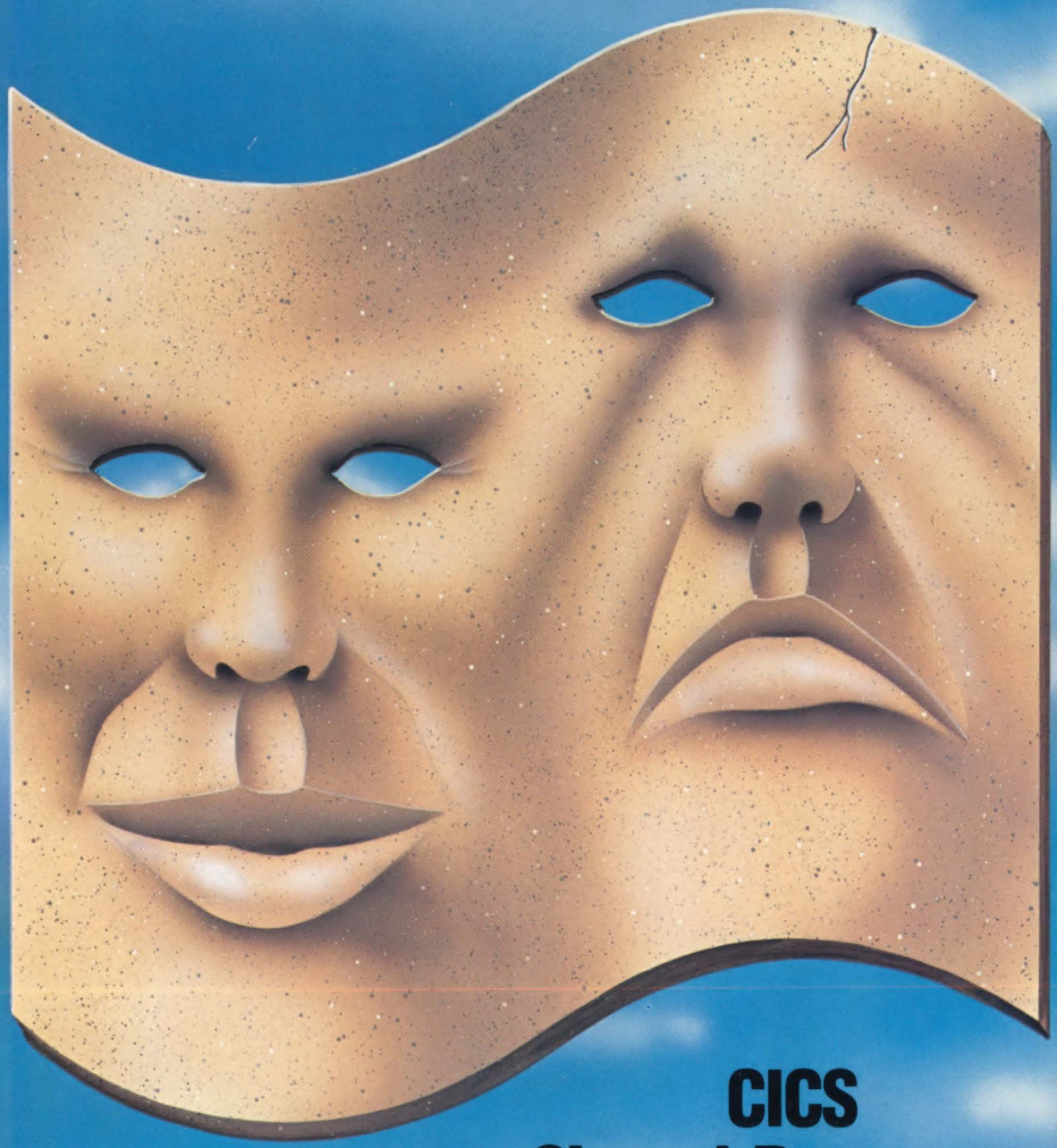
MAINFRAME JOURNAL

For Users of IBM System/370 Architecture & Compatible Systems

April 1989



CICS Shared Resources Triumphs And Tragedies

We've done more than change our name.

We've grown from a brilliant specialist into a high-energy resource for multiple environments.

As VM Software, we built an unmatched reputation for delivering powerful solutions to the problems of VM data center managers.

Today we're still #1 in VM. But we're a whole lot more: An innovator in network administration. A source of SQL/DS and DB2 database tools. A leader with Network Data-Mover products for multiple environments.

Our new name—Systems Center—reflects this expanded capability. It also signals our determination to be a continuing focal point of energy and creativity for our industry.

Through the years to come, we will continue to provide superior products for VM and networked environments. And we will continue to champion the cause of systems professionals everywhere by developing and marketing appropriate, effective technology.

Systems Center. It's a name to count on. A company to grow with. A high-energy resource for an emerging era. For more information write or call today: Systems Center, Inc., 1800 Alexander Bell Drive, Reston, VA 22091, telephone (703) 264-8000.

1-MFJ-8904

Something's killing CICS... and it's about to strike again.

NEW RELEASE!

eyewitness®

SOLVE THE MYSTERY

Look to Eyewitness® for help— complete CICS dump diagnostics.

Another CICS dump—it's enough to make you scream! But wait. Now there's Eyewitness, the revolutionary achievement in system and transaction failure analysis. Systems and applications programmers get superior **system, transaction, and hard-copy support**—all in one product.

For the first time, your applications programmers can go directly from a dump to their own failing source statements. The result? Self-reliance. Efficiency. Productivity.

Systems and operations people benefit, too. They spend less time on maintenance tasks because Eyewitness provides CICS architecture and internals. It's at their fingertips. No more agonizing hours searching for clues. Get answers in **minutes**. And with its rapid dump processor, you'll slash system dump time and downtime by up to 90%. Eyewitness has all of the facilities for **both** system and transaction dumps. Consolidate all of your dump management packages and save on maintenance costs. Best of all, Eyewitness bears the signature of Landmark Systems Corporation, makers of The Monitor for CICS™.

There's nothing like it. Analyzing CICS dumps doesn't have to be murder. Call us today for a FREE, 30-DAY TRIAL at **1-800-227-8911** or 1-703-893-9046.

LANDMARK™

Landmark Systems Corporation
8000 Towers Crescent Drive
Vienna, Virginia 22182-2700

CIRCLE #16 on Reader Service Card ▲

CONTENTS

DEPARTMENTS

April 1989

Volume IV
Number 3

COVER:

LSR is a powerful tool for improving CICS performance if it is used correctly. However when used incorrectly, it loses its effectiveness and degrades response time. Therein lie its triumphs and tragedies. For the effective implementation of LSR, turn to Richard G. Nikula's article on page 18. Cover illustration by David Kramer.

MAINFRAME JOURNAL© (ISSN 0895-5751) is published monthly by Thomas Publications, Inc., 10935 Estate Lane, Suite 375, Dallas, TX 75238, (214) 343-3717. Second class postage paid at Dallas, TX.
POSTMASTER: Send address changes to: MAINFRAME JOURNAL, P.O. Box 551628, Dallas, TX 75355-1628.

FEATURES

ADABAS 5: more than more performance.

ADABAS 5's leading position as a high-performance DBMS has been verified once again by a series of standard, fully scaled benchmarks. Each was conducted on a National Advanced Systems AS/EX™ 100 (equivalent in power to an IBM 3090 500S). And audited by Coopers & Lybrand.

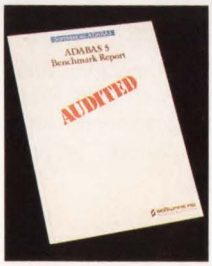
In the **standardized TP1 debit-credit benchmark,** ADABAS 5 worked with a data base containing **1 million accounts.** The results: a **record-breaking 388 transactions per second** (tps) – 99.3% serviced in under one second!

For the first time, an authentic **ET1 debit-credit benchmark** was conducted with ADABAS 5 managing **10 million accounts.** The results: **167 tps** (from terminal in/to terminal out) – 99.5% serviced in under one second!

These figures represent major savings in time and money. But they're not surprising – at least not to the thousands of organizations which already use ADABAS 5.

They expect more performance. Plus the benefits that come from 18 years' experience in DBMS technology: – relational functionality which allows adaptable data structures and fast information retrieval based on multikey criteria – document management and free text retrieval – 24 hour processing in a fully integrated DB/DC environment – portable applications across various hardware and operating systems – distributed processing – entity relationship data bases with recursive retrieval functions for knowledge-based systems.

Discover how much more profitable your DP organization can be. **Conduct your own ADABAS 5 benchmark,** using your own data and application profile in your own production environment. The facts will speak for themselves.



Demand the performance and functionality only ADABAS 5 can offer.

To order your free copy of the ADABAS 5 Benchmark Report, call toll-free: 1-800-843-9534 today.

SOFTWARE AG
Programming Business Success

™ AS/EX is a trademark of National Advanced Systems Corporation.

CIRCLE #75 on Reader Service Card ▲

EDITOR'S COMMENTS

Carol M. Hoag

A friend of mine and her husband left for a night out recently. She and her husband have two boys — a teenager and an 11-year old. While their parents were gone, the brothers ordered a pizza but did not eat it right away. When they were ready to eat it, they decided to warm it up in the oven. So, they turned on the oven and put in the pizza. Pretty soon, flames started shooting around inside the oven and smoke began billowing out. They had left the pizza in the box! When my friend asked her teenager why he had left the pizza in the box, he replied that was the way all of his friends warmed up pizza!

This startling example leaves no doubt that less-than-perfect human intervention involves some risk. Howard Miller in his article, "Unattended Computer Center Operation: 50 Questions and Answers," explains that unattended operation means implementing tools and techniques to eliminate or reduce the dependency on human intervention, thus improving quality. The objective of unattended operation is quality, not expense reduction according to Miller.

Unattended computer center operation is an intriguing subject. Interest in unattended operations was overwhelmingly apparent at the recent AFCOM (Association For Computer Operations Management) conference held in San Diego. Attendance at sessions focusing on unattended operations was standing-room only.

Expanded Storage

The interest in quality operation is pervasive. For those of you who are wondering what the increased use of expanded storage will mean to the MVS operating system, Bill Mullen offers his experience and insight in his article on page 8, "3090 Expanded Storage: Evolution Of A New Technology."

Local Shared Resources (LSR)

While LSR is a powerful tool for improving the quality of CICS performance, it has to be used correctly. Richard Nikula's article on the triumphs and tragedies of LSR on page 18 presents a method for effectively implementing this tool that can improve response time.

Why Prototype?

Prototyping guarantees user satisfaction according to Jon Pearkins whose article begins on page 44. He explains what it is and how the user can test the final system early in the design phase.

DB2 or IMS

Quality of operation often involves converting to a new technology. This is the case with IMS shops that must recognize DB2 as a vastly superior database management system, according to Jack Olson in his article, "DB2 In An IMS World," on page 82.

Tutorials

Learn how CPENABLE can be used to tune a logically-partitioned processor running under IBM's PR/SM and Amdahl's MDF from Mark Friedman in his article on page 38. Steve Eckols in his article on page 32 teaches how to use the text editor that comes with VM: the System Product Editor called XEDIT. "How VTAM Works" by Beverly Weaver begins on page 48 and Ted Keller examines basic internal workings of CICS' Temporary Storage and how they relate to performance on page 54. Marc Fey on page 69 offers guidelines for selecting productivity tools for development and maintenance.

Viewpoint — Last But Not Least

The next-to-last page of MAINFRAME JOURNAL offers you a special interest column called "Viewpoint" that will be a regular occurrence. Candle Corporation President Aubrey Chernick is the first guest columnist who presents his viewpoint of "Systems Application Architecture — Its Roots and Future."

With these and other articles in this month's issue, MAINFRAME JOURNAL continues to feature quality articles by knowledgeable writers. Our goal is to help you achieve a quality operation in all areas of mainframe computing.

MAINFRAME JOURNAL
For Users of IBM System/370 Architecture & Compatible Systems

Publisher and Editor-in-Chief
Bob Thomas

Associate Publisher
Martha Thomas

Editor
Carol M. Hoag

Copy Editors
Judy Beller
Pat Warner

Art Director
David Kramer

Assistant Art Director
Ken Buerer

Marketing Services
Sally Webb

Circulation Manager
Janice Porter

Assistant Circulation Manager
Nancy Crawford

Administrative Manager
Marian Davenport

Associate Publisher/Advertising
Mark A. Cauto

Advertising Manager
Denise Thomas

BPA

SUBSCRIPTION RATES & INQUIRIES: Subscriptions are free within the USA and Canada. One-year foreign subscriptions are $96.

All subscriptions, remittances, requests and changes of address should be sent to MAINFRAME JOURNAL at 10935 Estate Lane, Suite 375, Dallas, TX 75238, (214) 343-3717.

The Power Package!

Take the shortcut to fast response times. DATAPACKER/II, the CICS transmission optimizer from H&M, will reduce outbound and inbound data streams by 50-85%. The advantages speak for themselves: considerably improved response times, no screen blinking, lower line costs and free capacity for additional terminals and new applications.

DATAPACKER/II
CICS · TRANSMISSION · OPTIMIZER

CIRCLE #118 on Reader Service Card

H&M
SOFTWARE'S FUTURE

H&M Systems Software, Inc., 25 E. Spring Valley Ave., Maywood, N.J. 07607-2150, 1-201-845-3357, 1-800-FOR-DEMO

3090 Expanded Storage

Evolution Of A New Technology

By J. William Mullen

Introduction of expanded storage with the availability of the 3090 processor complex opened the door to a new era of storage usage and management by MVS operating systems. Initial use of expanded storage was as a high speed paging and swapping device. Increased processing power in the 3090 processors and availability of increased central storage sizes allowed for an increase in the number of tasks concurrently active in the configuration. Increased page movement for paging and swapping from the increase in concurrency of active tasks dictated a requirement for a mechanism that would handle page movement in a faster more efficient manner than auxiliary storage transfer and the use of page and swap datasets.

Expanded storage as initially implemented was not directly addressable by the user. Page transfers to and from expanded storage are handled in 4K blocks (page size) with no direct I/O capability. Transfer of pages to and from expanded storage and auxiliary storage requires movement of the pages to central storage in the transition. Page movement is synchronous which means that control of the processor is retained during page movement from inception to completion. The question, "Why expanded storage?" could be asked. With the above conditions and the use of central storage as a mechanism for page transfer, it could create potential central storage shortages resulting in paging problems in the configuration. We will address the architectural changes that occurred with the introduction of expanded storage, answers to the questions that arise from implementation of expanded storage and further use of the technology with the MVS/ESA operating system.

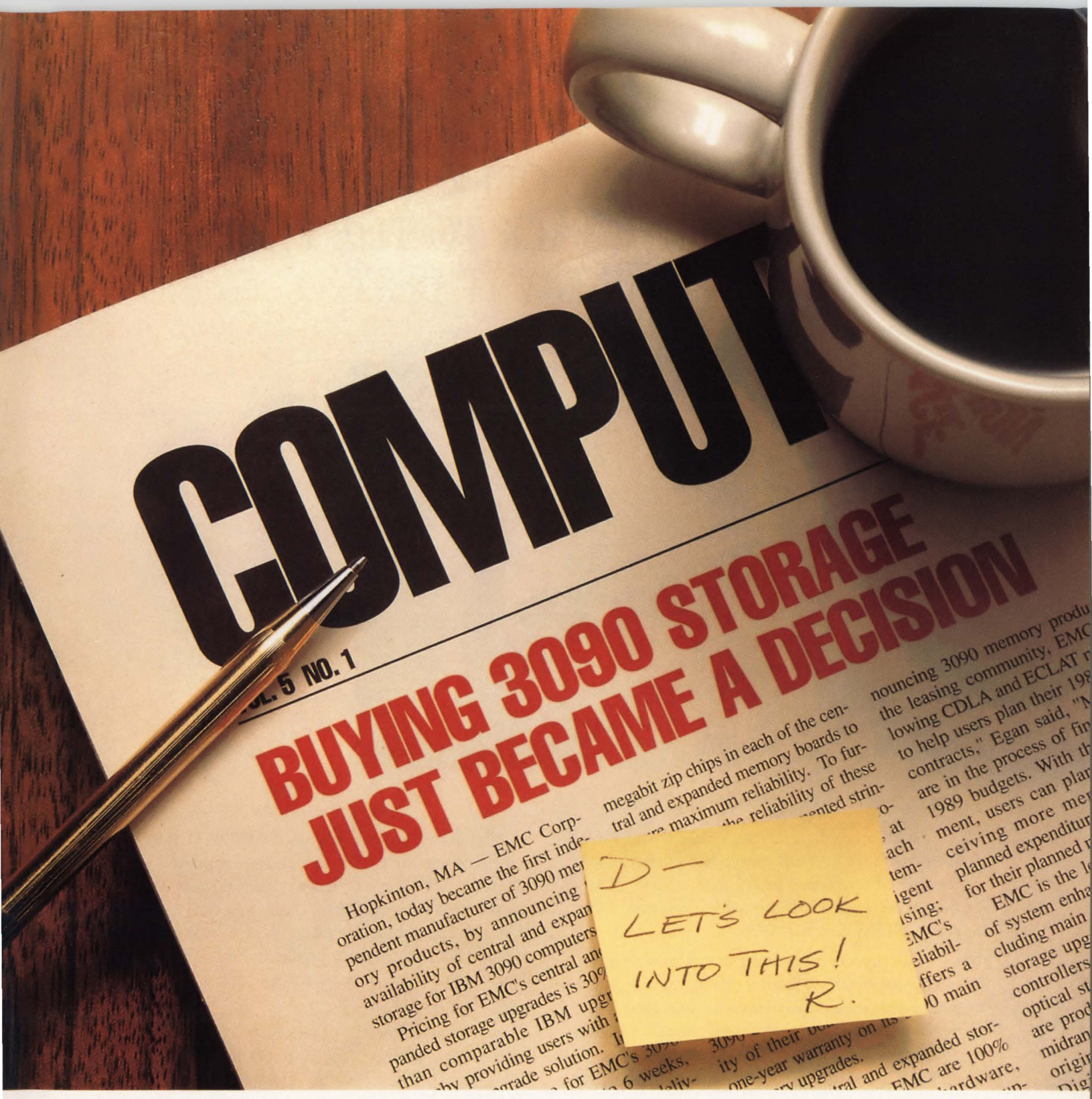
Processor Storage Swap

The concept of *processor storage swap* was introduced with the MVS/SE2 operating system and the logical swapping function. The first application of the concept was for MVS Time-Sharing Option (TSO) users as a mechanism to handle increased number of active users, minimize use of auxiliary storage for swapping of user pages and maintain response time levels for TSO transactions as transaction activity increased. TSO users were allowed to remain in central storage if there were sufficient frames to support their working set requirements and they exhibited an *interactive* mode of processing. The interactive mode of processing was based upon a new variable termed *think time* that was applied to both the individual TSO user and as an indicator of system central storage availability. As central storage was more heavily utilized and available frame counts reduced, the system *think time* was adjusted downward to allow less processor storage or logical swapping in the configuration. TSO users that were *a little less than interactive* would then be swapped to auxiliary storage during their terminal wait periods. As the number of TSO users increased, so did the size of auxiliary paging and swapping configurations required to handle page transfers.

For many installations, lack of availability of additional channel paths and DASD devices prohibited expansion of the auxiliary paging and swapping subsystem and, as the number of TSO users increased, transaction response times became worse with the only alternative for solving the problem being the addition of central storage. For those installations with the maximum amount of central storage installed on the processor complex, the only alternative was the addition of a processor complex and shifting a portion of the TSO workload to the new processor.

MVS Storage Management

Central storage management in the MVS operating system consists of paging, swapping and page stealing. For each central storage page in the configuration, a Page Frame Table Entry (PFTE) was cre-



Announcing 3090 Central and Expanded Storage from EMC Corporation.

As a 3090 user you now have the choice to make a smart business decision. Our 3090 Central and Expanded Storage upgrades offer you the reliability, performance, local and remote support that you demand at a price 30% lower than that of IBM.

Actually, this good news is old news. For 10 years we have been offering users cost-effective, state-of-the-art storage solutions. Our strong background in mainframe storage solutions has been recognized by Fortune 500 companies worldwide. These companies have realized that we provide a better business decision for their mainframe storage needs. Now, we have built this mainframe experience into our 3090 product offerings.

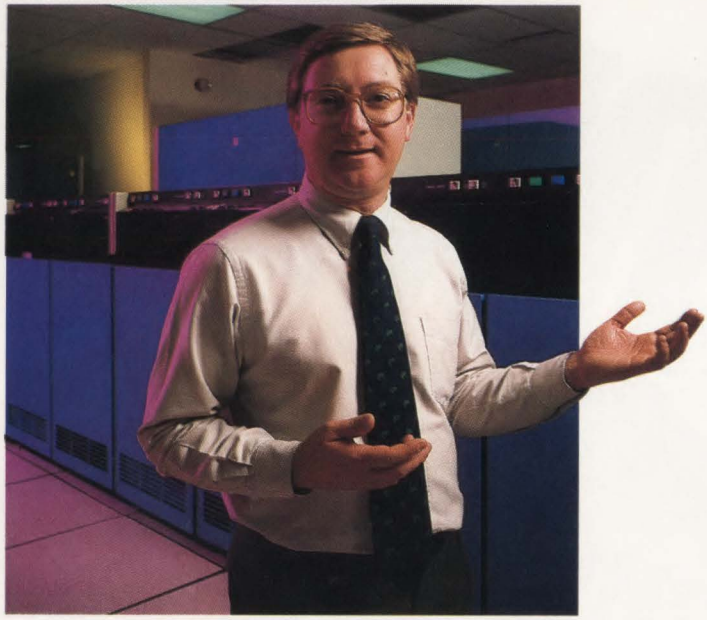
For additional good news on how EMC Corporation's 3090 storage upgrades can help you reach your business objectives call or write: EMC Corporation, Hopkinton, MA 01748-9103 **1-800-222-EMC2.** (In MA, 508-435-1000.)

EMC² The System Enhancement Company.

IBM is a registered trademark of International Business Machines Corporation.

CIRCLE #95 on Reader Service Card ▲

AUTOMATED OPERATIONS? NOT UNTIL WE TALK TO THE PEOPLE WHO KNOW IT BEST.



With more than 500 sites already using our IMS AutoOPERATOR, Boole & Babbage knows automated operations better than anyone. Now that experience is incorporated in our new AutoOPERATOR for MVS.

Our experience has shown us that the real power of automated operations lies in the ability to integrate automated console operations with your other performance software to optimize operator efficiency and system availability across the board.

As a standalone product, MVS AutoOPERATOR gives you everything you look for in automated operations software. Message suppression, message event automation, timed event automation, complex logic capabilities: all designed to relieve your overburdened operators. ISPF and EXCP support to make NetView easier to use. Even a powerful outboard capability with its own presentation facilities.

But the real power of MVS AutoOPERATOR is something you can't get anywhere else: seamless integration with our wide range of performance management software. Which means you can anticipate and respond automatically to a wide range of performance threatening problems. You can even manage remote and multiple systems from a single terminal, in a single session.

Get the real power of automated operations from Boole & Babbage. Call Marty Johnson today. In California: 800-624-5566. Outside California: 800-822-6653.

Boole & Babbage, Inc., 510 Oakmead Parkway, Sunnyvale, California 94086.

Boole & Babbage

International sales and support provided through The European Software Company and a worldwide distribution network.

CIRCLE #181 on Reader Service Card ▲

ated at system initialization time to contain the status information for each available central storage frame in the configuration. In MVS/SP, a PFTE required 16 bytes of storage that was allocated as part of the nucleus. Allocation of PFTEs was changed in MVS/SP Release 1.3.5 so that storage would be obtained in the extended nucleus above the 16MB line. For MVS/XA and MVS/ESA, PFTEs require 32 bytes of storage for each installed central storage frame with the storage allocated in the extended nucleus above the 16MB line.

Page usage of central storage frames is determined through the updating of the page Unreferenced Interval Count (UIC) that indicates the amount of time the frame has not been referenced by the owning task. The MVS Real Storage Manager (RSM) periodically evaluates all central storage frames in the configuration and increments a frame's UIC if it has not been referenced. The UIC value is then used by the MVS System Resources Manager (SRM) in determining Multi-Programming Level (MPL) adjustments and as a factor, along with available central storage frame counts, in the adjustment of system *think time* to increase or decrease the amount of logical swapping in the configuration. For TSO users whose individual *think time* is less than the system *think time*, processor storage or logical swap is allowed. It should be noted that batch type tasks can be logically swapped. Batch type tasks have a fixed *think time* of five seconds that eliminates their candidacy for logical swap during periods of low central storage frame availability. The UIC value is also used by the SRM to determine which frames can be stolen when the RSM indicates that the available central storage frame count has fallen below the frame count low threshold.

The availability of expanded storage required new mechanisms for storage management and interactions between the SRM and RSM.

Expanded Storage Architecture and Management

The Expanded Storage Table Element (ESTE) was introduced for managing expanded storage frames. At MVS system initialization, an ESTE is created for each available expanded storage frame. An ESTE requires 32 bytes of storage. ESTEs are allocated in the Extended System Queue Area (ESQA) above the 16MB line and are page fixed. Thus, for each 64MB of expanded storage added to a configuration, there is a reduction of approximately

FIGURE 1

Expanded Storage Criteria Table (ESCT)

ESCT	Class of Storage 0	1	2	Rule For Access to Expanded Storage
PAGED-OUT				
Changed (POC)	100	100	100	MA > CA
Unchanged (PUC)	100	100	100	MA > CA
STOLEN				
Changed (STC)	0	20	15	MA > CA
Unchanged (STU)	0	20	15	MA > CA
SWAP-OUT				
Changed (SWC)	100	100	100	UIC + MA > CA
Unchanged (SWU)	100	100	100	UIC + MA > CA
SWAP WORKING SET				
Ready (SWWS)	100	100	100	UIC + MA > CA
Terminal Wait				UIC + MA − TT > CA
VIRTUAL I/O				
All Pages (VIO)	900	900	900	MA > CA
VIRTUAL FETCH				
IMS Only (VF)		15		UIC + MA > CA

MA—Expanded Storage Migration Age
CA—ESCT Criteria Age For Access
TT—TSO User 'Think Time'

Class of Storage: 0—Privileged, Non-Swappable, and Common
1—Other Not 0 or 2
2—TSO Swap and TSO Other

512K bytes (128 frames) in the number of available central storage frames.

A new value, Migration Age (MA), is used for determining the *page life* of expanded storage frames. This value is an average residency time for all expanded storage frames and, unlike the UIC value for individual central storage frames, is not an indicator of an individual frame's residency time in expanded storage. The MA value is an indicator of availability of expanded storage frames in the configuration and controls access to expanded storage.

The process of page migration is the mechanism used by MVS, and specifically the RSM, to replenish expanded storage frames. The RSM determines if migration processing should occur based upon low and high frame availability thresholds for expanded storage frames. The MA value is not the criteria for expanded storage page migration. When the available expanded storage frames drop below the low threshold, the RSM migration routines evaluate expanded storage frames for eligibility to be migrated and sent to auxiliary storage. This migration process requires communication with the SRM where a swapped out task is to have its pages on expanded storage migrated to auxiliary storage. The migration process and RSM's evaluation of expanded storage frames affects the MA updating process. The SRM storage management routine, a timed routine, evaluates migration processing on a one second interval. If migration is not occurring, the MA is increased by 1.5 seconds for each elapsed second. If the RSM has evaluated all expanded storage frames for migration purposes, the MA is reset and the indication of average *page life* starts anew.

Concern over use of central storage as a migration mechanism has been a question often asked. The RSM migration routines have a threshold on the number of central storage frames that may be acquired to facilitate migration of expanded storage frames to auxiliary storage. Thus migration activity will not cause a shortage of central storage frames or induce paging problems due to frame shortages.

Access to Expanded Storage

Control of the use of expanded storage is governed by values in the Expanded Storage Criteria Table (ESCT) that can be modified through entries in the MVS Optimizer Parameter Table (IEAOPTxx) member of SYS1.PARMLIB. The ESCT is analogous to the Logical Swap Control Table (LSCT) introduced with the logical swapping function. Values in the ESCT are elapsed seconds of time and represent the Criteria Age (CA) value for comparison to the current MA value to determine eligibility of pages to be sent to expanded storage.

The use of the ESCT to control access

to expanded storage involved changes to the MVS swapping mechanism. The first was the definition of a task's working set for swap purposes. Swapping to auxiliary storage utilized a single working set definition of the pages to be retained by the task. Swapping to expanded storage introduced the concept of a primary and secondary working set. The primary working set is defined as LSQA pages, fixed pages and the first page of each allocated virtual segment for the task. The latter is needed for reconstruction of the virtual address space at swap-in time. Secondary working sets are all pages owned by the task but not included in the primary working set.

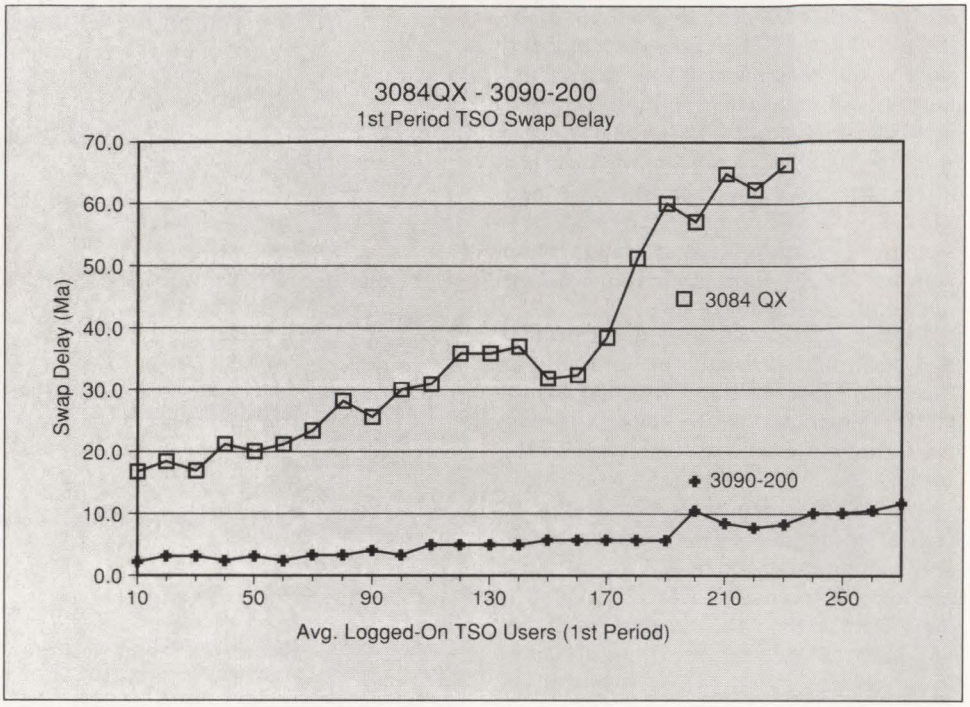
The second change introduced was the definition of task classes in determining access to expanded storage. Task classes of 0, 1 and 2 were introduced to identify Privileged, Other and TSO pages in determining if the pages were eligible for expanded storage residency. A further definition of the type of page for the various task types finalizes the ESCT definition.

When a task's pages are to be removed from central storage, the CA for the class of task and page type(s) is compared to the current MA for expanded storage frames. If the CA is less than the MA, the task's pages are sent to expanded storage. If the CA is greater than the current MA, the task's pages are sent to auxiliary storage. Figure 1 illustrates the ESCT *matrix* for task classes and page types with their respective default CA values.

Review of the default values for CA values shown in Figure 1 reveals that certain task classes and types of pages are favored in the control of access to expanded storage. Privileged (Class 0) pages are favored over TSO (Class 2) pages that are favored over Other (Class 1) pages. Within the task classes, note that certain types of pages are favored, in particular, stolen pages. This is done for two reasons. Pages stolen due to central storage shortages have a high probability of being returned in a short time interval and thus will experience a faster page fault resolution time from expanded storage than from auxiliary storage. Stolen pages are usually in the four to 10 page range, thus making it inefficient to send them to auxiliary storage when paging and swapping to 3380 devices and the slot allocation algorithm is utilized to manage page slots.

TSO working sets are favored in the swap page category to try to assure quick return of the working sets at swap-in time

FIGURE 2



that reduces swap delay and transaction response time. Figures 2 and 3 show the effect of reduced swap delay on TSO transaction response time when transitioning from a 3084-QX processor with 64MB of central storage to a 3090-200 processor with 64MB of central storage and 64MB of expanded storage. As the average number of logged-on users increases, notice the increase in the TSO first period transaction swap delay and the increase in the TSO first period transaction average response times corresponding to the increased swap delay on the 3084-QX processor. On the 3090-200 processor, the swap delay remains fairly constant as the number of average logged-on users increases, resulting in reduced transaction response times.

It is somewhat intuitive that you can affect the use of expanded storage through modification of the ESCT values. For example, if you wanted to reduce the use of expanded storage by batch task swapping, you would increase the Class 1 ESCT CA value for working set (ESCTSWWS) from its default value of 100 seconds (the MA must be one-and-a-half minutes or greater to allow the pages to be sent to expanded storage) to a value of 300 seconds (the MA must be five minutes or greater to allow the pages to be sent to expanded storage). Expanded storage pages currently being consumed by batch task swapping activity would then be available for the other task classes and batch swapping to expanded storage would resume when the average page life of the expanded storage frames was five minutes or greater. You can thus bias the use of expanded storage frames.

Note that the ESCTVIO parameter values were not present in the initial implementation of expanded storage and Virtual I/O (VIO) frames were not sent to expanded storage. Movement of VIO pages to expanded storage became available for MVS/XA in May 1988 and is a standard feature of MVS/ESA.

A final word on the swapping process in the expanded storage environment. Swap processing to auxiliary storage involves swapping out the complete swap set which must be totally swapped back in for the task to be eligible for dispatch. Swap to expanded storage requires only the primary working set be swapped back in for the task to be eligible for dispatch with pages in the secondary working set being page faulted into central storage when needed. Measurements indicate that the primary working set consists of approximately 10 to 13 percent of the pages in the task's total working set. Swap in from expanded storage thus requires fewer pages to be moved to allow a task to become dispatchable and makes more efficient use of central storage through the requirement for fewer pages. If a task is swapped to expanded storage and subsequently migrated to auxiliary storage, at swap-in time the primary working set is swapped back into central storage first and then the secondary working set is swapped back into

ANNOUNCING

A New More Productive Era in Systems Programming

The Systems Programming Environment Only in the SAS/C® Compiler

Until now, higher-level languages just couldn't hack it in the systems programming world. Too many issues stood in the way—inefficiency, poor access to low-level system services, bulky and intrusive library requirements, and inflexibility in addressing the IBM 370 architecture.

But now you can write systems-level routines faster and maintain them better than you ever could with assembler—with the SAS/C compiler's exclusive Systems Programming Environment (SPE).

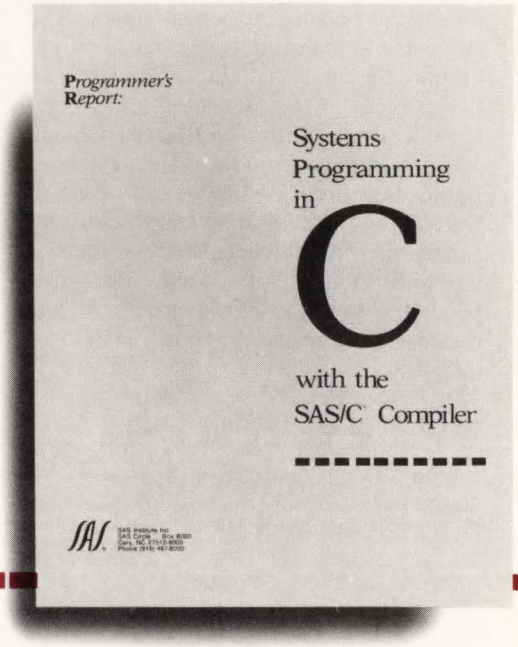
SPE is an extension to the C language that greatly simplifies the coding of user exits, tools, and utilities for JES2, VTAM, CICS, TSO, GCS, and other systems software. Included are support routines that allow you to write and execute C programs and a compact runtime library that features both general purpose and system specific functions for memory management, interrupt handling, low-level I/O, and more. There's also a utility that translates assembler DSECTs into C structure definitions—an enormous time-saver when you're writing programs that interface with assembler.

Together these tools provide a freestanding C environment designed to interact with the operating system the same way assembly language programs do. With SPE, your C programs can:

■ call existing assembler routines ■ generate any machine instruction in-line ■ easily access system data and control blocks ■ exploit BSAM or CMS file system I/O ■ process asynchronous events and interrupts ■ directly use SVCs and DIAGNOSEs

Then, at compile time, the SAS/C compiler's global optimizer will compress your code to produce routines that rival assembler for speed and efficiency.

With frequent updates and knowledgeable technical support—both provided free—the SAS/C compiler is the best investment you can make toward greater systems programming productivity.



Learn More in a FREE Programmer's Report

To find out more about systems programming with the SAS/C compiler, simply clip the coupon below and mail today. We'll send you our new **Programmer's Report: "Systems Programming in C"**. Or call us today to find out how you can receive the SAS/C compiler for a free, 30-day evaluation.

SAS Institute Inc.
SAS Circle □ Box 8000
Cary, NC 27512-8000
Phone (919) 467-8000
In Canada, call (416) 443-9811

The SAS/C compiler runs under MVS (370, XA, and ESA) and VM/CMS on IBM 370/30XX/43XX/937X, and compatible machines.

___ **Yes, send me a FREE copy of "Systems Programming in C".**

___ **Contact me with details of a FREE 30-day trial of the SAS/C® compiler.**

Please complete or attach your business card.

Name _____

Title _____

Company _____

Address _____

City _____ **State** _____ **ZIP** _____

Hardware System _____ **Operating System** _____

Mail to: SAS Institute Inc., Attn: ME, SAS Circle, Box 8000, Cary, NC, USA, 27512-8000. Telephone (919) 467-8000.

MJ0489

central storage in parallel on the number of available swap paths. Tasks, such as TSO tasks that fail the *think time* criteria for logical swap but pass the criteria for swap to expanded storage, are logically swapped in the transition to expanded storage.

Expanded Storage Transfer Time

Movement of pages between central and expanded storage occurs in microseconds versus the milliseconds required to move a page to and from auxiliary storage. The number of microseconds of transfer time given by IBM is 70 to 75 microseconds per page. Approximately one-third to one-half of this time (25 to 35 microseconds) is actual hardware transfer time and the rest is instruction path length to allocate receiving frames and initiate page transfer. The hardware bus used for page transfer is rated at a speed of 220K bytes per second which would give 18 microseconds of actual hardware transfer time. This time is increased by hardware protocol for fetching data from the sending memory and storing data to the receiving memory.

My experience indicates that of the pages sent to expanded storage approximately 98 percent are returned to central storage either through page faults or migration processing. Using a guideline of five percent of the processor complex allowable for page transfers (the old demand paging guideline), you see in Figure 4 that a significant number of pages can be transferred between central and expanded storage within this guideline. Multiple transfer times are used in Figure 4 since the actual page transfer time may increase slightly due to bus queuing as the page transfer rate increases.

Central or Expanded Storage?

When memory constraints surface in a configuration, many times a difficult question to answer is, "Do I add central or expanded storage?" There are currently no rules-of-thumb and each configuration has to be evaluated with respect to both workload usage of memory and service level requirements. We can make an estimate through review of the nature of workloads or subsystems active on the processor complex. If the primary subsystems active on a processor complex are non-swappable, such as CICS/VS, IMS/VS and DB2 workloads, the primary use of expanded storage in a non-ESA environment will be for demand paging and stolen pages when central storage becomes constrained. In this case, addition of central storage will probably provide the greatest improvement in subsystem performance. Demand and stolen pages going to expanded storage may eventually be migrated thus requiring the subsystems to retrieve the pages from auxiliary storage and defeating the purpose of expanded storage.

