIVE-MAP-EXIT
END-IF.
PERFORM Ø82Ø-INITIALIZE-DISPLAY THRU
    Ø82Ø-INITIALIZE-DISPLAY-EXIT.
IF WS-MAP-CMMND > SPACES
    PERFORM Ø60Ø-PARSE-COMMAND THRU
        Ø60Ø-PARSE-COMMAND-EXIT
END-IF.
MOVE WS-START-KEY          TO WS-ERRLOG-KEY.
Ø199-INITIALIZE-EXIT.
EXIT.
Ø200-PROCESS.
EVALUATE WS-AID
    WHEN DFHENTER
        PERFORM Ø30Ø-ENTER-KEY THRU
            Ø30Ø-ENTER-KEY-EXIT
    WHEN DFHPF2
        PERFORM Ø31Ø-PF2-KEY THRU
            Ø31Ø-PF2-KEY-EXIT
    WHEN DFHPF5
        PERFORM Ø32Ø-PF5-KEY THRU
            Ø32Ø-PF5-KEY-EXIT
    WHEN DFHPF7
        PERFORM Ø33Ø-PF7-KEY THRU
            Ø33Ø-PF7-KEY-EXIT
    WHEN DFHPF8
        PERFORM Ø34Ø-PF8-KEY THRU
            Ø34Ø-PF8-KEY-EXIT
    WHEN DFHPF1Ø
        PERFORM Ø35Ø-PF1Ø-KEY THRU
            Ø35Ø-PF1Ø-KEY-EXIT
    WHEN DFHPF11
        PERFORM Ø36Ø-PF11-KEY THRU
            Ø36Ø-PF11-KEY-EXIT
    WHEN OTHER
        MOVE 'INVALID KEY PRESSED'
            TO WS-ERROR-DISPLAY
        SET WS-88-DISPLAY-ERROR TO TRUE
END-EVALUATE.
EVALUATE TRUE
    WHEN WS-88-SCROLL-FORWARD
        MOVE Ø          TO WS-SUB
        PERFORM Ø40Ø-READ-FORWARD THRU
            Ø40Ø-READ-FORWARD-EXIT
    WHEN WS-88-SCROLL-BACK
        MOVE 16        TO WS-SUB

```

```

        PERFORM 0500-READ-BACKWARD THRU
                0500-READ-BACKWARD-EXIT
    WHEN OTHER
        MOVE 'INVALID SCROLL VALUE'
                TO WS-ERROR-DISPLAY
        SET WS-88-DISPLAY-ERROR TO TRUE
    END-EVALUATE.
    IF WS-88-PROCESS-ERROR
        GO TO 0299-PROCESS-EXIT
    END-IF.
0299-PROCESS-EXIT.
EXIT.
0300-ENTER-KEY.
* scroll to top of file by setting start key to zeroes and read.
    IF WS-88-SCROLL-TOP
        MOVE ZEROES                TO WS-START-KEY
                                    WS-END-KEY
                                    WS-ERRLOG-KEY
        MOVE 'TOP OF ERROR LOG'    TO WS-MAP-MESSAGE
        SET WS-88-SCROLL-FORWARD TO TRUE
        GO TO 0300-ENTER-KEY-EXIT
    END-IF.
* scroll to bottom of file by setting start key to zeroes and read
* until end of file and then display a page.
    IF WS-88-SCROLL-BOT
        MOVE ZEROES                TO WS-START-KEY
                                    WS-END-KEY
        PERFORM 0420-READ-TO-END THRU
                0420-READ-TO-END-EXIT
        SET WS-88-SCROLL-BACK TO TRUE
        MOVE 'N'                    TO WS-ERRLOG-FLAG
        GO TO 0300-ENTER-KEY-EXIT
    END-IF.
* scroll to a time by setting the start key to zeroes and read
* until the time or portion of time is matched or greater than the
* time entered then display a page.
    IF WS-88-SCROLL-TIME
        MOVE ZEROES                TO WS-START-KEY
                                    WS-END-KEY
                                    WS-ERRLOG-KEY
        PERFORM 0430-READ-TO-TIME THRU
                0430-READ-TO-TIME-EXIT
        SET WS-88-SCROLL-FORWARD TO TRUE
        GO TO 0300-ENTER-KEY-EXIT
    END-IF.
* scroll to a task by setting the start key to zeroes and read
* until the task is matched then display a page.
    IF WS-88-SCROLL-TASK
        MOVE ZEROES                TO WS-START-KEY
                                    WS-END-KEY

```