If the workloads on the processor complex are largely swappable, such as TSO and batch workloads, the addition of expanded storage should significantly improve the performance of the configuration. Even with a low percentage of Logical Swap Effective as reported on the RMF Swap Activity Report, addition of expanded storage should increase the most important percentage: Logical and Expanded Effective (these are the percentages of tasks swapped to their respective locations and returned to central storage without being transferred to auxiliary storage).

A detailed analysis of central and expanded storage usage by tasks and subsystems active on a processor complex requires reduction and analysis of the measurement data found in the SMF Type 30 accounting records and the RMF Monitor II Type 79, subtype 1 (ASD) records.

MVS/ESA and Expanded Storage

Recent announcements and availability of MVS/ESA components have shown intent to make significant use of the expanded storage technology through use of dataspaces and hiperspaces. Though dataspaces are created in central

FIGURE 3

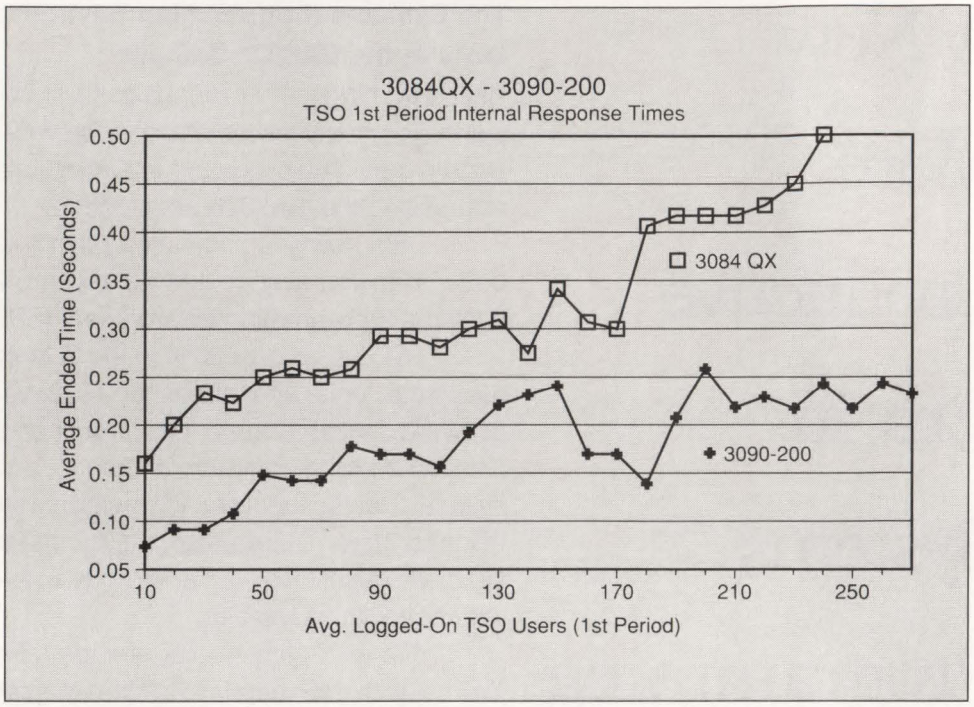


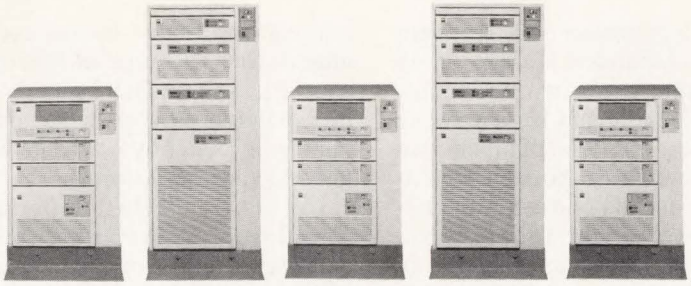
FIGURE 4

Expanded Storage Page Transfer Guidelines

(Five Percent Of A Processor Complex)

	70 Microseconds	80 Microseconds	100 Microseconds
3090-200	1,428	1,250	1,000
3090-400 (E)	2,856	2,500	2,000
3090-600 (E)	4,284	3,750	3,000

The values in the table are thresholds based on five percent use of the total processor complex for page transfer and should not be taken literally. The amount of page transfer activity expanded storage that your configuration can sustain and still meet service level objectives is a value you must determine.



One good decision leads to another.

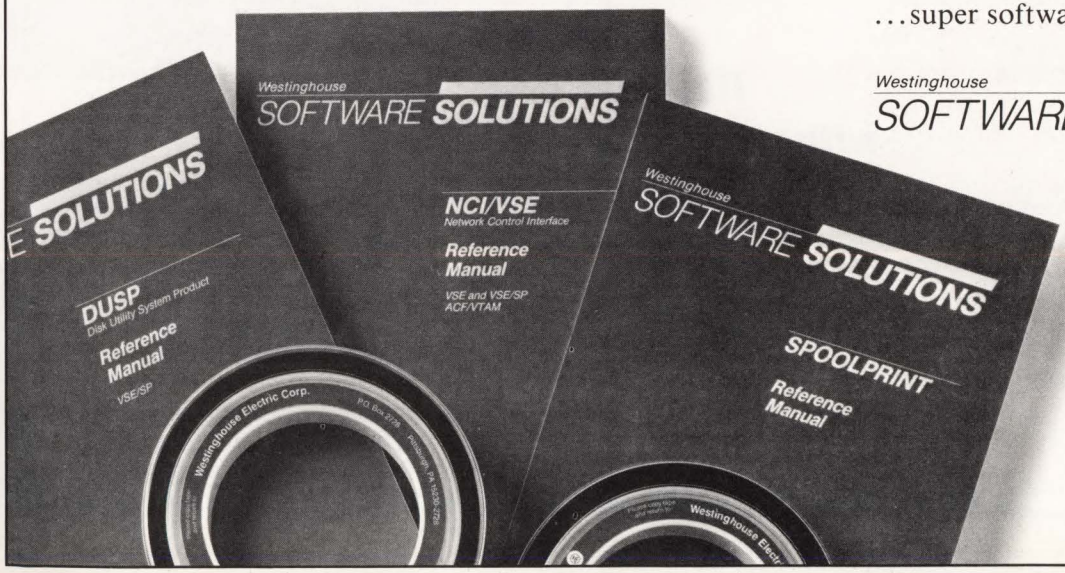
IBM 9370 and Westinghouse Software Solutions

If you're expanding your computer capabilities with the new IBM 9370 Supermini computer...congratulations! We can help you get that new system up and running quickly... to realize the productivity benefits in your data center right away.

Westinghouse has been providing Software Solutions for over 18 years. We have a family of proven systems software for MVS, VSE and VM operating systems...to perform the basic tasks like backing up data...to the complex tasks like tying a group of 9370s into your network and getting a single system image. Our broad range of products includes a network access and control package, multiple session manager, disk utility system, the WESTI TP monitor, disk space manager, job accounting package and many more.

If you want to get on a fast track to computer productivity, you don't need to go any further than Westinghouse...we're a name you can be sure of.

Westinghouse Software Solutions ...super software for that Supermini.



Westinghouse
SOFTWARE SOLUTIONS

Westinghouse

Management Systems Software
P.O. Box 2728
Pittsburgh, PA 15230
(800) 348-3523
(412) 256-2900 in PA

IBM is a registered trademark of International Business Machines.

CIRCLE #59 on Reader Service Card ▲

storage, their pages will be sent to expanded storage if the creating task is swapped or the pages are stolen. A special type of dataspace, termed a Disabled Reference Storage (DREF) dataspace, can be created by authorized tasks. When the DREF pages are sent to expanded storage, they are not migratable. This allows authorized tasks to take page faults while in a disabled operating state without having to retrieve the page from auxiliary storage.

The use of hiperspaces, whose pages reside only in expanded storage or on auxiliary storage, takes two directions. The first is the creation of Class 1 or cache type hiperspaces. This type of hiperspace can only be created by authorized tasks, is used as a cache facility and is not migratable except under certain special circumstances and will allow the retention in expanded storage of data that requires frequent and immediate access. The second is Class 2 or scroll type hiperspaces that can be created by any task but are migratable. This type of hiperspace will allow a task to retain frequently used data in expanded storage while pages containing infrequently used data will be migrated to auxiliary storage.

Use of Large Virtual Buffering and the direction of VSAM LSR subpools to dataspaces and expanded storage could enhance the performance of many existing application systems and particularly those applications in the CICS/VS, IMS/VS and DB2 arenas.

Another article will explore dataspace and hiperspace creation and use of the DSPSERV and ALESERV macros as well as the new data windowing services.

Summary

Expanded storage technology will continue to extend the horizons of the MVS operating system and provide new capabilities for both application designers as well as subsystem implementations. Enhancements in processor storage page and swap capabilities as well as data storage capabilities will bring about larger memory sizes for both central and expanded storage with an expectation of expanded storage being by far the largest on most processor configurations. As uses of expanded storage increase, you can expect to see more changes in MVS internal algorithms for control, access and management of memory. These changes will be discussed in future articles.

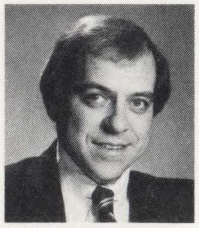
EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
240 (Interesting/Helpful), 241 (Too Technical), 242 (Too Basic)

2.) Would you like more articles on the same subject?
243 (Yes), 244 (No)

ABOUT THE AUTHOR



J. William Mullen is a senior consultant for BGS Systems, Inc. He has more than 20 years of data processing experience and has concentrated on the performance measurement and capacity planning area for the last 15 years. Mullen is a past officer and director of SHARE, Inc. and CMG, serving for the last eight years as editor of the CMG Transactions.

IBM mainframe application software has a new DB2 champion

We're Lawson, the IBM mainframe business application software company with the confidence to challenge the industry's Goliaths. And hit them with new ideas. Like low cost, low maintenance—yet full-featured—business application software.

Lawson: Leader in software technology.
Our new technology is rapidly showing the "big guys" a few things about PINSTRIPE® low cost, low maintenance software. Forget "other" IBM mainframe software. With its high prices, old code, time-consuming installation and expensive upkeep.

New technology made our lower costs possible. It also made us first with application software for the DB2 relational DBMS. Now you get a choice: DB2 or VSAM. With newly developed code that's easy to install. Simple and inexpensive to maintain.

Solid application technology.
Comprehensive, well-thought-out application features reflect the experience of our package designers. Seasoned accountants, CPAs and human resource professionals who know what features and functions your users need.

And features they request. In the last four years, 84 percent of user-group suggested enhancements have been added to Lawson PINSTRIPE® packages.

Service and support — cutting the "big guys" off at the knees.
We're big enough to get the job done—but small enough to be sensitive to our clients' needs. With the best service and support available on the IBM mainframe market today. Including your personal software consultant, an expert who knows your needs *and* your application.

A full range of optional Support-Plus services is available, too. Like implementation consulting, data conversion, technical and application consulting. And a post-sale audit to make sure you're utilizing every feature we've built into PINSTRIPE® packages.

Unique needs? We'll adjust our packages to fit. And guarantee the work. If you'd rather do the job yourself, you'll find Lawson software easy to modify. And we'll still support the rest of the package. Other application software companies won't.

Lawson, IBM mainframe software's new champion.
We've put it all together. New technology with fresh code, including both VSAM and DB2 expertise. Full-featured application technology. Backed up by support that sets a new industry standard.

All in low cost, low maintenance application software from Lawson. No wonder we can take on the "big guys." And win.

Lawson Associates Inc. provides a full range of feature-rich PINSTRIPE® business application software packages for the IBM mainframe hardware environment. Available in the U.S., Canada and overseas.

Accounting
• General Ledger • Accounts Payable
• Accounts Receivable • Fixed Assets
• Project Accounting

Human Resources
• Payroll • Personnel • Benefits

Distribution/Procurement Management
• Order Entry • Inventory Control
• Purchase Order

LAWSON

...WHATEVER IT TAKES!

For more information, call
Sue Weinacht
(612) 379-0258
LAWSON ASSOCIATES INC.
1300 Godward Street, Minneapolis, MN 55413

CIRCLE #32 on Reader Service Card ▲

CICS Shared Resources

Triumphs And Tragedies

By Richard G. Nikula

With the introduction of Local Shared Resources (LSR) into CICS, IBM added a powerful tool for improving CICS performance. However, when used incorrectly, it is easy to reduce the effectiveness of LSR and even degrade response time. This article will present LSR to the user and introduce a methodology for its effective implementation.

Most articles on CICS performance mention the use of LSR. Why? When implemented correctly, LSR can be as close to a no-lose situation as you will find in performance tuning. It can mean a reduction in virtual storage, channel and DASD usage, string waits, OSCOR-related problems and potentially, real storage. But most important, it means response time improvement. Before I continue, I need to take a few minutes to compare performance with and without the use of LSR.

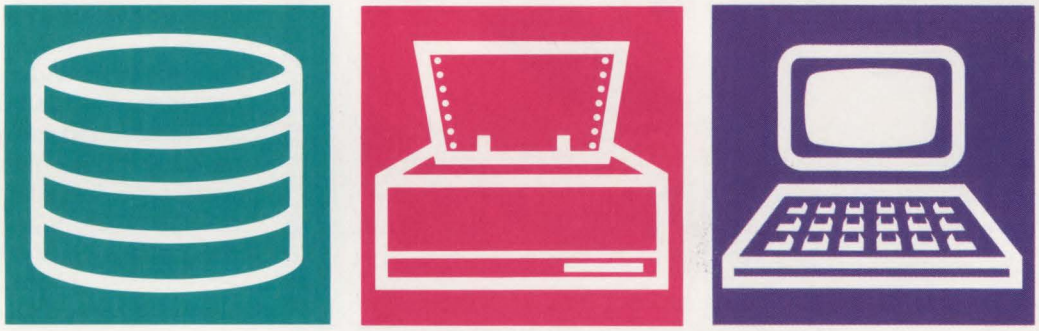
First, I will explain the sample application used for this analysis. It is a fairly simple application that manages an automotive parts file. It has a large user base and normally processes about three transactions a second. The application programs perform a variety of file services with the largest percentage being direct read by part number. It also performs updates (about one for every five direct reads) and browses — when the part number is not known (about one for every 15 direct reads). Since some cars have more problems than others, some parts are referenced more often than others.

Figure 1 shows a common buffer arrangement for non-LSR file definitions. Three buffer configurations and their varying effects on our application will be explained.

Case One

This case presents a definition with minimal data and index buffers specified — that is, one data and index buffer per string with one additional buffer for splits. In this case, all request types will operate similarly. Using one of the buffer sets associated with one of the strings, CICS will retrieve the highest level index first and will then retrieve subsequent lower-level indexes until the sequence set record is read. At this point, the required data control interval will be read into the associated data buffer. For a file with three index levels, this would require three I/Os for the index plus one for the data. Since I used the minimum buffer allocation, you can see that a lot of I/O had to be done to get the records desired. Worse yet, assume that another operator requests the same part at about

BUNDL®
DELIVERS!



Automated report distribution the way you want it.

Department A gets the whole report. Location B receives only its pages. Supervisor C needs the location summaries from all applications combined — online. BUNDL delivers exactly that. On time. Every time.

System Managed Output

Just automating MVS report distribution is not enough. BUNDL is an all-in-one product that brings automation to the production, archiving and online viewing of reports.

BUNDL puts MVS to work and puts you in control. It eliminates manual tasks and unneeded reports. It eliminates reruns. It bundles reports from multiple applications.

WYNIWYG

What You Need Is What You Get. BUNDL is effective because it gets users involved. BUNDL's online View facility lets them look at reports. Rearrange information. Produce reports on paper or fiche. Or archive reports for printing or reprinting later.

Leading in Automation Technology

Multi-image Manager in 1981. AutoMate in 1987. CheckOut/VM in 1988. Now BUNDL in 1989. We've become the recognized leader in the technology and support of Automated Operations.

Don't wait for automated report distribution. For delivery, call **412-323-2600** and ask for your Duquesne Systems account representative.

DUQUESNE SYSTEMS

Two Allegheny Center
Pittsburgh, PA 15212

Remote printing on a PC without a single compromise.

Until now you've had to rely on a S/36 or AS/400® to deliver remote printing. Now a PC with BARR software and adapter sustains print speeds of 6,000 lines-per-minute and line speeds of 128,000 bits-per-second. Only BARR maintains all this performance with the reliability and ease of use PC users expect.

BARR/SNA RJE or BARR/HASP software drives up to five printers from a single PC. What's more, you can enter data, print, and receive output all simultaneously – without interruption. BARR's advanced multi-tasking software easily manages even the most complicated tasks, including LAN access, tape support, file transfer, and special forms printing. In addition, BARR offers one year of friendly, dedicated customer support with each purchase.

Communications adapters and software are available for the IBM PC, PS/2, and compatibles.

Try BARR for 30 days. We've helped thousands save millions. Call 800-BARR-SYS.

Protocols		
SNA RJE with Multiple Sessions		
HASP – BSC Multileaving		
Host Systems		
JES2	DOS/POWER	CDC NOS
JES3	RSCS	VS1/RES

BARR

BARR SYSTEMS INC.
2830 NW 41 Street, Gainesville, FL 32606
800-BARR-SYS, 904-371-3050, FAX: 904-371-3018

AS/400 is a trademark of IBM Corp.

CIRCLE #91 on Reader Service Card ▲

the same time. This entire process will be repeated. Even if both requests are performed simultaneously, both requests will do all four I/Os.

There is no sharing of buffers allocated to different strings. In fact, a record could be in the process of being updated in one buffer and being read from another at the same time. Obviously, there must be a better way.

Case Two

This case presents the same definition with two changes. First, the number of index buffers is increased to eight. Second, the number of strings is increased to four. The latter was done to provide additional concurrent processing. The extra index buffers will be used to keep the high level index and some lower level indexes in storage to reduce the need to re-do the index I/O as in Case One. The highest level index will always remain in storage. Lower level indexes will be overlaid as needed. However, you will still read the same data record multiple times. It may still exist in multiple buffers and be updated in one of them, as in Case One.

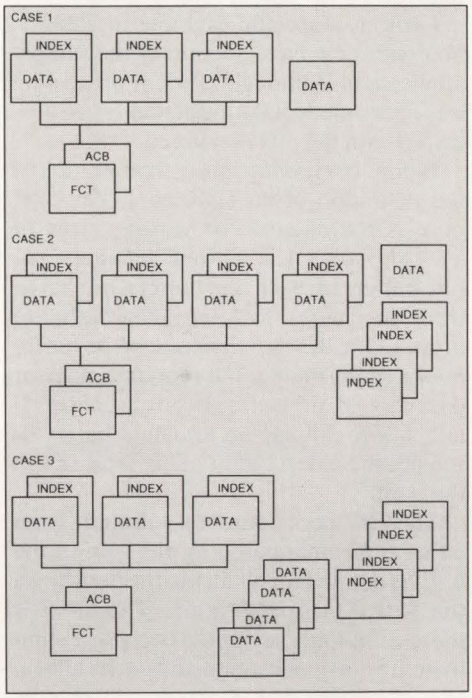
If only direct requests are processed, this is a much better implementation than the previous case. However, as the number of index levels and index records increases, the need for additional index buffers increases as well. The number of index buffers can be increased until all index records are in storage. However, the application also did some sequential processing. The next case will address this.

Case Three

This case is similar in definition to Case Two. The extra string was removed. The number of data buffers was increased as well as the number of index buffers. These extra data buffers will not be used to bypass reading data buffers as with index buffers. Instead, they will be used for chaining I/O for sequential processes (in this case — browsing). This is based on the anticipation that multiple browse requests will be issued and the required data will already have been retrieved. However, if the application reads only a few records or uses skip sequential processing, the extra I/O may actually be wasted. The buffers remain allocated to the string until the request terminates. If a transaction makes a sequential request while another is still active, it will not be allocated any additional buffers. As with Cases One and Two, the same record may exist in multiple buffers.

You have now seen three common buffer

FIGURE 1

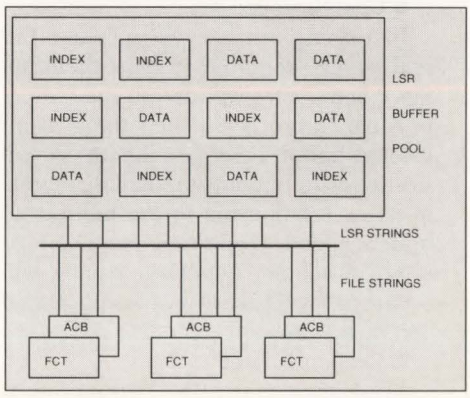


configurations for non-LSR files. Figure 2 shows the configuration when these files are defined as part of an LSR pool. I will now look at the effect of LSR on each of the cases discussed. This is a simplified discussion; we will go into more detail shortly.

Case One

This case produced the worst performance in non-LSR mode. If we cannot improve on it, then there is not much point in continuing. Fortunately, we can. LSR introduces the concept of a look-aside. A look-aside occurs when, rather than reading an index or data CI from DASD, the CI is located in a buffer in storage. Buffers are managed using a least-recently-used algorithm based on all other buffers of the same size sharing the buffer pool. This technique will obviously have a significant

FIGURE 2



effect on Case One that did no look-aside processing at all. One important change is that a record may no longer exist more than once in the buffers.

With non-LSR, you can have multiple copies of a CI being read by multiple strings. Only one string can update the CI. With LSR, only a single buffer is used for all requests. Because of this, any request needing control of the buffer will lock out any other request for the same buffer. For example, if a record is read for update, an exclusive request for another record in the same data buffer cannot be made until the prior request is complete. This applies to update and browse as well.

Case Two

This case will not be affected as significantly as was Case One. Some look-aside processing was done for index buffers. No data buffer look-aside was done so you should get improvement here. Case Two did define an extra string. Since one potential problem with adding strings is increased contention in the I/O subsystem, the reduction in I/O brought about by use of LSR may allow extra strings to be added without an I/O increase. Sequence set index records perform differently in true LSR than in this case. The sequence set record is read into the buffer associated with the string and will not be reused except by the next request on the same string.

Case Three

In this case, extra data buffers were added to allow for chaining of sequential requests. LSR makes no provision for this chaining process. That is, sequential requests will be allocated only a single data buffer in which to do I/O. However, sequential processes may still benefit from look-aside processing for data and index buffers. Depending on the request and usage pattern, LSR may still be more efficient than non-LSR. However, due to the least-recently-used buffer management process, a browse can flood the LSR pool.

At this point, I will summarize what has been seen so far, even with this simplified example. First, you have seen how to reduce the number of physical I/Os due to look-aside processing. Second, there is a potential for reduction in virtual storage as the need to allocate additional data and index buffers is replaced with a pool of buffers shared among multiple files using a least-recently-used algorithm. Also, additional strings could be allocated with less impact than with non-LSR. But of course,

the most important thing I mentioned was that you would reduce response time.

Look at a specific example in Table 1. You will see a cross section of the sample application. Although random processing was performed, 500 transactions were processed with the mix as defined.

Before continuing, have another look at the definition of an LSR pool. An LSR pool is a collection of various sizes of VSAM buffers. Rather than defining separate buffers for each file, buffers are shared (thus, the name). To determine whether to do an I/O, CICS will first look in the buffer pool to determine if the required index or data control interval is already available. If not, CICS will use an available buffer or replace the oldest buffer of the same size in the pool.

The LSR pool definition will come from either a formula based on the sum of the buffers defined for all files in the pool or via the SHRCTL specification. The latter is more commonly suggested because it eliminates the over allocation and under allocation of buffers due to activity differences not taken into account by the formula. Care must be taken; if a file uses a Control Interval (CI) Size for which there is no buffer, the next greater size will be used, that could waste significant buffer storage.

As you might expect, there are limitations to the effectiveness of LSR when adding additional buffers. Here are some examples.

■ Look-aside Processing

Initially, added buffers will noticeably increase the look-aside percentage. However, at some point, the increased buffers will add only a small percent. Do not over allocate buffers.

■ CPU Processing

Look-aside processing replaces I/O with processor time. As the number of buffers increases, the time spent searching for the correct buffer also increases. This is especially true prior to DFP 2.3 that introduced a hashing technique.

■ Real Storage

Increasing buffers will increase the real storage required to back them up. Since active files will use all the buffers possible, there is no such thing as an inactive buffer. Thus, you can expect real storage requirements to increase almost in direct relationship to the number of buffers added. If additional real storage is not available, increased paging will occur; you will have lost what you gained and then some.

■ Response Time

Most importantly, the bottom line is response time. Your greatest improvement will come from the initial implementation of LSR with added buffers bringing less significant improvement.

TABLE 1

	Minimum Buffers	Extra Index Buffers	Local Shared Resources
Total Transactions	500	500	500
Average FC Calls	1.5	1.5	1.5
Average I/O Waits	1.5	1.5	0.8
Average EXCPS Wait	2.7	1.5	1.0
Average Service Time	.039	.035	.032
Average Response Time	.251	.179	.166
Improvement vs #1	—	29%	34%
Improvement vs #2	—	—	7%
Index EXCPS	1,278	370	4
Data EXCPS	739	738	398
Improvement vs #1	—	45%	80%
Improvement vs #2	—	—	64%

Essentially, you will need to add buffers, track the effect, increase or decrease buffers and repeat the process to determine the best specifications for your environment. Depending on system resources (CPU, real storage and so on) you will need to decide what you are willing to commit.

You might be asking yourself, "Is it safe to put all of my applications into LSR?" Unfortunately, the answer is no. There are several reasons why you may not or cannot put an application into LSR.

The first case is the famous browse/update problem. This is the case in which an application program browses a file looking for records to update. This will work fine in the non-LSR case as the record being browsed and the record being updated are actually using separate strings. In the LSR case, the browse will get shared control of the buffer containing the record. When the application requests exclusive control of the same buffer to do the update, it will wait infinitely on itself. In order to resolve this, the application must be changed to either end the browse prior to the update or to do direct reads (with greater than or equal). You may decide that these changes are not feasible, in which case you must use a non-LSR definition.

The second case is applicable if you are on CICS 1.6.X. In this environment, you must not place an alternate index path or a base dataset with an alternate index and upgrade defined into LSR. Doing so compromises CICS exclusive control and can cause corruption on the datasets. This restriction is lifted with CICS 1.7.0 and above.

The third case deals with applications that are sequential and not random in nature. LSR is most effective when records are requested randomly. For example, if an application sequential reads many records at a time, each new buffer needed will have

to eliminate an older buffer in the pool. Since buffers just read by the application are not going to be oldest, other buffers for other applications will be affected. In the worst case, the sequential application could flush all buffers. A true sequential application should probably not be put in LSR; rather, use non-LSR approach number three.

A fourth case is a large dataset that, because of its size or structure, would undermine the look-aside concept.

A fifth case is a dataset that has a lot of split activity, particularly Control Area (CA) splits. If this is the case, allocating extra data buffers for a non-LSR file will improve the CA split processing more than LSR will. Of course, it is still better to eliminate the excessive split activity by allocating free space or redesigning applications' use of the file.

One final case, often overlooked, is an application that requires specific tuning or has unique buffer requirements. Since LSR provides limited tuning capabilities, an application may not be well suited for LSR or vice versa. For example, a critical application may run only infrequently. However, when it does run, you want it to run as fast as possible. If it is part of an LSR pool, the infrequent access may cause every access to the file to be physical I/O. This application may get better response time when separated into its own non-LSR buffers. Additionally, CICS 1.7.0 and above offers another alternative that will be discussed later in this article.

At this point, you should be convinced (or confused) about implementing LSR. Before you run right out and implement it at your site and then blame me for any problems, go over a possible methodology for attacking it. If you already have implemented LSR, perhaps this will give you some pointers on what to look for when you get back.

By far, the most important thing for successful implementation of LSR is standards. Without some kind of standards, you may improve performance but you may also worsen it. What kind of standards are required? The single most important one is definition of acceptable values for CI Sizes for data and index components. Without this standard, you will not be able to predict buffer requirements, understand the effect of buffer specifications or tune buffer requirements. Commonly, the following standards are used.

Index Control Interval Size

Either 512K or 1,024 is suggested most commonly. This is normally as large as index CIs need to be. However, there are cases in which large keys and poor key compression have made 2,048 practical to avoid premature splitting of the control area. LSR also makes 2,048 a good choice. Index CI standards must not specify any values that are also data CI Sizes. Doing so will reduce the effectiveness of LSR since index and data will then have to contend for the same buffers.

Data Control Interval Size

Common recommendations here are 2,048, 4,096 or 8,192. The number, 2,048, is small and normally recommended to reduce the I/O time when reading a single record. Since you are in LSR, look-aside processing will potentially eliminate the I/O and more data in the buffer increases the probability of getting a look-aside hit. Because of this and the possibility of needing 2,048 for index buffers (remember, this is a shop-wide standard), 2,048 should be avoided for data CI Size. The number, 8,192, is normally suggested for sequential applications since it means obtaining more data with each I/O.

As I have already discussed, LSR and sequential applications do not mix well. But 8,192 may still be viable as a means of isolating applications from each other within the same LSR pool. Larger CI Sizes may also be used but you should evaluate the needs and effects of each one. For example, if an infrequently used file requires a CI Size of 28K and it is the only one you have, placing it in LSR means allocating 28K buffers that are used infrequently. It would be better to redefine the file into an already used buffer size. I would suggest using 4,096 for most applications. For the same reasons as mentioned above, data CI standards must not overlap with index sizes.

Here are some general notes on defining CI Sizes. Stick with common sizes, even though values like 7K and 10K are valid CI Sizes. The obvious reason is that this does not require you to maintain a lot of different LSR buffer sizes. Another reason is that LSR supports only buffers of specific sizes. Using other sizes is permissible for VSAM but LSR will use the next larger supported (or defined) increment that is a waste of buffer space. Prior to DFP2.2, some sizes also cause additional I/O (albeit chained) and extra DASD space to be used. This is because VSAM will choose a physical block size based on CI Size and device type. For example, a CI Size of 7K will use a physical record size of 1K. 2K

multiples will use 2K physical records and 4K multiples will use 4K physical records. Thus 12K will be somewhat more efficient than 10K.

I previously stated that not having standards affects your ability to manage your CICS system. Look at some reasons for this.

Inability To Predict Buffer Requirements

If you cannot anticipate buffer sizes, you really should not use the SHRCTL macro to define buffers. The best you could do is to take the default 50 percent of all buffers or a percentage set via the RSCLMT parameter. In either case, you are not taking into account any unique requirements.

If a new application is defined to your CICS system, you cannot predict how many and what size buffers to expect without doing something like LISTCAT against all fields to determine CI Size. It is not impossible but it is time consuming and can change at any time if an application changes CI Size.

Inability To Understand the Effect of Buffer Specifications

Trying to understand what effect LSR is having is difficult if you do not know what applications are using what buffer sizes. If applications change CI Sizes from one day to the next, look-aside percentages could change drastically.

FIGURE 3

CICS REGION 1

FILE NAME	DATASET NAME	DATA CI SIZE	DATA EXCPS	DATA BFRS	INDEX CI SIZE	INDEX EXCPS	INDEX BFRS
FILEA	TEST.VSAM.FILEA	4,096	18,864	5	1,024	27,812	4
MASTER	PROD.VSAM.MASTER	4,096	23,678	4	2,048	62,172	3
AUDIT	TEST.VSAM.AUDIT	8,192	10,112	8	1,024	1,189	7
QUIKFILE	TEST.VSAM.QUICK	10,240	5,121	12	4,096	2,561	6
TRACK	PROD.VSAM.TRACKER	2,048	28,165	7	512	63,117	3
CUSTMST	PROD.VSAM.CUST	12,288	66,859	5	2,048	87,787	4

CICS REGION 2

FILE NAME	DATASET NAME	DATA CI SIZE	DATA EXCPS	DATA BFRS	INDEX CI SIZE	INDEX EXCPS	INDEX BFRS
FILEA	TEST.VSAM.FILEA	4,096	11,134	5	1,024	3,121	7
MASTER	PROD.VSAM.MASTER	4,096	14,564	4	1,024	2,202	8
AUDIT	TEST.VSAM.AUDIT	4,096	23,843	8	1,024	4,569	8
QUIKFILE	TEST.VSAM.QUICK	8,192	6,101	4	512	1,211	6
TRACK	PROD.VSAM.TRACKER	4,096	25,894	7	512	7,814	7
CUSTMST	PROD.VSAM.CUST	8,192	72,898	7	2,048	1,163	12

TABLE 2

	DATA CI SIZE	DATA BUFFERS	INDEX CI SIZE	INDEX BUFFERS	TOTAL FILE CONTROL CALLS
FILEA	4,096	5	1,024	4	20,000
FILEB	4,096	5	1,024	4	22,000
FILEC	8,192	5	1,024	4	15,000
FILED	8,192	5	1,024	4	500
FILEX	8,192	10	1,024	4	7,500

Difficulty In Tuning Buffer Specifications

Again, if you do not have any way of predicting buffer requirements or any influence over buffer size, you will not be able to tune buffer specifications. That is not to say you cannot do anything. However, new applications and changes in existing applications make it a difficult task requiring constant analysis. Having standards in place simplifies this process.

To get control of your CIs, you will need two basic standards. First, applications must conform to a standard CI Size based on file characteristics and usage. You must know what CI Size an application is using in order to allocate additional buffers, if necessary, prior to its going into the production LSR pool. Second, an application must not change the size of the CI Size without first notifying you so that the effect on buffer allocation can be evaluated.

Another standard you may want to implement deals with the use of browse versus direct reads. Due to the nature of LSR, direct reads are somewhat better than browse, especially when the number of records retrieved is large but the number actually processed is small. A direct read requires control of the CI briefly while browse holds it until the browse progresses to the next CI or ends. Since browsing can be imitated using a direct read with greater than or equal specified, this can be used instead.

If an application will be doing a significant number of browse requests, it may not be a good candidate for LSR; this should be determined prior to placing it into an LSR pool. These issues could be discussed at application design review time to give the designer the opportunity to re-evaluate if possible.

Assuming that you put these standards in place or choose to use "pot luck," the next stage is to evaluate your existing environment. There are several tasks involved. Look at each of these.

Understand Your Existing Distribution of CI Sizes

You need to determine what is currently being used. With LSR, the need to know CI Sizes is more important than with non-LSR. If you have set up standards, you will need to verify which applications conform. Figure 3 is a sample report showing two CICS regions prior to LSR implementation. Region 1 is not ready for much LSR implementation. Region 2 is in much better shape with a few exceptions.

Set Up the LSR Buffer Pools

To determine a starting point for allocating LSR pool buffers, you may want to take the quick way initially and not use the SHRCTL macro. This will most likely over allocate some buffers and under allocate others. You can then obtain CICS statistics or numbers provided by almost any CICS performance product to evaluate where to increase and decrease buffers. Another option is to do some up-front analysis — that is, determine the number of buffers currently allocated for files going into LSR and their respective CI Sizes. Use the following case in Table 2 (you will have more files to evaluate).

Evaluate File Usage

The purpose here is to determine which files make good LSR candidates and which do not. Figure 4 shows a sample report that could be used to make such a determination. This information is also available from CICS shutdown statistics. The point here is to spot files in which browse activity is high compared to direct requests. These files may be better left out of LSR. Low utilization files are also good candidates for LSR, assuming they are not part of a highly critical application. A percentage of the buffers used by these files can be placed into the LSR pool to be used by other applications.

Starting with the index buffers, you currently have a total of 20 index buffers all using a CI Size of 1,024. If you just use the default 50 percent, CICS will allocate 10 buffers. This is too low. These are probably cookbook definitions: that is, four index and five data buffers are always allocated regardless. Since the index definition is probably not performing well to begin with, cutting the buffers in half will not make any friends. In fact, I would encourage increasing the index buffers. Since the 1,024 buffers are small, virtual storage will not add up too quickly. Again, once we have defined them, you can look at the statistics to determine if you have over-allocated them.

Now for the data buffers. Two sizes are used — 4,096 and 8,192. For the 8,192 CI Size, allocating the default 50 percent results in 10 buffers. For the 8,192 buffers, this may be a good starting point. However, both FILED and FILEX have limited activity. FILED really does not need any dedicated buffers of its own. FILEX has over-allocated data buffers for sequential processing. Since the number of buffers being discussed is small, you may let this extra buffer become part of the pool. However, if we had a lot of files and many were over-allocated, the end result would be a surplus of buffers.

For the 4,096 buffers, 50 percent may be too low again. Since these two files have equal activity, they would both try to keep active buffers in the LSR pool. Thus, each would continually be doing some I/O to place buffers in the pool. Of course, as the number of files increases, the larger the

number that 50 percent becomes. Likewise, the probability that not all files will be active files also increases. You may be conservative with data buffers to improve virtual storage constraint but you will need to add buffers if buffer waits occur and if it is necessary to improve look-aside percentages. The point of this analysis is that while the default allocation of 50 percent may work, it may also result in too few or too many buffers. With a little more work, a lot of grief and time explaining what happened may be avoided.

Determine Number of LSR Strings

When defining your LSR requirements, one other decision to make is the number of strings. As with buffers, you have the option of defining the number of strings with the SHRCTL macro or a percentage of the total strings defined. But unlike buffers, the string parameter in the file entry and the value for the LSR pool both have significance. At the LSR level, the string limit defines the number of concurrent requests that may be active for all files (including extra strings for alternate index processing). At the file level, the string limit defines the number of concurrent requests that may be active for that particular file even though additional strings may still be available at the LSR level. So it is possible to get string waits at both levels. String waits serve a purpose as they reduce the load on the LSR pool and reduce DASD activity. However, they should occur in only a small percentage of all requests. Remember that browses, updates and mass inserts hold strings longer than direct requests.

One other important issue not to forget when implementing LSR is the browse/update problem. If an application with this scenario is moved to LSR, it will hang. So, you need to look for this type of situation.

You may still miss some potentials (the only-on-a-full-moon syndrome) but trying to catch them is worth the effort. Figure 5 shows an example of a report selection and sample output that could be used to spot potential problem transactions. This uses CICS MANAGER (Boole & Babbage) performance reporting language but any tool that provides detailed transaction information can be used. As you can see, the point is to search for transactions doing both browse and update activity. Once potential applications are spotted, they can be reviewed on your test system or reviewed for actual function prior to implementation into production.

A final word of advice. DO NOT IMPLEMENT LSR ALL AT ONCE. It is better to implement LSR in stages for several reasons. First, it means less analysis at each stage (although the total analysis time will be greater). Second, it means that you will be able to review the effect that a given stage had on performance and tune if necessary before continuing to the next stage. Third, if something does go wrong, only one specific group or application is affected making communication easier (although probably not less emotional). Fourth, since implementation is done in stages, so is your testing. You can take a more active role to ensure that testing is done meaning you should worry less about problems.