```

                                WS-ERRLOG-KEY
PERFORM 0440-READ-TO-TASK THRU
        0440-READ-TO-TASK-EXIT
SET WS-88-SCROLL-FORWARD TO TRUE
GO TO 0300-ENTER-KEY-EXIT
END-IF.
* scroll forward until the find string is matched then
* display a page
  IF WS-88-SCROLL-FIND AND WS-88-SCROLL-FORWARD
    PERFORM 0460-READ-FIND-NEXT THRU
          0460-READ-FIND-NEXT-EXIT
    GO TO 0300-ENTER-KEY-EXIT
  END-IF.
* scroll backward until the find string is matched then
* display a page
  IF WS-88-SCROLL-FIND AND WS-88-SCROLL-BACK
    PERFORM 0560-READ-FIND-PREV THRU
          0560-READ-FIND-PREV-EXIT
    GO TO 0300-ENTER-KEY-EXIT
  END-IF.
  IF WS-88-SCROLL-BACK
    MOVE WS-END-KEY          TO WS-ERRLOG-KEY
  END-IF.
0300-ENTER-KEY-EXIT.
EXIT.
0310-PF2-KEY.
* toggle hex mode
  IF WS-HEX-MODE = 'ON '
    MOVE 'OFF'              TO WS-HEX-MODE
  ELSE
    MOVE 'ON '              TO WS-HEX-MODE
  END-IF.
0310-PF2-KEY-EXIT.
EXIT.
0320-PF5-KEY.
* if PF5 is pressed, the COMMAREA should contain a find string
  IF WS-FIND-STRING = SPACES OR LOW-VALUES
    MOVE 'ENTER FIND STRING' TO WS-ERROR-DISPLAY
    SET WS-88-DISPLAY-ERROR TO TRUE
    GO TO 0320-PF5-KEY-EXIT
  END-IF.
* PF5 was pressed and the scroll direction is forward
* read until the string is found then display a page
  IF WS-88-SCROLL-FORWARD
    SET WS-88-SCROLL-FIND TO TRUE
    PERFORM 0460-READ-FIND-NEXT THRU
          0460-READ-FIND-NEXT-EXIT
    GO TO 0320-PF5-KEY-EXIT
  END-IF.
* PF5 was pressed and the scroll direction is backward

```

```

* read until the string is found then display a page
  IF WS-88-SCROLL-BACK
    SET WS-88-SCROLL-FIND TO TRUE
    PERFORM 0560-READ-FIND-PREV THRU
      0560-READ-FIND-PREV-EXIT
    GO TO 0320-PF5-KEY-EXIT
  END-IF.
0320-PF5-KEY-EXIT.
EXIT.
0330-PF7-KEY.
* MAX was entered on the command line so position to the top
* of the file
  IF WS-88-SCROLL-MAX
    MOVE ZEROES                TO WS-START-KEY
                                WS-END-KEY
                                WS-ERRLOG-KEY
    MOVE 'TOP OF ERROR LOG'    TO WS-MAP-MESSAGE
    SET WS-88-SCROLL-FORWARD TO TRUE
    GO TO 0330-PF7-KEY-EXIT
  END-IF.
* an amount was entered on the command line so scroll backward
* that many records
  IF WS-88-SCROLL-AMT
    MOVE WS-END-KEY          TO WS-ERRLOG-KEY
    SET WS-88-SCROLL-BACK TO TRUE
    PERFORM 0550-READ-AMT-BWD THRU
      0550-READ-AMT-BWD-EXIT
    GO TO 0330-PF7-KEY-EXIT
  END-IF.
* PF7 was entered without an argument so we will read backward
* one screen full
  MOVE WS-START-KEY          TO WS-ERRLOG-KEY.
* we are scrolling backward
  SET WS-88-SCROLL-BACK TO TRUE.
  IF WS-ERRLOG-KEY > 0
    SET WS-88-SCROLLING TO TRUE
  END-IF.
0330-PF7-KEY-EXIT.
EXIT.
0340-PF8-KEY.
* MAX was entered on the command line so position to the bottom
* of the file
  IF WS-88-SCROLL-MAX
    MOVE ZEROES                TO WS-START-KEY
                                WS-END-KEY
    PERFORM 0420-READ-TO-END THRU
      0420-READ-TO-END-EXIT
    SET WS-88-SCROLL-BACK TO TRUE
    MOVE 'N'                    TO WS-ERRLOG-FLAG
    GO TO 0340-PF8-KEY-EXIT

```

```

        END-IF.
* an amount was entered on the command line so scroll forward that
* many records
    IF WS-88-SCROLL-AMT
        SET WS-88-SCROLL-FORWARD TO TRUE
        PERFORM Ø45Ø-READ-AMT-FWD THRU
            Ø45Ø-READ-AMT-FWD-EXIT
        GO TO Ø34Ø-PF8-KEY-EXIT
    END-IF.
* PF8 was entered without an argument so we will read forward
* one screen full
    MOVE WS-END-KEY                TO WS-ERRLOG-KEY.
* we are scrolling forward
    SET WS-88-SCROLL-FORWARD TO TRUE.
    IF WS-ERRLOG-KEY > Ø
        SET WS-88-SCROLLING TO TRUE
    END-IF.
Ø34Ø-PF8-KEY-EXIT.
EXIT.
Ø35Ø-PF1Ø-KEY.
* MAX was entered on the command line so move to the
* left-most byte
    IF WS-88-SCROLL-MAX
        MOVE 1                        TO WS-DISP
        GO TO Ø35Ø-PF1Ø-KEY-EXIT
    END-IF.
    IF WS-88-SCROLL-BACK
        MOVE WS-END-KEY                TO WS-ERRLOG-KEY
    END-IF.
* a number was entered so scroll left that number of bytes
    IF WS-88-SCROLL-AMT
        IF WS-DISP NOT > WS-SCROLL-AMT
            MOVE 1                        TO WS-DISP
        ELSE
            SUBTRACT WS-SCROLL-AMT FROM WS-DISP
        END-IF
        GO TO Ø35Ø-PF1Ø-KEY-EXIT
    END-IF.
* PF1Ø was entered without an argument so scroll left one
* screen full
    IF WS-DISP < 8Ø
        MOVE 1                        TO WS-DISP
    ELSE
        SUBTRACT 79 FROM WS-DISP
    END-IF.
Ø35Ø-PF1Ø-KEY-EXIT.
EXIT.
Ø36Ø-PF11-KEY.
* MAX was entered on the command line so move to the
* right-most byte

```

```

IF WS-88-SCROLL-MAX
  MOVE 29951 TO WS-DISP
  GO TO 0360-PF11-KEY-EXIT
END-IF.
IF WS-88-SCROLL-BACK
  MOVE WS-END-KEY TO WS-ERRLOG-KEY
END-IF.
* a number was entered so scroll right that number of bytes
IF WS-88-SCROLL-AMT
  COMPUTE WS-DISP-CALC = WS-DISP + WS-SCROLL-AMT
  IF WS-DISP-CALC > 29951
    MOVE 29951 TO WS-DISP
  ELSE
    ADD WS-SCROLL-AMT TO WS-DISP
  END-IF
  GO TO 0360-PF11-KEY-EXIT
END-IF.
* PF10 was entered without an argument so scroll right
* one screen full
IF WS-DISP < 29872
  ADD 79 TO WS-DISP
ELSE
  MOVE 29951 TO WS-DISP
END-IF.
0360-PF11-KEY-EXIT.
EXIT.
0400-READ-FORWARD.
PERFORM 0700-START-BROWSE THRU
  0700-START-BROWSE-EXIT.
IF WS-88-PROCESS-ERROR
  GO TO 0400-READ-FORWARD-EXIT
END-IF.
IF WS-88-SCROLLING
  PERFORM 0710-READNEXT THRU
    0710-READNEXT-EXIT
  END-IF.
PERFORM 0410-PROCESS-FORWARD THRU
  0410-PROCESS-FORWARD-EXIT.
* save the ending key
MOVE WS-ERRLOG-KEY TO WS-END-KEY.
PERFORM 0730-END-BROWSE THRU
  0730-END-BROWSE-EXIT.
0400-READ-FORWARD-EXIT.
EXIT.
0410-PROCESS-FORWARD.
PERFORM 0710-READNEXT THRU
  0710-READNEXT-EXIT.
IF WS-88-PROCESS-ERROR
  GO TO 0410-PROCESS-FORWARD-EXIT
END-IF.

```

```

IF WS-88-ERRLOG-EOF
    GO TO Ø41Ø-PROCESS-FORWARD-EXIT
END-IF.
ADD 1                                TO WS-READ-COUNT
                                    WS-SUB.