FIGURE 4

FILE NAME	BROWSE COUNT	READ COUNT	GET UPDATE	UPDT COUNT	ADD COUNT	DELETE COUNT	CA SPLITS
TEST	1,123	98,124	21,123	21,123	514	0	0
MASTER	18,123	5,122	1,122	1,122	2,123	0	1
AUDIT	0	0	0	0	6,123	0	4
CUSTMST	0	88,543	123	16	1,567	87	0
TRACK	5,812	6,232	0	0	0	0	0

FIGURE 5

```
SELECT TYPE 6E RECORDS FROM CMRDETL
  WHERE T6EFCPC > 0
    AND T6EFCBC > 0
USING T6ETRID T6PGNM T6EFCGC T6EFCPC T6EFCBC T6EFCAC T6EFCDC REPORT
```

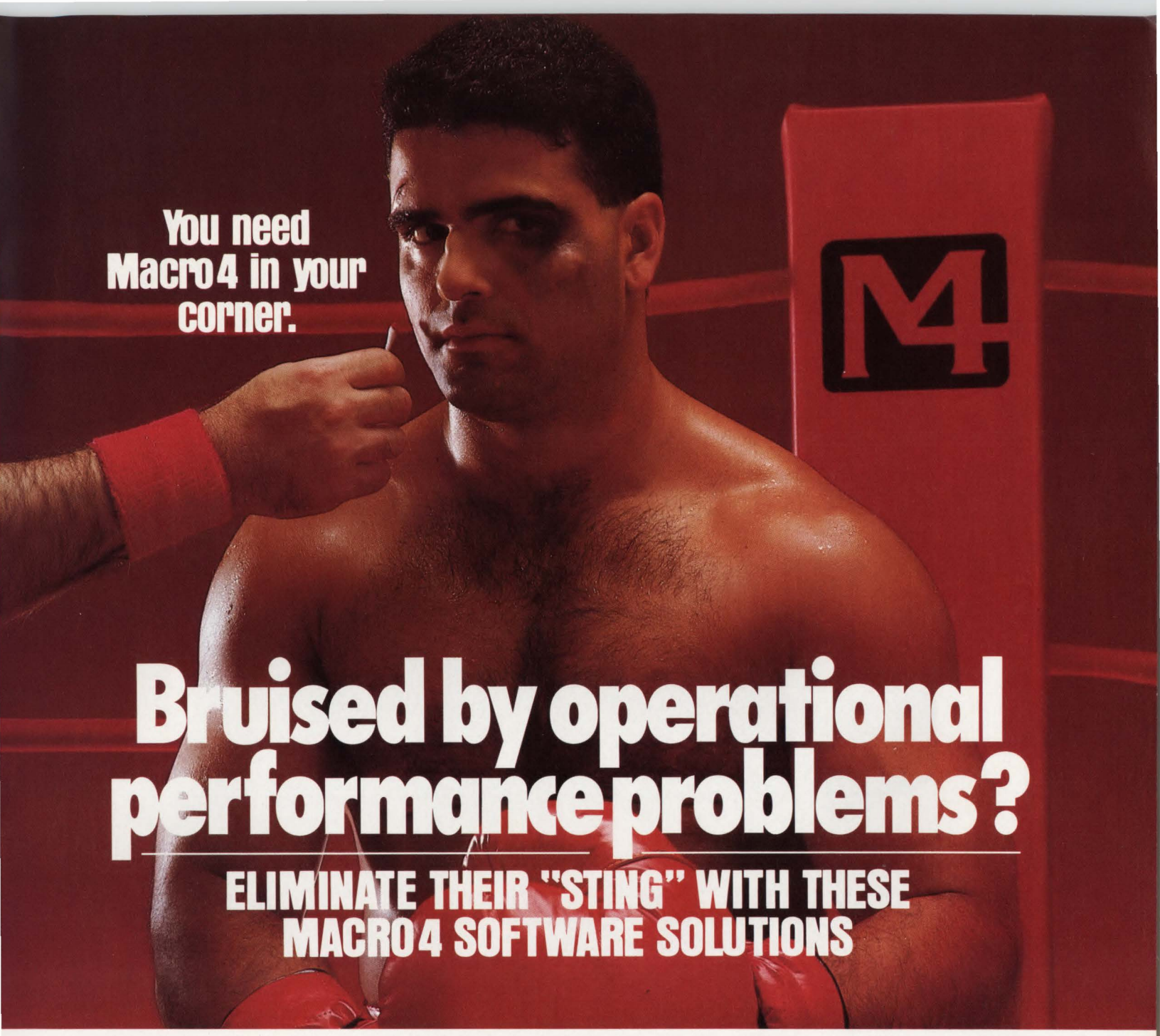
TRAN ID	PROGRAM NAME	GET COUNT	PUT COUNT	BROWSE COUNT	ADD COUNT	DELETE COUNT
AB01	AB019012	0	3	3,056	0	0
TR01	TR01CR12	5	1	16	0	0
AB01	AB019012	0	5	2,123	0	0

CICS 1.7.0 and Above

At this point, I am going to go off on a tangent. I am going to look at the changes that CICS 1.7.0 and above brought about in this area.

Most significant is that LSR is now the default (actually LSR pool #1 but I will discuss this shortly). Some people have been interpreting this incorrectly. They seem to believe that this is the go-ahead from IBM to throw everything into LSR.

As I have discussed, not everything should be in LSR. It is more like an urging to use the facility rather than a blessing. If you are a non-LSR environment, migrat-



Bruised by operational performance problems?

ELIMINATE THEIR "STING" WITH THESE MACRO 4 SOFTWARE SOLUTIONS

DUMPMASTER — eliminates the time-consuming problems of program debugging and dump handling by offering you:

- simplified on-line dump management
- on-line debugging via CICS, CMS, TSO or ICCF
- handles both batch program and CICS abends
- reduction of dump printing costs and programming time

LOGOUT — offering unrivaled DOS/VSE system control, this superb console manager gives you:

- sub-console facilities at CICS or VM terminals
- rapid access to multiple system consoles from any terminal for precise, single-point control
- full support of all SYSLOG and POWER commands
- built-in help facilities
- time limiting and progress reporting
- instant access to data in POWER queue files

CICSPRINT — designed to maximize your spooling, and report handling capabilities by providing:

- POWER to CICS, CICS to POWER and VM to POWER spooling
- improved and simplified end-user facilities
- greater flexibility and control with minimal operator intervention
- unmatched versatility to effectively save print time and reduce costs.

Don't settle for mere "band aid" solutions. Let the power of Macro 4 help protect your system from operational performance "bruises"!

TUBES: World's Best Session Management Tool • VSAM/TUNE: The Leading VSAM Performance Tool • VPAC: The Unbeatable VM Performance Analyzer and Control Facility

WE'LL HELP YOU DELIVER YOUR OWN KNOCK OUT PUNCH!

CALL 800-223-0414

AND PUT THE POWER OF MACRO 4 IN YOUR CORNER.

We've been providing powerful solutions to MIS system problems on an international scale since 1968 — with a full range of software solutions for VSE, VM and MVS operating system requirements.

Macro 4

Systems Software

Brookside Plaza, P.O. Box 187, Mt. Freedom, NJ 07970

(201) 895-4800 • (800) 223-0414

INTERNATIONAL OFFICE: MACRO 4 PLC Crabbet Park House, Turners Hill Road, Worth, Crawley, West Sussex, RH10 4SS England (0293)886060
OTHER OFFICES: France, Italy, Switzerland, Belgium, West Germany, Netherlands, Spain, Scandinavia, Israel, Australia, Japan and Pacific Basin.

CIRCLE #199 on Reader Service Card ▲

How to lose wait.

```
SELECTION CRITERIA - XASC21.DXPSAMP.SELCRIT(INVFILEC) ------------- LINE 00001
COMMAND ---> _                                                    SCROLL ---> CSR
CRITERIA NO  1
-------LEVEL NUMBER/DATA-NAME------- -FORMAT- RO ---------FIELD VALUE----------
01  INVENTORY-RECORD
  05  PART-NO                        C  15
  05  DESCRIPTION                    C  40
      (POS 31-40)
  05  STOCK-INFO
    10  STATUS                       C   6    EQ B/O
    10  UNIT-OF-MEASURE              C   2    EQ EA
    10  UNIT-PRICE                   PS  5  2
    10  WAREHOUSE                    C   3
    10  QTY-ON-HAND                  PS  5    BT 0;5
    10  QTY-RESERVED                 PS  5
    10  QTY-BACKORDERED              PS  5
    10  FILLER                       C   1
  05  REORDER-INFO
    10  REORDER-POINT                PS  5
    10  REORDER-QUANTITY             PS  5
    10  LEAD-TIME-DAYS               Z   3
  05  VENDOR-INFORMATION
      OCCURS     3 TIMES
  05  VENDOR-INFORMATION(1)
```

With powerful productivity tools.

The time wasted trying to manipulate data with time-consuming utilities and one-time programs is enough to spoil anyone's appetite.

We designed the XPERT Series to allow you to easily, accurately and quickly manipulate data in MVS data files, and IMS and DB2 data bases.

Quick access to data made the way you like it.

DATA-XPERT, IMS-XPERT and DB2-XPERT all have an ISPF-like design, so you'll learn faster and produce results quicker.

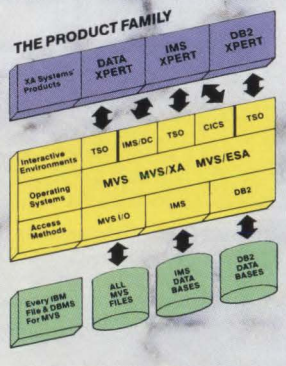
You'll easily access data files such as VSAM, and IMS and DB2 data bases.

Working with any size file or data base, you can selectively edit, browse, extract, load, convert, reformat and print any size file or record to get just the data you want to work with.

Try it, you'll like it.

Find out why the XPERT Series is rated #1 over the competition.* Call today for more information or a free 30-day trial of DATA-XPERT, IMS-XPERT or DB2-XPERT.

Call 1-800-344-9223;
in Canada, 1-800-344-9224.



The XPERT Series from XA. You may never look better.

XA SYSTEMS CORPORATION

The Essential Software Company

*Hancock Survey, January 1989. DATA-XPERT and IMS-XPERT are registered trademarks of XA Systems Corporation. DB2-XPERT and The Essential Software Company are trademarks of XA Systems Corporation. ISPF, IMS and DB2 are products of IBM.

CIRCLE #134 on Reader Service Card ▲

ing to CICS 1.7.0 and LSR at the same time will make your task more difficult although not impossible. I would suggest migrating to LSR and CICS 1.7.0 at different times.

Another significant change (although available to an extent with MVS/XA DFP, CICS 1.6.1, and IMS 1.3) is that the DL/I buffer pool and the CICS LSR pool are no longer one and the same. This has made it more difficult to tune LSR due to the addition of the DL/I activity. It could also seriously affect the effectiveness of LSR and CICS when DL/I activity is high. Note that this facility is available only with MVS/XA DFP even for CICS 1.7.0 and above.

A useful facility added as of CICS 1.7.0 is the ability to define up to eight LSR pools. The default for a file is LSR #1. Again this facility is available only with MVS/XA DFP. Allowing files to be put into different pools adds additional tuning issues and some solutions to problems created by a single LSR pool. Here are a couple of examples.

Pool Hogging

This is the case of an application for which placement in LSR is beneficial. However, due to activity and access patterns, it dominates the LSR pool making other applications suffer. Without multiple pools, the only option would be to define it using unusual CI Sizes so that it would be isolated to a given set of buffers. With multiple LSR pools, it can be moved to its own pool and segregated from other applications.

Application Splitting

This is the case of an application that has particular performance objectives (either critical or non-critical). Placing it into the LSR pool creates the potential that its own buffers may be stolen (for the non-critical case). This made it a poor candidate for LSR in the past. But with multiple LSR pools, the application can be moved to its own pool and segregated from other applications.

Read Prior to Add

A change as of CICS 1.7.0 that may go unnoticed with LSR but noticed without LSR is the solution to the famous duplicate record DELETE problem. Prior to 1.7.0 if a recoverable application issued an ADD, received a duplicate record condition and then either abended or issued a rollback request, the original record was deleted.

To get around this problem, CICS will attempt to read the record prior to adding it. Without LSR, the read request causes potential I/O as does the corresponding add. LSR reduces the I/O potential, as the read request simply primes the buffers required.

Additional SHRCTL Specifications

Since multiple pools are defined, the discussion on estimating buffers and strings increases in complexity to some degree. Analyzing statistics for multiple pools requires that you understand the segregation of applications or workloads by LSR pool.

Pool Selection

Since you have eight pools, how do you decide which pool to put a new file into? My advice is that you typically use LSR pool #1 and use the other pools for special purposes only. After all, a single pool can do a lot on its own. Without DFP 2.3, some CPU consumption may be eliminated by using multiple smaller pools rather than a single large one.

Deferred Pool Creation

Another feature as of CICS 1.7.0 is that the LSR pools are created at first open for a file in the pool and deleted at last close. In CICS 1.6.1, the pool was created at CICS startup and remained present until CICS shutdown. In fact, one file had to be defined as open-initial or the LSR pool was not created.

If SHRCTL was not coded, CICS would have to read the catalog for each entry which could significantly increase startup time. In addition, no provision existed for dynamic allocation so if the dataset was not allocated at startup, allocations could be different than expected.

With CICS 1.7.0, the LSR pool is not built until a file in the pool is opened (note that CICS 1.7.0 also supports a new form of deferred open). As with CICS 1.6, if a SHRCTL is not coded, CICS must interrogate the catalog to determine the buffer requirements. It also allows for dynamic allocation (through the FCT, not via other vendor products) when obtaining pool requirements. This may appear to the end user as a fairly lengthy response time.

The pool will remain allocated until all files in it have been closed. Subsequent opening of another file in the pool will repeat the process. One thing in particular to watch out for is an application designed to do its own opens and closes. One in particular is CEDA (on-line resource definition).

CEDA opens and closes the DFHCSD dataset each time it is invoked. If DFHCSD is part of an LSR pool (which, of course, it typically is), it can cause the pool to be created and deleted many times (assuming other files in the same pool are not also open).

Okay, now back on to the original track. As you can see, LSR is a useful facility. At the beginning of this article, I stated that LSR is almost a "no-lose" situation. Well, there is no free lunch. Although LSR provides a lot of benefits, it requires a lot of analysis up front for correct implementation. But we are not done yet.

No CICS region is static. Applications come and go. They are changed and new features added. A new level of CICS and operating system code is introduced. All of these play a part in day-to-day life. You cannot just define LSR and then leave it. Even if your applications do not change, you may encounter problems due to the integral relationship between CICS and DFP. Yes, there have been APARs in this area. But your applications do change and you need to keep on top of them. Here are a few suggestions for ongoing analysis.

Application Profiling

I am a strong supporter of doing ongoing application profiles. An application profile is an analysis of the normal process and resource usage by transaction or workload. The intent is to spot transactions that are not within common norms for your installation. You should already be doing this type of profile. If you are not, here are a few examples of information you might include:

- Transaction or Workload Identifier
- Response Time
- Resource Usage (CPU, Storage, Paging and so on)
- File Control Call Type and Counts
- File Control Request Time per Transaction
- String and Buffer Wait per Transaction
- Temporary Storage Requests and Timings.

As you saw in the earlier example, each improvement in file access affects these numbers. By comparing profiles of LSR applications with themselves and other applications, you will be able to see any inconsistencies or trends introduced by application or tuning changes.

File Profiling

This is another type of profile that very few people do. Like an application profile, the intent is to see trends and irregularities. A file profile should contain information such as the following:

- File name
- Dataset name
- Volume information
- Data and index CI Size
- File control request types and counts
- VSAM request types and counts
- Data and index EXCP counts
- Data and index buffer specifications
- File string information
- LSR pool number (if 1.7.0 and above, only if LSR in 1.6).

The more pertinent information kept with the profile, the easier to understand the cause when numbers start changing.

Local Shared Resource Data

Data at this level need not be kept at the level required for files and applications. This is a high level indicator to give an overall feeling for LSR definition. When doing detailed analysis, the information at the file level is better for determining changes required.

Control Interval Reporting

Just a simple report to let you know that everyone is following standards. This information could also be obtained from the profile but a profile may be done weekly or less frequently. This report should be run daily, even during the day. The purpose is to spot any deviation as soon as possible and correct it.

Summary

First of all, you have seen how LSR may reduce the resources required to complete a simple I/O request. Since I/O request time can account for a significant percentage of transaction response time, you can improve internal response time even while reducing resource usage. I did point out, however, that adding resources will reach a point of diminishing returns and the decision must be made whether to continue to apply resources. With the possibility of being accused of heresy, I have even identified cases in which LSR is not applicable.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
245 (Interesting/Helpful), 246 (Too Technical), 247 (Too Basic)

2.) Would you like more articles on the same subject?
248 (Yes), 249 (No)

ABOUT THE AUTHOR

Richard G. Nikula is a senior consulting software engineer at Boole & Babbage. He is one of the authors of the CICS MANAGER product. He has had previous experience as a CICS systems programmer working with IMS and DB2. Boole & Babbage, 510 Oakmeade Parkway, Sunnyvale, CA 94086, (408) 735-9550.

MAINFRAME JOURNAL
For Users of IBM System 370 Architecture & Compatible Systems

Multiple VAE Mode Machines & VM

In the article, "Multiple Address Spaces in VSE" (March 1989), the author correctly describes the advantages of running VSE in VAE mode. However, what I tend to disagree with are the points about running a VAE mode machine under VM.

The author said that it was possible to do this but both VM and the guest VSE would page, causing double paging. This can be eleviated by running the VAE mode machine in a V=R area under VM; in a V=R machine only the VAE guest does the paging. Also CCW translation can be turned off, since VM does not need to do the translation. He also said, "There is only one VAE guest allowed per VM system." This is incorrect; one can run multiple VAE mode machines under one VM system. What cannot be done is to run more than one V=R area. Therefore, if more than a single VAE machine is running under VM, double paging and double CCW translation would occur only in these.

What we have done at our shop is dedicate 7MB of real storage to our V=R VAE production machine (out of 16MB real storage on a 4381 Model 3). This allows us to run other non V=R VAE mode machines strictly for testing and development purposes. For our own tech support testing, we also bring up a VSE machine running in VM mode.

This method has allowed us the comfort of 40MB multiple address spaces on both the production and main development machines and also lets the production machine in the V=R area run at top performance by eliminating double paging and double CCW translation.

As you can see, I am a big fan of VAE mode but I'm also a big fan of VM. Therefore, I cannot see throwing out VM in order to get the many advantages of running VSE in VAE mode; nor can I see overlooking VAE if one desires to run multiple VSE machines under VM.

P.S. Of all the magazines that come across my desk, yours is by far the most informative and my favorite, thank you.

Danny Mastre, Coordinator of Technical Support
Pioneer Mutual Life Insurance Co., Fargo, ND

Lighten Up!

With April comes the only day specifically set aside for foolishness and good humor. In honor of April Fool's Day, we offer this submitted piece intended to give you a chuckle.

IBM SYSTEM/370-XM (EXTENDED MNEMONICS)

XA, ESA, ES/370? What does it all mean?

When IBM introduced "XA" or Extended Architecture technology, we observed an event of such magnitude that it finally allowed us to soar above the hum-drum life of 24-bit addressing. What? A gigabyte instead of a megabyte? What an incredible concept! More power and storage than we could ever imagine!

Well, obviously our imagination could have gone much further. Along came the magical world of ESA and behold, we could now entertain the notion of a terrabyte! IBM had skillfully listened to its customers' needs and demands and had provided many growth paths to fully utilize the super-power of the high-end mainframe environment.

SOMETHING MISSING

XA, ESA are noble concepts. However, there was obviously something missing. We had extended architecture, sure, but a part was missing that we desperately needed to *unlock the power* of these behemoths!

Well, as our user community has so skillfully illustrated in the past, when IBM omits (or forgets, or refuses!) to implement a certain facet of our system that could increase power substantially, we are ready to step up to the plate and provide the answer (thanks, Pete!).

As a member of the IBM user community, we stand ready to furnish the answer that will truly provide the extended functionality we require.

THE SOLUTION

We already have XA. What about "XM"?? XM or Extended Mnemonics will provide the power to truly drive systems to their full potential. These extended System/370 instructions are available immediately, require nothing more than imagination to implement and immediate benefits may be realized. These are just a few of the possibilities I have come up with over the years.

These enhancements are available free of charge for the asking. I will do my best to provide support as needed. However, "No representation as to the merchantibility or fitness for a particular purpose . . ."

I welcome all suggestions and additions to the extended mnemonics list that follows. Please forward all ideas to *MAINFRAME JOURNAL* and the staff will ensure that I receive your input. When I compile the final list, I will approach IBM about inclusion in a future release (once again, thanks Pete).

S/370 EXTENDED MNEMONICS

MNEMONIC	MEANING
AI	ADD IMPROPER
ARX	ADD AND RESET TO ZERO
BBI	BRANCH ON BLINKING INDICATOR
BCBE	BRANCH ON CHIP BOX EMPTY
BCBF	BRANCH ON CHIP BOX FULL (NATURALLY)
BCBM	BRANCH ON CHIP BOX MISSING (OF COURSE)
BCR	BACKSPACE CARD READER (VALID FOR 2540 ONLY)
BD	BACKSPACE DISK
BH	BRANCH AND HANG
BIPBS	BRANCH IF PER BIT SET
BODT	BURN OUT DISPLAY TUBE
BPO	BRANCH ON POWER OFF
BSP	BACKSPACE PRINTER
BSPP	BACKSPACE PRINTER AND PUNCH (SIMULTANEOUSLY)
BST	BACKSPACE AND STRETCH TAPE (NOT FOR PAPER TAPE)
BX	BRANCH EXCLUSIVE
CCBR	CORRUPT CORE BEYOND RECOGNITION
CM	CIRCULATE MEMORY
CRN	CONVERT TO ROMAN NUMERALS (IBM ITALY ONLY)
CSO	CANCEL SYSTEM ORDER
CVM	CONVERT TO METRIC
CVU	CONVERT TO UNARY (BASE 1)
DO	DIVIDE AND OVERFLOW
DVC	DIVIDE AND CONQUER
ED	EJECT DISK (MULTIPLE OPERANDS ALLOWED)
EM	EMULATE 407
EO	EJECT OPERATOR
EROM	ERASE READ ONLY MEMORY
FSRA	FORMS SKIP AND RUN AWAY
HCF	HALT AND CATCH FIRE
IIB	IGNORE INDICATOR AND BRANCH
IR	INCLUSIVE OR
MC	MOVE CONTINUOUS
MDB	MOVE AND DROP BITS
MLR	MOVE AND LOSE RECORD (ONLY FOR MASTER FILES)
MTI	MAKE TAPE INVALID
MWC	MOVE AND WRAP CORE
PD	PUNCH DISK (NO REFERENCE TO ALTERNATE TRACKS)
PIHC	PUNCH INVALID HOLE COUNT
PRF	PUNCH READ FEED
RASC	READ AND SHRED CARD
RBT	REWIND AND BREAK TAPE
RCR	REWIND CARD READER
RCSD	READ CARD AND SCRAMBLE DATA
RD	REWIND DISK
RICS	READ INVALID CARD SIZE
RIRG	READ INTER-RECORD GAP
RNR	READ NOISE RECORD
QNR	QUIET NOISE RECORD
RP	READ PRINTER
RPB	REVERSE PARITY AND BRANCH
SRSD	SEEK RECORD AND SCAR DISK
SRZ	SUBTRACT AND RESET TO ZERO
SSJ	SELECT STACKER AND JAM
SSLTC	SHORT SLT CARD
TDB	TRANSLATE AND DROP BITS
TDCS	TEAR DATA CELL STRIP
UER	UPDATE AND ERASE RECORD
WDC	WRITE DATA CHECK
WEC	WRITE EQUIPMENT CHECK
WMNM	WRITE MORE NEW MNEMONICS
WWLR	WRITE WRONG LENGTH RECORD
XN	EXCLUSIVE NEITHER
XOL	EXCLUSIVE OR LONG
XT	EXCLUSIVE AIN'T (TEXAS 370 ONLY)

Eric L. Vaughan, President
Smartech Systems, Inc., Dallas, TX

VM/CMS XEDIT

How To Invoke And Interpret Its Display

By Steve Eckols

This article is intended to teach you how to use the powerful text editor that comes with VM: the System Product Editor, called XEDIT for short. Although you can easily learn to do simple things with XEDIT, the fact that it is a sophisticated program means you need to learn many commands and techniques to get the most out of it. That is the purpose of this article: to help you learn not only the easy, obvious commands but also those that are less obvious — so you will be a more productive user of your system.

If you work on an IBM mainframe, the chances are good that you either use VM now or will in the near future. Although VM has been around for more than a decade, it is growing in popularity because it is easier to use than other operating systems for IBM mainframes. That is due in large part to its being designed from the ground up as an interactive operating system. In addition, it can make more efficient use of a system's hardware.

As a VM user, you need to be able to work effectively with XEDIT. Even though VM has more of an interactive flavor than IBM's other mainframe operating systems, you still need to be able to create and maintain files that contain text, whether that text is test data, word processing-like files, program source code, job streams for other operating systems or procedures for the CMS environment. Although a number of different text editors are available for IBM mainframes, XEDIT is widely used because it is an integral part of VM and because it is such a powerful program in its own right.

So whether you are a programmer, a systems analyst, an administrator, a data entry operator or a manager on a VM system, you will benefit from learning how to use XEDIT. After you have, you will find that your job will be easier to do and that you will get more out of your system.

To use XEDIT, the first thing you have to do is start it. So that is what you will learn how to do here. Then, after the editor has started, you have to be able to identify the different areas of the XEDIT screen to work effectively.

How to Start XEDIT

Usually you will invoke the editor by simply entering the CMS command XEDIT followed by the identifier of the file you want to edit. For example, I would enter XEDIT LISTMAST COBOL B to edit a COBOL source file named LISTMAST on my B-disk. Because you can abbreviate the XEDIT command as simply X, the command X LISTMAST COBOL B is equivalent to the first command. Figure 1 presents a simplified syntax format for the XEDIT command.

You will almost always enter the first two operands — the filename and filetype of the file you want to edit. If you do not, XEDIT looks to a special file of editor specifications for a default name. For now, just assume that you will always supply the fielname and filetype operands when you invoke XEDIT.

With XEDIT, you can either access an existing file or create a new one with the filename and filetype you supply. If XEDIT finds a file with the filename and

DON'T BURY YOUR INFORMATION IN PAPER

Gain instant on-line access to vital reports.

The Innovator—WSF2: True Report Management Software

Your mainframe was not designed to be a printing press. It's there to provide vital information. As fast as you need it. And on-line or on paper, no report management software system lets your mainframe do its job as quickly and conveniently as WSF2.

Now you can selectively distribute reports so each department gets only the specific printouts it needs—even just a page or a line, if that's all that's required.

Plus, WSF2 provides instant on-line access to any part of any report, so your people can simply call up information they need on their monitors. Either way, productivity skyrockets.

WSF2 gives MVS users true flexibility:

- Selective distribution of any part of a printed report, resulting in up to 40% savings on paper costs
- Timely, accurate on-line access to any part of any report
- User-friendly menu and help screens
- Complete, secure archival and retrieval system
- The most sophisticated search and find facility available

Discover the savings of *true* report management. For more information about WSF2, fill out the coupon below or call toll-free for a free brochure. **1-800-777-WSF2.**

RSD AMERICA INC.

100 Merrick Road
Rockville Centre, NY 11570

GENEVA • NEW YORK • LONDON • MADRID
SÃO PAULO • SINGAPORE • SYDNEY • TEL AVIV

RSD AMERICA, INC.
100 MERRICK ROAD, ROCKVILLE CENTRE, NY 11570

Please send me complete information on WSF2.

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

CIRCLE #42 on Reader Service Card ▲

filetype that match those you specified, it opens it and lets you work on it. However, if XEDIT cannot find a matching file, it creates a new, empty one for you.

Where XEDIT looks for a file you specify depends on the value you supply for the filemode operand. If you specify a filemode letter, XEDIT searches only the minidisk associated with it. For example, the command XEDIT LISTMAST COBOL B causes the editor to look only on the currently accessed B-disk for the file LISTMAST COBOL. Even if a file with the same filename and fieltype exists on the currently accessed A-disk that comes before the B-disk in the normal CMS search sequence, it will not be opened by the editor. If you supply a filemode letter and the specified file does not exist, XEDIT creates a new file with that identifier.

On the other hand, if you omit the filemode operand when you enter the XEDIT command or specify an asterisk for it, the editor looks through your currently accessed minidisks in the standard CMS search sequence until it finds a file with the filename and filetype you specified. So if you enter the command XEDIT LISTMAST COBOL or XEDIT LISTMAST COBOL * and files with that filename and filetype reside on both your current A-disk and B-disk, the one on your A-disk will be opened. If there are no files with that filename and filemode on any of your accessed minidisks, XEDIT creates a new, empty file with the filename and filetype you specified on your A-disk.

Figure 1 also includes one option of the XEDIT command, NOPROF. It has to do with actions that occur when you start XEDIT. You can customize your XEDIT environment by storing statements to alter the editor's appearance or functions in your *XEDIT profile,* a CMS file called PROFILE XEDIT. When you start XEDIT, it normally retrieves PROFILE XEDIT and executes the statements the file contains. If you want to bypass the XEDIT profile, you specify the NOPROF option when you invoke the editor. For example, the command X LISTMAST COBOL A (NOPROF invokes XEDIT with its defaults in effect.

FIGURE 1

The XEDIT Command

The XEDIT command

XEDIT [filename filetype [filemode]] [(NOPROF]

Explanation

XEDIT	The XEDIT command invokes the System Product Editor. You can abbreviate the command as X.
filename	The filename component of the file identifier of the file to be edited. This operand is required, along with filetype, unless the file identifier is specified in your XEDIT profile.
filetype	The filetype component of the file identifier of the file to be edited. This operand is required, along with filename, unless the file identifier is specified in your XEDIT profile.
filemode	The filemode letter of the minidisk that contains the file to be edited. If you specify filemode and the file does not exist on that minidisk, XEDIT creates it. If you omit filemode, XEDIT searches all of your accessed minidisks for the file identified by filename and filetype; if that search is unsuccessful, XEDIT creates the file on your A-disk.
NOPROF	Specifies that the statements in the XEDIT profile (a CMS file called PROFILE XEDIT) should not be executed when the editor is invoked.

The XEDIT Screen

If you invoke XEDIT with its defaults in effect, the XEDIT screen will look like the one in Figures 2 through 9. The XEDIT screen has several components: (1) the file area, (2) the current line, (3) the scale line, (4) the command line, (5) the prefix area, (6) the file identification line, (7) the status area and (8) the message line. This section describes each of these components.

If you are following along at your terminal as you read, you may find that the screen XEDIT displays when you start the editor is arranged differently from the examples in Figures 2 through 9. That is nothing to worry about. It is simply due to statements stored in your XEDIT profile that customize your screen's appearance. (If you want to work with the screen layout in the figure and yours is different, invoke XEDIT with the NOPROF option; remember to include the left parenthesis that separates the option from the rest of the command.)

The File Area

The largest component of the XEDIT screen is the *file area,* shaded in Figure 2. It is where the contents of the file you are editing are displayed and where you can key in text directly over existing text to make changes. You can issue commands, either directly or through PF keys, to alter data displayed in the file area or to scroll the text that is displayed. You will learn how to perform basic editing functions and other, more advanced functions in later articles.

Within the file area, one line is always the *current line,* even though the XEDIT display is able to show multiple lines. In Figure 3, the current line is the one that contains the following text:

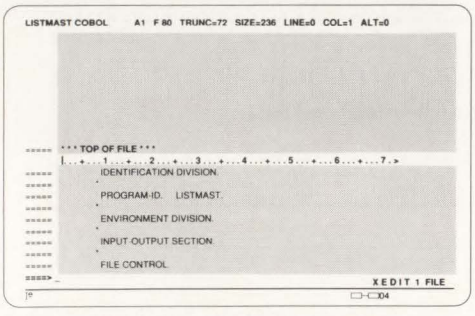
* * * TOP OF FILE * * *

The current line is easy to identify because it is displayed in high intensity characters. (In screen images like the ones in Figures 2 through 9, I represent high intensity characters with bold face type.)

Incidentally, the TOP OF FILE line and a similar END OF FILE line that appears after the last line of text in your file are not actually in your file. XEDIT simply displays them to let you know when you have reached the beginning or end of the file.

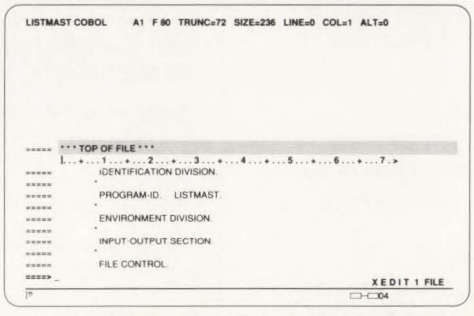
The current line's default position is in the middle of the screen. As the file lines that are displayed scroll up and down during an editing session, the line that is current changes. However, the line that appears at the middle of the screen in high intensity characters, regardless of its content, is the current line. (As with the other characteristics of the XEDIT screen, you can change the default position of the current line. I will show you how to do that in a later article.)

FIGURE 2

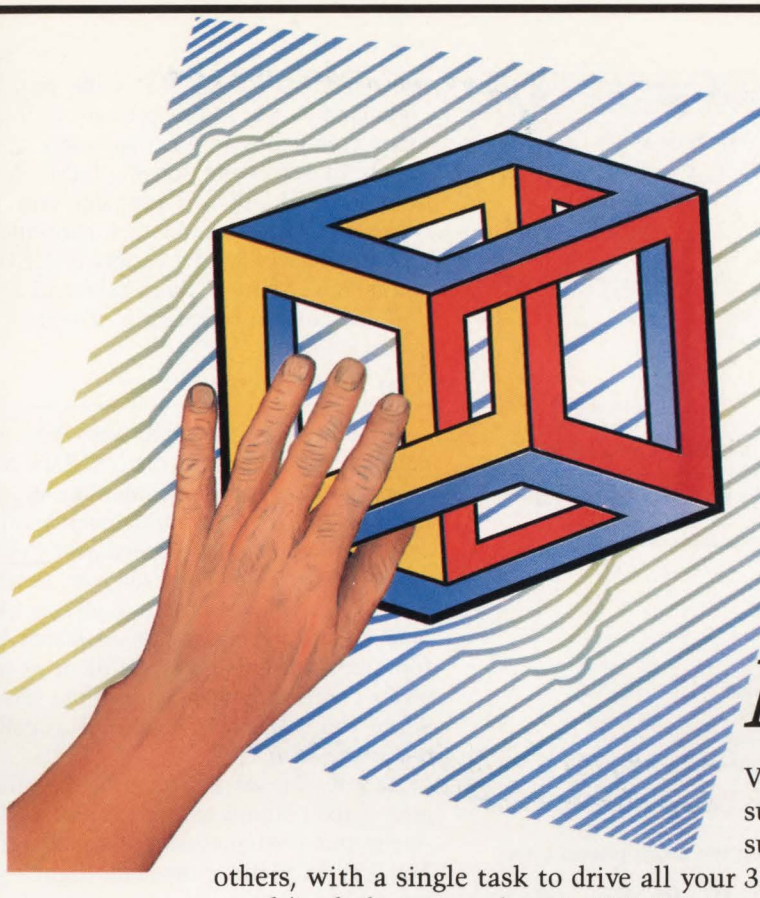


The XEDIT Display's File Area

FIGURE 3



The XEDIT Display's Current Line



VPS Support Of Network Printers On MVS Is A Complex Task Made Simple.

VPS has been used to replace other products, such as: IBM's 328x/ADMPRINT/DSPrint, CICS supported printing, SASWTR®, RJE and many others, with a single task to drive all your 3270 family printers directly from the JES spool (including cross domain VTAM printers).

1700 Sites use VPS as their shop standard 3270 family printer driver.

- Printers supported include the full array of 3174/3274 attached printers, including IPDS support for 42×4's, HP and Xerox lasers, plotters and PC printers.
- VPS runs as a VTAM application. NO system modifications. NO JES maintenance.
- Automatic forms control, full FCB support, dial-up PC printers, printer pooling and "hot" printers are all supported.
- Full screen "ISPF-like" command interfaces for CICS, TSO and ROSCOE® permit end-user control of printers with totally menu driven command entry.

CALL or write for more information, or to arrange for a free TRIAL – ATTN: Marketing Dept.

LRS

Levi, Ray & Shoup, Inc.

Specializing in Computer Systems Software

2387 West Monroe ◆ Springfield, Illinois 62704 ◆ (217) 793-3800
Telex (510) 60-00675

*SASWTR is a registered trademark of SAS Institute, Inc., Cary, NC.

*ROSCOE is a registered trademark of Applied Data Research, Inc.

The current line is important because many XEDIT commands affect it or operate on other lines relative to it. You can think of the current line as your "position" in the file as you edit it. Even though a full screen of text appears on the XEDIT screen, the current line is precisely where you are located.

The idea of the current line has its roots in the history of XEDIT. Although XEDIT is a full-screen editor, it is a development of earlier line-oriented editors that were designed to work with teletypewriter terminals. Because a teletypewriter terminal can only handle one line of data at a time, line-oriented editors are designed so operations you request are performed on a single line or on a group of lines positioned relative to a single line. So, the current line is really a carryover from line-oriented editors. (By the way, a term you may come across that is also a carryover from line-oriented editors is *line pointer;* it is synonymous with current line.)

The Scale Line

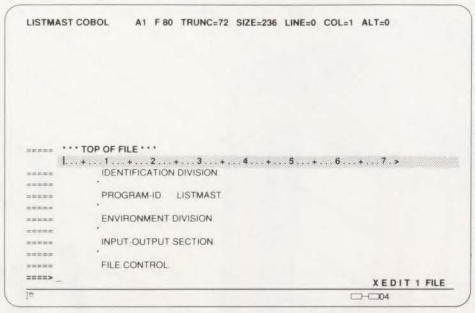
The second major component of the XEDIT screen is the *scale line*. It is the line in Figure 4 below the current line that looks like a ruler. Like the current line, the scale line is displayed in high intensity characters. Its default position is in the middle of the screen under the current line. I will show you how to change its default position in a later article.

In a sense, the scale line is a ruler. You can use it to "measure" the columns in your file. Columns that are even multiples of 10 (10, 20, 30 and so on) are identified with single digits in the scale line. For example, the digit one in the scale line marks column 10 in the file being edited. Plus signs identify columns that are even multiples of five between multiples of 10; the plus sign between the one and two identifies column 15. Other columns are identified by periods in the scale line.

Notice two special characters in the scale line. First, look at the vertical bar in column one. It marks the *current column,* the column where some special column-oriented XEDIT commands will take effect. Most of the time, you do not need to be concerned with the current column.

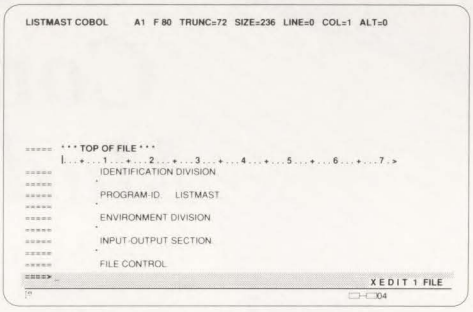
Also, notice the greater than sign (>) on the far side of the scale line. It marks the column beyond which editing changes will not apply. That is, it marks the right side of the editing zone. Here, that is the same as the truncation column that, as you can see in the line at the top of the

FIGURE 4



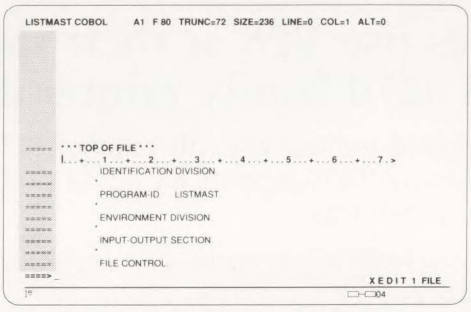
The XEDIT Display's Scale Line

FIGURE 5



The XEDIT Display's Command Line

FIGURE 6



The XEDIT Display's Prefix Area

screen, is set at column 72. You can set both the editing zone and the truncation column to suit your needs. Usually, though, XEDIT picks the appropriate values for the kind of file you are editing based on the filetype you specify when you invoke the editor.

The Command Line

Another area of the XEDIT screen is the *command line*. Its default position, illustrated in Figure 5, is at the bottom of the screen below the file area. The high intensity characters ====> identify it. Like the scale line and the current line, you can change the default position of the command line.

You use the command line to enter a variety of commands to control XEDIT operations. (Strictly speaking, these are *subcommands* — XEDIT is the primary command — but in this article I will call them commands.) You can enter commands to scroll the screen, locate text, tailor the XEDIT environment and retrieve and save files. In each subsequent article, you will learn a variety of XEDIT commands. Many of them have a distinct line and column orientation, a vestige of earlier editors.

The Prefix Area

In addition to the commands you can enter on the command line, XEDIT also supports a set of commands that let you take advantage of its full-screen editing environment. They are called *prefix commands* because you enter them as prefixes to the lines you want them to affect. (Actually, they are *prefix subcommands* but for simplicity I will refer to them as just prefix commands.) The part of the screen where you enter prefix commands is called the *prefix area*.