IF WS-READ-COUNT = 1
    MOVE WS-ERRLOG-KEY              TO WS-START-KEY
END-IF.
* move the display portion of the record to a hold area and
* convert non-display to '.' one byte at a time then move
* it to the map
MOVE WS-LOG-RECORD(WS-DISP: 79)
                                    TO WS-HOLD-LINE.
PERFORM VARYING WS-CHAR-SUB
    FROM 1 BY 1
    UNTIL WS-CHAR-SUB > 79
    MOVE WS-LOG-BYTE(WS-CHAR-SUB)
                                    TO WS-CHAR-TEST
    IF WS-LOG-BYTE(WS-CHAR-SUB) ALPHABETIC OR
    WS-LOG-BYTE(WS-CHAR-SUB) NUMERIC OR
    SPECIAL-CHAR
        MOVE WS-LOG-BYTE(WS-CHAR-SUB)
            TO
            WS-DISPLAY-CHAR(WS-CHAR-SUB)
    ELSE
        MOVE '.'
            TO
            WS-DISPLAY-CHAR(WS-CHAR-SUB)
    END-IF
    IF WS-88-HEX-ON
        MOVE WS-LOG-BYTE(WS-CHAR-SUB)
            TO WS-HOLD-BYTE
        PERFORM Ø84Ø-CONVERT-LINE THRU
            Ø84Ø-CONVERT-LINE-EXIT
        MOVE WS-DISPLAY-BYTE(1: 1)
            TO WS-HEX1-CHAR(WS-CHAR-SUB)
        MOVE WS-DISPLAY-BYTE(2: 1)
            TO WS-HEX2-CHAR(WS-CHAR-SUB)
    END-IF
END-PERFORM.
MOVE WS-DISPLAY-LINE                TO WS-MAP-LINE(WS-SUB).
* if hex display, move the hex values
IF WS-88-HEX-ON
    ADD 1                                TO WS-SUB
    MOVE WS-HEX1-LINE                TO WS-MAP-LINE(WS-SUB)
    ADD 1                                TO WS-SUB
    MOVE WS-HEX2-LINE                TO WS-MAP-LINE(WS-SUB)
END-IF.
* if hex display, we can only fit 5 input records (plus two hex
* line for each) on the display
IF WS-88-HEX-ON AND WS-READ-COUNT = 5

```

```

        GO TO 0410-PROCESS-FORWARD-EXIT
    END-IF.
* if this not hex mode, there is room for 15 records
    IF WS-88-HEX-OFF AND WS-READ-COUNT = 15
        GO TO 0410-PROCESS-FORWARD-EXIT
    END-IF.
    GO TO 0410-PROCESS-FORWARD.
0410-PROCESS-FORWARD-EXIT.
    EXIT.
0420-READ-TO-END.
* read to the end of the file
    PERFORM 0700-START-BROWSE THRU
        0700-START-BROWSE-EXIT.
    IF WS-88-PROCESS-ERROR
        GO TO 0420-READ-TO-END-EXIT
    END-IF.
    PERFORM 0710-READNEXT THRU
        0710-READNEXT-EXIT
    UNTIL WS-88-PROCESS-ERROR OR
        WS-88-ERRLOG-EOF.
* save the key to the last record
    MOVE WS-ERRLOG-KEY          TO WS-END-KEY.
    PERFORM 0730-END-BROWSE THRU
        0730-END-BROWSE-EXIT.
0420-READ-TO-END-EXIT.
    EXIT.
0430-READ-TO-TIME.
* the key is positioned to the first record and we will read until
* the time found is equal to or grater than what was eneterd
    PERFORM 0700-START-BROWSE THRU
        0700-START-BROWSE-EXIT.
    IF WS-88-PROCESS-ERROR
        GO TO 0430-READ-TO-TIME-EXIT
    END-IF.
    PERFORM 0710-READNEXT THRU
        0710-READNEXT-EXIT
    UNTIL WS-LOG-RECORD(18: WS-TIME-SUB) =
        WS-SEARCH-TIME(1: WS-TIME-SUB) OR
        WS-LOG-RECORD(18: WS-TIME-SUB) >
        WS-SEARCH-TIME(1: WS-TIME-SUB) OR
        WS-88-PROCESS-ERROR OR
        WS-88-ERRLOG-EOF.
    MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
    PERFORM 0730-END-BROWSE THRU
        0730-END-BROWSE-EXIT.
0430-READ-TO-TIME-EXIT.
    EXIT.
0440-READ-TO-TASK.
* the key is positioned to the first record and we will read until
* the task found is equal to what was entered

```

```

PERFORM 0700-START-BROWSE THRU
        0700-START-BROWSE-EXIT.
IF WS-88-PROCESS-ERROR
    GO TO 0440-READ-TO-TASK-EXIT
END-IF.
PERFORM 0710-READNEXT THRU
        0710-READNEXT-EXIT
UNTIL WS-LOG-RECORD(47:14) = WS-SEARCH-TASK OR
        WS-88-PROCESS-ERROR OR
        WS-88-ERRLOG-EOF.
MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
PERFORM 0730-END-BROWSE THRU
        0730-END-BROWSE-EXIT.
0440-READ-TO-TASK-EXIT.
EXIT.
0450-READ-AMT-FWD.
* this will read forward the number of records entered as a
* command argument
PERFORM 0700-START-BROWSE THRU
        0700-START-BROWSE-EXIT.
IF WS-88-PROCESS-ERROR
    GO TO 0450-READ-AMT-FWD-EXIT
END-IF.
ADD 1 TO WS-SCROLL-AMT.
PERFORM 0710-READNEXT THRU
        0710-READNEXT-EXIT
        WS-SCROLL-AMT TIMES.
MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
PERFORM 0730-END-BROWSE THRU
        0730-END-BROWSE-EXIT.
0450-READ-AMT-FWD-EXIT.
EXIT.
0460-READ-FIND-NEXT.
* read forward searching for the find string
PERFORM 0700-START-BROWSE THRU
        0700-START-BROWSE-EXIT.
IF WS-88-PROCESS-ERROR
    GO TO 0460-READ-FIND-NEXT-EXIT
END-IF.
* PF5 was pressed
* if a find column was entered, read another record and try to
* match if no find column was entered, add one to the current
* displacement and try to match the string
IF WS-AID = DFHPF5
    IF WS-FIND-COLUMN > 0
        PERFORM 0710-READNEXT THRU
                0710-READNEXT-EXIT
    ELSE
        ADD 1 TO WS-DISP
END-IF

```

```

END-IF.
COMPUTE WS-FIND-DISP-MAX = 30029 - WS-FIND-LENGTH.
PERFORM 0470-FIND-STRING THRU
    0470-FIND-STRING-EXIT
    UNTIL WS-88-STRING-FOUND OR
        WS-88-PROCESS-ERROR OR
        WS-88-ERRLOG-EOF.
MOVE WS-ERRLOG-KEY TO WS-START-KEY.
PERFORM 0730-END-BROWSE THRU
    0730-END-BROWSE-EXIT.
0460-READ-FIND-NEXT-EXIT.
EXIT.
0470-FIND-STRING.
* if we are doing a find by column, we have already done a primer
* read and will now start at the next record
* if we are doing a free-form find, this read will position us at
* the record last read on the previous execution, one byte past
* where we left off
    PERFORM 0710-READNEXT THRU
        0710-READNEXT-EXIT.
* a find column was entered in the initial find command so only
* look beginning at that column
    IF WS-FIND-COLUMN > 0
        IF WS-LOG-RECORD(WS-FIND-COLUMN:WS-FIND-LENGTH) =
            WS-FIND-STRING(1:WS-FIND-LENGTH)
            MOVE WS-FIND-COLUMN TO WS-DISP
            SET WS-88-STRING-FOUND TO TRUE
        END-IF
    ELSE
* no column argument was entered
        PERFORM 0480-SEARCH-LOGREC THRU
            0480-SEARCH-LOGREC-EXIT
        END-IF.
0470-FIND-STRING-EXIT.
EXIT.
0480-SEARCH-LOGREC.
* bump through the record one byte at a time trying to match on
* the string
    IF WS-DISP > WS-FIND-DISP-MAX
        MOVE 1 TO WS-DISP
        GO TO 0480-SEARCH-LOGREC-EXIT
    END-IF.
    IF WS-LOG-RECORD(WS-DISP:WS-FIND-LENGTH) =
        WS-FIND-STRING
        SET WS-88-STRING-FOUND TO TRUE
        GO TO 0480-SEARCH-LOGREC-EXIT
    ELSE
        ADD 1 TO WS-DISP
        GO TO 0480-SEARCH-LOGREC
    END-IF.

```



```

Ø48Ø-SEARCH-LOGREC-EXIT.
    EXIT.
Ø50Ø-READ-BACKWARD.
    PERFORM Ø70Ø-START-BROWSE THRU
        Ø70Ø-START-BROWSE-EXIT.
    IF WS-88-PROCESS-ERROR
        GO TO Ø50Ø-READ-BACKWARD-EXIT
    END-IF.
* if we are scrolling through the file, do a primer read to
* position at the next record
    IF WS-88-SCROLLING
        PERFORM Ø72Ø-READPREV THRU
            Ø72Ø-READPREV-EXIT
    END-IF.
    PERFORM Ø51Ø-PROCESS-BACKWARD THRU
        Ø51Ø-PROCESS-BACKWARD-EXIT.
* move the key of the last record read to the start key
    MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
    PERFORM Ø73Ø-END-BROWSE THRU
        Ø73Ø-END-BROWSE-EXIT.
Ø50Ø-READ-BACKWARD-EXIT.
    EXIT.
Ø51Ø-PROCESS-BACKWARD.
    PERFORM Ø72Ø-READPREV THRU
        Ø72Ø-READPREV-EXIT.
    IF WS-88-PROCESS-ERROR
        GO TO Ø51Ø-PROCESS-BACKWARD-EXIT
    END-IF.
    IF WS-88-ERRLOG-EOF
        MOVE WS-START-KEY          TO WS-END-KEY
        SET WS-88-SCROLL-FORWARD TO TRUE
        GO TO Ø51Ø-PROCESS-BACKWARD-EXIT
    END-IF.
    ADD 1                          TO WS-READ-COUNT.
    SUBTRACT 1 FROM WS-SUB.
    IF WS-READ-COUNT = 1
        MOVE WS-ERRLOG-KEY          TO WS-END-KEY
    END-IF.
    IF (WS-88-HEX-ON AND WS-READ-COUNT = 5) OR
        (WS-88-HEX-OFF AND WS-READ-COUNT = 15)
        IF WS-ERRLOG-KEY > Ø
            CONTINUE
        ELSE
            SET WS-88-SCROLL-FORWARD TO TRUE
        END-IF
    END-IF.
* move the display portion of the record to a hold area and
* convert non-display to '.' one byte at a time then move
* it to the map
    MOVE WS-LOG-RECORD(WS-DISP: 79)

```