As you can see in Figure 6, the prefix area is the column on the left side of the screen that contians rows of equal signs. You key in prefix commands right over the equal signs.

Other Areas of the XEDIT Screen

In addition to the areas of the XEDIT screen I have already described, there are three other components that supply you with information as you work. They are illustrated in Figures 7, 8 and 9: (1) the file identification line, (2) the message line and (3) the status area.

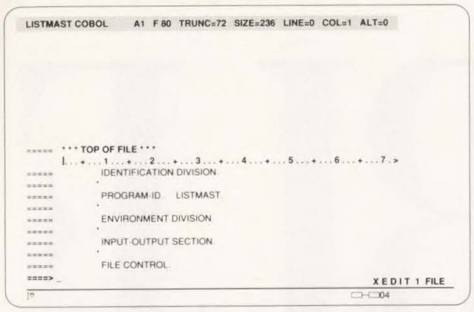
The file identification line

The first line of the XEDIT screen, shaded in Figure 7, is the *file identification line*. It contains the identifier of the file being edited, some of the file's characteristics and current status information for your editing session. In Figure 7, the file LISTMAST COBOL is beng edited.

After the file identifier, the file identification line contains two items of interest about the file you are working on: record format and record length. If the file has fixed-length records, the letter F appears, followed by the length of the records. In Figure 7, the values F 80 mean the file LISTMAST COBOL has fixed-length records that are 80 characters long. If you edit a file with variable-length records, the first value will be a V followed by the maximum record size for the file.

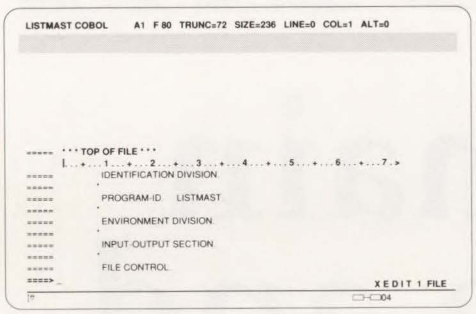
The last five items in the file identification line give you status information about your editing session. The TRUNC item indicates the column after which data

FIGURE 7



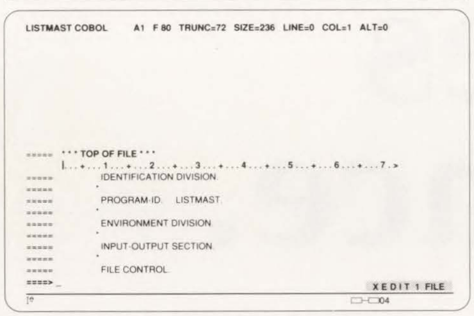
The XEDIT Display's File Identification Line

FIGURE 8



The XEDIT Display's Message Line

FIGURE 9



The XEDIT Display's Status Area

will not be added to lines of the file you are editing. In Figure 7, that is column 72 that is appropriate for a COBOL source language file.

The SIZE item tells you how many records are currently in the file you are editing. In Figure 7, the file contains 236 records. You should realize that this number refers to the work file the editor uses. If you open a file, then add records to it with XEDIT, the new records are stored in XEDIT's work file. When you save your work, they become part of the permanent file named in the file identification line.

The LINE and COL values tell you where the current line and current column are located in the file. In Figure 7, the current line is at the top of the file; it is

See VM/CMS XEDIT *page 43*

CPENABLE

limits domain interference and improves performance.

By Mark Friedman

In this month's column, I will continue the discussion from last month of CPENABLE, an infrequently used and little understood tuning parameter in the OPT member of SYS1.PARMLIB. I will show how CPENABLE can be used to tune a logically-partitioned processor running under (depending on your mainframe vendor) either IBM's PR/SM or Amdahl's MDF.

First, look at an environment in which both PR/SM and MDF excel. Suppose you have an important transaction-processing workload that has grown to use 90-100 percent of a single-engine processor on its own. As the machine approaches its capacity, performance begins to suffer. If it is a well-tuned application and system and the problem is simply one of running out of capacity, there is no alternative but upgrading the processor environment.

Consider the upgrade possibilities. The processor you are currently running is already the fastest in the manufacturer's line. You cannot switch to another vendor's offering because it would mean writing off the company's investment in the current machine at a considerable loss. (This is what someone in accounting tells you. But you figure he knows his business when the salesman from that other vendor tells you the same thing.)

You can "MP" the machine, literally upgrade the computer system from a single CPU to a multi-processor configuration with two or more CPUs, all sharing the same memory and channels. When you double the number of processors in the machine available to do work, you are virtually doubling the current capacity.

When you first "MP" the system, the resulting system is likely to be considerably larger than what would be, strictly speaking, *required* for the workload. Nice when you can swing it but often when you are spending the company's money, you cannot. After all, the new configuration will be significantly underutilized for some period of time. In a chargeback environment, it will be difficult to recover the costs of the upgrade.

But wait, you also find a growing TSO and batch test environment that needs some additional capacity. It is not a big

workload, requiring less than 40-60 percent of a single-engine processor, so together the two workloads would fit nicely into the new machine. You are now in business because the upgrade can be cost-justified. But how can you ensure that you can run them together without the ups and downs of the test environment affecting the stability of a critical production system?

Enter PR/SM and MDF. With PR/SM and MDF you can run each workload under a separate copy of the operating system. Each operating system is totally isolated from the other. From an availability standpoint, the special-purpose PR/SM (or MDF) hardware guarantees the isolation of the two workloads. Each logical partition or domain has dedicated memory and channels but can share the processors. You can set processor-sharing targets so that the production workload gets everything it needs while the test workload is serviced at a lower priority from the ample capacity left over.

PR/SM and MDF Performance

PR/SM or MDF is the logical configuration choice but one that still does have some costs associated with it. Look at PR/SM and MDF performance in more detail. From a performance standpoint, there are some costs associated with running either PR/SM or MDF. There are MP effects in the new environment, PR/SM effects and the additional cost in memory due to running multiple copies of the operating system in one machine.

Simply doubling the number of CPUs in your system will never result in a doubling of capacity due to MP effects. MP effects are the result of two or more processors sharing memory. For the most part the processors execute in parallel but whenever there is a possibility that separate parallel processes may both attempt to update the same location in main storage at the same time, the processes must be serialized. Hardware and software locks are used to guarantee the integrity of instructions and data "critical sections" that are subject to updating by multiple processes.

Serialization and locking are important performance factors in an MP because only one processor at a time can safely run a volatile chain of system control blocks. The performance implication is that when one processor is accessing the dispatcher queue, for instance, another processor needing access to the same resource is "locked out" until the first pro-

cessor is finished. A second performance implication is that the hardware serialization instructions themselves, such as Compare and Swap, take longer to execute on an MP because instruction execution cycles literally must serialize.

Even though CPU cache memory is typically dedicated to a processor, there are performance implications for the CPU cache in an MP. Since most instructions literally execute from cache and not from main memory, it is necessary to reflect main storage updates by one machine in the caches of the other engines in an MP.

In IBM's architecture, a separate cache manager is notified whenever main memory is updated. Before memory in cache can be accessed, the cache manager must be called to verify that it is current. If the memory locations were modified elsewhere, a current copy of the memory contents must be retrieved. The additional cache communication requirements in an MP cause instruction execution cycles to elongate.

PR/SM effects are two-fold. First, there is the obvious impact of an additional workload, the PR/SM dispatcher itself that manages the processor time-slicing. PR/SM overhead is relatively slight but as the highest priority workload in the system, it comes right off the top. The more frequently the PR/SM dispatcher is invoked, of course, as you increase the number of partitions that must be managed, the higher the overhead. As you add partitions, the overhead multiplies.

CPU Cache Performance

The second area of impact is more subtle. In last month's column, I discussed CPU cache performance. When a process begins execution, its ''working set'' of data and instruction areas is acquired slowly, one cache miss at a time. Once the working set is resident, however, there are fewer cache misses and fewer costly main memory accesses. Instruction execution throughput is increased.

The difference in cache performance between a workload that is ''cold-started'' and has to build its cache working set through cache misses and one that is ''warm-started'' with portions of its working set already resident is significant.

Using cache, you would like a program that begins execution to stay in execution as long as possible. On the other hand, when a program needs to perform I/O, it is a good idea to make the CPU available to another program. During the time it takes to do a conventional I/O operation, today's top-of-the-line mainframes are capable of executing hundreds of thousands of instructions. The whole idea behind multiprogramming is to overlap I/O processing where the CPU would normally be idle with the CPU instruction execution cycles of other jobs.

Multiprogramming is operating system support for switching among multiple programs executing simultaneously. Each program being executed is in a different stage of execution and is either ready to use the processor or waiting on I/O completion. Multiprogramming has been a feature of IBM processors since OS/360, and, of course, MVS is a multiprogramming operating system. In a multiprogramming environment, I/O operations are executed in parallel with resynchronization required only at I/O completion which is designed to interrupt the currently executing process.

Multiprogramming causes you to switch the processor back and forth between ready processes, as well as handle I/O completion interrupts asynchronously. For instance, suppose Job A is running and an interrupt to signal completion of an I/O for the higher priority Job B occurs. When the interrupt occurs, the interrupt processor pre-empts the currently executing process (Job A). The IOS interrupt routines are invoked. Once they verify that the I/O has completed successfully, they notify Job B. When IOS finishes, it executes a call to the MVS Dispatcher.

Since the I/O that Job B was waiting on has been processed and completed, Job B is now ready. Job A, that was interrupted while it was in execution, is also ready. Since the highest priority task on the dispatcher ready queue gets control, Job B gets the processor ahead of Job A. After Job B's turn at the processor, Job A is again selected for execution. By now, how much of Job A's frequently referenced instruction and data areas remain in the cache from the previous execution cycle? Little or none.

In fact, you can see that servicing the interrupt as a pre-emptive interrupt initiated a cycle of three different task ''contexts'' being cold-started. Each cold-started process suffers through a period of degraded cache performance while it loads its execution context of frequently referenced instructions and data areas. Frequent ''context-switching'' (IBM's term) is a major cause of performance degradation in today's mainframes.

MVS software is designed to optimize cache hardware performance by controlling the amount of context-switching that occurs. The CPENABLE parameter, as discussed last month, is used in an MP to restrict I/O interrupts to a single processor. The processor that is enabled for interrupts runs an IOS context and little else. Since Job A, running on its own processor, is never interrupted, it is allowed to execute until it relinquishes control of the CPU voluntarily — all the while with its instruction execution working set resident in cache.

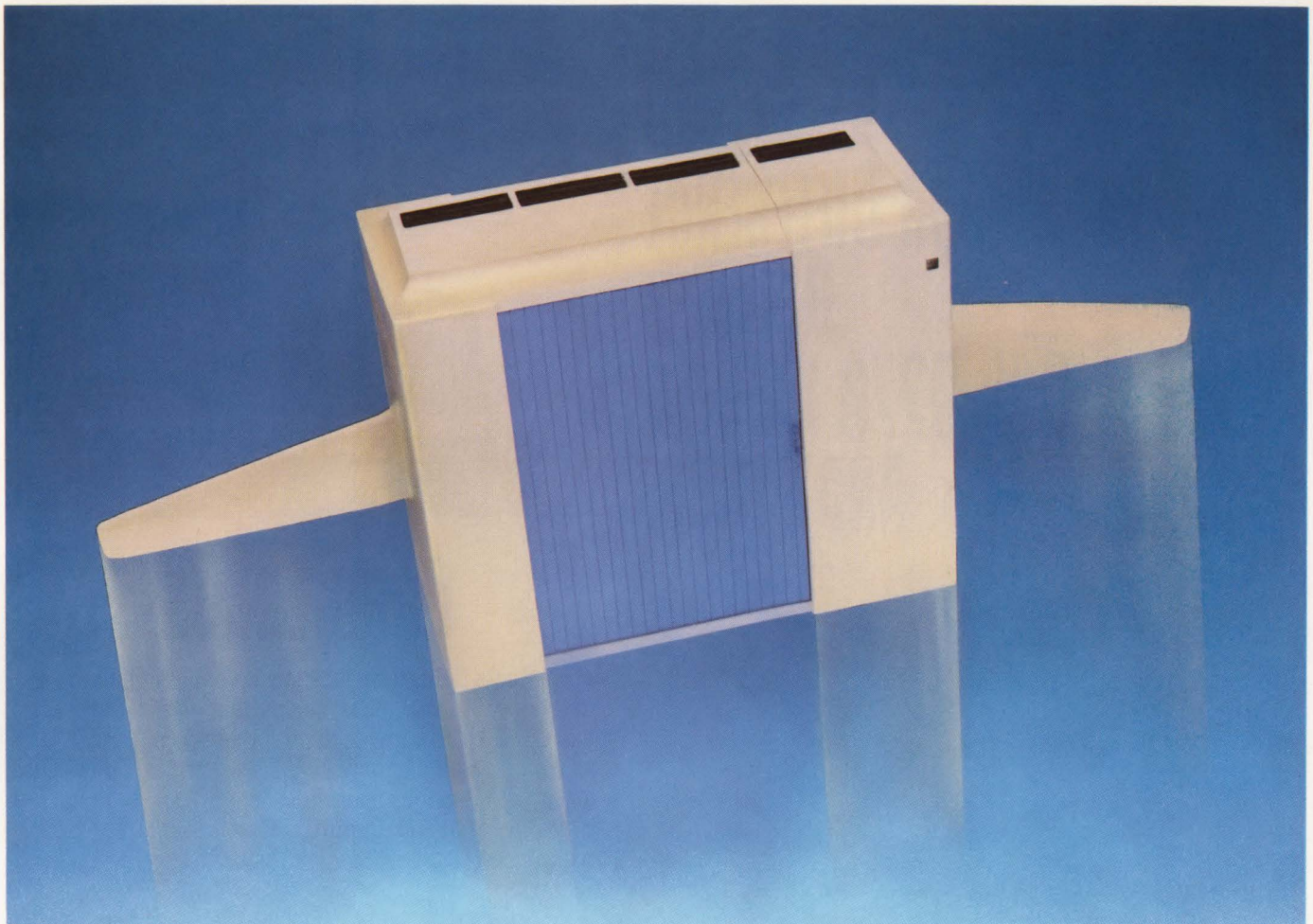
MDF's hardware dispatcher introduces another layer of switching on top of and independent from MVS and therein lies the problem. MDF's dispatcher gets control at the end of a time-slice to find another domain that is ready to run. In Amdahl's implementation, a large cache memory is provided to limit the negative impact of more frequent context-switching on cache performance. Memory segments in cache are actually tagged with the domain number of the operating system that owns them.

A key point is that MDF will not allow an interrupt to occur if the domain that handles it is not active. I/O operations are tagged so that MDF knows which operating system needs to be notified when the I/O completes. Because I/O interrupts are delayed in the channel, I/O service time tends to elongate somewhat.

Since the domain time-slice is normally about five milliseconds by itself, the amount of delay introduced by keeping the interrupt pending is not too significant for conventional DASD I/O that performs in the neighborhood of 20-30 milliseconds. Workloads that require better I/O performance using faster devices like cached controllers and dedicated solid state disks might find the additional delay time prohibitive, however.

Another serious consequence is that when the domain is finally dispatched, all the interrupts that are pending occur at once. Instead of interrupts occurring at random, they get stacked up. Due to CPENABLE they also get backed up because only a single processor in an MP is enabled for interrupt processing by the operating system. The result is the worst kind of queuing behavior possible for the arrival and servicing of I/O interrupts.

An I/O bound workload typically waits on the completion of one I/O event before it can get on to the next. Stacking up the I/O interrupts serializes the entire execution cycle of the workload and defeats most of the benefits of multiprogramming



Now, Zebb Takes Your Data Center To New Heights With Automated Rerun Management.

ALTAI AUTOMATES. With Zebb, the new standard in automated rerun management software. Whether you've never used a rerun system – or you've used a competitive product for years – Zebb can mean tremendous savings to your MVS data center in computer time and resources. Because Zebb takes rerun management out of the 1970s and into the 1990s.

Zebb will automatically recognize that a job is being rerun, and restart that job at the point of failure – without JCL changes. This feature alone gives your data center an enormous productivity boost.

Plus, Zebb prevents "NOT CATLG-2" and duplicate data set errors. Provides comprehensive job tracking. Allows convenient, full-screen interrogation. And Zebb can be evaluated without disrupting your present environment.

For a 30-day, no-obligation trial, call Altai Software today at **800/227-7774**.

ALTAI

ZEKE *The Scheduler*

ZACK *The Operator*

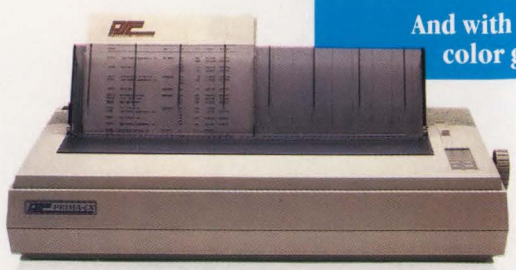
ZEBB *The Rerun Manager*

624 SIX FLAGS DRIVE ♦ SUITE 150
ARLINGTON, TEXAS 76011

CIRCLE #114 on Reader Service Card ▲

AT $1,595 THE MAJOR DIFFERENCE BETWEEN OUR 3270 PRINTER AND THEIRS.

NOW 50% FASTER!
And with optional color graphics.

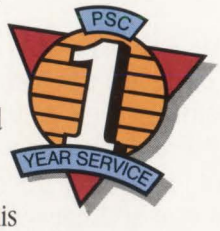


OURS EXISTS.

That's right. Take a minute and flip through this magazine. You won't find another mainframe printer that gives you as many features for anywhere close to this price. Starting at only $1,595, the Prima-CX is designed for industrial-grade applications, not PC-type printing. It attaches to IBM mainframes as well as 4300 and 9370 mid-ranges — via IBM 3174, 3274, and 3276 cluster controllers.

So whether you're producing memos or listings, there's a Prima-CX model (80- or 136-column) that's for you. The IBM-type front panel puts control at your fingertips. You can feed cut sheets without removing continuous forms. Program tear-off and RPQ options. Cancel print jobs from the printer. Take advantage of the 324 cps draft speed. You'll also enjoy the new optional GDDM and PSS color graphics. All this in the smallest footprint of any coax-compatible printer.

No one else offers a first-year, on-site service contract for only 49 cents.



They'd be afraid to. But, buy a Prima-CX professional printer and we'll give you a comprehensive, *first-year, on-site* service contract for *only* 49 cents!* We can make this incredible offer because our design ensures maximum reliability. Contact us today for complete details.

PSC

Printer Systems Corporation
9055 Comprint Court
Gaithersburg, Maryland 20877

1-800-638-4041
(1-301-258-5060, in Maryland)

IBM is a registered trademark of International Business Machines Corporation. Referenced model numbers are the products of International Business Machines Corporation.

*Subject to PSC terms and conditions.

CIRCLE #50 on Reader Service Card ▲

and multi-processing. Amdahl reports that for transaction processing workloads with a heavy I/O load running under MDF, the delay in processing interrupts can be serious.

One solution to the problem of stacking the I/O interrupts is to narrow the MDF dispatcher interval. However, this also increases MDF overhead. In a multi-processor, it is worth experimenting with lower threshold values for CPENABLE or turning it off completely. When an MVS system domain does get control, turning off CPENABLE would allow it to use all available resources to work off the backlog of interrupts that need servicing.

In IBM's PR/SM there is an option that allows interrupts to be handled immediately. Because an interrupt is serviced promptly, I/O processing does not stack up in PR/SM like it does in MDF. In PR/SM, since I/O interrupts occur immediately, I/O bound workloads should see little or no degradation. The preferred way to run PR/SM is to service interrupts immediately.

Other than the way they handle interrupts for inactive partitions, PR/SM and MDF are quite similar. There are also some differences in the way PR/SM and MDF perform due to underlying mainframe hardware differences. IBM mainframes provide a larger number of (slower) CPUs, up to six on a 3090-600, with smaller individual caches compared to Amdahl.

There are performance considerations in the PR/SM approach. The impact of PR/SM can be substantial on CPU instruction throughput and, consequently, CPU bound workloads are subject to degradation under PR/SM.

Whenever PR/SM dispatches a new logical partition, it flushes the cache. It assumes there is little likelihood of returning to the previous partition with any of its CPU cache working set resident. Each time a partition receives control under PR/SM, it is cold-started.

Overall, the amount of context-switching also increases due to PR/SM partition dispatching and interrupt processing by domains waiting to be dispatched. Since many more processes are cache cold-started, the result is a considerable reduction in effective CPU capacity.

CPENABLE can be helpful. If an I/O bound workload appears to be eating into your effective CPU capacity under PR/SM, consider using CPENABLE to slow it down. Raising the CPENABLE thresholds will help to stack the interrupts from the I/O bound workload behind a single processor in an MP, keeping the remaining processors free to handle your other workloads. This will be especially effective in an MP with more than two CPUs (3090-300s and up).

I find it interesting that after all these years it is worth fiddling with CPENABLE threshold values in OPT. CPENABLE was conceived and implemented for MVS/XA to optimize CPU cache performance in a multi-processor by providing a throttle for excessive interrupt-processing induced context-shifting. It is used to manage and configure an n-way processor dynamically for optimal balance of interrupt processing and CPU instruction execution throughput. It has seldom proved necessary to adjust the factory settings for CPENABLE in a normal MP environment.

Under PR/SM and MDF, the CPENABLE mechanism provides a useful control to limit domain interference and to improve performance. I would like to hear from any readers who are experimenting with CPENABLE in a PR/SM or MDF environment. Write me in care of MAINFRAME JOURNAL and I will share your results with other readers in a future column.

I am indebted to David Young of Amdahl and his publication, "MDF: Planning for Capacity," for a detailed performance analysis of MDF and the I/O stacking problem. The publication is available through your local Amdahl representative.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
255 (Interesting/Helpful), 256 (Too Technical), 257 (Too Basic)

2.) Would you like more articles on the same subject?
258 (Yes), 259 (No)

ABOUT THE AUTHOR

Mark Friedman is a director in the Technical Division of Landmark Systems. He is a frequent speaker on MVS performance topics at SHARE and CMG meetings. Friedman holds a Master's degree in Computer Science and has 10 years experience as an MVS performance analyst and a developer of performance analysis software tools. Landmark Systems Corp., 8000 Towers Crescent Dr., Vienna, VA 22180, (703) 893-9139.

VM/CMS XEDIT

VM/CMS XEDIT *from page 37*

before the first record. As a result, the current line is shown as 0. When I scroll the file, the value shown for LINE will change. The current column in Figure 7 is one; that agrees with the position of the current column indicator (the vertical bar) in the scale line. The last item in the file identification line, ALT, shows you how many times the file has been changed since it was last saved.

The message line

From time to time, XEDIT needs to send you messages to advise you of processing information or of errors. When it does, it uses the *message line*. By default, the message line is located in line two of the screen, immediately below the file identification line. In Figure 8, the message line is empty.

As with the other elements of the XEDIT screen, you can move the message line. And because the message line is used relatively infrequently, you can specify that it share the same line as the command line.

The status area

The last informational area of the XEDIT screen is the *status area,* located in the lower right corner of the display. In Figure 9, it shows

XEDIT 1 FILE

Most of the time, this is what the status area will contain. However, if you are editing more than one file or using an alternative XEDIT processing mode, the value of the status area will help you keep track of what you are doing.

I will discuss how to use the essential XEDIT commands in the next article.

This article is an excerpt from Eckols' book titled VM/CMS: XEDIT Commands and Features, *published by Mike Murach & Associates, 4697 W. Jacquelyn Avenue, Fresno, CA 93722, (800) 221-5528; in California (800) 221-5527.*

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
255 (Interesting/Helpful), 256 (Too Technical), 257 (Too Basic)

2.) Would you like more articles on the same subject?
258 (Yes), 259 (No)

ABOUT THE AUTHOR

Since 1980, Steve Eckols has written nine books on IBM mainframe subjects ranging from system development to VSE to IMS to VM. He is currently teaching "Computers in Education" courses at San Diego State University.

Why Prototype?

Prototyping is still the only application development technology that guarantees *user satisfaction.*

By Jon E. Pearkins

In an average week, a friend of mine spends an hour each with ten application development managers discussing the problems and challenges they face. During 1988, he spoke with 250 of these managers, most of them more than once. What they identified most frequently as their most significant problem is enlightening.

The application development manager of a large East Coast hospital perhaps sums it up the best, "We go through all the effort and expense of analysis, design and implementation, making sure we have close user involvement all through the project, only to deliver the final system to the user and have it fail acceptance testing." During the postmortem, he hears the user manager say something like, "I didn't know this system wouldn't work until my staff tried it in our day-to-day departmental environment."

The simple answer would be to let the user's staff test the final system as early as possible in the application development life cycle. *Prototyping* attempts to address that seemingly impossible challenge.

Prototyping Defined

Data processing is almost as famous for buzzwords as it is for acronyms. Each segment of our industry seems to have one buzzword that stands head and shoulders above the rest in terms of popularity. After a meteoric rise, its popularity tends to fall off quickly after six months or a year. For applications development, prototyping reached number one just over four years ago. But today, prototyping forms a standard part of the methodology or life cycle of very few installations. Why?

Whenever my friend mentions prototyping to applications development managers, almost all claim to know exactly what it is but few would agree on a definition. If they do not know what prototyping is, they can hardly be expected to use it.

The buzzword of four years ago, made famous by the books and lectures of people like James Martin and Bernard Boar, focused on the *user interface*. The goal is to develop an interactive application that looks and feels, to the user, exactly like the final system but costs at least one order of magnitude less.

Prototyping is the natural outgrowth of the successful *structured walk-through* technique. The project leader sits down with the user and walks through the document describing the user interface. Capitalizing on the adage, "A picture is worth a thousand words," most of the document is mock-ups of what the user will see on the terminal. Because there are so many possibilities and because creating them is an extremely tedious, time-consuming chore, the diagrams typically reflect less than 10 percent of the possibilities the user would see in the final system.

The problem with structured walk-throughs is the requirement for *visualization* on the part of the user. First, the user representative attending the structured walk-through is typically a manager many years removed from the day-to-day operations for which the system is being

developed. Second, visualization is a technique few people have even heard of. It is a difficult concept to understand, let alone practice. Why do you think the few athletes who have mastered it have been so successful?

Prototyping, then, can be thought of as the automation of the structured walkthrough. Given an appropriate tool, it is both possible and practical to process the user response and produce the resultant screen display for each possible scenario.

In short then, prototyping gives the user the opportunity to test, early in the design phase, a full-function working model of the user interface for the final system.

How Prototyping Works (Best)

As early as possible in the design phase, development work should begin on the user interface. Typically, preliminary design of the format of each screen can be done by the systems analyst directly with a *screen painter,* eliminating the time-consuming "draw it on paper" process. This approach assumes access to an application development system or standalone design tool that provides a complete set of prototyping functions.

Although the screen design could be reviewed at this stage by the user, a rudimentary *flow table* is normally created for each screen to define what user action will see control transferred to which screen. An *action* would be the entry of a value in one or more fields on the screen, followed by the Enter key or any PF or PA key. Hitting one of these action keys, without entering a value in any field, would also be an acceptable action.

Newer applications, especially those based on Systems Application Architecture (SAA), also use the position of the cursor when the action key is hit, to determine where to transfer control. For example, a single choice SAA menu panel must provide a one-character field just to the left of numbered choices with the cursor initially positioned beside choice one. The user can then make a choice in one of two ways:

- By typing the *number* of the choice in the field beside choice one and hitting Enter or
- By moving the cursor *beside* the desired choice and hitting Enter *without the need to key a value into the field.*

An effective prototyping tool will also provide default field edits and messages so that the analyst can safely give no thought to error conditions at this point in the process.

The prototype would then be shown to the user for initial reactions. Remember that this is the first draft of screen design, more a tool to get the user and the analyst thinking about what the on-line portion of the system should look like. As a result, expect a lot of suggestions for change from the user. Prototyping works best as an iterative process. Think of it as a user-analyst cooperative development approach, not acceptance testing; that comes later, as will be seen shortly.

After the initial user review, the analyst should endeavor to make the changes to screen design quickly and immediately review the results with the user. Remember, review meetings typically end with a consensus between the user and the analyst: each had a clear, hopefully identical, view of what the screens should look like. It is important to capitalize on this consensus by incorporating the changes in the prototype while they are still clearly in focus and fresh in both of their minds. Once there is agreement on the wording, format and appearance for each screen, it is time to prepare for *user acceptance testing*.

Throughout this process, every effort should be made to build a prototype that provides the same user interface that the user will see when the system goes into production. Unless the final system will be delivered using microcomputers as terminals, use of a PC-based prototype can only lead to misunderstandings. The 3270 is a block mode terminal while the PC normally operates on a character-by-character basis. But, more important, the keyboards are completely different.

The keys are not just in different places but many of the keys themselves are completely different. You will never find Clear or Erase EOF on a PC keyboard and keys like <-' have the same appearance but very different purposes. On the 3270, <-' is the New Line key, moving the cursor to the leftmost field on the next line of the screen that contains a field. No transmission to the computer is initiated. On the PC, the <-' is like the Enter key on the 3270, the most common way to initiate an action. Formerly called the Return key, many new PC keyboards call it the Enter key, acknowledging the PC's growing affinity with 3270-based mainframes.

Expectations can be the biggest problem, however. Try telling a user he is going to be using a lowly terminal after he expects to be using a microcomputer like the one he used during prototyping.

Edits of input field data should be fully implemented, including any customization of error messages that may be required. Any missing flow table actions should also be included along with help screens.

Test data is an obvious but often neglected area of focus for the analyst's attention. Although only a minimal amount is required, it should be carefully created to bring out the different scenarios that could exist. But it does not mean having to load and then access the test data from your DBMS. That adds an unnecessary level of complexity to the whole process; let your prototyping tool handle the test data itself.

After a preliminary review by the user's representative on the project, it is time to get the application tested by the user's departmental staff who will be using the system in the day-to-day work. If feasible, allow these individuals to do a hands-on test of the system at their desk on the same type of terminal they will be using after the system is fully implemented.

Ideally, use two people to test each function of each screen. Be sure that they are performing these functions on a regular basis as part of their job. Nothing fails worse than an acceptance test by a user for a task he is not currently doing on a day-to-day basis. The rationale behind two users is to differentiate their comments between flaws/failings in the system and opinions/differences in work habits.

And do not forget training for these people. You cannot expect users to test a system they do not understand. Training also minimizes the likelihood of compromising the power of the acceptance test by having the analyst do everything for the users "over their shoulders." The users must test the system completely themselves not with coaching or having the analyst type the appropriate key sequence whenever they are confused.

Prototyping and Productivity

John Toellner & Associates did an interesting study a few years ago. The study indicated that *changes* to an application made during *acceptance testing* cost 75 times as much as those same changes made during *requirements analysis*. Maybe that explains why so many installations have had to hide the costly redesign and retrofitting expenses that occur when an application fails acceptance testing.

Because acceptance testing is the last major step before a system goes into pro-

duction, the project budget is typically 95 percent exhausted, if not already over budget. Redesign and retrofitting magically becomes maintenance, that catch-all for more than 50 percent of the budget of installations that have large applications that were written more than 15 years ago. For other installations, redesign and retrofitting becomes an entirely new development project.

Prototyping is still the *only* application development technology that *guarantees* user satisfaction. As such, it delivers productivity in a way that is different from the traditional data processing way of thinking of it. 4GLs made their mark by getting the job of programming done faster. Instead of getting the job done faster, prototyping *eliminates* the job of redesign and retrofitting altogether. Or perhaps the job has been done faster by moving it back into the design phase. Toellner's study went a step further to conclude that the earlier in the application development life cycle you can make a change, the less it will cost you.

Whichever way you look at it, prototyping delivers big in the area of productivity. Three years ago, while 4GLs were boasting of productivity boosts over their programming language predecessors, James Martin loved to point out that programming was becoming an increasingly smaller part of the whole application development process, that is, cutting costs in half in a small portion of a project has only a minimal effect on the total cost of the entire project.

The Future of Prototyping

IBM's vision of application development in the SAA world of the mid-1990s is one where users will do the initial design of applications themselves. In IBM's view, prototyping is the most important technique they will use.

To illustrate the importance IBM attaches to prototyping, it is revealing to note the emphasis during a recent industry conference. During a one hour SAA presentation, more than 10 minutes were dedicated to describing the type of application development system IBM expects to deliver in the next three to four years. It will allow users to do much of the design and prototyping will be at its heart.

With the standardization provided by SAA, IBM will be developing a new generation of work stations. These work stations will actually execute the user interface (that is, do the screen handling) on the work station's own CPU. The application development system will ship a copy of the user interface program for the complete application to each work station at the beginning of a session for use during the session.

Choosing the Right Tool

Prototyping is only as successful as the tool you use is suitable. Because prototyping is carried out entirely in the design phase, the tool must be oriented toward the systems analyst, not the programmer. That means no jargon and no coding. For maximum analyst productivity, the tool should eliminate the need for pen and paper in the design of the user interface. Given menu-driven, fill-in-the-blanks panels with intelligent defaults, the analyst could do his design directly from his own terminal.

As was said above, the initial prototype requires productive ways to create screens and define the flow between screens based on user key sequences. Reasonable de-

faults are required for field edits and resultant error messages so that the analyst need not be involved with them initially.

Once the initial prototype has been reviewed by the user, the need to rapidly make changes exists. Compiler-based tools often deliver highly productive ways for the analyst to make these changes online. However, they involve time-consuming, multi-step batch compilations before the changes can be reviewed by the analyst, let alone the user. On the other hand, the software industry has shied away from interpreter-based application development systems because of their poor performance in a production environment with high transaction rates.

Technologies do exist that combine the benefits of both compilers and interpreters. Immediate implementation of changes and performance is possible with a table-driven approach, for example. Specifications for the prototype can be stored compactly in a table permanently resident in a DASD dataset but stored in virtual memory during execution of the application. Tightly-written, Assembler-based routines that implement the specifications provide the speed during production.

As you can see, it is important to select a tool that was designed for prototyping because the design criteria for an effective prototyping tool is not the same as the design criteria for an effective 4GL. Not fully understanding what prototyping was all about, some vendors added prototyping in name alone to their product without taking the time to provide a tool specifically designed for prototyping. Back when it was the buzzword of the month, prototyping became just another feature added to the benefits column of the brochures and sales manuals like CASE is today. Maybe that would explain why a major 4GL vendor found that only 4.5 percent of its clients used its product for prototyping.

Typically an organization will iterate through the application development process *five times* before the user gives up and accepts the system delivered by MIS. Used effectively, prototyping can help you avoid that same situation.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
260 (Interesting/Helpful), 261 (Too Technical), 262 (Too Basic)

2.) Would you like more articles on the same subject?
263 (Yes), 264 (No)

ABOUT THE AUTHOR

Jon E. Pearkins, CSP, CDP is president of Certified Software Specialists Ltd. Although his 15 years of data processing have been spent mostly in IBM mainframe technical support, he has experience in all phases of development from programmer to project leader for both applications and systems software. Pearkins also spent three years as a data processing manager, building an IBM mainframe installation in an organization with no previous computing capability. Certified Software, 54015 Range Road 212, Ardrossan, Alberta, Canada T0B 0E0, (403) 998-0607.

HOW VTAM Works

A terminal that is controlled by VTAM can access any application in the network regardless of its physical location.

By Beverly J. Weaver

It never ceases to amaze me how little is known about what other systems programmers do. All of us are so busy and involved in our own work that we find everyone else's work mysterious. Consequently, many people in systems programming as well as applications programming consider VTAM to be difficult to understand, if not pure magic.

What Is VTAM?

VTAM (Virtual Telecommunications Access Method) is a part of IBM's Systems Network Architecture (SNA); the other principal software component of SNA is the Network Control Program (NCP). VTAM runs on your host computer (3090, 3083, 4381 and so on) along with the operating system (MVS, DOS/VSE or VM) and other subsystems such as IMS, CICS, JES, POWER, RSCS and TSO. NCP runs on the communications controller (3705, 3725, 3720, 3745).

VTAM and NCP control most of the devices that use subsystems, including 3270-type display terminals and printers, distributed processors and SNA/RJE stations. NCP does most of the work to send and receive data for "remote" devices (those that are attached to the communications controller) while VTAM delivers data to "local" 3270-type devices (channel-attached to your host processor) and to programs. Because the communications controller offloads a large portion of the work for remote devices, it is often called the Front End Processor (FEP).

VTAM is the controlling point in the entire SNA network. To display devices or to make devices or programs usable in the network, issue commands to VTAM. Even the NCP and all its resources are controlled by VTAM.

A Collection of Files

Like most subsystems in a mainframe environment, VTAM consists of a set of files and a startup procedure. In MVS this procedure resides in SYS1.PROCLIB. If you would like to see just how your VTAM system is defined and how it all works together, begin by looking at the VTAM startup procedure. The member in PROCLIB has the name used when starting VTAM. Since VTAM is started after each IPL, look for the start command being issued during that time, for example:

```
S VTAM17,,,(LIST=17)
```

In this example, the member name would be VTAM17. The default name is NET for VTAM and if you have a small network, you may be using NET for the VTAM procedure name.

A typical startup procedure might contain what is shown in Figure 1.

The STEPLIB is used to read in the program which is first called (ISTINM01) and the utility programs which are used to load the network control program into storage in the communications controller or to dump the storage contents of the communications controller if necessary. If you have no STEPLIB statement, you will

find the required modules in the system library link list.

The VTAMLIB DD card in the JCL points to a file containing executable modules, including those provided when VTAM is installed. If you browse the directory of this file, you will find many modules beginning with the prefix IST. Most of these are programs provided by IBM. You may also find other modules with names like USSxxxx, MODExxx and so on. These are user-created tables for Unformatted System Services (USS) and for session parameters to be used when devices are logging on to application programs (MODETAB). More about those later.

Another DD card is used to point to the VTAMLST file that contains the source listings for definition statements and start parameters. You must define in the VTAMLST file all the resources that will be using your network and/or methods to search to find resources as needed. In addition, you will have path tables to tell VTAM what physical routes exist to send data from this VTAM to the NCP or to other VTAM systems.

Startup Parameters

There are two special types of members in your VTAMLST file. One is used to specify parameters for your VTAM system each time you start the system. The name will begin with ATCSTR and end with a suffix that is supplied when VTAM is started. As shown above, you might start your VTAM as follows:

S VTAM17,,,(LIST=17)

If you specify the "LIST" value at startup, the two characters are used as a suffix for ATCSTR to make up the member name where you have stored startup parameters. If you do not specify "LIST=," the default member name is ATCSTR00. The named member (for example, ATCSTR17) is merged with ATCSTR00.

Browsing the members in your VTAMLST file that begin with ATCSTR will show you how your VTAM is started. Inside a member you might find a value specified for the CONFIG parameters — for example, CONFIG=17. This gives you a suffix for a member in your VTAMLST file that begins with ATCCON and contains a list of member names in your VTAMLST file. These members will be automatically activated every time VTAM is started.

This value (CONFIG=17) could be overridden when VTAM is started as could all the values in the start parameter list. For example:

S VTAM17,,,(CONFIG=17,LIST=17)

If nothing is specified for the CONFIG parameter, you will be reading the ATCCON00 member and activating that list of members of your VTAMLST file when VTAM is started.

FIGURE 1

```
//VTAM17     PROC
//VTAMSTEP   EXEC   PGM=ISTINM01
//STEPLIB    DD     DSN=SYS1.VTAMLIB,DISP=SHR
//VTAMLIB    DD     DSN=SYS1.VTAMLIB,DISP=SHR
//VTAMLST    DD     DSN=SYS1.VTAMLST,DISP=SHR
//SYSPRINT   DD     SYSOUT=A
//SYSOUT     DD     SYSOUT=A
//SYSABEND   DD     SYSOUT=A
//*    THE FOLLOWING ARE NEEDED FOR CONTROLLING NCP
//NCPLOAD    DD     DSN=SYS1.NCPLOAD,DISP=SHR
//NCPDUMP    DD     DSN=SYS1.NCPDUMP,DISP=MOD
```

FIGURE 2

```
          VBUILD TYPE=LOCAL
P17A342   PU     CUADDR=342          3174 AT CUU 342
L17A000   LU     LOCADDR=2,          LOCAL ADDR FOR PORT 0          X
                 DLOGMOD=S3278M4,    DEFAULT LOGMODE ENTRY          X
                 MODETAB=MODESNA3,   IN THIS LOGMODE TABLE          X
                 PACING=0,           NO PACING FOR DISPLAY DEVICE   X
                 SSCPFM=USSSCS,      USE CHARACTER CODED LOGONS     X
                 USSTAB=USSSNA1,     FIND THEM IN THIS USS TABLE    X
                 VPACING=0           NO VPACING ON DISPLAY DEVICE
```

Major and Minor Nodes

To find out which resources defined in your VTAM have been activated, display the list of major node definitions.