```

                                TO WS-HOLD-LINE.
PERFORM VARYING WS-CHAR-SUB
  FROM 1 BY 1
  UNTIL WS-CHAR-SUB > 79
  MOVE WS-LOG-BYTE(WS-CHAR-SUB)
                                TO WS-CHAR-TEST
  IF WS-LOG-BYTE(WS-CHAR-SUB) ALPHABETIC OR
    WS-LOG-BYTE(WS-CHAR-SUB) NUMERIC OR
    SPECIAL-CHAR
    MOVE WS-LOG-BYTE(WS-CHAR-SUB)
                                TO
                                WS-DISPLAY-CHAR(WS-CHAR-SUB)
  ELSE
    MOVE '.'
                                TO
                                WS-DISPLAY-CHAR(WS-CHAR-SUB)
  END-IF
  IF WS-88-HEX-ON
    MOVE WS-LOG-BYTE(WS-CHAR-SUB)
                                TO WS-HOLD-BYTE
    PERFORM Ø84Ø-CONVERT-LINE THRU
    Ø84Ø-CONVERT-LINE-EXIT
    MOVE WS-DISPLAY-BYTE(1:1)
                                TO WS-HEX1-CHAR(WS-CHAR-SUB)
    MOVE WS-DISPLAY-BYTE(2:1)
                                TO WS-HEX2-CHAR(WS-CHAR-SUB)
  END-IF
END-PERFORM.
* if hex display, move the hex values
  IF WS-88-HEX-ON
    MOVE WS-HEX2-LINE
                                TO WS-MAP-LINE(WS-SUB)
    SUBTRACT 1 FROM WS-SUB
    MOVE WS-HEX1-LINE
                                TO WS-MAP-LINE(WS-SUB)
    SUBTRACT 1 FROM WS-SUB
  END-IF.

  MOVE WS-DISPLAY-LINE
                                TO WS-MAP-LINE(WS-SUB).
* if hex display, we can only fit 5 input records (plus two hex
* line for each) on the display
  IF WS-88-HEX-ON AND WS-READ-COUNT = 5
    GO TO Ø51Ø-PROCESS-BACKWARD-EXIT
  END-IF.
* if this not hex mode, there is room for 15 records
  IF WS-88-HEX-OFF AND WS-READ-COUNT = 15
    GO TO Ø51Ø-PROCESS-BACKWARD-EXIT
  END-IF.
GO TO Ø51Ø-PROCESS-BACKWARD.
Ø51Ø-PROCESS-BACKWARD-EXIT.
EXIT.
Ø55Ø-READ-AMT-BWD.
* this will read backward the number of records entered as a

```

```

* command argument
  PERFORM 0700-START-BROWSE THRU
    0700-START-BROWSE-EXIT.
  IF WS-88-PROCESS-ERROR
    GO TO 0550-READ-AMT-BWD-EXIT
  END-IF.
  ADD 1 TO WS-SCROLL-AMT.
  PERFORM 0720-READPREV THRU
    0720-READPREV-EXIT
    WS-SCROLL-AMT TIMES.
  MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
  PERFORM 0730-END-BROWSE THRU
    0730-END-BROWSE-EXIT.
0550-READ-AMT-BWD-EXIT.
  EXIT.
0560-READ-FIND-PREV.
* read backward searching for the find string
  PERFORM 0700-START-BROWSE THRU
    0700-START-BROWSE-EXIT.
  IF WS-88-PROCESS-ERROR
    GO TO 0560-READ-FIND-PREV-EXIT
  END-IF.
* PF5 was pressed
* if a find column was entered, read another record and try to
* match if no find column was entered, subtract one from the X
* current displacement and try to match the string
  IF WS-AID = DFHPF5
    IF WS-FIND-COLUMN > 0
      PERFORM 0720-READPREV THRU
        0720-READPREV-EXIT
    ELSE
      SUBTRACT 1 FROM WS-DISP
    END-IF
  END-IF.
  PERFORM 0570-FIND-STRING THRU
    0570-FIND-STRING-EXIT
    UNTIL WS-88-STRING-FOUND OR
      WS-88-PROCESS-ERROR OR
      WS-88-ERRLOG-EOF.
  MOVE WS-ERRLOG-KEY          TO WS-START-KEY.
  PERFORM 0730-END-BROWSE THRU
    0730-END-BROWSE-EXIT.
0560-READ-FIND-PREV-EXIT.
  EXIT.
0570-FIND-STRING.
* if we are doing a find by column, we have already done a primer
* read and will now start at the next record
* if we are doing a free-form find, this read will position us at
* the record last read on the previous execution, one byte past
* where we left off

```

```
PERFORM Ø72Ø-READPREV THRU
      Ø72Ø-READPREV-EXIT.
* a find column was entered in the initial find command so only
* look beginning at that column
  IF WS-FIND-COLUMN > Ø
    IF WS-LOG-RECORD(WS-FIND-COLUMN: WS-FIND-LENGTH) =
      WS-FIND-STRING(1: WS-FIND-LENGTH)
      MOVE WS-FIND-COLUMN TO WS-DISP
      SET WS-88-STRING-FOUND TO TRUE
    END-IF
```

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

Henry Cable
Senior Developer
Ahold Information Services (USA)

© Xephon 2003

December 2000 – November 2003 index

Items below are references to articles that have appeared in *CICS Update* since Issue 181, December 2000. References show the issue number followed by the page number(s). Subscribers can download copies of all issues in Acrobat PDF format from Xephon's Web site.

3270 bridge	203.37-44, 216.28-49	Display	184.5-10
722	209.3-5	Documenting programs	188.9-17
Abend	209.3-5	Dynamic workload management	204.21-30, 205.25-37, 206.10-17, 207.38-50, 208.13-17
Affinities	200.7-22	E-business	181.32-47
Batch	198.27-42	ECI	198.11-26
BMS	187.46-51, 188.18-55, 206.3-9, 208.7-12	Edit	205.12-25, 206.17-30
Book review	207.50	EJB	191.3-7
Browse	205.12-25, 206.17-30	EJB Server	214.3-13
Case	203.44-47	e-mail	203.14-19, 210.19-33, 213.14-37, 214.44-47
Case translation	197.11-13	EXCI	214.28-44
CEDA	190.11-12	Exit-enabling	196.42
CEMT	212.3-6	Faxing	189.25-35
Chargeback	202.18-39, 203.20-36	File control	193.3-9, 204.8-13
CICSPLEX SM API	207.9-28, 208.22-43	Forward recovery	181.14-28
CICSplex SM	194.19-28, 205.25-37, 206.10-17, 207.38-50, 208.13-17, 210.3-4, 212.22-51, 213.37-47	GLUE	182.28-35
CICS TS 1.3	188.3-4, 192.25	Help	216.12-15
CICS TS 2.2	197.3-11	Hot-pooling (HPJ)	181.3-13, 192.3-7
Closing files	203.3-14	HSM	197.13-17
Cold start	182.11-16, 184.10-11	Hung terminal	198.3-10, 202.7-18
Communication	207.3-8	IIS	193.9-22
Compiler	200.22-30	Integrated Translator	189.3-7
Conflicts	212.6-15	Interval control	201.6-11
COPY	181.28-31	ISC	182.17-28
Copy	206.3-9	ISC connections	204.30-44
CPSM	192.8-10	Java	181.3-13, 192.3-7, 198.11-26
CPSMCONN	209.14	Journalling	193.3-9
Cryptographic co-processor	197.13-17	Journal management	208.18-22
CSD	182.35-47, 183.3-6, 183.19-38, 184.36-47, 185.12-32, 187.6-40, 190.3-10	JVM	214.3-13
CVDA	185.3-11	LE options	201.3-5
CWA	197.18-30, 199.7-12	LE/370	187.40-45
DB2	182.35-47, 196.20-41, 197.31-51, 215.23-40	Libraries	190.16-24
DB2ENTRY	191.28-47	Log exit	205.3-12
Deactivating	201.24-	Log Manager	211.4-11
DFHRMUTL	209.6-13	Log	196.3-10, 214.15-27
DFHSHUNT	200.3-7	Log stream	180.6-9
		Logging	193.3-9
		Maps	185.32-47, 202.3-7
		Messages	186.9-22

Messaging	215.16-22	Sort	192.26-45, 193.32-43
Monitoring	182.17-28, 183.6-18, 190.12-16, 190.24-47, 191.22-28, 199.12-40, 206.30-42, 207.29-38, 210.16-18, 213.6-14, 214.13-14	Starting	208.3-6, 209.6-13, 212.3-6
MQSeries	203.37-44, 204.14-21, 216.28-49	Start-up type	204.3-8
MRO	182.17-28	Statistics	199.12-40
Named Counter Service	195.14-26	Stopping	212.3-6
Opening files	203.3-14	Storage	199.3-7
OTE	201.40-47	Storage use	186.23-32
PCL	196.10-20	Stress	182.3-11
Performance	195.12-13, 198.27	SYSIN checker	202.40-46
Performance Monitor	213.3-6, 216.15-28	System logger	190.24-47
PL/I	189.3-7, 189.7-24, 194.10-18	Task activity	182.3-11
PL/I transactions	187.3-6	TCP/IP	194.10-18
Printer assignment	183.39-47, 184.11-36	Templates	199.40-49, 200.30-42
Printing	194.29-43, 195.26-33, 196.10-20, 202.3-7	Temporary storage	191.8-11, 192.11-24, 194.3-9
Programming	195.3-11	Temporary storage queues	186.3-9
QMF	203.14-19, 205.37-43, 211.34-46, 212.15-22	Terminals	186.33-47
Questions and answers	195.41-43, 196.43, 197.51, 198.43, 199.50-51, 201.47, 202.47, 203.47, 204.44-45, 205.43, 207.51, 210.43, 211.47, 212.51, 213.47, 214.47	Testing	189.7-24