D NET,MAJNODES

This display, along with all the VTAM commands, can be issued at your operating system console or at a terminal connected to an automated operator facility like NCCF or NetView.

A major node is merely a collection of minor nodes all within one member of the VTAMLST file. The major node name is the member name. A minor node is a resource definition for a line, a program, a cluster controller (3274, 3174 and so on), a terminal, a printer and so on.

To see what is defined in each major node, you can browse the member by that name in the VTAMLST file or you can issue a display command using the name of the major node.

D NET,E,ID=majornodename

When you are browsing VTAMLST members to see the resources defined in a major node, you will find on the first line inside the member VBUILD TYPE=xxx, where xxx describes the type of resource being defined. Of course, there are exceptions to every rule; the most obvious exception is the member that is the NCP major node. In this member you will find the entire source used to generate the load module that is executing in the communications controller. There should be a PCCU definition statement and a BUILD statement near the top of this member.

Defining Resources

Look at two types of major nodes that are common and easy to understand — local SNA controllers and applications.

Local SNA Controllers

You may be sitting in front of a 3270-type display terminal connected to a 3174 cluster controller that is channel-attached to the host computer. If so, your terminal and the controller are both defined to VTAM in a LOCAL major node. Inside that member in your VTAMLST file you might find the listing shown in Figure 2.

In this example, the cluster controller (PU) is connected to the operating system at the channel address 342 with a name of P17A342; your terminal (LU) is connected to the cluster controller in the first port and it is named L17A000.

Defining Programs

The major node definition that is going to be most interesting to you if you support any subsystem using VTAM to control its terminal devices is the application major node. Any program that uses VTAM to send and receive data and commands between it and another program or a terminal device is considered to be a VTAM application program.

In a complex environment, you will have several application major node members in VTAMLST. There might be definitions for JES, NJE, NetView or

Succeed

PLATINUM *technology, inc.*
The DB2 Company™

PLATINUM *technology, inc.* is the only full service software company specializing exclusively in DB2. All our effort and energy is focused on delivering the broadest array of quality solutions for the DB2 user. As a result, IBM® has designated PLATINUM a Business Partner through the Authorized Application Specialist program specifically for DB2.

PLATINUM PRODUCTS

PLATINUM offers a complete line of software tools, education, and published products to ensure your success with DB2.

Software Products

The PLATINUM product portfolio consists of a complete family of administration, development, and end user DB2 software products. All are compatible with DB2 V1.3 & V2.1. The software products include:

- **RC/Query™** - A comprehensive DB2 catalog query tool.
- **RC/Update™** - The industry's best DB2 object management and data editing tool.
- **RC/Migrator™** - A complete object and data migration tool.
- **RC/Secure™** - An extensive DB2 security management tool.
- **PLATINUM Database Analyzer™** - The DB2 database and DASD analysis, audit, and management tool.
- **PLATINUM Report Facility™** - The DB2 query and reporting system for developers and end users operating in TSO and CICS environments.

DB2 Education Courses

A complete series of hands-on DB2 training courses. PLATINUM courses cover all aspects of DB2, QMF, and CSP. All courses are available either at your facility or at our Corporate Education Center.

- Introduction to DB2
- DB2/SQL Application Programming
- DB2 Application Planning and Database Design
- DB2 Database and System Administration
- Using DB2 and QMF
- CSP Application Programming

Published Products

The most recognized and authoritative DB2 standards, methods, and guidelines for DB2 implementation.

- **PLATINUM DB2 Guide/Online™** - The industry's leading standards manual for design, development, and administration of DB2 systems.
- **The PLATINUM Reference™ for DB2** - The quick, pocket-sized reference for DB2 information.

Support

All products and services carry our PLATINUM Quality Assurance Guarantee. Support is available 24 hours a day, 7 days a week via our toll-free hot line.

WORLDWIDE AVAILABILITY

PLATINUM's products and services are available around the world through our Affiliate Network. PLATINUM's full service capabilities include local support, education, and superior DB2 professional services.

with DB2!

PLATINUM *technology, inc.*
555 WatersEdge Drive
Lombard, IL 60148-9930
(312) 620-5000 FAX (312) 953-1923

For further information, in-house demonstration, or our exclusive no-obligation product evaluation call:

1-800-442-6861

IBM Business Partner

CIRCLE #7 on Reader Service Card ▲

NCCF, CICS, IMS, NPM, NPDA, TSO and many other programs. You can issue the following command to see which applications are active on this VTAM system:

D NET,APPLS

Application Names

If you look at the major node for the VTAM program you support, you will find a VBUILD statement at the top with TYPE=APPL. After it there will be an APPL definition statement for each Access Method Control Block (ACB) used to talk to VTAM. The ACB contains information used both by the application program and by VTAM.

You will also find a VTAM application name for your program. This is the name by which the SNA network recognizes your program. It is also called the minor node name and appears as the label on the APPL definition statement. There is a second name associated with each program — the ACBNAME. The ACBNAME is used when the program opens its ACB to begin communications with VTAM. If you have not specified an ACBNAME when you define the application, the minor node name is used by default.

When you start an application program, it passes a name to VTAM in the ACB when it is opened. VTAM searches in the active major nodes for that ACBNAME. If VTAM does not find an APPL statement that is active with that name for the ACBNAME, you will get a bad return code from the open and your program should put out a message. If this happens to you, check to be sure that the major node with the definition for your program is active, that no other program is already running using that definition and that the name you are using when you start your program is spelled the same as the ACBNAME on the APPL definition statement.

The ACBNAME is either stored in the program or given when the application is started. For example, TSO uses ACBNAME=TSO for its Terminal Control Address Space (TCAS) and TSO0001 for the first user to logon, TSO0002 for the second and so on. Since you may have more than one TSO system in your entire SNA network and resources need unique names in the network, your VTAM support person may have named the application minor node name with another name, such as TSOA. Then the application definition might look like Figure 3 (with only two users shown).

FIGURE 3

```
         VBUILD TYPE=APPL
TSOA     APPL   ACBNAME=TSO,AUTH=(PASS,TSO,NVPACE),EAS=1
TSOA0001 APPL   ACBNAME=TSO0001,AUTH=(PASS,TSO,NVPACE),EAS=1
TSOA0002 APPL   ACBNAME=TSO0002,AUTH=(PASS,TSO,NVPACE),EAS=1
```

The AUTH parameter gives this program certain authorization and the EAS is the estimated number of active sessions (terminals or other VTAM programs logged on). In the case of TSO, each user gets a separate APPL and a different application name. To see this in action, logon to TSO and then display your own terminal by its name:

D NET,E,ID=L17A000

You will see that you are in session with TSOA0001, or some such name, not "TSO." Of course, even the characters "TSO" may be changed. For example, if a TSO system runs in New York City, you could call the main APPL "NYC" and all the others could be NYC0001, NYC0002 or any other one to eight-character name.

SNA and VTAM were announced to allow resources to be shared.

For a VTAM application program like CICS or IMS that uses one ACB for the sessions it has with all its terminals, there is only one APPL definition statement for the whole system. You can specify the ACBNAME (as an APPLID) when CICS is initialized. The following definition would be for a CICS called CICSTEST that would normally have 300 maximum users:

```
         VBUILD TYPE=APPL
CICSTEST APPL   AUTH=(ACQ),EAS=300
```

No two programs can run on any VTAM system using the same ACBNAME and no two applications can have the same minor node name. Resource names must be unique. In addition, the major node name (the member name in the VTAMLST file) must also be unique. You may find, for example, that all the TSO APPL statements are in a major node name using a convention such as AMTSO or @@TSO.

Session Establishment

In order for VTAM to allow data and command traffic to flow between two resources, they must both be in "active" status. This means that the component within VTAM that controls the resources, the Systems Services Control Point (SSCP), must establish a session with each resource. For example, if you activate a major node for the SNA controller defined above, VTAM starts a session between its SSCP and the Physical Unit (PU) component of the microcode running in the cluster controller. Then VTAM would begin a session between the SSCP and the Logical Unit (LU) component that resides in the cluster controller microcode for each device.

These are called the SSCP-PU and SSCP-LU sessions. When these are complete, the resources show "ACTIVE" if displayed. At that point a "stick man" should appear in the box at the bottom left of your screen and you should receive the VTAM "good morning" message. This message is described as USS message #10 in the Unformatted System Services Table (USSTAB) assigned to your terminal and may be anything from a blank screen to an elaborate full-screen display.

When you request to be connected to a VTAM application at your terminal, you will probably do so by typing a character-coded command on the screen. The command is sent to the SSCP in the VTAM that is controlling your device. The SSCP replaces your characters with a standard logon command using the USSTAB that was defined for your device. That table lists various commands you are allowed to type on your terminal. For example, typing the characters "TEST" might cause you to be connected to CICSTEST.

When VTAM sends your logon request to the application, it also includes some session parameters from an entry (DLOGMOD) in the logon mode table (MODETAB) that had been defined for your terminal device. These are rules to be followed during the time your terminal and the application are communicating. These rules include the largest data that

the device and the program each can receive, the screen size of the terminal device, which SNA commands can be sent and so on. For more information on what is in a logon mode table entry, see "The Basics of ACF/VTAM Logmode Tables" by Lloyd A. Hagemo, Jr. in the November/December 1988 issue of MAINFRAME JOURNAL.

The application can refuse to let you logon, it can accept your logon and use the parameters supplied by VTAM or it can completely change these parameters.

The application gives the session parameters back to VTAM that then sends them to the logical unit component in the cluster controller for your terminal. If the parameters are acceptable, you get logged on. If not, you will get an error message and you will not be connected to the application.

When you are logged on to an application at your terminal, the LU component that is taking care of your device will be in session with the LU component of the program you are communicating with. This is called the LU-LU session and is where most of your work is done. At this point, the box on your terminal where the stick man appeared earlier would be filled in solid.

Active LU-LU Sessions

We already saw what applications were active within the VTAM system by issuing:

D NET,APPLS

If you would like to see how many resources are in session with a specific application, you can display it by name:

D NET,ID=CICSTEST

If you extend the display, you will also get the names of all the terminals or programs that are connected to CICSTEST:

D NET,E,ID=CICSTEST

Resource Sharing

SNA and VTAM were announced to allow resources to be shared. Before VTAM, terminal devices were attached to a particular host computer and to one application on that computer. Now, with all the necessary definitions in place, a terminal that is controlled by VTAM can access any application in the network regardless of its physical location. Where you might have needed two terminals to get to TSO and CICS, now you can logoff of one and logon to the other from the same terminal device.

Just remember every time you successfully logon to one program and then to another and issue commands or transactions and receive data — it would all be impossible without VTAM.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
265 (Interesting/Helpful), 266 (Too Technical), 267 (Too Basic)

2.) Would you like more articles on the same subject?
268 (Yes), 269 (No)

ABOUT THE AUTHOR

Beverly J. Weaver is a seminar leader for Verhoef Information Packages, Inc. that specializes in training and education with emphasis on MVS, CICS and data communications. Her primary areas of focus are data communications and IMS. Verhoef Information Packages, Inc., 799 Broomfield Ave., Verona, NJ 07044, (201) 239-1005.

CICS

Temporary Stor

Temporary Storage (TS) is CICS' scratch-pad facility. It is widely used by application programs, program products and internal CICS functions to hold data for relatively short periods of time. Because the use of TS is so pervasive, its performance can affect most applications.

Several years ago I attempted to tune temporary storage processing by adjusting CI Sizes and the number of buffers and strings. I anticipated that when I increased the number of buffers or the auxiliary storage CI Size, the number of physical reads to DASD would decline. As expected, these actions did reduce the number of physical reads. Surprisingly though, this tuning also reduced the number of writes, sometimes dramatically. Before tuning, I had anticipated that the growth of TS activity would require some kind of dedicated high-speed device (such as solid-state DASD) for some of our CICS regions. After tuning, DASD activity was reduced considerably, often by an order of magnitude or more. The number of I/Os to these same datasets is now small enough that DASD performance is no longer an issue.

In this article I will discuss some of the basic internal workings of TS and how they relate to performance.

CICS Storage Performance

By Ted C. Keller

Before addressing performance, I will review basic TS structure, some of the more important control blocks and how these control blocks are used to manage TS.

CICS supports two major classes of TS data. The first, *main temporary storage* (main-TS), resides in main storage. In MVS/XA systems this data will be stored above the 16MB line. Main-TS data cannot be recoverable and is used primarily to store small data elements with relatively short life spans.

The second class is *auxiliary temporary storage* (aux-TS). Data in aux-TS nominally resides in an Entry Sequenced Dataset (ESDS) maintained by CICS using special control interval processing. Larger amounts of data may be stored in aux-TS for longer periods of time. Data in the aux-TS dataset may be retained between executions of CICS with warm or emergency restarts.

In addition to its two major classes, TS data may also be categorized as *queued* or *non-queued*. Non-queued TS allows only one data element per data ID and is supported only via macro-level programming. Queued TS supports multiple data elements per queue ID and may be created and accessed with either command- or macro-level programs.

The *TS Common Area* (TSMAP) is one of the primary TS control blocks. In addition to serving as the repository for a number of statistics, TSMAP contains pointers to other control blocks including the *AUX Control Area* (TSACA) and the *auxiliary storage byte map* (byte map). The byte map contains one byte for each *Control Interval* (CI) in the aux-TS dataset. Each byte contains a value representing the number of free segments in the CI to which it corresponds.

Each CI in the aux-TS dataset is divided into fixed-length segments. If the CI Size is 16K or smaller, each segment is 64 bytes; if it is larger than 16K, each segment is 128 bytes. Each record stored in aux-TS has a header (approximately 20 bytes). Together, the record and its header always use a whole number of segments and begin on a segment boundary. Records that are too large to fit in a single CI will be divided into sections and placed in as many CIs as necessary.

The numbers of buffers and strings allocated are major factors in TS performance.

The TSACA contains a pointer to a chain of Buffer Control Areas (BCAs). There is one BCA for each TS buffer defined to CICS. In each BCA there is a flag indicating whether the buffer is being used for output. No more than one-half of the TS buffers may be used for output at any time. This ensures that at least one-half of the buffers will be available for read activity. The TSACA also contains pointers to a chain of VSWA Control Areas (VCAs). Each VCA is associated with one TS string. The number of buffers and strings is controlled in the System Initialization Table (SIT) or startup parameters. The numbers of buffers and strings allocated are major factors in TS performance.

Outputting Auxiliary Temporary Storage

Whenever a request is received to write a TS record small enough to fit in a single CI (spanned records will be discussed later), TS routines attempt to place the record in a buffer already in memory. The routines first scan the chain of BCAs to see if any CI in any output buffer has enough free segments to hold the record. If any output buffer in the chain has sufficient space and is not locked (that is, currently being read or written to DASD by another task), the record will be placed in that buffer.

If there is not enough space in any of the non-locked output buffers, the byte map will be scanned to locate a CI for output. If less than 75 percent of the CIs are utilized (contain data), an empty CI will be selected. Otherwise, the first CI with enough free segments will be used. If space cannot be found to hold the record, TS routines will attempt to expand the file (use secondary extents — releases 1.7 and beyond only). If this cannot be done, no space is available to hold the record. Either the requesting task will wait or control will be returned with the NOSPACE condition (on a conditional request).

When the CI selected for output is not already in a buffer, TS routines will select an appropriate buffer for use. If any buffer is free (does not contain a CI), it will be selected. Otherwise, the buffer that has been least recently referenced will be chosen, subject to the limitation that no more

than half of the buffers be used for output. If necessary, the CI currently in the buffer will be written to DASD. After this, if the selected CI is not empty (contains data), it will be read into the buffer. If the CI is empty, the buffer will be formatted without reading the CI from DASD.

Since an empty CI will be selected whenever less than 75 percent of the available CIs contain data and since empty CIs can be processed without being read into the buffer, it is important to allocate enough space in the TS dataset so that at least 25 percent of the CIs are always free. This can save physical I/O operations and reduce processing delays.

Once the CI and buffer have been selected, the record to be *written* will be moved to the buffer. A flag will be set in the BCA indicating that data has been changed in the buffer. If the *Temporary Storage Table* (TST) indicates that the TS ID is recoverable, another flag will be set indicating that the buffer contains recoverable data. After this, control will be returned to the requesting task without further delay.

Notice that no attempt is made to actually write the record to auxiliary storage when a write request is issued by the application. The data is merely placed in the buffer and control is returned to the requesting task. At a later time, when the buffer needs to be written for some other reason, the data will then be written to the aux-TS dataset. This allows the chain of buffers to serve as an output caching mechanism for aux-TS.

Even with write requests for recoverable TS IDs, the buffer into which the data is placed will not be immediately written. Instead, buffers containing recoverable TS data will be written later when the same task writes additional TS data for the same ID and that data needs to be placed in a different buffer. No more than one buffer will contain non-written data for the same TS data ID. This allows efficient use of TS resources while still minimizing the amount of activity required when a syncpoint is taken.

Typically, TS buffers will be written to DASD only when the following occurs.

- *The buffer needs to be flushed* (that is, forced out to allow it to be used to hold another CI). For output, this implies that there is insufficient room in any non-locked output buffer to hold the new record.
- *Recoverable TS data is forced out.* TS routines will not allow more than one buffer per recoverable ID to be in memory without being forced out to DASD.
- *When a syncpoint is taken,* the remaining buffers containing recoverable TS for the task will be written, unless they had already been written.
- *During a warm shutdown,* all buffers containing data not yet written to DASD will be output.

Retrieving Temporary Storage

TS storage queue IDs are located using a series of control blocks called *Unit Tables* (TSUTs). Each TSUT is one page long (usually 4K) and consists of a fixed header followed by a series of *Entries* (TSUTEs), each of which describes a single TS ID. The TSUTs and TSUTEs are maintained in ID order. For non-queued IDs, the TSUTE contains either the virtual storage address of main-TS data or the DASD address (CI number and segment number) of aux-TS data. For queued TS, the TSUTE contains the address of the first of one or more *TS Group IDs* (TSGIDs). Each TSGID contains a number of entries corresponding to the TS elements for the queued TS ID. Each TSGID entry contains either the virtual storage address or the TS DASD address of one element. The SIT parameter TSMGSET defines how many TSGID entries will be allocated initially for each queued TS ID.

When a request is received to read data from TS, the logic flow is approximately as follows: First, the chain of TSUTs is scanned to locate the one containing the correct eight-byte TS ID. Then a binary search is done on the table of TSUTEs within the TSUT to locate the applicable TSUTE. The address in the TSUTE (for non-queued data) or the TSGID (for queued data) will be used to locate the TS element. If the data is in main-TS, it will be moved to the appropriate user area. If it is in aux-TS, the CI number from the TS DASD address will be used to scan the chain(s) of BCAs to determine if the CI is already located in a buffer. If it is, the data will be accessed directly from the buffer.

If the aux-TS data being accessed is not in a buffer, a buffer will be selected from the pool. If there are any empty buffers in the pool, they will be selected. Otherwise, the input buffer that has been least recently referenced will be used. If no non-locked BCA can be found, the task will wait for a buffer. Next, the new CI will be read into the buffer from DASD. Finally, the data will be moved from the buffer to the appropriate user area.

The processing associated with the deletion of TS data is relatively minimal; the processing costs to reorganize a CI are borne when the space is needed for new data.

Deleting Auxiliary Temporary Storage

When a request is issued to DELETE, RELEASE or PURGE aux-TS data, the CIs containing the data are not actually referenced. Instead, appropriate bytes in the byte map are updated adding back the amount of space previously allocated to the freed elements. The space occupied by the freed TS elements will not actually be available until it is needed. Later, when new TS records are added to the CI, they will be placed in free space at the end of the CI. When the amount of contiguous space at the end of a CI is insufficient to hold a new record, the CI will be compressed — each record in the CI will be examined and compared to the active TSUTEs. Inactive records will be squeezed out as data is shifted to the beginning of the CI. All free space will then be consolidated in one area at the end of the CI. Thus, the processing associated with the deletion of TS data is relatively minimal; the processing costs to reorganize a CI are borne when the space is actually needed for new data.

When recoverable TS data is deleted, the space will not be released until a syncpoint is taken unless the queue was created by the same task in the same unit of work. This is done to ensure the integrity of the data.

When the last element in a CI is deleted (all segments are free), the buffer containing that CI (if the CI is in memory) is returned to the chain of available BCAs.

If the CI was not in a buffer, no special action would be taken. (The fact that the byte map shows that all segments are free is sufficient to indicate the CI is empty and available for reuse.)

Thus, it is quite possible for data to be written to an aux-TS, read from that CI and eventually deleted without ever incurring any physical I/O operations.

Spanned Temporary Storage

Spanned records are aux-TS data records that are too large to fit within a single CI. When a request is received to write such a record, the record is broken into sections approximately half the size of a CI, assigned a dummy key and written as a series of records. Similarly, when a spanned record is retrieved, all of the CIs containing any part of the record will need to be accessed. Spanned records obviously generate a considerable amount of processor overhead.

The primary reason the spanned record capability was introduced was to allow the aux-TS dataset to contain a smaller CI Size but still be able to hold occasional larger records. Smaller CI Sizes reduce both transfer time and channel and path contention when blocks are transferred to or from DASD. The greatest advantage of using smaller CI Sizes is improved performance in the I/O subsystem. However, it is important that no more than about 10 percent of all TS records be larger than CI Size, since each spanned record may require several I/Os whenever it is written or retrieved.

Recommendation One — Pool Size

It is normally desirable to have as large a pool of TS buffers as practical. Larger pool sizes will tend to reduce the number of physical I/Os and reduce application delays.

The larger the pool, the higher the probability that records can be read or written without incurring physical I/Os. However, since the temporary storage buffer pool is *not* above the line, the size of the pool will affect the amount of storage available to the CICS Dynamic Storage Area (DSA).

While there is no magic number of buffers that will be optimal in all situations, I might suggest that 50K to 250K be dedicated to the pool if at all possible. *I certainly do recommend a size larger than the default of three buffers in almost every situation.* Unless you are experiencing severe virtual storage constraints, it will usually be worthwhile to have at least 10 to 20 TS buffers. In some systems, 50 to 100 or more buffers might even be appropriate. Realizing that most CICS systems running in an MVS environment are using something close to the maximum workable region size, it is somewhat axiomatic that increasing the size of the TS buffer pool will necessitate a reduction of the DSA. The task of sizing the pool may be difficult, especially in a constrained environment. Nevertheless, additional buffers can improve performance considerably in most situations. The two following examples illustrate this.

While there is no magic number of buffers that will be optimal in all situations, I suggest that 50K to 250K be dedicated to the pool if at all possible.

Example One

As an example, consider a CICS region with about 700,000 write requests, 800,000 read requests to auxiliary storage per day. Let us assume a somewhat normal mix of data lengths varying primarily from 20 bytes to 3K with a mean of about 500 bytes; only about two percent of the data will be spanned and five percent recoverable. Also, suppose that 70 percent of the activity in this region occurs in an eight-hour period. This would translate into an average of about 17 logical write requests and 19 read requests per second over the eight-hour period. Assume also that this data had an average life expectancy of about 15 seconds. TS data in this region might be used primarily for pseudo-conversational communications and would be deleted as soon as it was referenced.

In this example, about 8,500 bytes of new data would need to be stored per second. With a 4K TS CI Size, this data would fill about 2.1 CIs per second (allowing for control information). If four buffers were allocated, a maximum of two buffers could be used for output. This would mean that output buffers would hold a little less than one second's worth of data on the average. With an optimal flow of data, about two buffers would need to be written each second. However, since some data is recoverable and some spanned and since records may need to be written before they are full (for example, a 3K record is to be written and there is only 2K free in each output buffer), it is more likely that four to six buffer writes would be required per second.

Considering the randomness of write and delete activity, it would not be uncommon to see data residing in 60 to 80 CIs at any given time. With only four buffers, it would seem that the chance of finding data in the pool when needed would be rather small unless it was reaccessed almost immediately after being written. Since much of the data in this pool would not be read for at least several seconds, 80 to 90 percent of the read requests might result in buffers being read from DASD. The net result would be a total of about 550,000 to 700,000 buffer reads and writes in the eight-hour period. This could be even worse if more data was either recoverable or spanned.

If the number of buffers in this pool was increased to 40, the results should be fairly dramatic. In this case, a large percentage of the most recently referenced CIs would usually be in memory. Additionally, with enough buffers, much of the data placed in the buffers would be deleted or purged before it was ever written to the device. Chances are high that with up to 20 output buffers in the pool, there would be adequate free space in the pool for most records added to aux-TS. Output buffers should need to be written a much smaller percentage of the time, except when recoverable data is involved. With at least 20 buffers reserved for read activity, chances are quite high that data will be found in the pool when needed. Additionally, with about 20 buffers reserved for output activity, chances are excellent that data will still be in the pool 10 to 15 seconds after it is written.

It would not be uncommon to see 80 to 90 percent of all read requests satisfied directly from the pool with a similar percentage of write requests satisfied without causing buffer writes. There will proba-

See CICS Temporary Storage *page 75*

Unattended Computer Center Operation

50 Questions and Answers

By Howard W. Miller

One of the obstacles to the rapid expansion of information technology is a lack of confidence on the part of computer center users, organizational management and even computer center staff that the computer center will be available when it is needed or that response time will be adequate to get work done. These groups have been conditioned to believe this from years of unacceptable quality. What has developed is a fundamental lack of confidence on the part of key groups of people that information technology in general and computer centers specifically can meet their expectations. The solution is to automate the computer center in order to implement unattended computer center operation. The objective of unattended computer center operation is *quality,* not expense reduction.

In an attempt to overcome some of the misunderstandings about unattended computer center operation, this article answers 50 commonly asked questions. The questions are grouped under four headings: understanding unattended computer center operation, computer center staffing issues, computer center user issues and technical obstacles to unattended computer center operation.

Understanding Unattended Computer Center Operation

Q. What is unattended computer center operation?

A. A computer center is defined in the broadest sense as a computer processing center without regard to computer size or computer vendor. Unattended operation is the total automation of all data center functions, a dark room environment in which computers run without human intervention. Unattended operation requires the elimination of such traditional and seemingly essential data center operation functions as computer operators, data entry, input/output control and media distribution. Furthermore, it calls for the elimination of such relatively new functions as librarians, production coordinators and help desks. The concept of unattended operation requires looking beyond solutions to today's problems and looking at future requirements of the data center as capacity expands, availability requirements increase and on-line processing becomes the only mode of processing.

Q. Has any computer center achieved unattended operation? If not, how do you know it can be achieved?

A. To the best of my knowledge, no computer center has achieved complete unattended operation. Some dedicated test computers operate unattended; software engineers are given operator capabilities to correct problems. Some major computer centers have been isolated into separate dark rooms with equipment that does not require continuous attention. Others are operating a shift or a weekend without human intervention. However no one seems to have achieved total unattended computer center operation.

Some computer centers have made considerable progress reducing their staff by as much as 50 percent and more while improving their service levels and improving the quality of life for their users. Those that have achieved this improvement have done so by identifying and removing the obstacles. These early advocates can attest that there is no insurmountable obstacle. Some obstacles cannot be resolved with today's technology. However the receptivity to this movement is so great, it is a certainty these final obstacles will be quickly resolved.

Q. What is a dark room computer operation?

A. The *dark room* or *lights-out operation* is the process of isolating equipment that does not require intervention and moving it into a dark room or unattended environment. Such equipment includes the central processor, disk drives, communication controllers and like equipment. Large data centers have done this for a long time.

A second approach to a *dark room* is clustering noncritical computer processing into a *lights-out* period and operating unattended during that period. In this approach, noncritical processing is clustered into periods such as weekends or the graveyard shift. During this period, the computer center is then operated without staff. If the work fails, it is left until morning or Monday and it is corrected and rerun at that time.

Q. How does a dark room differ from unattended computer center operation?

A. Unattended computer center operation, on the other hand, is the broader concept of implementing tools and techniques that eliminate or reduce the dependency on human intervention. Its objective is not to black out space or time but to implement tools and techniques to improve quality and to reduce and eliminate the fault points that are the points of human intervention. As the points of human intervention are reduced, the amount of staffing is reduced and the quality improves. Unattended computer operation and a *dark room* are addressing the same issue in a different way.

Q. Why do you want to remove all the people from the computer center? Is the benefit worth all the effort?

Only one automated operations product is a perfect fit.

Service level management began with OMEGAMON®. So we precision-crafted automated operations to fit perfectly with OMEGAMON. Because automation is just another part of performance management.

AF/OPERATOR™ learns about MVS, CICS, IMS, or DB2 problems from OMEGAMON. So as soon as an exception is triggered, AF/OPERATOR can issue commands to pinpoint the problem. Then speed a message directly to your best troubleshooter for fast resolution. It can even take action itself to improve service at machine speed.

Maybe the CICS claims application is way over its service level. AF/OPERATOR can issue OMEGAMON commands to determine which resource is delaying claims. Or if other jobs are causing the delay. It can then automatically track down the user to see if the conflicting job can be cancelled.

Of course, AF/OPERATOR lightens the load of your operations staff with console message management, automatic startup of time-dependent activities, reliable IPLs, and orderly shutdowns. And it simplifies your operations by becoming the single point of control for all your systems — MVS, IMS, and JES to name a few.

For over ten years, data center managers have trusted the performance of their systems to Candle. Now you can automate your data center with the same confidence. Expect AF/OPERATOR to fit your data center perfectly — now and in the future. For more information, call Terry Forbes at **(800) 843-3970**.

!Candle®

For prerecorded information 24 hours a day, call (800) 228-0631.

CIRCLE #120 on Reader Service Card ▲

A. The initial focus of unattended operation is on expense reduction and the reaction is that staffing does not represent a large portion of the computer center operating expense. Furthermore the staffing component, as a percent of the total expense of operating a computer center, is becoming smaller and conversely the hardware and software components are becoming an increasingly larger percent.

The computer professionals that ask, "Why remove the people?" are asking an excellent question. In most computer centers, the expense of the staff component is in the range of 30 percent, while the hardware, software and supply components are in the 70 percent range. In addition to the increasing demand for computer capacity and computer center automation software, the hardware and software expense elements are increasing at a faster rate than the need for staff to operate the computer centers.

However the objective of unattended computer center operation is not expense reduction. The objective is *quality*. One of the significant obstacles to the expansion of information technology as a whole is a lack of confidence on the part of computer center users that the computer center can provide quality services. Automation of the computer center improves quality.

Q. What will happen if I do not move in the direction of unattended computer center operation?

A. This is a tough question to answer diplomatically. For computer center management, unattended computer center operation is a career decision. If the computer center does not improve quality, the computer center users will go away. Computer center users will justify departmental machines — machines that already operate unattended. Another alternative is that organizational management will recognize the need for unattended computer center operation and bring in someone who will implement it. I do not think there is a real choice.

Q. What resources are required to implement unattended computer center operation? Will additional staff be required? Will the software or hardware budgets need to be increased?

A. The resources required to implement unattended computer center operation are already in the computer center. Sell the concept to management up-front. Show them what you are seeking to achieve and get a commitment to redirect the dollars you save back into the process. As you experience turnover in the computer center, do not replace the positions. Use turnover as an opportunity to eliminate that function and at every opportunity consolidate job functions. Target to eliminate key functions and use the staffing as implementers. This is an opportunity to train and implement at the same time. Make unattended computer center operation a self-funding project.

Q. How can I expect my computer center users to react to unattended computer center operation?

A. Computer center users will be suspicious, concerned and unbelieving. They will think you are trying to offload your work onto them. The users will be concerned that they will be left to flounder

about without any assistance and they will not believe that it can be done. You might be lucky and have computer center users that have a different perspective. However to be on the safe side assume they will be skeptical and put together a communication program up-front that specifically addresses these concerns.

Q. How can I effectively approach the computer center users with the unattended computer center direction?

A. Address unattended computer center operation from the quality perspective. Remember, these computer users have experienced up to 25 years of things not working quite right, of poor response time, of computer failures and incorrect reports. Emphasize that your objective, and your *only* objective, is to improve quality and that improved quality will make their life easier.

Q. What is the single biggest obstacle to achieving unattended computer center operation?

A. The human obstacles are the most diabolical. First, most data centers have yet to recognize that unattended operation is achievable. Second, if they have recognized it they do not consider it possible. The first obstacle is easily overcome through education. The second is more difficult because it becomes a self-fulfilling prophecy. Total unattended operation may not be immediately achievable, but partial accomplishment is immediately achievable. Remember, a partial meal is better than going hungry. Movement toward the goal will result in immediate gains and it positions the organization to take advantage of new solutions to traditional obstacles.

Unattended operation can be threatening. By definition it involves reassigning staff, re-educating and the like. The nature of information technology is such that there is always a greater demand for qualified staff than is available. Attack unattended operation from a positive perspective: emphasize the desire to improve service, to return management control to end users and to develop qualified staff.

Q. Can you guarantee that unattended operation will not fail?

A. Anything that the computer center does to automate itself improves reliability, availability and service. It reduces expense. If the computer center eliminates one error, if it installs one automated tool, if it improves service by one percent point, it has not failed. The improved computer center is better than it would have been if nothing had been done. Anyone can achieve these kinds of improvements. There are, however, no guarantees in computer processing and unattended computer center operation is no exception.

Q. How is success measured? How do I effectively prove that unattended computer center operation is working?

A. Are your reliability, availability and service better? Do you have fewer people in the computer center? Does the computer center user have more control over input, processing and output? If you can answer yes to these questions today and if the answers are still yes this time next month, the month after and so on then you are successful.

Q. How can I assist in making unattended computer center operation happen? What is the single most important action I can take to assist?

A. Commit to making unattended operation happen. Teach management, computer center staff and computer users alike that it can be done. Systematically identify and remove fault points, design new application systems to accommodate this direction and put pressure on hardware and software vendors (especially IBM) to design for unattended operation. There is no magic, just daily commitment, hard work and a lot of fun.

Computer Center Staffing Issues

Q. How can you remove all of the staff from the computer center? Will there be some staff in the computer center?

A. The staff that we are discussing is all of the staff that acts as an intermediary between the computer user and the computer. We are discussing data entry, operators, help desks, tape librarians, media handlers, quality control clerks and all the associated layers of management. The objective is to stop *inspecting* quality into the computer center and to start *building* quality. The objective is to automate the computer center and to give the computer user the ability to manage its computing requirements.

Yes, there will be staff in the data center, but it will be different. There will be computer operation analysts to analyze and install new hardware and communications as the computer grows, to analyze and replace obsolete hardware and software and to manage the multi-million dollar resource.

Q. Are there any disadvantages to removing all the people from the computer center?

A. There will be the same kind of disadvantages that we had when the phone company eliminated the phone operator to place a call. For some, it will be difficult to stop working through someone else to get the job done. It is always pleasant to deal with someone nice and helpful. The direction of many industries is self-service whether it is pumping gas, an automatic teller machine, a voice recognition telephone system or a pregnancy test and the computer center is no exception.

Q. What is going to happen to me? Are you going to put me and the other computer center operation staff out on the street?

A. Under no circumstance. To be successful with unattended operation, the computer center needs to enlist the assistance of the computer center personnel and the computer center user. These are the people who understand the computer center the best. Over the last 25 years, there has been a chronic shortage of computing personnel and there seems to be no solution to this shortage in the near future. No one will throw away good people.

The valuable training that the displaced staff receives will prepare computer center staff for better positions in other areas of computing. I am prepared to make the commitment to training as will all those who are successful with unattended computer center operation. The only people who will have any problem in this scenario are those who are not prepared to

learn and grow and that is not a personality trait of computing people.

Q. We are a union shop. How do I eliminate a union position without a strike?

A. There is no doubt that unattended computer center operation is eliminating bargaining unit positions. Unattended computer center operation is committed to improved quality, improved productivity and improved quality of work life. It offers staff the opportunity to expand job knowledge, horizons and salaries.

The alternative is the same as other industries that fail to automate — no job at all. Unions need to decide whether they are there to protect the interest of their members or their own interests. Unattended operation seems to clearly offer an opportunity to those who support it. Unions can play a vital role to ensure the intent is not subverted by the desire for short-term financial gains.

Q. What do I do with a computer center employee who feels threatened and is causing other employees to be concerned?

A. Spend more time communicating. Communicate one-on-one and in groups with others who see the light. This will not be a frequent occurrence and if it is frequent, assume you are doing something incorrect. Think through your tactics and try something else. If you remain committed to unattended computer center operation, if you re-educate the computer center staff and if you are tenacious, you are going to nibble away the obstacles to unattended operation.

Q. What does the computer center do with the staff that it displaces?

A. The computer center retrains them for better positions in other areas of information technology. Make a commitment to promote all positions from within and hire from the bottom. Train the displaced staff to fill these positions. Remember there are still operation analysts in the computer center and other departments who need people such as technical services, database services, application services, the information center and computer user departments. Look around, there are all kinds of people using computers and they are all looking for staff.

Q. What steps do I take to ensure that the computer center staff is able to be placed in other positions within the organization?

A. Put them to work making unattended operations work. When they achieve their objective, they are ready for more responsible positions.

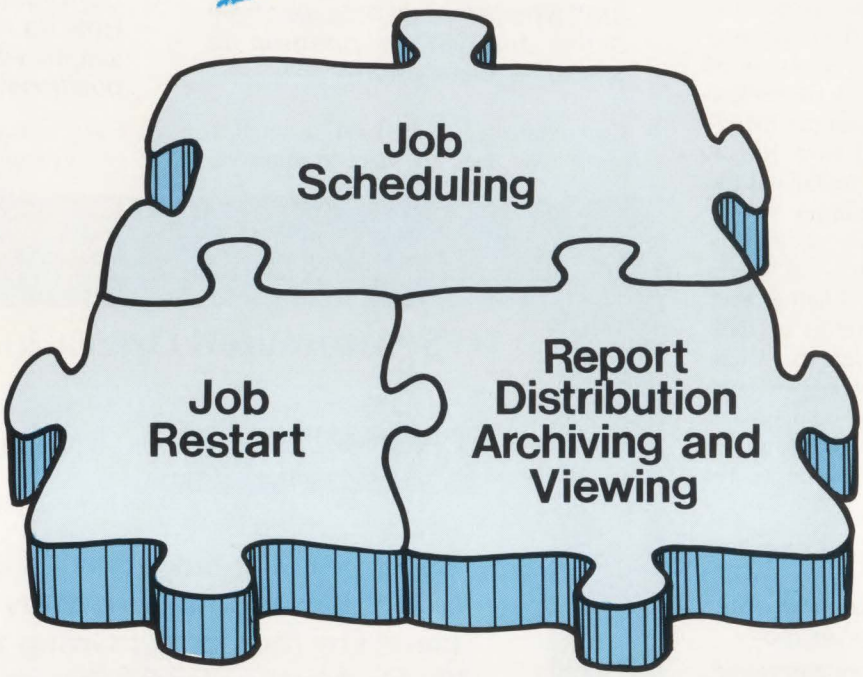
Q. What do I do with people on my staff who do not want to gain new skills, but whose position is slated for elimination?

A. Communicate. It is hard to picture someone who does not want to improve. If they do not want a better position, assist them to find something in another computer center. Ultimately, these people will have a problem that only they can solve. This will be an unusual situation.

Q. Our organization is not growing. How do I retrain people into positions when there are no new positions?

A. Hang in there. If it is just a question of not growing or growing very slowly,

Integrated Operations Architecture®
The Only Sensible Solution



To The Automated Operations Puzzle!