RCT	211.3-4	Threadsafe	216.3-12
RDO definition	188.5-9	Time Machine	197.18-30, 199.7-12
Reactivating	201.24-40	Tracing	215.40-47
Refresh programs	200.42-51	Transaction Gateway	209.15-29, 210.5-15, 211.12-34
Resource management	210.3-4	Transaction information	201.11-24
Scanner utility	209.29-47, 210.34-43	Transaction Server	209.15-29, 210.5-15, 211.12-34
Session reuse	200.3-7	TSO file	194.29-43, 195.26-33
Shutdown	208.18-22	Tuning	191.28-47
SIGNON	197.3-11	VB6	194.10-18
SIT parameter	184.3-5, 209.14	VSAM	181.14-28, 212.6-15
SOAP	215.3-16	VSE	181.32-47
Sockets	213.14-37	VTAM	206.42-47
		Web	193.23-31, 195.34-40
		Web documents	189.36-51
		Web interface	185.11-12
		WebSphere	193.9-22
		Web User Interface (WUI)	191.12-21
		XMEOUT	198.3-10, 202.7-18
		XMITIP	210.19-33, 214.44-47

Why not share your expertise and earn money at the same time? Articles for *CICS Update* can be of any length and can be sent or e-mailed to Trevor Eddolls at any of the addresses shown on page 2. A free copy of our *Notes for contributors* is available from our Web site at www.xephon.com/nfc.

CICS news

Serena Software has announced a new Version 4.1 of its Application Performance Manager, StarTool.

StarTool APM is a performance measurement and analysis system that helps to resolve OS/390 and z/OS job performance issues. Users can focus their activities on tuning specific areas of an application to boost productivity.

In addition to a redesigned architecture and user interface, Version 4.1 includes extended support for the z/OS platform through improved monitoring and reporting capabilities.

The improved architecture makes the product easier to customize and maintain. Version 4.1 also provides improved, real-time monitoring capabilities with extended support for a wider range of subsystems including CICS, IMS, and DB2.

In addition, Version 4.1 delivers better reporting by providing customers with more in-depth information on an application to help quickly solve performance problems and eliminate the need for further analysis.

For further information contact:
Serena Software, 2755 Campus Drive, 3rd Floor, San Mateo, California 94403, USA.
Tel: (650) 522 6600.
URL: http://www.serena.com/product/aa_st_apm_ov.html.

* * *

iWay Software has improved its data integration capabilities, which include new

cluster management, remote installation and management, tighter integration with Microsoft SQL Server, and new data adapters.

Cluster management makes sure resources are available; active agents distribute work to the appropriate machines; fail-over capability guarantees job conclusion; load balancing speeds completion; and all monitoring and development can be done from anywhere via remote console.

Remote installation reduces the manual tasks involved in a OS/390, z/OS, OS/400, Unix, and OpenVMS Server installation. Server installations are simplified because the installation files can be transferred and configured from a local laptop or a remote computer, eliminating the need for certain types of technical and operational support. Additionally, a Web-based management console enables adapters to be configured and managed, anywhere with an interface, from a common graphics-based front-end.

Intelligent adapters have been added and enhanced to support DB2 Version 8 and IMS Version 8, bringing the total number of data and application adapters to well over 250. Additionally, iWay's Adapter for CICS now supports RRS (Resource Recovery Services) and integrated two-phase commit.

For further information contact:
iWay Software, Two Penn Plaza, New York, NY 10121-2898, USA.
Tel: (212) 330 1700.
URL: <http://www.iwaysoftware.com/products/index.html>.



xephon