Meet The CONTROL Team

CONTROL-M

- *State-of-the-Art Job Scheduler*
- *NO system/JES hooks, SMF exits or SVCs*
- *2 hour Installation*
- *ISPF, ROSCOE or CICS On-Line Facility*
- *Forecasting/Simulation*
- *Automatic Date/Control Card changes*
- *Fastest schedule implementation*
- *Automatically Open/Close CICS files*
- *Can be integrated with CONTROL-D and CONTROL-R*

CONTROL-R

- *Automated JOB Restart System*
- *Eliminates manual intervention*
- *Automatic catalog/GDG adjustment*
- *Modifies JCL as required*
- *Eliminates lost time and the errors associated with reruns*
- *NO system/JES hooks, SMF exits or SVCs*
- *Integrated with CONTROL-M*

CONTROL-D

- *Automated Report Distribution, Viewing, and Archival System*
- *NO system/JES hooks, SMF exits or SVCs*
- *Easy On-Line report definition and viewing using ISPF, ROSCOE or CICS*
- *NO permanent database required*
- *True laser printer technology*
- *Printer workload balancing*
- *Can run independently or integrated with CONTROL-M*

Products Designed To Work Together

TONE® Software Corp.

1735 S. Brookhurst Street • Anaheim, California 92804 • (800) 833-8663 • (714) 991-9460 • TELEX: 4974583 • FAX: (714) 991-1831

then it may be necessary to carry some extra staff for a while. Most organizations have a minimum of five to 10 percent turnover annually. If you commit to filling positions from within, you should be able to place the displaced people. Remember you are saving monies. If you carry some people until a position opens, you are only deferring the savings. Talk to your human resources staff at the start and get its support. Do not surprise them. It is my experience that human resource and management people are supportive if given a chance.

Q. As a data center director, why should I voluntarily put myself in a compromising position by reducing my staff? I will lose my power and eventually my job.

A. If you do not, the computer center users will justify a departmental machine or machine that already operates unattended. Also, organizational management will recognize the need and bring in someone who will implement unattended computer center operation. This is not a question of choice. We are all in a footrace with professional obsolescence and if the data center director does not want to lose the race, he or she must move in the direction of unattended operation.

Q. How do you introduce the topic of unattended operation to the computer center staff?

A. Do it as a group. Go through the concepts and cover as many questions as you can anticipate. Hand out materials to go over leisurely with time to think about it. Emphasize that it is vital to the success of the computer center and the organization and ask them for suggestions. Schedule time to get back together and discuss suggestions and questions. It is also very helpful to get the computer center users involved.

Q. What can be done to minimize computer center staff morale problems while moving toward unattended computer center operation?

A. Communicate. Tell the computer center staff what you are achieving and why it is important. Encourage them to ask questions and to participate in the process. Ensure that everyone has the opportunity for an equal or better position. Use the conversion to unattended operation as an on-the-job training exercise. It gets the staff involved, provides in-service training and actually improves morale.

Computer Center User Issues

Q. What is unattended computer center operation going to do for my department?

A. Computer center users are the benefactors of this process. The early years of data processing were characterized by departments relinquishing some of their management reponsibilities to the computer centers. This was the only alternative available for realizing the benefits of the computer. The computer center assumed responsibility for data entry, computer processing, scheduling, report generation and report distribution. Unattended operation returns responsibility for the system to departmental management.

If computer center users view this process as unloading undesirable tasks on them, they will resist the process. User

involvement is essential. Emphasize productivity gains: enormous amounts of time that are expended in meetings and on memos addressing problems caused by human intervention are eliminated. Emphasize morale improvements: errors resulting in disruptions to work schedules, finger pointing and a reduction in the quality of work life are eliminated. Promote an awareness that unattended operation resolves these problems and improves morale.

Q. How does the computer center expect users to schedule work, key data and administer report distribution without additional personnel and budget?

A. The computer user is already doing these functions: developing a schedule, writing it down, hand carrrying it, following up, attending meetings to discuss why it did not work today when it has for the last twenty days and so on. The computer user is doing the same for data entry and report distribution. What is more, the computer center is doing a similar set of activities. Everything gets done at least twice. If the computer user does it directly, it takes less time than by doing it manually. It does require some changes in procedures and a little retraining.

Q. Since the computer center is going to save a lot of money by eliminating positions, why can't they transfer the funds to the user department budget?

A. The computer center is reducing the effort of the computer user. There is no budget increase for the computer user. In fact in many cases, the user's budget can be reduced. The computer user tends to be skeptical about these budget reductions. Find one prepared to make it happen and demonstrate that it does reduce effort.

Q. Why should the user take on all this new responsibility? Why should I be the one to take the blame when there is a problem, when I can now blame the computer center?

A. The computer center user is not taking on more work. Rather he is doing things in a more streamlined manner. The user is better able to service the mission. The user is being given the opportunity to demonstrate to management how quality and productivity can be improved. There is no blame here; we are discussing praise.

Q. Why eliminate the keypunch department? It can handle peak workloads and it saves everyone a lot of time.

A. Keypunch or batch data entry is an obsolete concept. It is much more effective for computer users to enter data at the source and to edit and correct it immediately. The information is available immediately. There is no transcription and there are no transcription errors. There is no lost paperwork. There are no batches that are processed incorrectly and there are no meetings to determine who shot John and so on. It is a more efficient process. It reduces data entry staff and it reduces computer user staff. Direct data entry improves quality and saves more time.

Q. Why would you want to have the on-line system available 24-hours-a-day? No one is ever around after six o'clock.

A. There is a lot of human intervention involved with managing an on-line network. Bringing the system up and down is an opportunity for error. It requires that the system be purged resulting in a period of computer under-utilization. The inability to run batch and on-line at the same time and the inability to run multiple systems concurrently are symptoms of rigidity. This lack of flexibility results in human intervention and error. It is amazing what a difference there is in a computer center that can run anything simultaneously with anything else.

Furthermore, we are moving toward self-service. Employees will want to work at home and at off-hours. Think about the implications to your computer center and you will see why you need 24-hour-a-day on-line systems.

Q. How does the nightly batch schedule get defined in an unattended computer center?

A. The computer user sets up his own schedule and the schedule is modeled to determine if there is sufficient time to complete the schedule. Changes to the daily, weekly or monthly schedule are not frequent and conflicts are less frequent. When conflicts occur, they will be with systems that cannot be run concurrently with other software systems or on-line processing. Other conflicts arise from ad hoc processing. If the computer user is given the ability to see his schedule modeled, he will make the necessary compromises.

Q. Who commits to the workload requests of the computer center user in an unattended computer center?

A. The information technology group, as part of its capacity planning group, ensures that there are sufficient resources to meet the needs of the organization. This does not change. Automated guidelines and procedures are established for handling ad hoc and abnormal processing. Under normal circumstances exceptions are rare and information technology management is still available to mediate.

Q. Who informs the computer users of processing problems?

A. The computer user can inquire about the status of his processing on-line via the scheduling system.

Q. Who assists the computer user in an unattended computer center?

A. The computer center operation analyst who is addressing the source of the remaining problems and applying a permanent correction is available for assistance. This person may be a part of the computer center or of some other group.

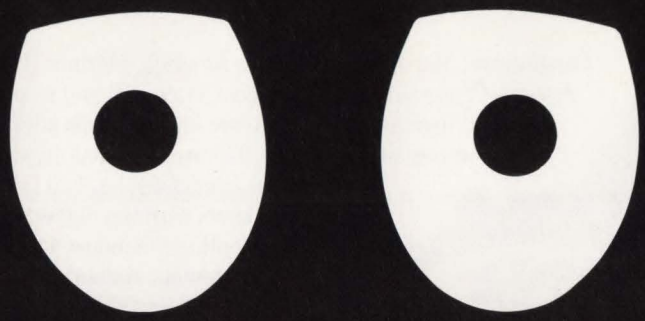
Q. The computer center staff inspects work requests and finished products for errors before they reach the computer user. With no one in the computer center, will this lack of inspection result in problems and worse service?

A. Many errors are created by transcribing data and passing documents to other staff. As the computer user assumes control of such functions, the need for transcription, handling and inspection goes away. The direct input of data requires an intelligent data-entry feature: data is edited and corrected on-line. Software is put in place to perform edits and to balance output. The whole objective of unattended computer operation is the identification and permanent correction of errors. Unattended operation results in improved quality, service, reliability and performance.

Q. What can be done to minimize computer center user morale problems while moving toward unattended computer center operation?

A. Communicate. Tell the computer center user what you are achieving and why it is important. Encourage him to ask

See Unattended Operation *page 88*



This is one place you'll never be when you buy our software.

When you choose any DTA software product there are many things you'll get. Like very effective, easy to work with, low-frustration products.

There is, however, one thing you won't get. You won't get left in the dark. That's because we support our software products with clear, exacting, thorough documentation. Documentation that many consider the best in the industry.

And there's more. You also get the DTA Straight Line, a direct customer service number that puts you person to person with DTA's Software Product Developers. So in the unlikely event that you hit a snag, you can get an answer to your dilemma straight from the horse's mouth.

Our software product developers work day and night fine tuning our products. Every detail is painstakingly covered.

And all new products are tested under tough, honest, fool-proof conditions at client sites.

So if you're looking for some select software products to improve your VSE, MVS or VM systems, give DTA a look. We're opening a lot of eyes with our attention to detail and support.

DTA Software...More efficient in every detail.

DTA®

Software Products Group
550 Waterford Park
505 N. County Road 18
Minneapolis, MN 55441
800-521-6773
612-591-6100

CIRCLE #129 on Reader Service Card ▲

The World's Most Successful Software Companies Are Sold On Software Management.

Compuware, Management Science America, Morino, Policy Management Systems and Systems Center are five of the most successful and respected names in the software industry. Together, their customer bases include almost every IBM site around the world—representing over 40,000 installed mainframe products.

These companies have all taken yet another step to provide their customers with the highest quality software. They are implementing the industry's only automated software management system—ENDEVOR.®

Using ENDEVOR, these vendors are able to manage software development and distribution within a unified environment. ENDEVOR automatically tracks every component of a software system from the time it's designed through testing and release. The results—shorter development cycles, fully reliable systems, on-time delivery to users and a more sophisticated approach to customer support.

Now you can bring all the benefits of the ENDEVOR Software Management System to your organization. With ENDEVOR, you can manage the creation, evolution, distribution and operation of your vendor-supplied and internally developed software—automatically. Whether it's traditional, 4GL, CASE or DBMS based.

The ENDEVOR Software Management System provides:

- Inventory and library management for classifying and tracking software components
- Change control that produces audit trails, source-to-executable links, and ensures standards enforcement
- Configuration management that automatically captures application interdependencies for point-in-time recreation and change impact analysis
- Release management that automatically controls the distribution of applications from test to production and to remote sites

If you're concerned about automating the management of your organization's application systems and are interested in improving the quality of production turnovers, software distribution, vendor application updates, software documentation, application stability, change implementation or change impact analysis, follow these software industry leaders.

Call us at (508) 870-1900 and ask us how ENDEVOR has already worked wonders at over 200 top installations like yours, and at the top software firms in the world.

BST

The Software Management Experts

Business Software Technology, Inc., Westboro Executive Park, 114 Turnpike Road, Westborough, MA 01581-9990
ENDEVOR is a registered trademark of Business Software Technology, Inc.

Guidelines for selecting productivity tools for development and maintenance

By Marc Fey

If you were to portion out the time programmers spend on various development tasks, you would likely find the greatest portion of the pie belongs to *testing* — some 30 to 40 percent by today's average among Fortune 1000 companies.

If you were to estimate the amount of time your organization now spends on maintenance, the figure of 50 percent would probably be close to the sum total of time devoted to program corrections and enhancements.

Those percentages alone should give DP managers a reason to pause and think: it is time to aim productivity tools at the costliest areas of the applications life cycle — namely testing and maintenance.

Ironically, given the amount of time that testing and maintenance consume, productivity tools for these areas are among the most *misunderstood*. As a result, DP organizations are still finding it hard to make headway in an application backlog that now averages 18 months in large organizations.

Misunderstandings about productivity programs

Several myths about productivity tools persist. The most troublesome ones are the following:

- **Myth #1:** Somewhere there exists one productivity tool that is all things to all programmers
- **Myth #2:** In-house developed productivity tools are best because they are custom tailored to the environment
- **Myth #3:** The chief advantage of productivity tools is that they automate what you now do manually.

Now for the facts.

- **Fact #1:** There is no single, magical tool that can do everything in testing and maintenance. Both testing and maintenance involve numerous tasks — creating and changing test data, running tests, identifying and correcting errors and evaluating results. In fact, there are four categories of productivity tools that address these four task areas.
- ***Data file utilities*** that allow you to allocate files, create test databases, edit test data, identify and fix bad data, validate, view and print output as well as perform other data intensive tasks inherent in both testing and maintenance.
- ***Data comparators*** that let you compare the differences between two files and thereby verify differences that should or should not be there.
- ***Program debuggers*** that help you locate and correct individual program errors by allowing you to watch the flow of your program as it is executing.
- ***Test coverage analyzers*** that provide critical test quality information such as what has and has not been tested and the percentage of code tested.
- **Fact #2:** In-house developed programs are not always conducive to greater productivity throughout your organization. They typically are tailored not to the entire organization, but to an ad hoc situation and a specific programmer. Productivity tools, on the other hand, should promote and support a consistent and methodical process. Ad hoc tools undermine that effort.
- **Fact #3:** The focus on productivity tools should not only speed up the process but also should allow you to improve development and maintenance in *strategic* ways.

Look at productivity tools this way: they hold the potential for improving program quality and letting programmers see what they could not see before (that is, errors that are both hard to detect and correct). In other words, the right combination of tools can provide two major advantages with respect to productivity.

- ***Improve operational productivity*** by replacing tedious and error-prone tasks with faster and more accurate functions.
- ***Create strategic productivity*** by reducing the type of errors that are not caught until the application reaches production or corrective maintenance.

Thus by improving *specific* tasks, productivity tools can have larger impact in development, maintenance *and* production. With those facts in mind, what should DP organizations consider to select the right combination of productivity tools for testing and maintenance?

In reality, all four categories of tools

mentioned are essential to the health of an application — much as a balanced diet is essential to one's health. However, DP organizations can and should prioritize their productivity needs and select tools in the order that will bring the greatest payback.

At the same time, tools should not be selected on a piecemeal basis. DP managers and product evaluators need to consider how the product will be integrated with existing and future products.

The following 12 guidelines are minimum requirements for selecting a tool that enhances productivity, program quality and programmer satisfaction. The guidelines include sample questions you might ask yourself.

■ Point #1: *Identify problematic tasks*

Productivity comes from people. When investigating a software package, ask the vendor what major tasks the product handles. Then ask your programmers how often they perform those tasks and how problematic they are now. The exercise is worth the effort. By looking into several tools this way, you also have a better idea about where your productivity drains are located.

■ Point #2: *Determine software compatibility*

First, this point requires further investigation within your own organization. Ask yourself a few questions. Which file structure do I use most often? In terms of application backlog, which could benefit the most by having productivity tool functions integrated with my current software environment? Next, consider compatibility with respect to installation requirements. Finally, consider the degree to which it is compatible with other software such as your on-line editor.

■ Point #3: *Ascertain people compatibility*

Productivity gain is the outcome of two multiples: functionality and usage. You can buy a product that does wonderful things. However if few programmers use it, your cost justification flies out the window. There are key factors that affect programmers' acceptance. If it is difficult to learn, programmers will likely conclude it is difficult to use. On the other hand, some productivity tools actually make it easier to bring new programmers on board, because difficult tasks become simple procedures. In other words, some of those tasks that required the special expertise of a systems analyst may now be performed by all levels of the programming staff, if desired.

■ Point #4: *Determine how much function for how much cost*

If you research what is available, you should find at least several products that purport to do the same thing. By asking the vendor how his product differs from that of vendor A or B, you will probably hear answers relating to one or more of the following: cost, type of features and depth of function. Naturally, you do not want to pay for features and functions you do not need. On the other hand, a lower cost product that does not address the full range of programmer needs in an important area could be a losing proposition.

■ Point #5: *Compare compatibility with your development process*

A powerful productivity tool enhances your development and maintenance proc-

POWER TOOLS.

FISCHER INTERNATIONAL

IOF is "the troubleshooter" for JES2 batch processing users under TSO and CICS. IOF Release 6.0 teams IOF's standard functions (such as quick access to the JES2 spool, the ability to modify both input and output job characteristics, and the ability to view JES2 batch job output to see whether the job worked), with a premier offering of new features.

IOF's major new enhancements include Sorting Features that allow users to sort the IOF Job List (or Output Group Display) based on any field on the menu. Release 6.0 also features the Exclude Function, which allows users to selectively exclude jobs (or Output Groups) on the IOF Job List (or Output Group Display) based on the job characteristics.

ACTIVATE THE POWER!

IOF™

(Interactive Output Facility)

RELEASE 6.0

In addition, utilities such as the Dump JES2 Control Blocks option let users browse through JES2 control blocks for a job quickly and completely. The new CLIST Interface option lets you "program" IOF to analyze your job queue. For example, CLISTs can be built to automatically analyze the results of production jobstreams.

IOF Release 6.0 also includes new options that provide easy maintenance and streamlined installation.

TSO-Rx offers TSO users immediate benefits—improved TSO response time, increased functionality, and better support. And, administrators will appreciate how easily TSO-Rx centralizes control over TSO functions, facilitates maintenance, and allows easy introduction of upgrades.

TSO-Rx reduces the demand on critical system resources such as catalogs, VTOCS, and frequently-used PDS directories, for more consistent response time throughout the day.

TSO-Rx has window-controlled multitasking to let users run CLISTs or invoke commands concurrently with normal ISPF functions. Command completion and maximum return are displayed through a window on the terminal screen.

TSO Rx™

THE POWER IS DYNAMIC!

The Help Desk utility, accessed with a simple two-word command, allows support for non-technical users.

TSO-Rx centralizes control over user attributes and resources. Each user can be given the access he needs without endlessly modifying logon PROCS or CLISTs.

TSO-Rx frees technicians from the time-consuming logistics of software installation and maintenance, thus allowing time to attend to more important functions.

Contact your sales representative for a complete introduction to IOF Release 6.0 and TSO-Rx.
Call toll-free: 800-237-4510. In Florida, call (813) 643-1500.

FISCHER INTERNATIONAL SYSTEMS CORPORATION

4073 Merchantile Avenue Naples, Florida 33942

CIRCLE #115 on Reader Service Card ▲

ess. It ensures that critical tasks are performed, promotes consistent standards and clarifies communication between managers and programmers. With both development and maintenance, you might ask yourself these questions. Does the software help me document the quality of the program? Does it give programmers a better understanding of the program and in the case of maintenance, what is being changed?

■ Point #6: *Look into training requirements and support*

Few organizations can afford training downtime, yet training is critical to ensure that the product is used. You should assess how much training is required to get up to speed on the product. An added plus is vendor-provided training that is designed to get your people using the product the same day. At the very least, there should be sufficient documentation to enable your staff to do its own training.

■ Point #7: *Evaluate the organization of the documentation*

Software documentation may not make for exciting reading, but the information should be well organized. It should lead the user to the right answer and it should be thorough and accurate. Think about the information you will need if you are fixing a crash in the middle of the night and you have never used the product's fix-it-yourself function before.

■ Point #8: *Assess the level of vendor support*

After reading the fine print of the contract, ask the vendor a few questions on the long-term use of the product. What is the level of that support? Will the vendor continue to work on future enhancements? Is the vendor willing to answer both product usage and technical questions?

■ Point #9: *Research customer satisfaction*

By looking at the vendor's track record in customer satisfaction, you get an in-depth look at the vendor's style of doing business. You will want to confirm that the product's benefits will meet actual DP organizational needs. Ask the vendor to provide reference letters from present customers. Read independent product

evaluations or obtain names of present customers who are willing to discuss the product.

■ Point #10: *Ask for a product trial*

Take advantage of a trial period, especially if it is free. This is the only way to see if the product lives up to your expectations in terms of functions, features, usage and benefits. The vendor can help you establish trial measurements and evaluation parameters. Your organization should provide specific objectives and additional cost justification criteria.

■ Point #11: *Do a cost justification*

Many products will show obvious and substantial payback almost immediately. However do not overlook strategic paybacks over the long-term. A product that saves you two hours per programmer per month may not seem worth the cost. But if that same product uncovers 50 percent more errors in less time, your maintenance and production costs are bound to plummet dramatically. You should develop a cost justification algorithm before a product trial or ask the vendor to provide statistics on time savings developed by other users of the product.

■ Point #12: *Confirm your commitment*

Selecting the right product and putting it into use can be a time-consuming process. Therefore, you should look at this checklist (or your own) and determine your organization's willingness to invest the required time to obtain internal agreement on product requirements from everyone including those who will use the product. The DP community already abounds with too many horror stories of costly products relegated to the role of dust catchers all because of lack of commitment and insufficient planning.

Testing and maintenance deserve a closer look. Improvement in those two areas alone could probably cut application backlogs in half. Given the range of productivity tools available, testing and maintenance are areas in which you can see improvement immediately.

It is a matter of choosing the right tools and using the right selection criteria. The bottom line boils down to two things. From an operational point of view, the productivity tools should help you control development and maintenance costs now. From a strategic point of view, they should help you build better quality software to reduce long-term maintenance and production support problems and thus, pay dividends for years to come.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

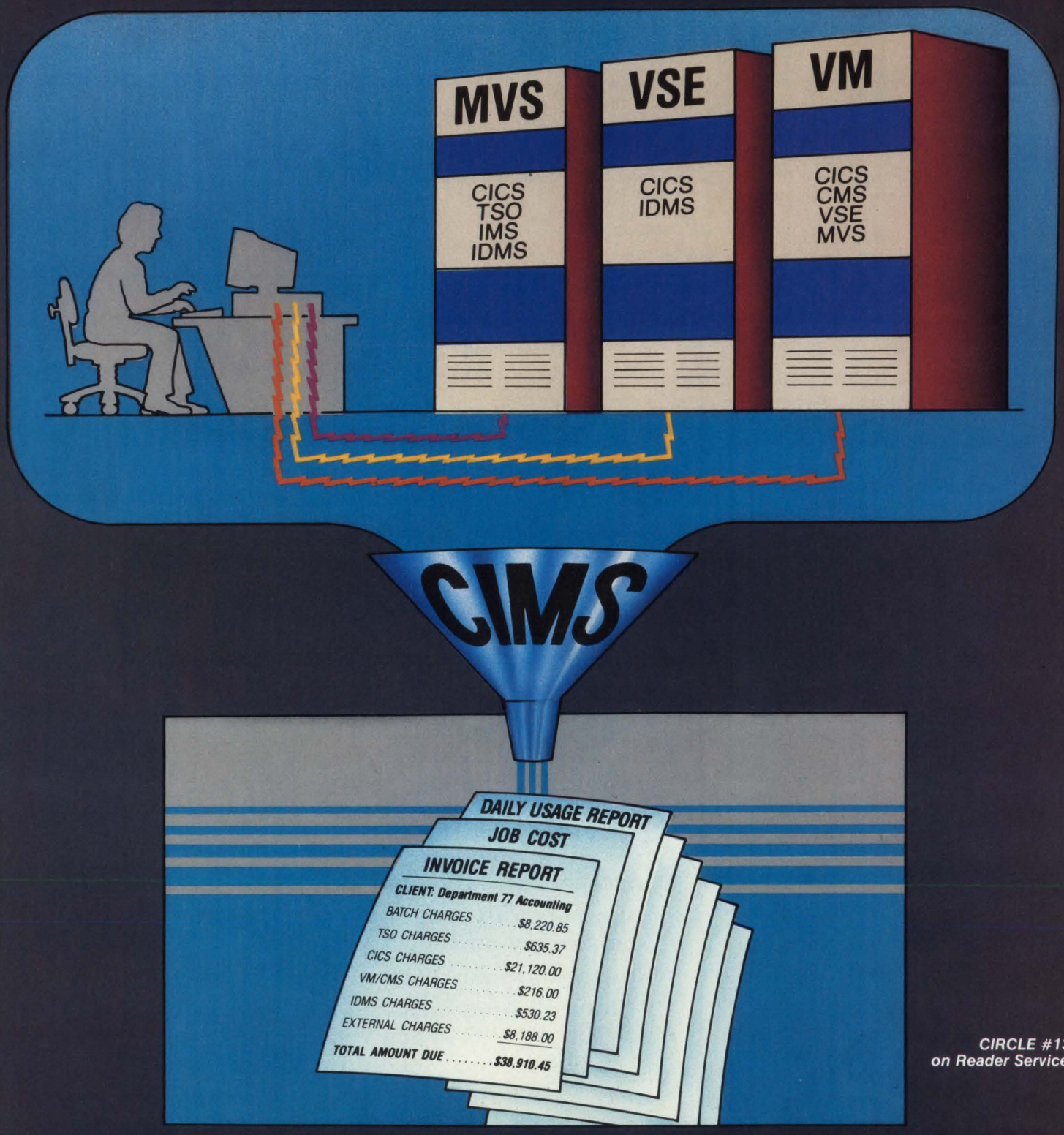
1.) This article was:
280 (Interesting/Helpful), 281 (Too Technical), 282 (Too Basic)

2.) Would you like more articles on the same subject?
283 (Yes), 284 (No)

ABOUT THE AUTHOR

Marc Fey is chairman of XA Systems, a provider of development and maintenance productivity tools for IBM MVS and MVS/XA environments. Fey often serves as an industry spokesperson on the subjects of software testing and maintenance. XA Systems, 983 University Ave., Bldg. D, Los Gatos, CA 95030, (408) 395-1800.

CIMS — *The Integrated Chargeback System*



CIRCLE #138
on Reader Service Card

Job Accounting • Chargeback • Cost Analysis
Resource Utilization Reporting

Call for additional information or a demo.

BMS Computer Inc. • ***Quality Products Since 1974***
375 North Wiget Lane, Suite 210 • Walnut Creek, CA 94598 • Tel. (415) 938-2620 • FAX (415) 946-0289 • Telex 469516

CICS Temporary Storage *from page 57*

bly be less than 150,000 total physical I/Os in the eight-hour period. In this example, a 144K increase in pool size (36 more buffers at 4K each) could yield a savings of about 500,000 physical I/Os over an eight-hour period (or about 17 I/Os per second to the aux-TS dataset).

Example Two

In this example, aux-TS is used primarily for internal purposes such as communicating with initiated tasks. The average size and level of activity will remain the same as Example One, but the life expectancy will be reduced to three seconds. In this example, eight buffers might be able to hold most of the more commonly needed data much of the time. With various spikes and valleys in activity, the percentage of logical to physical I/O requests might be quite reasonable with as few as eight buffers. With 15 to 20 buffers, the vast majority of all I/O requests should be eliminated (unless much of the data is defined as recoverable).

While the exact numbers shown in these two examples are fictitious, they are representative of real world experiences I have encountered. Pool sizes of 10 to 50 buffers have provided significant performance in many widely varying situations.

Another factor to consider when determining the number of TS buffers is the number of tasks that had to wait for buffers. A task will wait for buffers when the buffer it needs is locked (I/O is being done for another task) or all buffers are locked (waiting for the completion of I/O). In either case, the requesting task must wait for the completion of one or more I/O events. If practical, the number of buffers should be large enough to minimize the number of buffer waits.

Recommendation Two — Auxiliary Storage CI Size

As mentioned above, smaller CI Sizes are preferred if possible. Smaller sizes allow better performance in the DASD subsystem. A CI Size of 4K or 6K would be appropriate on most 3380-class devices. However, there are some situations in which larger CIs would be preferable. First, if many spanned records are written, it may be better to increase CI Size and reduce the number of buffers since every spanned record will occupy part of at least three CIs. In a system without enough buffers to materially reduce the number of buffer writes and reads, a large percentage of spanned records can force quite a bit of additional I/O activity. It is usually better to do one longer I/O than several shorter ones. It is best to use a CI Size large enough to keep the number of spanned records to a minimum.

Recommendation Three — Number of Segments

When using queued TS, each ID can have one or more elements associated with it. Whenever a new queued ID is written, a TSGID will be allocated containing the number of entries specified in TSMGSET. There is a trade-off between the virtual storage used by TSGID entries and the processing required to maintain multiple TSGIDs for a single queue ID. Generally speaking, the default value of four is

If you have article ideas, comments or suggestions concerning MAINFRAME JOURNAL, write or call: Bob Thomas, editor-in-chief, MAINFRAME JOURNAL, PO Box 551628, Dallas, TX 75355-1628, (214) 343-3717.

If...

you spend a lot of time tuning VSAM

Then...

you will find this is the best five hours you've ever invested.

Announcing Goal Systems' Spring VSAM Class Series

This class concentrates on VSAM (including DL/I and IMS VSAM) files, dataspace allocation and monitoring techniques which will assist in optimizing overall system performance. Techniques covered allow accurate tracking and effective tuning of VSAM files. This is a "how to tune" class designed for the practitioner. A brief Goal Systems' VSAM product line overview will follow the instruction.

San Diego, CA • Tuesday, April 4 • 8-12:30	Detroit, MI • Tuesday, May 16 • 12:30-5:30
St. Louis, MO • Wednesday, April 5 • 12:30-5:30	Rochester, NY • Wednesday, May 17 • 12:30-5:30
Ft. Lauderdale, FL • Thursday, April 6 • 12:30-5:30	Pittsburgh, PA • Thursday, May 18 • 12:30-5:30
Dallas, TX • Tuesday, April 18 • 8-12:30	Houston, TX • Tuesday, May 23 • 8-12:30
Cincinnati, OH • Wednesday, April 19 • 12:30-5:30	Hartford, CT • Wednesday, May 24 • 12:30-5:30
Charlotte, NC • Thursday, April 20 • 12:30-5:30	Cleveland, OH • Thursday, May 25 • 12:30-5:30
Phoenix, AZ • Tuesday, May 2 • 8-12:30	Anaheim, CA • Tuesday, June 6 • 8-12:30
Baltimore, MD • Wednesday, May 3 • 12:30-5:30	Denver, CO • Wednesday, June 7 • 12:30-5:30
Philadelphia, PA • Thursday, May 4 • 12:30-5:30	Milwaukee, WI • Thursday, June 8 • 12:30-5:30

(800) 537-4891 East/West • (800) 848-4640 Central
Classes are provided at no charge

Goal Systems

Goal Systems International • 7965 N. High Street • Columbus, Ohio 43235 • 800-848-4640
Goal Systems International S.A.R.L. • 88 avenue de Wagram • 75017 Paris, France • Phone: (1)42 67 55 55 • Telex: 641.094

CIRCLE #79 on Reader Service Card ▲

probably sufficient unless a significant percentage (perhaps 15 to 20 percent) of the queue IDs created require multiple TSGIDs to be created.

Recommendation Four — Number of Strings

A string is required whenever a buffer needs to be written or read to or from auxiliary storage. Each string provides a vehicle by which CICS can request I/Os to the access method and I/O subsystem. When a buffer needs to be written or read, a string will be allocated for the buffer. If no strings are available, the task will wait.

Since most aux-TS files are defined on a single volume, only one I/O can actually take place at a time. Any I/Os associated with additional strings will wait in the I/O subsystem queued for the device. Unless the CICS region is CPU constrained or the volume upon which the aux-TS dataset resides is heavily used by other files, there is little benefit associated with having more than three or four strings. The task will still need to wait for previously scheduled TS I/Os to complete. It should not make much difference whether it waits for strings in CICS or is queued for the device. Adding strings can be beneficial if there is a lot of competition for the device. In this case, the additional strings will allow the I/O requests to be scheduled sooner by the access method and be positioned earlier in the queue for the device.

If CICS is CPU constrained (that is, CPU demand is 60 percent or higher for the region — *CPU demand* is the sum of the times CICS is using or attempting to use the CPU), it may also be desirable to allocate additional TS strings. A CPU constrained region may not get around to recognizing the completion of external events in a timely manner and additional delays may be incurred by tasks waiting for strings.

It is possible to allocate the auxiliary storage dataset on multiple volumes to split up I/O activity. However, since the CIs at the beginning of the dataset will receive most of the activity, it may be difficult to make good use of additional extents. When a multiple volume aux-TS dataset is being used and I/O activity is occurring on both volumes, allocating additional strings may be beneficial. Of course, the primary reason for splitting the file is to reduce I/O activity on the volume. Normally this can be better achieved by increasing the number of TS buffers.

Recommendation Five — Virtual Storage Considerations

CICS uses a number of internal control blocks to manage its various functions. Most of these consume a relatively minor amount of virtual storage. There are really only two factors that can have a noticeable impact on virtual storage. The first that we discussed earlier is the number and size of buffers. If virtual storage constraint is a problem, one of the primary symptoms will be high I/O activity on the program library dataset (DFHRPL). Increasing the number of TS buffers will decrease the size of the DSA and may cause an increase in activity to the program library. The trade-off may be between I/Os to the program library and to the TS auxiliary storage dataset.

All IBM needs to do to bring TS processing into the 1990s is allow the TS buffer pool to move above the 16MB line and that should solve most TS-related performance problems.

A second area that may use a large amount of storage is the byte map. One byte is allocated for each CI up to a maximum of 32,767. If the dataset is significantly overallocated, the byte map may be larger than need be. However, it should be large enough that at least 25 percent of the CIs are free most of the time.

Recommendation Six — The Use of Main Temporary Storage

With MVS/XA, main-TS is stored in memory above the 16MB line. This means that an MVS GETMAIN must be issued each time a record is stored and a FREEMAIN done when the data is released. This would seem to be more expensive (CPU-wise) than the logic required to store data in a tuned aux-TS system. However, for data that is created once and referenced many times (such as a common table or reference data), main-TS is considerably more efficient than aux-TS. Even if the aux-TS pool size is large enough to eliminate most physical I/Os, the BCA chain must be scanned each time data is accessed to locate the CI containing the data. When the data is in main-TS, the data is accessed directly and the scanning is eliminated. Data that will be frequently referenced may be better stored in main-TS.

In the days before MVS/XA and Release 1.6.1 of CICS, strong restrictions were generally applied to the use of main-TS. In those days, main-TS was stored in the shared subpool of the DSA and was a major concern in storage constrained systems. Today, with storage above the line, there are many more opportunities to exploit this facility. The main issue is the amount of real storage that will be needed to map this data. Within reason, there are many uses of main-TS that might be justified. Remember, though, that main-TS data cannot be recoverable and cannot be retained between executions of CICS.

Of course, the trade-off between the use of aux-TS and main-TS is performance. If the aux-TS system is adequately tuned, the percentage of requests affected by physical I/O events can be relatively small. If main-TS is heavily exploited, the overall CICS working set size will increase and possibly cause additional paging. Unless page-faults are being serviced from expanded storage, paging is perhaps one of the most negative performance factors a CICS system can face. Therefore, the choice of whether to use main or aux-TS may be influenced by the amount of virtual storage that can be dedicated to the TS buffer pool and how much memory is available to satisfy an expanded CICS working set.

Conclusions

Several times over the past few years I have heard that IBM's plans are to establish a CICS architecture that will support systems running 500 or more transactions per second. The structure of aux-TS is typically cited as one of the limitations to this level of activity. If you look at the IBM performance guide for CICS, the recommendation is to use the minimum number of buffers for the aux-TS buffer pool to minimize the impacts of virtual storage constraint. With this philosophy,

temporary storage performance is doomed in high activity systems.

My experience has been that increasing the aux-TS pool will improve performance, often dramatically. If sufficient virtual storage exists almost all I/O activity can be eliminated except that attributed to recoverable data. It is my opinion that all IBM needs to do to bring TS processing into the 1990s is allow the TS buffer pool to move above the 16MB line. This should solve most TS-related performance problems for a long time to come. (The only reason it is still below the line is that DFP currently does not support CI processing above the line and CI processing is necessary for processing efficiency.) Most of the remainder of TS processing appears to be relatively efficient and capable of withstanding significantly higher levels of activity without stress.

The TS program was rewritten in Release 1.6.1. After that, performance was significantly upgraded via a PTF to 1.6.1 that was made standard in Releases 1.7 and 2.1. This article has addressed TS storage for Releases 1.6.1 (with the performance upgrade), 1.7 and 2.1. Anyone running older releases of CICS or Release 1.6.1 with an old maintenance level will not receive performance as good as that cited here and may not be able to take advantage of some of the tuning suggestions.

TS can provide the performance needed to service high activity systems if tuned and used wisely.

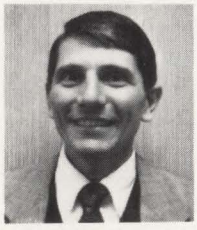
EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
270 (Interesting/Helpful), 271 (Too Technical), 272 (Too Basic)

2.) Would you like more articles on the same subject?
273 (Yes), 274 (No)

ABOUT THE AUTHOR



Ted C. Keller is the manager of a group responsible for CICS systems support, performance management and capacity planning at Yellow Freight System, Inc., 10990 Roe Ave., Overland Park, KA 66211, (913) 345-3274. He has worked in various data processing jobs for more than 21 years.

VSAM/Easy

Makes The VSAM Decision Easier

By Elizabeth L. Morgan

Database Management Systems (DBMS) are comprehensive, powerful and flexible data management tools but they also imply much higher overhead. Many times this higher overhead can be justified by DBMS' greater functionality and easy access via fourth-generation languages. In fact, some DP shops have embraced DBMS as a panacea for all applications and VSAM has been relegated to the position of "unsophisticated solution."

VSAM Lives On

Two major forces still breathe life into VSAM. First, a tremendous amount of data resides in VSAM files and the cost of migration to a DBMS can be prohibitive. Second, as many shops have discovered the hard way, a DBMS is not the best solution for all applications. At Penn State University, the data processing budget is in excess of $1 billion per year. DBMS performance sometimes degrades to unacceptable levels when handling the University's large volumes of data requiring intensive access. As Ken Blythe, director of Management Services at the University, sums it up, "We use a mix of DBMS and VSAM for optimal throughput and performance and have no intention of eliminating VSAM."

VSAM Strengths

VSAM utilizes inverted tree technology and because of its simplicity, VSAM can often process large volumes of records with high usage more quickly and efficiently than a DBMS. If you require simple, keyed files with summarized data, VSAM is more efficient and less costly than a DBMS.

Frank Garcilaso, manager of Systems Support for the City of Denver points out, "VSAM works better than a DBMS for small, highly interpretive-type tables and for files requiring few keys. VSAM does a good job if you do not try to use too much alternate indexing and do a good job of design."

VSAM Limitations

Though VSAM is clearly the most efficient and least expensive solution for many simple applications, one major limitation is a significant drawback. VSAM applications are usually coded in difficult-to-use, third-generation languages such as COBOL and PL/I. VSAM files are not generally accessible by 4GLs such as Software AG's Natural and Information Builders' Focus. Programming in COBOL and PL/I is much more cumbersome and time consuming than with the newer 4GLs. Randy Muller, director of Data Processing for Edison Brothers Stores (St. Louis, MO), estimates that developing applications in Natural vs. COBOL decreases his development time by as much as 500 percent.

A second drawback of using VSAM is that it has no effective data recovery capabilities.

Data processing shops have long wished for a way to utilize VSAM's inherent strengths and lower overhead without having to do their programming in COBOL.

VSAM Access Made Easy

Many VSAM users, including Penn State University, Edison Brothers Stores and the City of Denver, have discovered a valuable tool in VSAM/Easy, a standardized VSAM interface from MB Solutions, Inc. in Denver, CO. VSAM/Easy is designed to permit consistent VSAM access across languages, applications and processing environments. It supports full function VSAM access from programs written in most fourth-generation languages such as Natural and Focus. Also it provides easier and more efficient accessing of VSAM from programs written in third-generation languages such as COBOL and PL/I.

Many 4GLs will only read VSAM. Only a few will allow users to both read and write VSAM. Interviewed users of VSAM/Easy say that they are using the product with their fourth-generation programming languages to access VSAM files with full power and exceptional reliability. "VSAM/Easy makes working with VSAM files just as easy as working with DBMS files," says Muller. It manages all the complexities involved with accessing VSAM files. Inexperienced and proficient programmers alike are freed from considering the technical aspects of using VSAM. Programmers report that they need to know fewer commands and handle fewer errors when they access their VSAM files with VSAM/Easy.

Penn State University

Penn State's data processing environment involves tremendously large amounts of data and intense access to that data.

Penn State finds that with this level of volatility, its DBMS performance can degrade to the point that the computer has sometimes been brought to its knees. Blythe says, "As long as we are working in a record-to-record mode with small amounts of data, our DBMS response is adequate. However, because of our security requirements, when a user wants to obtain just a few records from a much larger database, response time to the request for information is not acceptable."

VSAM has helped resolve this issue. For end-user reporting needs, the current practice at Penn State is to extract information from the DBMS, copy the information to VSAM or sequential files and then make that extracted information available to end users for their reporting needs.

Says Blythe, "Programmers like VSAM/Easy. It has lightened the load on the computer and users are able to obtain reports on-line that they were previously able to get only from batch runs done overnight and they are able to obtain results from on-line requests faster because of the on-line application's use of VSAM rather than our DBMS."

Edison Brothers Stores

Edison Brothers Stores is implementing a new Merchandising Inventory Tracking System (MITS). The system involves inventory tracking for 1,200 stores and 8,000 stocking units in an average of 20 sizes and/or colors. It is estimated that the files contained in the system will have up to 125 million records, averaging about 200 bytes per record. On the average, each file will take approximately one 3380 DASD device.

When evaluating the appropriate access methodology for this application, Edison Brothers found that none of the DBMS' considered would perform efficiently against the processing level estimated for MITS. In addition, DBMS' do not permit simultaneous batch and on-line updates under CICS. However, with VSAM, Edison was able to develop its MITS with response time that is within acceptable ranges and to achieve the multiple threading it needs to assure fast response times. Edison can also perform the desired simultaneous CICS/on-line and batch update functions.

VSAM/Easy permitted Edison to develop MITS using a fourth-generation language to perform VSAM access tasks rather than having to use COBOL. This resulted in an estimated improvement in programmer productivity of five to one. VSAM/Easy gives Edison a standard way of performing VSAM access. The company is using the product both from its fourth-generation language (which has no native VSAM capability) and from COBOL.

In addition to the savings in time associated with its development efforts, Edison Brothers performed a benchmark to measure the efficiency of using VSAM/Easy to access VSAM from COBOL programs. The benchmark was conducted by performing 20,000 batch update transactions and 400 add transactions against a master file containing 25,000 master records. Table I shows the results of that test. It compares times required to run the jobs using native COBOL and times required to run the same jobs using COBOL but with VSAM/Easy being called from the COBOL program to accomplish all VSAM operations.

City of Denver

Mike Czyzewski, database administrator for the City of Denver, reports, "VSAM has tremendous usefulness, particularly in the area of table handling and small files. Our DBMS can only bring in one record at a time and bringing in a DBMS-based table (a table of unit codes for example) would require many read operations. Bringing the same table from VSAM would quite often (depending on the size of the table) require only one read operation. Additionally, because of 'Control In-

See VSAM/Easy *page 89*

SQL/DS AND DB2 USERS:

It Takes the Right Tools to Create a Masterpiece

Like all great works of art, creating an efficient database requires the right tools. Using Systems Center's, full line of database management tools, you can turn your database into a masterpiece.

With the creative edge that Systems Center database products provide, you can improve the performance of your database, increase the productivity of your programmers, and protect the integrity of your data.

- **DB/REORGANIZER** for **SQL/DS** (formerly VMSQL/REORG) allows the Database Administrator to improve the performance of SQL/DS by quickly and easily altering and reorganizing database objects.
- **DB/REPORTER** (formerly VMSQL/REPORT) for both SQL/DS and DB2 allows programmers to write reports in a fraction of the time required of conventional programming languages.
- **DB/OPTIMIZER** for **DB2** (formerly DB/OPTIMIZE) reports on DB2 access path selection problems, identifies potential problem areas, and recommends possible solutions.
- **DB/EDITOR** for **SQL/DS** (formerly VMSQL/EDIT) makes it easy to maintain SQL/DS tables and to build database entry applications.
- **DB/SECURE** for **DB2** provides a comprehensive means of controlling access to DB2 systems, resources, and data.

With Systems Center's database products, you can quickly master the art of database management. Call Systems Center today to get the tools you need to create a masterpiece!

(800) 359-5559
(703) 264-8000

6-MFJ-8904

SYSTEMS CENTER
FORMERLY VM SOFTWARE, INC.

NETWORK ADMINISTRATION PRODUCTS ■ RELATIONAL DATABASE PRODUCTS
VM SOFTWARE PRODUCTS ■ NETWORK DATAMOVER PRODUCTS

1800 Alexander Bell Drive, Reston, Virginia 22091

CIRCLE #5 on Reader Service Card ▲

DB2 In An IMS World

By Jack E. Olson

Most IMS shops recognize the eventual dominance of relational technology but continue to operate from strategies that inhibit its implementation.

By now, most data processing installations that have depended on IMS as their primary Database Management System (DBMS) have installed DB2. These veteran IMS businesses have a remarkable similarity to each other in their use of DB2 and stand in sharp contrast to those who have installed DB2 without ever having been an IMS shop.

Within most IMS shops, DB2 has a relatively insignificant presence. Where they may be using 10 licenses of IMS, they will have one or two DB2 licenses. The number of DB2 applications in production as well as their size and importance is small. It is not unusual to find a company that has had DB2 on site for three or more years and is still not using it for significant production applications.

In contrast to this situation are the non-IMS shops that bring in DB2. They are highly aggressive in bringing major applications to the DB2 environment and show little reluctance to convert major business-related applications to DB2.

The difference in these two situations undoubtedly results from the fact that IMS has been a highly successful and satisfactory DBMS. It has served its customer base well over the last 15 to 20 years and has encompassed most of the major business-related applications for those users.

However, DB2 is part of a new technology far superior to that of IMS and promises much more value to users in the future. Most IMS shops recognize the eventual dominance of relational technology but continue to operate from strategies that inhibit its implementation. These strategies keep them in an IMS-DB2 coexistence period too long and slow orderly movement to DB2.

DB2 Strategies

IMS shops tend to go through strategy stages like those in Figure 1. They install DB2 and run a small pilot project. DB2 has to prove itself to the IMS community before it makes a commitment to real applications. This phase typically lasts for six to nine months but sometimes it lasts years.

Another stage for many shops is that in which DB2 is an Information Center (IC) tool only. MIS managers move large amounts of data to the DB2 environment from the IMS environment on periodic schedules to execute report generation and ad hoc query functions. Although this is an excellent environment for IC operations, it significantly underutilizes DB2 for that customer.

Stage 2 strategies appear after shops recognize that relational technology is the technology of the future. This strategy statement appears to give a strong favor to DB2 and management appears to believe it is being aggressive in meeting new technology. This strategy statement is the current position of *most* IMS shops having DB2.

Stage 2 strategy, however, has a serious flaw in it. Although it gives DB2 new applications, most of the important business data has long ago been implemented in IMS applications. This situation leaves DB2 on the outside of a company's major activities unless significant large changes to an application get approved. When these rewrites are planned, inter-relationships among multiple applications tend to discourage converting to DB2 regardless of policy. Although Stage 2 appears to be an aggressive DB2 strategy, it is, in fact, an inhibitor for growth of DB2 use.

Stage 3 strategy is almost never a stated strategy. Applications are moved to a combined IMS/DB2 environment to allow faster migration to DB2. It is clearly a step in the right direction. However, it occurs as an unplanned exception to Stage 2 and not as an explicit strategy. Recent discussions of this issue with six IMS accounts revealed that all six had a Stage 2 strategy while all six had applications in Stage 3.

This apparent violation of Stage 2 happens because the DB2 community is eager to participate in the major functions of the business and actively seeks to move applications away from IMS in order to do so.

Stage 4 exists when a company has decided to commit the future to relational technology and explicitly plans to move most applications from IMS to DB2. There are few companies in this stage, but they do exist.

The greatest career advancement you could make this year isn't on the job...

It's in an Amdahl systems education course.

Let's face it, as a systems programmer your greatest asset to your employer is your knowledge. When you want to move up in the company, it's not who you know, it's what you know that counts.

Yet, in the ever-changing world of systems programming, keeping your knowledge current and up-to-date is no easy task. You can wade through dull textbooks, try to follow complicated interactive video courses or attend classes taught by instructors more interested in selling products than teaching knowledge.

Or you can take an Amdahl systems education class. Our programs are designed to bring you up to speed in an intensive, hands-on environment.

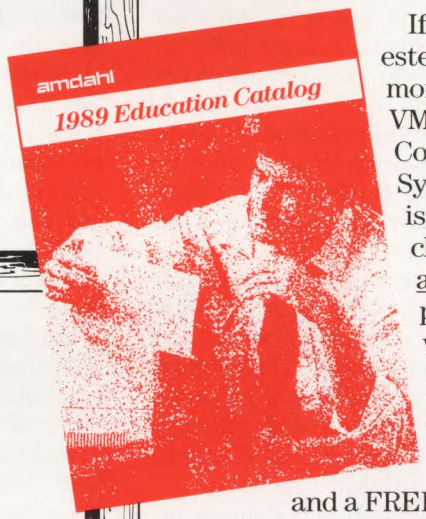
Professional, individual instruction in all 370 system architectures.

Amdahl is the only education service that caters to all 370 system architectures, *regardless* of vendor hardware. And we're the only center staffed by veterans in the field who bring a technical superiority and unparalleled expertise to every course. Plus, we keep our classes small to encourage an open forum where information is readily exchanged.

Fully-equipped Education Centers nationwide.

All Amdahl courses are taught in full-service Education Centers located all across the country with on-site labs for hands-on application. We schedule over 500 classes in 50 different subjects each year at Centers in Chicago; New York; Columbia, MD; Atlanta; Houston; Orange County and Santa Clara, CA; Washington, DC; Hartford, CT.

Request a FREE 1989 Education Catalog, today.



If you're interested in learning more about MVS, VM, Data or Communications Systems, Amdahl is the smart choice for you and your company. Find out what we've got going in 1989. For Express Enrollment information and a FREE 1989 Education Catalog, just give us a call. Or, complete and mail the postage-paid card attached for your Catalog. Don't delay, classes fill up early.

Take the next step. Call today:

1-800-233-9521
ext. 133

amdahl®
The SMART Choice

CIRCLE #60 on Reader Service Card ▲

Why you should want your data on DB2

There are four primary reasons for wanting the important business data on DB2 instead of IMS.

The first reason is that applications can be developed much faster with DB2 than with IMS. The company saves money in development and gets an earlier return from the application. This reason initially appears to be irrelevant to existing IMS applications. However, important data continuously becomes a part of new applications. Having data positioned in IMS inhibits new applications that use that data from being developed in DB2 or encourages an application to be developed as a MIXED APPLICATION where some of the data is in IMS and some in DB2.

The second reason is that an application is easier to maintain in DB2 than in IMS. Since most applications undergo significant maintenance over their lifespan, this factor is important. An application in DB2 sustains changes faster and cheaper than in IMS. The ability to change more responsively improves the value of the application to the organization. Maintenance costs are an excellent justification for converting an existing IMS application to DB2 without the impetus of a major rewrite.

The third reason is the exceptional ability of relational DBMS' to deal with ad hoc requests for information without the need for writing programs. The ability to respond to unexpected needs for information is becoming more and more crucial in business environments. With DB2 you have this; with IMS you do not.

The fourth, and most important reason, is that future technology advances will become available to DB2 and not to IMS. Distributed database capabilities will evolve on a relational base and not an IMS base. The enormous potential for distributed databases to change the way corporate computing is done will bring strong rewards to those who grow with it as opposed to those who do not.

The relational implementation of DB2 is better able to take advantage of future hardware advances in memory and parallel processing than is IMS. DB2's explosive performance improvements are not over. Major improvements are possible in the future through the synergy of relational concepts and operating system and hardware advances.

DB2 is in the SAA and ANSI mainstream; IMS is not.

Other emerging technologies such as CASE, end-user computing capabilities, expert systems and object-oriented databases will be built from relational platforms and not from IMS platforms. The integration of spreadsheet and database, text editors and database will all be available for relational databases and not IMS databases. These are exciting new capabilities that will add significant value to organizations that use them. If you have your data in IMS, you will not be able to play.

The situation for most IMS shops is that DB2 is the better DBMS. Most of the important business data is in IMS; therefore, the data is in the wrong place. The longer an organization resists conversion plans, the more value the organization loses over time.

So why not move all of the data to DB2?

There are many reasons why a migration to DB2 cannot be realistically undertaken in a short period. The cost of redeveloping the applications is significant. The availability of staff and the relative priority value of conversions to new applications must be considered. The conversion period is disruptive to operations and end users. It takes time to tune the DB2 version of applications to the same or better efficiency as their IMS counterparts.

DB2 requires much different administration and control. New procedures must be established and incorporated into the fabric of the users' operations.

Applications must be redesigned for DB2. IMS data design generally results in bad design for DB2. Since the user will plan to use DB2 for many years into the future, taking the time to redesign the data structures is essential.

These negatives seem overwhelming. Since IMS appears to be doing the job adequately today, companies are not motivated to move data even though failure to do so will cost them over time.

Users concentrate only on the *cost of conversion* without giving consideration to the *cost of not coverting*. These costs include the cost of maintaining both DB2 and IMS and all of the tools related to both, the lost opportunity costs for an application being on the wrong DBMS and the excessive cost and risk of applications being caught between IMS and DB2 environments.

Over time, the cost of conversion will go down as users gain more experience with DB2 and conversion tools emerge. The penalty cost of coexistence will go up as the DB2 environment becomes more valuable. When all costs are considered, an aggressive conversion strategy is the only one that makes sense.

FIGURE 1

Typical Coexistence Strategies

STAGE 0 — Pilot program to evaluate DB2
STAGE 1 — DB2 is IC (Information Center) tool only
STAGE 2 — DB2 gets all new applications plus major IMS rewrites
IMS applications are hands off
STAGE 3 — DB2 and IMS mixed applications allowed
STAGE 4 — Conversion plans for all IMS applications

It's time to reexamine strategy

For most IMS shops there will be a long period of coexistence. Most of them are operating from a strategy of coexistence that fails to recognize the advances made in DB2. Companies need to reexamine their strategies and establish more aggressive plans to take advantage of DB2 and the future that is coming with it. Coexistence should be a period of conversion and not a permanent relationship. Those shops that move through it will be better off than those that do not. Those that recognize and deal with the problems in conversion will be better off than those shops that let it occur unplanned.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.
1.) This article was:
290 (Interesting/Helpful), 291 (Too Technical), 292 (Too Basic)
2.) Would you like more articles on the same subject?
293 (Yes), 294 (No)

ABOUT THE AUTHOR

Jack E. Olson is responsible for product architecture and design of DB2 products for BMC Software, Inc. (Houston, TX). Working out of BMC's Austin, TX facilities, he has 20 years of experience in software development, 17 of which were in IBM development labs. His experience as a programmer, manager and architect have centered around database and data management environments. At IBM, Olson worked on both CICS and IMS.

Westinghouse Management Systems Software

VENDOR PROFILE

Westinghouse Management Systems Software is committed to improving productivity and quality. That commitment has resulted in several product breakthroughs, including the first disk utility system in 1969; the first commercial VSE teleprocessing monitor, WESTI, in 1971; and the first multiple session manager in 1984.

Today, Westinghouse has a comprehensive line of productivity software for IBM mainframe users, including products for data management, network management and productivity enhancement for use on MVS, VSE and VM operating systems.

Early History

Westinghouse Management Systems Software was founded in 1969 with the development of a disk utility (backup system) product at the Westinghouse Telecomputer Center (Pittsburgh, PA). Called Dump/Restore, the software was targeted for the IBM market.

Over the years that one product has been utilized at approximately 7,000 installations. It remains a well-known product in the DOS/VSE market and was instrumental in providing Westinghouse with a substantial customer base by meeting a previously unfilled niche in the IBM mainframe service market.

Throughout the 1970s, Westinghouse added new products to the DOS product line. The second product introduced was a communications product called WESTI, similar to IBM's CICS product. WESTI has remained a popular product among users who find it easier and more efficient than CICS.

In 1978, Westinghouse Management Systems Software began offering productivity enhancement software for the IBM VM operating system. In 1985, MULTSESS and NCI were added as the group entered the large-scale IBM MVS market.

New Product Introductions

Earlier this year, Westinghouse integrated its popular Network Control Interface (NCI) and Multiple Session Manager (MULTSESS). Also, the company expanded its network access and control product portfolio to produce a powerful, integrated network solutions package capable of running across all IBM mainframe operating systems.

Company officials feel the Westinghouse Integrated Network Solutions (WINS) product line is one of the most advanced, truly integrated network product portfolios available. In essence, WINS comprises an architecture for implementing VTAM applications similar in concept to IBM's Systems Applications Architecture.

WINS simplifies user access and control of complex networks by providing users with a broad range of powerful functions integrated with uniform panels. The WINS portfolio includes NC-SPE (Single Point Entry), MULTSESS, NC-NIM (Node Information Manager), NC-NVI (Network VTAM Interface), NC-VSAM (a VSAM I/O manager) and NC-Mail.

The WINS package features a common application architecture utilizing a central administration file.

The WINS package features a common application architecture utilizing a central administration file. The latter allows all the products, including MULTSESS, to share the same information, eliminating duplicate administration and minimizing use of computer resources.

In 1988, Westinghouse introduced a new line of IMS products including BORIS and SPICE. BORIS offers users a powerful on-line tool for reporting on IMS transactions. It can be used for chargeback, IMS tuning and monitoring. SPICE is an IMS checkpoint/restart utility. It prevents delays in starting the on-line system and loss of on-line transactions resulting from rerunning batch work after an initial failure.

Mark A. Potenzone, manager of the North American Group, oversees the MSS sales and marketing effort as well as customer support, contract administration, research and development. He is also responsible for acquisitions of new software products from third-party vendors.

European Offices Are Strategic

Five key offices in the United Kingdom, France, Switzerland, Germany and the Netherlands sell directly to the European market. Additionally, MSS utilizes the services of professional agents to cover areas where they do not function directly. Each office functions as a profit center and consists of sales and technical support personnel.

Leading the European effort is Chris Warren. As manager of European sales and marketing for WEMSSA, Warren is responsible for a staff of 50 employees as well as the professional subcontractors who represent Westinghouse products outside of the geographical area served by the key offices.

Quality Technical Support

Westinghouse stands behind its customers with quality technical support from detailed documentation to on-site consultation. The Westinghouse support staff works closely with the development engineers to acquire a thorough knowledge of all software products. Besides answering customer questions, they are trained to advise clients on implementation strategies.

Headquartered in Pittsburgh, PA, the company utilizes the sophisticated hardware at Westinghouse Corporate Computer Services to duplicate most operating environments in order to troubleshoot customer application problems. Branch offices are also maintained in Cherry Hill, NJ and Los Alamitos, CA.

Westinghouse Management Systems Software is ranked among the top 200 national software and service suppliers.

Vendor Profile is a regular forum whereby a vendor is given the opportunity to introduce the company and its products to MAINFRAME JOURNAL readers.

SYSD

SYSD Offers CICS Users An ISPF Alternative

By John Kador

Almost 90 percent of MVS data centers use IBM's ISPF (Interactive System Productivity Facility) under TSO (Time Sharing Option) for on-line programming development and editing. So where does that leave a tool like SYSD from H&W Computer Systems (Boise, ID), the subject of this month's product review? There are two equally valid ways to answer that question. You can say, "Only five percent are using it" or "As many as five percent have found it superior to TSO/ISPF." In any case, almost 500 data centers rely on SYSD.

SYSD offers a number of advantages over the IBM alternative. First, the users we talked to report SYSD is far more economical both in its acquisition and its overhead than TSO/ISPF. SYSD handles multiple users with considerably less overhead than its TSO counterpart. In addition, SYSD provides better performance and faster response times than TSO.

Second, users report it is easier and faster to use. With TSO/ISPF, specific commands are available only in specific modes and the system is totally menu-driven so that users have to go through menu layers to get to the function they want to perform. Under SYSD, all commands are accessible from a single mode.

Third, since SYSD is a CICS-based system, many CICS users will be able to stay in CICS without having to switch back and forth to TSO. These users have the functionality of ISPF at their fingertips but with only a fraction of the overhead. By being able to selectively move user groups away from TSO and into the more efficient environment of CICS, many companies are giving more users access to the CPU.

SYSD offers CICS users the following functions:

- An ISPF/PDF-like editor
- Job tracking from job submission, to input queue, to execution, to output queue
- On-line report viewing (spool display facility)
- Printing reports on CICS printers
- Job submission
- CICS management utilities
- On-line dataset utilities
- Optional editor interface to Pansophic's Panvalet libraries.

```
------------------ CPMS/SYSD PRIMARY OPTION MENU ------------------ (1/1)
INPUT ===>                                                  SCROLL: CSR

                                                    USER     - CAJ
  0 - PARMS     - Specify CPMS/SYSD parameters      DATE     -  12/26/88
  1 - BROWSE    - Display source data               TIME     -  11:38:07
  2 - EDIT      - Create or change source data      TERMINAL - L5E2
  3 - UTIL      - Perform utility functions
  4 - ACTIVE    - Display active jobs
  5 - N         - Display jobs in the input and output queue
  6 - O         - Display jobs in the output queue
  7 - PRINTER   - Display/change a printers status
  C - CICS      - Enter CICS transactions
  T - TUTORIAL  - Display information about CPMS/SYSD
  U - USER      - Perform user file maintenance
  X - END       - Terminate CPMS/SYSD session

PRESS END KEY TO TERMINATE CPMS/SYSD.
```

SYSD follows the standard ISPF hierarchical menu structure. It also comes with its own internal security system that can be controlled by a single administrator or delegated to reflect a specific organizational structure. Exits are provided to support RACF, Top Secret and ACF2.

Three Conditions

There are three general conditions under which a data center might consider implementing SYSD. The first reason is to open up the ISPF and Spool Display Search Facility (SDSF) services to more users. Frequently, users other than professional programmers require the power of ISPF and SDSF. However, because of normal shop limits or limits in TSO address space, this service cannot be offered. By using SYSD, which is CICS-based, this shop can now expand service without concern over incremental CPU resource consumption. For example, Texasgulf Chemicals reported a situation in which 8 to 12 concurrent TSO users brought a CPU to a standstill. When the ISPF load was shifted to CICS with SYSD, more than 150 concurrent SYSD sessions were active with no degradation in response time.

The second condition under which SYSD might make sense is for a CPU-constrained data center that has a growing ISPF and SDSF user base. By moving to the more efficient editor, the data center retains the functionality of ISPF and SDSF but CPU cycles are preserved avoiding a costly CPU upgrade.

DOS/VSE shops converting to MVS represent a third condition. The overhead necessary just to function in the MVS world is a constant surprise to users acclimated to functioning under VSE. Many shops are unprepared for the overhead consumed by TSO/ISPF and quickly find themselves CPU-bound. By exploiting CICS and SYSD, these shops often report getting more horsepower from their hardware investments.

Substantial Benefits By Avoiding TSO/ISPF

The benefits of avoiding TSO/ISPF can be substantial, as St. Paul, Minnesota-based MSI Insurance can attest. MSI operates an IBM 3090 under MVS/XA. The company succeeded in boosting programmer productivity 25 percent and saving $500,000 per year by avoiding the purchase of a CPU to handle additional processing that TSO would have required. According to Hicri Koroglu, technical services director, these savings were the result of MSI's decision to install SYSD rather than TSO/ISPF.

By avoiding TSO/ISPF, the savings were even more substantial, Koroglu asserts, noting that the CICS-based editor

is considerably more efficient than TSO-based ISPF. He is convinced the decision avoided the need to purchase another 3090-class processor for its 80 developers. "The decision to install SYSD saved about $500,000 a year. For seven years since we converted to MVS, that is $3.5 million! SYSD gives us 99 percent of TSO/ISPF but at only 25 percent of the cost," he notes.

CICS More Efficient Than TSO

Ingersoll Milling Machine Company based in Rockford, IL manufactures large-scale custom-made metal cutting and composite forming machining systems. The $400 million company operates a NAS-XL/70 (equivalent to an IBM 3090 Model 200E) under MVS/XA and CICS 1.7.

According to Clyde Webster, manager of Software Support, SYSD is instrumental to Ingersoll's commitment to build rather than buy applications software. All of Ingersoll's applications programmers (more than 100 at last count) use the editor's source program update feature for submitting jobs. Ingersoll was one of the earliest users of SYSD. It acquired the tool to replace a custom-built editor. The company had considered running TSO/ISPF but it was convinced that SYSD under CICS was more efficient. "In addition to its considerable advantages over TSO, we find it offers significant benefits," he says, noting such features as an ISPF-like editor, job tracking and submission capabilities, spool display facilities and on-line dataset utilities.

For example, applications programmers in the Cutting Tool Division use SYSD to program the numerical controlled machining centers. From engineering drawings, these programmers write the actual instructions that control how the lathe or milling machine blade moves over the part to be fabricated. The editor is also used to manipulate all source statements for submitting the job for processing and for reviewing the results. Programmers simply type in an engineering drawing number to create the members of a Partitioned Dataset (PDS) that allows the part programmers to start updating and writing the part program. They submit the job for processing and review the results. If the job is not correct, they make changes and submit it again, much like an applications programmer.

Ingersoll takes advantage of SYSD's many exits to beef-up system performance. Instead of putting all SYSD users into the same work dataset, it splits up the users. The company wrote some logic that assigned half the users to one dataset and half to another. The datasets were put on different disk packs and different channels, thereby cutting I/O in half, reducing contention and improving overall performance.

"SYSD has evolved into a product of strategic importance for Ingersoll Milling. If we lost TSO tomorrow, it would not be anywhere near as painful as if we lost SYSD. We have made it a company decision not to use TSO because of its overhead requirements. We would not be on the CPU today if we did all of our work on TSO instead of SYSD. We would have constraints and storage problems. It's clear we'd need a new CPU," Webster points out.

Conversion Leads to Problems

The Farm Family Insurance Companies based in Albany, NY provides life, health, disability and casualty insurance products to members of Farm Bureaus in 10 Northeastern states. In 1985, having outgrown the limitations of OS/VS1, Farm Family converted to MVS/SP. To meet the DP requirements of its users, it is currently running an IBM 4381 Model P14 under MVS/SP and CICS.

Among the features that resulted in an increase in productivity for the users is the split-and-swap feature.

Under OS/VS1, Farm Family had been using Pansophic's Panvalet library management system and Pan/Online as a line editor to access the Job Entry Subsystem (JES) spool. With the conversion, the company discovered that Panvalet/Online did not support JES2 spool access. "Losing JES spool access presented a major problem for our users. Users were accustomed to Pan/Online to review their job output. Everyone relied on being able to conveniently access the JES spool," Bob Smith, Farm Family's systems programming manager, reports.

Acknowledging the undesirability of eliminating such a widely-accepted function, Smith began looking at alternatives. With continued access to Panvalet libraries and to the JES spool as his key criteria, Smith quickly identified SYSD that has a Panvalet interface. When Farm Family Insurance Companies installed the system in 1986, users again had convenient access to Panvalet libraries and the JES spool. According to Smith, it is extremely efficient, leading to a number of unanticipated benefits. For example, SYSD uses about one-third of the amount of resource of another CICS region running production transactions as measured by RMF. Although Farm Family is running a lot of canned software in that CICS region, Smith does not believe that fact alone could account for the difference. "Some of the difference must be due to the efficiency of SYSD," he adds.

Among the features that resulted in an increase in productivity for the users is the split-and-swap feature. SYSD enables users to define up to four concurrent sessions. Moving between sessions is accomplished with a single operation. When one session is active, another can be called up simply by keying in the function. SYSD automatically accomplishes the split. When two or more sessions are active, movement between the sessions is done with a single key stroke. "The split-and-swap features were real gains for us. Most of our use here is making changes to a few lines of code and then checking the output. With the split-and-swap feature, we can make the change, run the job and check the output and go right back to the program and make more changes — and do all of this with a few key strokes," Smith explains.

The importance of the split-and-swap feature is echoed by other end users at Farm Family Insurance Companies. Paul Ziobrowski, associate actuary, explains that the actuaries use SYSD for updating premium rate tables for the company's multiple lines of insurance. "It's really convenient to use the split screen," Ziobrowski says. He often uses up to four sessions to enter and review data. He may have sessions with the JCL, the program or a subroutine and want to check the output in the JES spool. "I can enter a few new lines of code or some changes, switch to the JCL and submit the job and then check the output. If it isn't what I want, I can switch back and make more changes and repeat the process, all with a few simple keys," he adds.

Before settling on SYSD, Smith did consider other options. One of them was to use TSO with the Panvalet Option. A test using a stand-alone MVS machine confirmed his suspicion that using TSO would require too much overhead. "The trial supported my expectations that using TSO would be too costly in terms of resource use," he says. In fact, SYSD uses about the same amount of system resource as would be required by a single TSO user. (In total, SYSD requires 1752K of storage; individual TSO users require 1500K each. With 35-40 simultaneous users, the CPU would grind to a halt if TSO was applied to this activity.) "We simply couldn't afford to add all of the people who had been using Pan/Online as TSO users," notes Smith.

SYSD is available from H&W Computer Systems, Inc., PO Box 15190, Boise ID 83715, (208) 385-0336.

ABOUT THE AUTHOR

John Kador is a free-lance writer and a frequent contributor to MAINFRAME JOURNAL.

Unattended Operation *from page 66*

questions and to participate in the process. Deal with a user who is receptive and advertises successes.

Q. In an unattended computer center, how are hardware and communications problems identified and resolved on a timely basis?

A. Security and environmental monitoring devices are available to monitor the vital aspects of the data center in the absence of computer room staff. Such equipment can recognize failing equipment or intrusions and phone designated staff on an exception basis using voice synthesizers. Furthermore the devices can be queried by cautious or inquisitive management. Some on-call procedures are required but they may be activiated by hardware and communications vendors.

Q. Once achieved, what will be the role of the unattended computer center? What services can I expect from the computer center?

A. The unattended computer center will be a utility available for you at your convenience. The computer center services will be the same except that quality will be much higher.

Technical Obstacles to Unattended Computer Center Operation

Q. How do you expect to eliminate tape processing?

A. Tape processing is a major fault point and although there are replacement media available, tape is likely to be with computing centers for a long time. The most immediate alternatives to tape are the StorageTek 4400, Automated Cartridge System and the Masstor Systems M860, Storage Management System. Masstor Systems Corporation located in Santa Clara, CA does not have a large presence in the United States, but is making a substantial dent in the European market. It appears to be a good potential alternative for tape.

In addition, there are commercially available optical disk systems and the increased density of disk, the reduced expense and the ability to locate the physical devices at considerable distance from the mainframe makes disk-to-disk backup appear feasible as it was with the IBM 2311 and 2314 disk drive systems (removable disk).

Eliminating tape is difficult. Dependence on tape can be mitigated by reducing its use. Computer centers can identify tape that is obsolete and a roadblock to unattended operation and consciously reduce its use. When it is used it will be a conscious deviation and the computer center will be cognizant of the consequences. Nibble away at this media.

Q. How are you going to eliminate the tape library?

A. Unattended operation requires the elimination of tape since tape is the most rudimentary of manual computer center functions. However tape will be with us for a long time and tape management software helps to improve reliability and reduce direct labor associated with the use of this media. Identify tape as an obsolete media and as a roadblock to unattended computer center operation. Consistently reduce its use.

Tape dataset stacking software should be used in concert with tape management software. Many tapes are backups that are rarely used. Furthermore, many of these backups use only a fraction of the tape volume. By stacking these kinds of tape datasets, the data center can reduce the physical handling of tapes, reduce the volume of tape inventories, decrease offsite storage cost and improve cost containment. Purchased utilities are available to do tape dataset stacking; an example is Tape Data Stacking Utility (TDSU) from Alltran in Denver, CO.

Q. How do you expect to eliminate printing?

A. Report management and distribution software are available. This software directs reports to a disk device rather than to a printer. Once on disk, it can be retained for a predefined period, viewed and if necessary, printed under the control of an end user. This is not a substitute for on-line queries, but it is an outstanding intermediate step.

Q. How are you going to eliminate console operator interaction with the computer?

A. Identify all computer center procedures that require computer operator intervention. Divide the results of this evaluation into those procedures that are easy to eliminate and those that are difficult to eliminate. Further, divide the difficult into those that can be resolved with installed software and those that require new software.

Establish a plan that defers the difficult changes and implements the easy changes. The easy changes provide ample opportunities for reducing operator intervention and once accomplished, establish a presence and foundation for proceeding.

Next, determine the requirements for new software. New software requires a long lead time; get this process started early. Proper utilization of existing software is also a factor. In many cases, software has been purchased and installed but is neither properly nor fully utilized. With these two plans in place, it is much easier to go back and address the more difficult changes.

Q. The only automated rerun recovery system on the market today needs to be manually activated to perform the automated rerun. How is unattended operation achieved with a manual tool?

A. Nothing is perfect. Implement what you have and make the most of it. Look for opportunities to extend the features by writing routines that extend the functionality without impacting the integrity of the software. Bring the software supplier into the fold; make sure he understands what you are seeking to achieve and convince him that it is in his best interest to extend the features. Look for solutions that will solve your problem and assist them to make a profit.

Q. How do you handle a physical security problem or a problem with the physical support plant (air conditioning, fire, water chiller or electrical problems)?

A. Security and environmental monitoring devices are available to monitor the vital aspects of the computer center in the absence of computer room staff. Furthermore there is equipment that can recognize failing equipment or intrusions and phone designated staff on an exception basis using voice synthesizers. The devices can be queried for status. In today's world, these may not be one and the same solution, but I am confident that better solutions are on the horizon.

Q. What are you going to do when a critical process fails (an application software system or operating software) with no one in attendance at the computer center?

A. An environmental monitoring system can recognize interruptions to application processing as well as security breaches

and failing computer or ancillary equipment. Such equipment recognizes a failing process and phones designated on-call staff.

Q. How do you perform routine data library management functions in an unattended computer center?

A. If they are routine, then automate these functions. If they are not routine, question why you are doing them. There may be a different alternative. If all else fails, there is human intervention from operation analysts, technical services and database services. Remember, these other operations are also overhead functions and information technology is seeking to automate these functions. That is a different article.

Q. How are you going to procedurally accept new jobs into "production status" in an unattended computer center?

A. Automate the job turnover process and decentralize it to the computer user and application services. Installing new jobs into production status is a significant source of error in most computer centers. Moving programs into the correct libraries, changing JCL and so on are labor-intense and error-prone processes. The computer center adds no value to the process and only acts as an inspector.

Q. Can a Local Area Network (LAN) play a role in implementing unattended computer center operation?

A. A LAN is not an integral part of the unattended scenario. However anything that extends the sphere of influence of the central computer to the computer user and leverages the base of installed equipment is beneficial to the process. The LAN falls into that category.

Q. Are there tools that assist in the implementation of unattended computer center operation such as electronic mail and electronic forms authorization?

A. Definitely install electronic mail and electronic forms authorization. Neither tool is integral to the unattended computer operation process, but they make the transition much easier. Electronic communication makes communications easier and quicker during the transition period. Experiment with electronic conferences: try to get computer users to share hints and experiences via electronic mail.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
275 (Interesting/Helpful), 276 (Too Technical), 277 (Too Basic)

2.) Would you like more articles on the same subject?
278 (Yes), 279 (No)

ABOUT THE AUTHOR

Howard W. Miller, CDP, CSP, is responsible for administrative computing at Boston University. He has held senior-level positions in systems management for more than 20 years.

VSAM/Easy

VSAM/Easy *from page 80*

TABLE 1

COBOL vs COBOL + VSAM/Easy Benchmark (CPU Seconds)

	COBOL	COBOL + VSAM/Easy	Time Saved	Percent Savings
20,000 updates	42.23	40.52	1.71	5%
400 adds	8.90	2.60	5.59	68%
Combined run	50.42	43.19	7.25	17%

terval Processing,' an entire table can be read once and then remain in memory for the duration of the run, saving additional I/Os.''

The City of Denver also extracts information from its DBMS, copies it to temporary VSAM or sequential files and then makes that extracted information available to its users for reporting needs. Use of VSAM in this fashion has reduced the load on the database and allowed the division to develop some applications for on-line use rather than requiring that the application be run at night in batch mode. It also made it faster and easier for end users to obtain ad hoc information.

How VSAM/Easy Works

A program uses VSAM/Easy through a standard CALL or LINK to one of its two I/O modules (VNAT or VNATB) with associated parameter values that are defined in the Control-Block. Most file attributes are automatically determined by examining the VSAM catalog and/or the CICS File Control Table.

In addition to defining the Control-Block that tells VSAM/Easy what to do, a Record-Area must be defined that is large enough to hold the longest record anticipated to be returned from a record retrieval. This area is where data to be written to VSAM or data received from VSAM will be contained.

After the CALL has been successfully completed, a VSAM record will have been read, added, modified or deleted, depending on the parameters supplied in the Control-Block. If a record write operation was requested and successfully completed, a VSAM record will have been written to the VSAM file. If a record retrieval was requested and successfully completed, the retrieved record will be contained in the Record-Area.

Regardless of the success or failure of any requested operation, a three-character Return-Code indicating the exact results of the operation will be presented to the program in the Return-Code field of the Control-Block. This code may be interrogated by the calling program for conditions specified in the list of VSAM/Easy Return-Codes. Depending on the results of an operation and the analysis of the Return-Codes, appropriate actions may be taken under program control.

The selection of VSAM or a DBMS as the access methodology of choice for any particular application must be based on the specific requirements of the application. Each methodology has its strengths and weaknesses which must be carefully evaluated during the decision process. VSAM/Easy is available from MB Solutions, Inc., 2525 West Main St., Suite 205, Littleton, CO 80120, (303) 794-1740.

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
285 (Interesting/Helpful), 286 (Too Technical), 287 (Too Basic)

2.) Would you like more articles on the same subject?
288 (Yes), 289 (No)

ABOUT THE AUTHOR

Elizabeth L. Morgan is a free-lance writer specializing in DP-related areas.

PRODUCT UPDATE

IOA Enables Integrated Automation of Operations

Tone Software Corp. (Anaheim, CA) just announced the Integrated Operations Architecture (IOA). IOA is the underlying software architecture for Tone's CONTROL family of data center operations productivity software products: CONTROL-M Production Control and Job Scheduling System, CONTROL-D Report Distribution and Management System and CONTROL-R Automated Job Restart and Recovery System. IOA is said to be the design architecture that enables the integrated automation of all data center functions resulting in a single unified application for the control of the entire computer operations environment.

For more information
CIRCLE #200 on the Reader Service Card

Zebb, "The Rerun Manager," Introduced

Altai Software (Arlington, TX) recently announced a major new addition to its family of data center automation software. Zebb, "The Rerun Manager," was designed to bring automated rerun management for today's on-line environment into the 1990s. When Zebb is installed, it recognizes when a job is being rerun and automatically restarts that job at the point of failure or at the proper restart step if, for example, temporary datasets are in use. By automatically bypassing completed job steps that do not need reprocessing, Zebb is said to save an enormous amount of CPU and personnel time.

For more information
CIRCLE #201 on the Reader Service Card

BUNDL Provides Automated Report Distribution

Duquesne Systems (Pittsburgh, PA) now markets the BUNDL software product. BUNDL, named for its report "bundling" capabilities, provides an automated report distribution system for MVS environments. It is said to offer the first and only "system managed output" solution to data centers. In addition, BUNDL manages on-line viewing for users and automates the report distribution and printing process for data center operations by bundling and routing output to local or remote destinations.

For more information
CIRCLE #202 on the Reader Service Card

VIA/CENTER Provides Intelligent COBOL Reengineering

VIASOFT, Inc. (Phoenix, AZ) just introduced VIA/CENTER, a platform that is the foundation for an integrated suite of intelligent COBOL reengineering products. The VIA/CENTER products are based on a revolutionary reengineering platform that extracts comprehensive information about how programs work and stores it on-line for programmer use throughout the reengineering cycle. Included within VIA/CENTER is VIA/Insight, said to be the only interactive code analyzer on the market, and VIA/Smart Test, an interactive tester/debugger that integrates comprehensive testing facilities with program analysts.

For more information
CIRCLE #203 on the Reader Service Card

KEYFAST/DE4 Data Entry System Allows On-line Update

H&M Systems Software's (Maywood, NJ) KEYFAST was reportedly the first data entry system fully CICS, CMS and PC compatible. Recently H&M announced the newest version, KEYFAST/DE4. It allows on-line updating of user files by permitting data entry personnel to set up complete on-line applications providing both file inquiries and updates. KEYFAST/DE4 is fully integrated into the KEYFAST system and requires no training of application developers or end users. It operates in MVS, VSE and VM environments.

For more information
CIRCLE #204 on the Reader Service Card

ENDEVOR/MVS Now Has Configuration and Release Management

Business Software Technology, Inc. (BST) in Westborough, MA has enhanced the ENDEVOR/MVS change control platform to include an Automated Configuration Manager (ACM) and Software Control Language (SCL). It is now possible to capture and maintain a complete cross reference of an application system's components including all common program modules and their interrelationships. Once a software package is defined, it can be distributed automatically throughout the various stages of the application development life cycle, including remote sites in a network. ACM automatically captures program configuration at source compilation regardless of the program language. SCL enables users to define the logical relationships among software packages and to distribute them.

For more information
CIRCLE #205 on the Reader Service Card

SAS System Now Interfaces IDMS/R & DATACOM/DB

SAS Institute Inc. (Cary, NC) now provides SAS System's analysis and presentation capabilities for users of two of the more popular databases. The SAS/ACCESS Interface to IDMS/R links the SAS System under MVS with Cullinet Software's IDMS/R, while the SAS/ACCESS Interface to DATACOM/DB links the SAS System under MVS with Computer Associates' DATACOM/DB. SAS/ACCESS software is a modular component of the SAS System, an integrated software system for data management, analysis and presentation. Capabilities within the SAS System include data entry, retrieval and management; report writing and graphics; statistical and mathematical design and analysis; business planning, forecasting and decision support; project management and operations research and applications development.

For more information
CIRCLE #206 on the Reader Service Card

MVS AutoOPERATOR Recently Announced

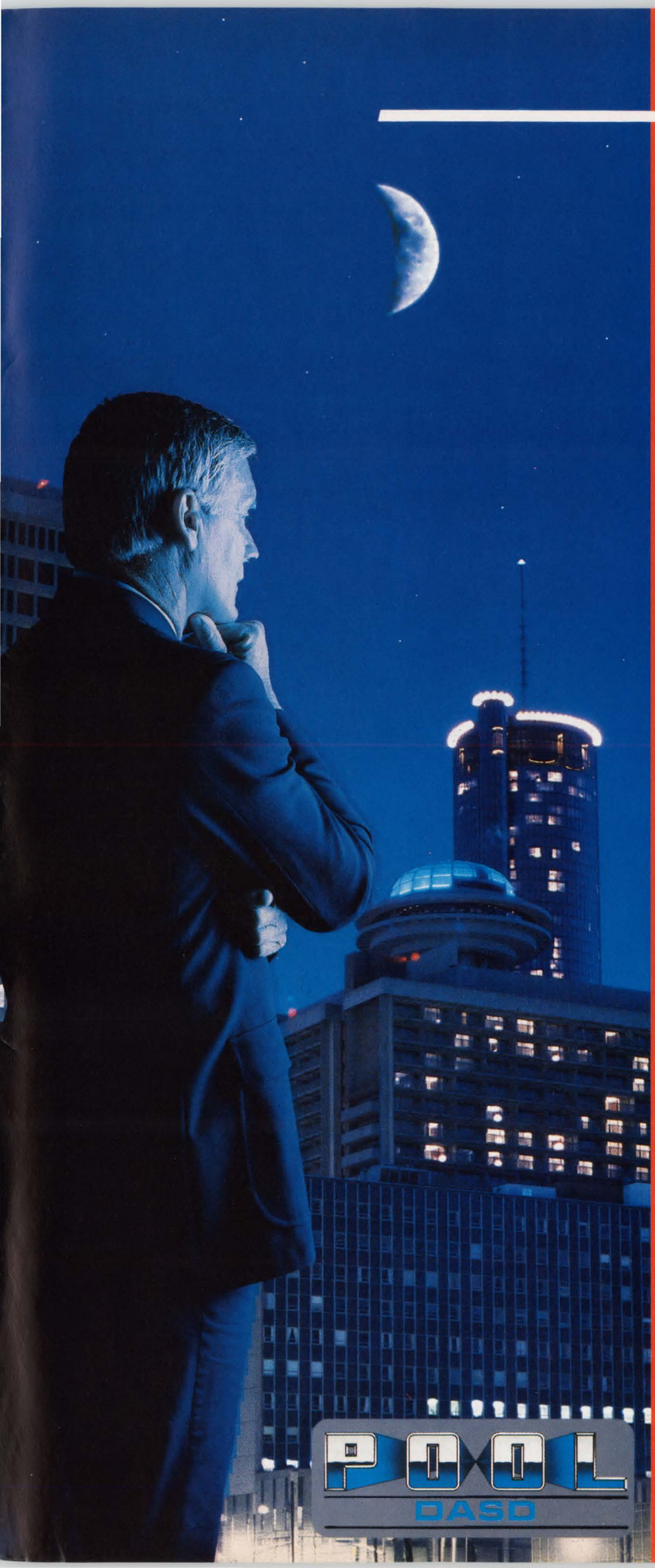
Boole & Babbage's (Sunnyvale, CA) MVS AutoOPERATOR is a Computer-Assisted Operations (CAO) tool for the MVS SP, XA and ESA environments. Designed to streamline data center operations for greater productivity, accuracy and control, MVS AutoOPERATOR offers the ability to manage all MVS resources through full-screen, panel-driven, ISPF-like displays. Remote and multiple systems, subsystems and workloads can be managed from a single location at a single terminal session. MVS AutoOPERATOR provides the operator with instant status information on system behavior as well as an integrated ALERT display that highlights critical problems, events and messages and takes decisive, pre-planned action to guarantee operational efficiency.

For more information
CIRCLE #207 on the Reader Service Card

BUDGET-DASD Provides "Realtime" Space Budgeting

Empact Software's (Conyers, GA) newest product, BUDGET-DASD, dynamically provides "realtime" DASD space budgeting and enforcement. It can control the total amount of DASD space (prior to allocation) that a user, group, division or company may allocate. BUDGET-DASD also offers choices of 24 fields to tailor budget categories including high-level qualifier, job account information, RACF user ID or a variety of other fields.

For more information
CIRCLE #208 on the Reader Service Card



"POOL-DASD has solved our major concern, controlling the allocation of datasets." –Technical Support, ISA

POOL-DASD is an IBM MVS mainframe software product that dynamically controls dataset pooling and enforces installation dataset standards.

Customer Comments:

*POOL-DASD has allowed us to go from **user-managed storage** to **system-managed storage** and enabled us to position ourselves for the future —TODAY!*
—Technical Services Manager, *Financial Services Co.*

POOL-DASD's flexibility and enforcement capabilities returned control of all DASD allocations back to where it belongs — with ME – I like being in control!
— DASD Administrator, *Bank*

Benefits / Features of POOL-DASD:

- ▲ Makes Pooling easy
- ▲ Provides Management Control
- ▲ Provides non-specific VSAM POOLING
- ▲ No EDTGEN's or IPL's to change pools
- ▲ Overrides or corrects JCL prior to allocation
- ▲ Offers over 50 fields as selection criteria for pooling and standards enforcement
- ▲ Sophisticated IF/AND/OR selection capability
- ▲ Enforces standards prior to allocation
- ▲ Installs in 30 minutes

Are you ready to take control? Call Empact Software (404) 483-8852

EMPACT software
A Boole & Babbage Company
Corporate Headquarters
1803 Overlake Drive, Conyers, Georgia 30208

CIRCLE #119 on Reader Service Card ▲

TECH ADVISOR

The question and answer below were published in the November/December 1988 issue. The response addressed the problem only from the MVS/SP aspect as Lee Barnes of Lee Data Corp. in Minneapolis, MN pointed out. So as not to mislead MVS/XA users, Barnes' response also is published for complete coverage of the issue.

Q *Could you elaborate on STEPLIBs versus LNKLST?*

A The LNKLST is for authorized programs. If you put the authorized program library in the LNKLST, then you can execute the program without the need for a STEPLIB (for that program only).

If you want to run an authorized program without putting the library in the LNKLST, then you must use a STEPLIB for that program and you need to put the library in the APFLIST. But if you put an authorized program in the LNKLST and then (for whatever reason or mistake) you execute the program using a STEPLIB, you lose authorization.

The use of the LNKLST can be a performance issue since the more libraries that are in the list, the longer the search chain becomes for all executable programs.

A Your answer was true for MVS/SP but the first and last paragraph are in error for MVS/XA. The first paragraph states, ''The LNKLST is for authorized programs'' and the last paragraph says, ''The use of the LNKLST can be a performance issue since the more libraries that are in the list, the longer the search chain becomes for executable programs.''

The LNKLST is not just for authorized programs. The system parameter LNKAUTH of IEASYSnn specifies whether the link datasets in the LNKLST are authorized. The LNKAUTH systems parameter has two values; LNKAUTH=LNKLST and LNKAUTH=APFTAB. When you code LNKAUTH=LNKLST (the default) the system treats all link datasets in the LNKLST concatenation as authorized but when you code LNKAUTH=APFTAB, the system treats only those link datasets in the APF table, IEAAPFnn as authorized.

To include link datasets that are not APF authorized in the LNKLST concatenation you should do the following.

1. Place the parameter LNKAUTH=APFTAB in the IEASYSnn member of SYS1.PARMLIB.
2. Include in the member LNKLSTnn all desired datasets.
3. Include in the member APF table, IEAAPFnn, all datasets to be authorized, including those already in LNKLSTnn. Exclude the datasets from the APF table that are not to be authorized but are in the LNKLSTnn.

The reason for using LNKAUTH=APFTAB to include non-APF authorized link datasets in the LNKLST is for performance. An otherwise high I/O link dataset in the LNKLST reduces the I/O for the directory of the link dataset because the directory resides in memory.

The LNKLST lookasize (LLA address space) function creates and maintains in memory a directory of modules in the LNKLST concatenation. Because the BLDL will find the program in the LLA directories in memory, STEPLIBs can then be eliminated for executable programs in link datasets of the LNKLST concatenation. The LLA directory is hashed and resides in the LLA address space.

Using the LLA has several advantages.

1. The LNKLST does not need to be tuned for concatenation for optimal performance, nor do BLDL lists need to be maintained. The order in which datasets are concatenated does not affect the time required to search hashed directories. BLDL lists and STEPLIBs are not necessary; the LLA maintains the directories of the LNKLSTnn in memory.
2. With the directories of the LNKLSTnn in memory, channel and device concatenation that otherwise occurs when searching PDS directories is eliminated.
3. Datasets in the LNKLST concatenation no longer have to be APF authorized. Unauthorized link datasets can be included in the LNKLSTnn that were formerly included in STEP, JOB and TASK libraries.
4. The LNKLST concatenation can include up to 123 datasets.
5. The LLA directory can be updated without performing an IPL. Members can be added, deleted, or updated in a dataset in the LNKLSTnn concatenation by issuing the F LLA, REFRESH command or by issuing the P LLA command followed by the S LLA command.

In summary, the best reason for including non-APF authorized link datasets in the LNKLSTnn concatenation is to reduce the I/O required to fetch modules in any highly-used PDS. Placing a highly-used PDS in the LNKLSTnn eliminates the need for STEPLIBs or JOBLIBs.

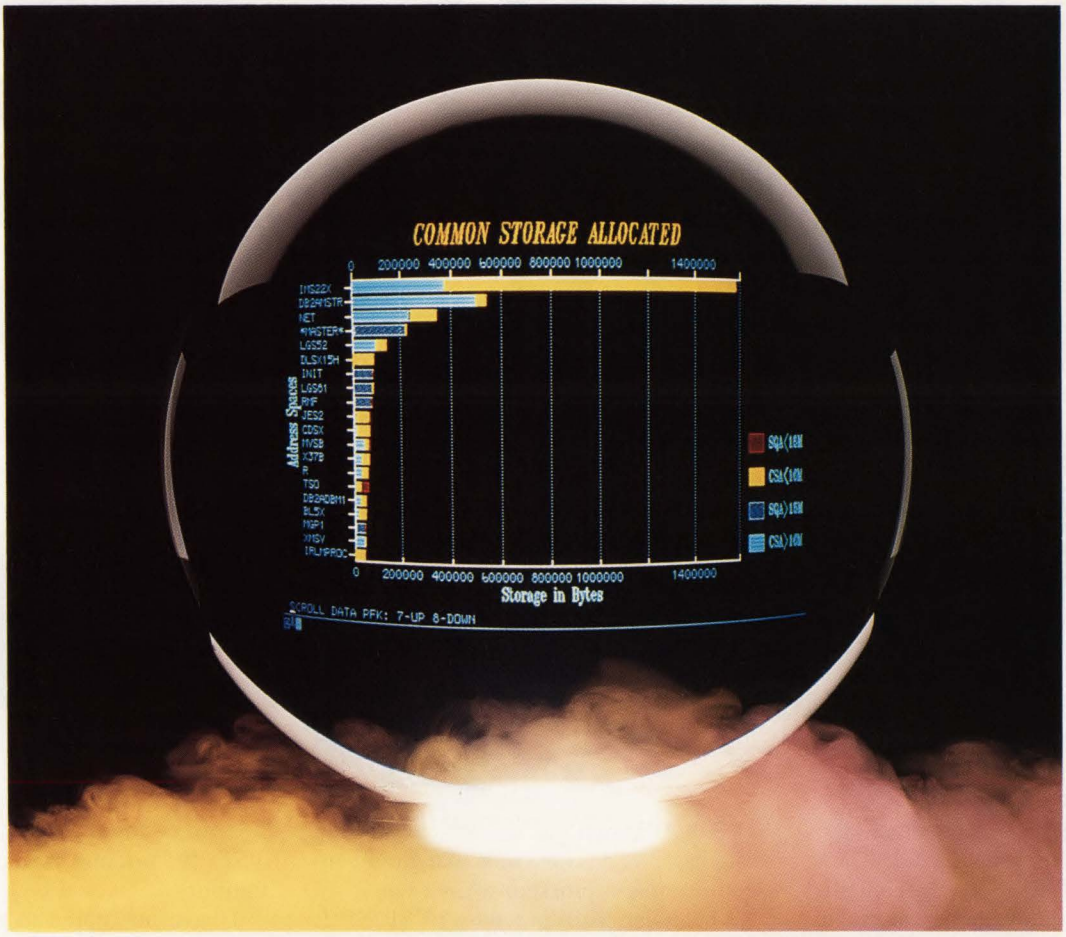
Q *With the release of a new Principles of Operation (SA22-7085-1), there are two new System 370-XA instructions. Could you explain the function and use of the CFC (Compare and Form Codeword) and UPT (Update Tree) instructions and their relationship to each other?*

A I looked up these instructions in the Principles of Operation manual you mentioned and I was quite surprised to find a three-page description and a flow chart for the UPT (Update Tree) instruction. The CFC (Compare and Form Codeword) instruction also has a lengthy description. Since I had not heard of these instructions before I received your questions, I began asking around to see if anyone knew about them. As you already know, I could not find anyone who knew how these instructions are used.

The speculation was that the CFC instruction will be used by sorts to improve processing times since the manual directly references sorts and the UPT instruction may be an instruction that ESA will be using. Exactly how they work or if they inter-relate I do not know. I suspect that a product developer may know this answer.

Questions are answered by consultants and instructors from Davis, Thomas & Associates (Minneapolis, MN), the largest technical services firm in the upper midwest. Please address your technical questions to: The Tech Advisor, MAINFRAME JOURNAL, PO Box 551628, Dallas, TX 75355-1628.

INTRODUCING THE FIRST REAL-TIME MONITOR FOR CSA.



Introducing the new Common Storage Monitor for RESOLVE PLUS, the first in-depth tool for monitoring users of CSA and SQA on MVS/XA. And the most effective way to reduce time consuming, and costly, IPLs caused by common storage constraints.

Control CSA creep.
Common Storage Monitor helps you control "CSA creep", the slow, hidden increase in common storage allocation that can ultimately result in system degradation and even failure. It is the only monitor that allows you to account for common storage usage by individual user. So you can identify applications that are abusing common storage and recover wasted CSA held by terminated tasks.

Pinpoint CSA usage.
Common Storage Monitor displays allocated storage for each address space by job name. Operating in either ISPF or command mode, you can display as much detail as you need to identify the user responsible for the allocation, and to analyze how the storage is being used. Password-protected RESOLVE PLUS services are also provided to free areas of CSA once they have been identified. Online color charts give you an easy-to-interpret overview of common storage allocation.

For more information on the Common Storage Monitor for RESOLVE PLUS Version 3.0.0, call Marty Johnson today. In California: 800-624-5566. Outside California: 800-822-6653. Boole & Babbage, Inc. 510 Oakmead Parkway, Sunnyvale, California 94086.

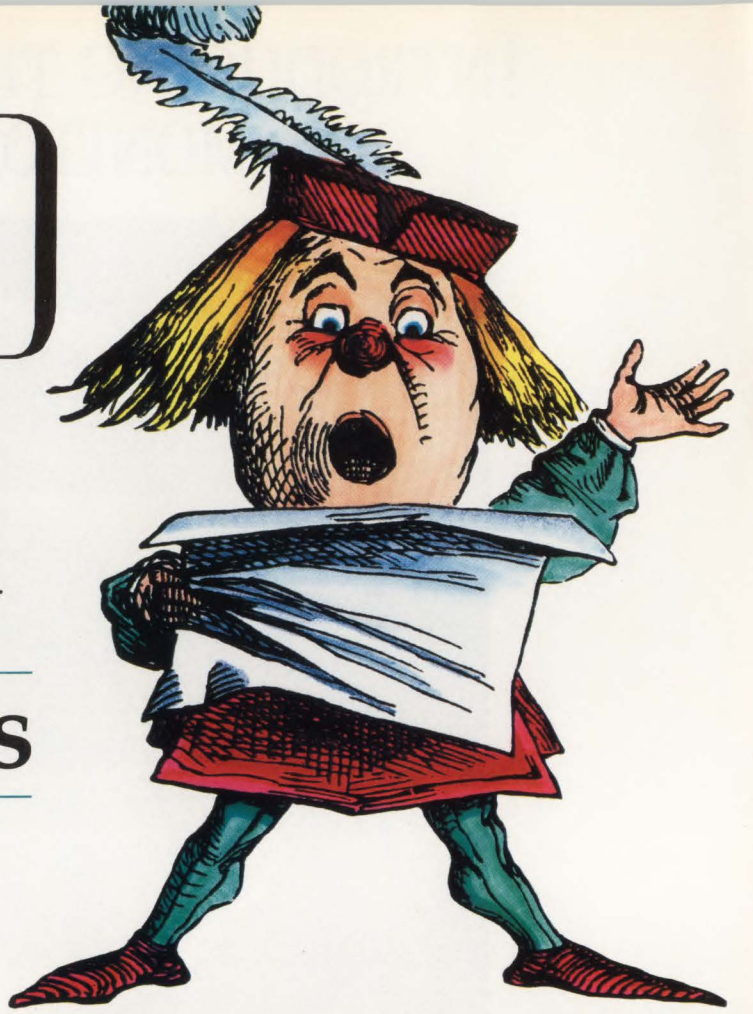
Boole & Babbage

International sales and support provided through The European Software Company and a worldwide distribution network.

NOW AVAILABLE IN A STANDALONE VERSION!

CIRCLE #160 on Reader Service Card ▲

VSE/SP Announcements



By Pete Clark

The IBM announcements made during the fall of 1988 included new levels of VSE/SP. Specifically, VSE/SP 3.2 became available December 1988; VSE/SP 4.1.0 will be available June 1989 and VSE/SP 4.1.1 will be available December 1989.

After making my way through the verbage and listening intently to IBM presentations on the new levels of VSE/SP, I am convinced that IBM is serious in its intention to "maintain the vitality of VSE."

It appears that VSE/SP 4.1 contains enhancements for a broad spectrum of VSE users. These enhancements are so broad that some current VSE users have indicated that VSE/SP 4.1 might not contain enough extentions and enhancements to entice them to install the new release.

VSE/SP 4 does contain an increase in license fees or an upgrade charge depending on whether you currently lease or have purchased VSE. Current mainstream VSE users are concerned that it may be difficult for them to present a realistic cost/benefit analysis to management for moving from VSE/SP 3 to VSE/SP 4.

More than one VSE user has pointed out that perhaps IBM is trying to address the substantial cost difference between VSE and MVS by increasing the cost of VSE rather than decreasing the cost of MVS. Although it is an interesting thought, it is an approach that might indeed become a problem of its own.

When Olan Mills purchased VSE/SP 2, the cost of the complete package for a group 30 license was approximately $60,000. At the VSE/SP 4 level, that figure is now approximately $107,000 — a rather substantial increase. While I certainly believe that VSE/SP 3 and 4 do contain substantial additional functions, I do not believe that VSE/SP 3 and 4 contain functions and features that make it twice as valuable to Olan Mills' installation when compared to VSE/SP 2.

It is obviously to IBM's advantage for its customer base to move to new levels of code as soon as possible and I believe that it is also an advantageous move for users. The advantages for IBM are reduced back levels of code to support, reduced education requirements for personnel, reductions in problem support requirements, improved revenue and so on. User advantages are new functions, new facilities, better system stability, improved performance, better support and so on.

In today's environment, users are typically not moving to new levels of code as rapidly as they should and a less than attractive cost/benefit ratio will only slow this movement. Pricing should encourage users to upgrade to later levels of VSE/SP; pricing should not become an inhibitor.

I have attempted to outline the major enhancements of the three VSE announcements as I understand them. Additional information is available from your local IBM office through announcement letters.

The VSE/SP 3.2 functional enhancements are the following.

- Support for new ES/9370 CPUs.
- Support for new DASD attached to 9370 processors; that is, 9332-600 and 602.
- Support for up to nine address spaces.
- Support for up to 128MB of virtual storage. Note that only 40MB of VIO is supported. (User patch available for 128MB VIO.)
- Some minor improvements to VSAM listcat output.
- Labeled tape support for tapes produced by the VSE librarian.
- A more appropriate way of handling dates contained within the VSE library directory during restores.
- A VSE console display of partition and system GETVIS status.

The VSE/SP 4.1.0 functional enhancements are the following.

- Basic cache support including support of read operations, error recovery and commands to initiate cache function.
- 3745 communication controller device support via a new level of NCP code.
- Additional new 9370 CPU support.
- SQL guest sharing support when operating VSE under VM that allows VSE and CMS to share access to SQL databases under VM control.
- A new level of DITTO that includes enhancements in full screen data view/update, additional browse functions, ASCII/EBCDIC translation facilities with tape functions, new CMS dump/load file functions and a group of *productivity* enhancements.
- A new level of POWER including support for an output exit that permits modification of output records, a task dispatch trace for problem determination, parallel asynchronous subtask processing, highest priority relief (that is, VTAM/CICS under control of POWER can be at a higher priority), mode set support for 3480 tapes, improvements in CPDS print support to ease restarting of output (that is, record count incrementing), network account number for account billing information when jobs are sent to a JES host for processing, increased from 20 to 60 the number of sessions available when POWER shared spooling is participating in a network and improved partition dump format.
- A new level of VSAM that includes improved open/close error information (that is, catalog name, module id, additional verbage), bufni and bufnd parameters available on the // DLBL JCL statement that should result in improved buffer space utilization, dynamic assignments to the same DASD will use the same LUB and IDCAMS print infile dump will print keys in character and hex.
- A new level of DL/1 that includes a new status code returned for certain error conditions when processing with procopt GO, reports that contain time and date stamps, a partial database load option with limited read access, extended CI Sizes up to 30K. Improved blocksize on tape for backup/restore processing (that is, up to 32K) and return code support from utilities and application programs to VSE conditional JCL.
- A new level of VTAM that contain some additional function and facilities but apparently contains a requirement for additional shared storage that may be a problem for many. I suspect that anyone who is currently having problems with the shared storage size of VTAM may have a real problem with the new VTAM. I believe that IBM is aware of the impact of the continuing VTAM storage increases that are occurring with each new release and is actively searching for a solution. IBM, please hurry, many of us are severely restricted by VTAM's need for large amounts of shared storage.

The VSE/SP 4.1.1 functional enhancements are the following.

- A new level of POWER including support that allows POWER to schedule by day and time when a job should be executed.
- Removal of the requirement for a dedicated VSE system console. Enhanced unattended node support, including problem determination from a central site and enhancements for distributing PTFs via DSX/DSNX from a central support site.
- Some improvements for continuous operation, including automatic system restart after either a hardwait or a system breakdown and automatic subsystem restart after an abnormal termination.
- Automatic reorganization of the ICCF library when contiguous free space has been exhausted.
- Procedures that provide for a fully-automated system startup and shutdown.
- Some enhancements to OCCF, allowing NetView to exist in a non-shared partition, improving unattended node and continuous operation support.

When I look at the new function and facilities of VSE/SP, they appear to logically group into three areas. One is enhanced distributed VSE node support, another is new device support specifically 9370 oriented and third, extensions for the current VSE user base. IBM appears to be trying to make VSE appeal to a broad spectrum of users both current and new. I submit to you that is a good approach.

The difficulty is in balancing the needs of the various different user types when developing function and facilities. It is up

to you and me to furnish IBM with concise appropriate information as to needs of the user base and it is also up to you and me to be sure that we are heard and that appropriate actions are taken. Various estimates place the VSE user base at 22,000 to 30,000 licenses — a sizeable contingent and a user base that should be a significant directional force.

If these latest VSE announcements do not include areas of major concern to your particular installation, it is time for you to inform IBM. Utilize IBM PASRs and user organization requirements to present the information that you feel is appropriate to IBM. From my perspective, the latest announcements appear to have been too biased toward distributed VSE nodes and I would suggest that additional emphasis be placed on resolving situations and product extensions for current VSE users.

VSE/SP 3.2 and VSE/SP 4.1.0 contain the significant enhancements for our installation. They are: extended addressing, POWER enhancements, VSAM enhancements and cache support along with the VSE/SP 4.1.1 POWER time and day scheduling feature. It is nice to see significant VSE enhancements announced and delivered. I am certainly looking forward to VSE enhancements in the future. (I hope.) I do have some concern about the size of the cost increase of VSE/SP.

I would suggest to you that when all the announcements are reviewed in proper context and as a group, the bottom line is IBM is "continuing to maintain the vitality of VSE."

EDITORIAL EVALUATION

Please circle the appropriate numbers on the Reader Service Card.

1.) This article was:
295 (Interesting/Helpful), 296 (Too Technical), 297 (Too Basic)

2.) Would you like more articles on the same subject?
298 (Yes), 299 (No)

ABOUT THE AUTHOR

Pete Clark has been in data processing for 25 years, the last 11 with Olan Mills, Chattanooga, TN. Clark is a recognized authority on the VSE operating system. He has worked many years to extend the limits of VSE.

Taking a Hit?

LET OUR THREE "POWER PUNCHES" PROTECT YOUR SYSTEM PERFORMANCE!

Taking a hit in system resource monitoring and accounting? Turn to Macro 4 Systems Software for your performance solutions.

VPAC—the unbeatable VM performance analyzer that puts automatic control of resource allocation at your fingertips, providing you with:

- graphic displays and comprehensive reporting
- early detection of system bottlenecks
- continuous exception monitoring
- dynamic control of system resources

VSAM/TUNE—world's leading VSAM management and performance tool that:

- tunes clusters for optimum I/O performance
- dynamically allocates buffers
- provides on-line analysis and modeling
- plans and simulates for system growth without impact
- simplifies IDCAMS listings and generates control statements
- offers a single-product solution without the hassle of multiple VSAM software packages

SYSTEM ACCOUNTING—the perfect partner for the DOS/VSE environment with comprehensive analysis that supplements IBM data with accurate statistics, giving you:

- comprehensive optimization tuning
- detailed load monitor and capacity planning
- precise accounting for both CICS applications and system resources

Look to these Macro 4 "power punches" when your needs call for total precision in system performance monitoring.

TUBES: World's Best Session Management Tool • CICSPRINT: Spooling Package for CICS, JES and Power
DUMPMASTER: Dump Management and Analysis Package • LOGOUT: Premier Console Manager for DOS/VSE

macro 4
Systems Software

Brookside Plaza, P.O. Box 187, Mt. Freedom, NJ 07970
(201) 895-4800 • (800) 223-0414

INTERNATIONAL OFFICE: MACRO 4 PLC Crabbet Park House, Turners Hill Road, Worth, Crawley, West Sussex, RH10 4SS U.K. (0293)886060
OFFICES ALSO IN: France, Italy, Switzerland, Belgium, West Germany, Netherlands, Spain, Finland, Austrailia and Japan

WE'LL HELP YOU DELIVER YOUR OWN KNOCK OUT PUNCH!

CALL 800-223-0414

AND PUT THE POWER OF MACRO 4 IN YOUR CORNER.

We've been providing powerful solutions to MIS system problems on an international scale since 1968—with a full range of software solutions for VSE, VM and MVS operating system requirements.

CIRCLE #39 on Reader Service Card ▲

SAA – Roots And Future

By Aubrey G. Chernick

Aubrey G. Chernick, President, Candle Corp.

One of the most significant developments in information processing in the last 20 years is IBM's Systems Application Architecture (SSA). Although there are many reasons for this, the bottom line is that SAA extends the centralized mainframe benefits of the 1960s and 1970s and extends the PC benefits of the 1980s by transparently marrying the benefits of both. It accomplishes this through layered system software that enables application software to be processed on a portable, distributed and cooperative basis on diverse computing platforms.

What are the roots of SAA and where is it headed? I believe five major forces have shaped the evolution of SAA with its benefits and impacts evolving in at least three foreseeable ways.

We can look at the forces shaping SAA from many perspectives, including the ''theoretical drivers,'' ''personal drivers,'' ''competitive drivers'' and two that are ''application drivers.''

SAA Root I — Theoretical Drivers

IBM periodically cycles around to using ''systems'' approaches to drive its product direction. SAA becomes the latest attempt at a system solution for the marketplace — an evolution beyond System 360.

SAA Root II — Personal Driver

Earl Wheeler is a key driver behind SAA. The benefits that Earl (a VTAM expert) observed based on the layered nature of SNA are now brought forward to provide a layered approach to information processing (SAA layers).

SAA Root III — Competitive Driver

IBM wants to use SAA competitively against DEC, AT&T and the Japanese.

SAA Root IV — Customer-Written Application Drivers

Customers want a growth path for their own applications. IBM wants SAA to help grow S36 and S38 users into ultimate mainframe users.

SAA Root V — Packaged Software Drivers

IBM has become concerned about application software houses writing for DEC platforms. IBM tries to woo these application software companies to IBM and away from DEC. SAA is the carrot — code to SAA and sell your applications on everything from PCs to 3090. (Roots IV and V suggest that IBM has relearned that software sells hardware.)

Of all the forces that have shaped SAA, I believe the last two ''application drivers'' are the most significant from an industry point of view. Here now are what I see as future impacts.

SAA Impact I — Portability

The first impact of SAA as it relates to applications is portability. This allows an application (user developed/''investment'' or vendor packaged) to be easily ported to a larger or smaller hardware system. SAA ''layers'' protect the application from being ''sensitive'' to the hardware (see Figure 1). Portability enables the application to be shifted between systems. ''Inventory'' can run on PC *or* S38 *or* 3090.

SAA Impact II — Distributed Data Processing

At this level, SAA enables applications (on one platform) to access data from various platforms. IBM sells data management technology (ESA-SMS, optical, DB2, DASD farms) at the mainframe, mid-range and PC levels in response to new applications that are being enabled (desktop publishing, image, voice).

SAA Impact III — Cooperative Processing

This evolution of impact goes well beyond the initial portability benefits (and even goals) of SAA. It allows the processing of single transactions to be performed (cooperatively) across more than one system.

Part of the processing is performed ''close'' to the user (PC). Other parts of the processing (same transaction) may be processed at a mainframe (update or access to the corporate database).

The cooperative processing environment becomes more important as a company tries to integrate intelligent workstations with centralized information databases. It is also important in on-line transaction processing environments where greater availability and higher throughput (tps) can theoretically be achieved.

SAA Impact IV — Systems Staff

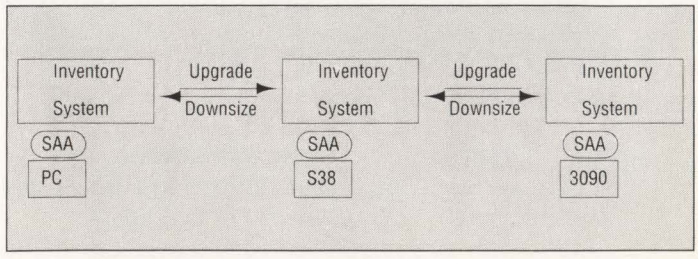
The evolution of SAA impacts beyond portability will affect technical support personnel. Very simply, if the processing of single transactions is spread either via data (Impact II) or via function (Impact III) across multiple systems, they all have to be working.

Data center executives look forward to this future ''enterprise'' model of computing (3090, mid-range, PC). But data center staff will have to deal with the increased complexity of ensuring that all ''component'' systems involved in all transactions are running (availability) and all are performing well (performance and response time).

When there is a problem and service levels are not met, the enterprise computing environment that the tech support staff will have to analyze will be awesome.

Regardless of the motivation, SAA truly will be a boon to enabling application processing within different IBM platforms. At the same time, the future distributed and cooperative processing benefits that will be felt by application development and end users will cause operational and performance complexities that will be a challenge for tomorrow's data center employees.

FIGURE 1



DOCS™

THE BEST-KEPT SECRET IN VSE CONSOLE AUTOMATION.

Data centers today are serious about automating their VSE operations. Most realize the system console is the logical first step. But what many don't realize is that the solution for VSE console automation is already in use at hundreds of data centers worldwide.

TOTAL MESSAGE CONTROL

DOCS from SMARTECH Systems actually operates as the VSE console, which means you have total control of *all* console messages. And because DOCS is not dependent upon an online system, you always have access to multiple consoles, *local or remote*.

What's more, with DOCS' auto-reply capabilities, you can practically automate your entire system operation by responding to anticipated messages *before* they appear. You can pass CICS or VTAM commands from batch or even automate the system startup procedure.

Plus, DOCS' message suppression and routing capabilities allow you to customize each console to display only the messages *you* require, eliminating messages that don't need attention. You can even operate multiple VSE consoles *and* the VM operator console from one CRT — giving you a comprehensive console automation solution.

AUTOMATING YOUR KEYSTROKES

DOCS has a wealth of features to automate your keystrokes such as programmable function keys, multi-line input, automatic data insertion, last-line recall and screen recall. These features mean you accomplish more — *in less time*.

INSTALLS IN JUST 30 MINUTES

After a simple 30-minute installation without any customization, DOCS becomes an invaluable part of your operation.

So if you are looking for the secret to VSE console automation, find out what hundreds of our customers already know.

For more information on your console automation needs, call 1-800-53-SMART. (Outside the U.S., call 214/956-8324.)

SMARTECH SYSTEMS, INC.

Turning high technology into SMART TECHnology.

10015 W. Technology Blvd., Dallas, TX 75220
FAX: 214/357-6338 Telex: 9102503110

International representatives:

France: Futurs Systemes et Technologies • Poitiers, France
Tel: 33-49-01-7374, FAX 33-49-01-7351

Australiasia: Mycroft Systems Ltd. • Auckland, New Zealand
Tel: 64-9-817-7673, FAX: 64-9-817-3640

For additional international representative information, contact SMARTECH Systems, Inc.

VSAM DATA COMPRESSION

Without the CPU Overhead

IAM REDUCES THE SIZE OF YOUR VSAM FILES BY 30 TO 70%

IAM'S FILE STRUCTURE — SAVES 20 TO 40%
IAM uses an advanced file structure which is far superior to VSAM. IAM's supercompressed index, freespace concepts and blocksizes make much more efficient use of disk space.

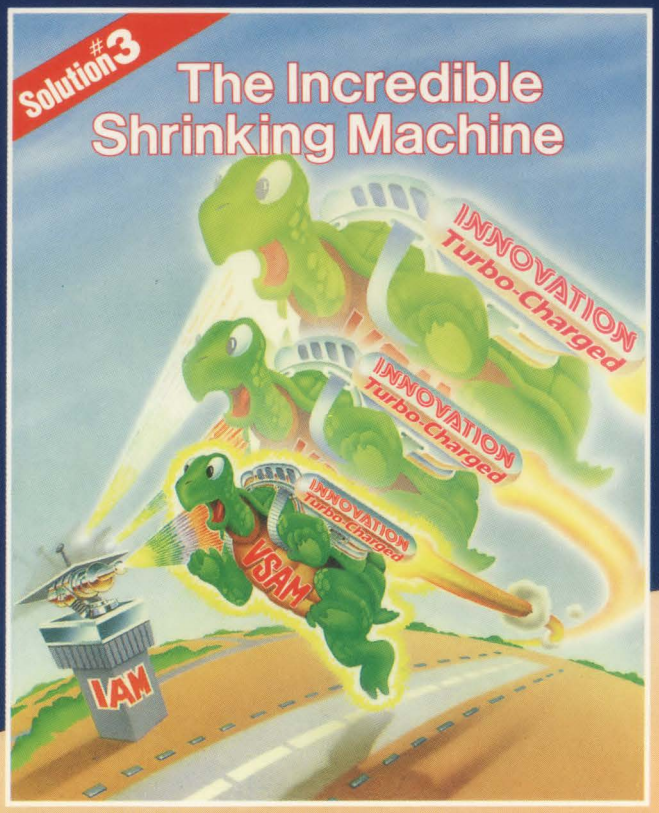
IAM'S DATA COMPRESSION
SAVES AN ADDITIONAL 20 to 50% DASD SPACE
Most files contain records with unused fields or repeating sets of characters. When IAM applies its proprietary compression techniques, the result is an additional 20 to 50% reduction in file size.

IAM's CPU time is dramatically less than competing compression products. In fact, since IAM's CPU time is normally much less than VSAM, IAM with data compression takes less CPU time than normal VSAM processing.

TRANSPARENT
Online systems (CICS), BATCH jobs, TSO, SMP/E and other applications make extensive use of key indexed VSAM (KSDS) files.

IAM is a transparent alternative to VSAM KSDS files, which substantially reduces the impact of VSAM processing in your installation. There are no modifications to programs or JCL to use IAM files in place of VSAM.

AUTOMATIC RELEASE OF UNUSED SPACE
IAM takes the guessing game out of VSAM space allocation. Large amounts of disk space are wasted when users overestimate how much space VSAM requires or how many records a file will contain. VSAM cannot release overallocated space.



FREE VSAM SPACE SAVINGS ANALYSIS*

SPACE SAVINGS

DATA SET NAME	VSAM TRKS ALLOC	VSAM TRKS USED	IAM TRKS STD	IAM TRKS COMPR	% SAVINGS STD	% SAVINGS COMPR	TOTAL RECORDS	AVERAGE LARGEST	MAX LRECL
BIG.CLUSTER	37155	37155	27855	15600	25	58	4754670	233 502	580
CICS.FILE.MASTER. TABLE.CLUSTER	21000	19005	12720	9495	33	50	5068165	150 150	150
NAME.ADDRESS.FILE	9315	8985	6465	1875	28	79	428529	680 2090	2090

The VSAM simulation report displays the current size and data characteristics of your VSAM files and how much space IAM will save you with and without data compression.

To see your VSAM usage, send for INNOVATION's free VSAM reporting programs.

Call for a Free No Obligation 90 Day Trial

Supports MVS, MVS/XA and MVS/ESA

Makers of FDR and ABR

INNOVATION DATA PROCESSING

275 Paterson Avenue, Little Falls, NJ 07424 • (201) 890-7300

CIRCLE #97 on Reader Service Card ▲