All Things Data

In this issue...

- Daylight Saving Time and your installation
- IMS DLIModel utility
- DB2 and DFSORT
- Recommendations for:
  - Stand-alone dump
  - System test
  - File systems

Kershaw Mehta
Senior Technical Staff Member, System z Strategy and Design Group
We’re very excited.

Hot Topics has a new look. From our cover shot to some stylin’ illustrations inside, we’ve unveiled what we think are terrific new graphics. Of course, we continue to offer the same great information on technical topics that matter, from some of our best and brightest contributors. Like Kershaw Mehta, zSeries Strategy Manager and Globalization and Unicode Architect. That’s Kershaw standing in front of one of our new System z9 servers on the cover. He is co-author along with Nick Carbone of our lead feature “Spring ahead without falling behind. What DST changes mean to you,” all about enhancements to data processing for recent changes to extend Daylight Saving Time in the U.S. This issue is highlighting all things data because we know how important data and applications are to your business. And we hope with our new look and feel that you will continue to be entertained as well as enlightened.

They say change is good and in this issue you’ll find plenty of changes to enhance z/OS and the processing power of its applications. You’ll find that our emphasis in this issue is on managing your data whether for the z/OS operating system or for z/OS products and functions like Language Environment, IMS, DB2, DFSORT, and WebSphere Application Server.

There are also recommendations for users of stand-alone dump, tips for system testing, tips from a new z/OS user about getting started with tools, and a migration tool for V1R8. And we haven’t forgotten you z/OS UNIX System Services users with topics that range from a discussion of serializing your data with byte range locking to information on migrating file systems. Want to brush up on your z/OS skills? Learn how to become a certified z/OS master with info about the IBM System z Entry Level for z/OS System Programmer Mastery test.

Speaking of changes we’ve had a few ourselves with the z/OS Hot Topics Newsletter team. We say goodbye to Tracy Williams and Tara DiMaggio and welcome Ken High, our new Art Director. As always, we strive to provide our z/OS users the best and most useful technical information.

Our goal is to please, so we’d like to hear from you — what you like, what you don’t like, what you’d like to see. Do you want more regular newsletter departments like “Level 2 with me?” Are there favorite products or functions that we’ve overlooked? Interested in more articles about System z hardware? Any ideas for future newsletter covers? More snakes on a mainframe? Please let us know by dropping us a line at the address below.

The Editors
newsletr@us.ibm.com
Contents: Featured articles

4  Spring ahead without falling behind  What DST changes mean to you  
   BY KERSHAW MEHTA AND NICHOLAS CARBONE
3  The latest thrilling celebrity hook-up  DB2 and DFSORT!  
   BY R. DAVID BOENIG II
18  Overlooking something?  Seven tips for improving your system tests—and your luck  
   BY JOEL MASSER AND NANCY FINN-DENAPOLI
22  From h to z  HFS to zFS file system migration  
   BY AHILAN RAJADEVA AND BILL SCHOEN
38  Every picture tells a story  Best practices for stand-alone dump  
   BY ED MEKEEL AND GLENN GARRISON
40  Overcoming obstacles  IMS DLIModel utility  
   BY KEVIN WANG AND DANNY NGUYEN

6  Snakes on a mainframe  Message flood automation  
   BY W. KEVIN KELLEY
9  I didn’t know you could do that?  Little known secrets of ShopzSeries  
10  Perform amazing new data processing tricks with DFSORT  
   BY FRANK YAEGER
14  Moving right along  z/OS V1R8 migration checker  
   BY MARNA WALLE AND JIM BECKER
16  Is there ESP in your future?  The z/OS Early Support Program  
   BY KATHY KULCHOCK AND LAUREL MORSHAUSER
20  It’s a lock  Evolution of byte range lock manager  
   BY MICHAEL COX AND KERSHAW MEHTA
24  Restart z/OS UNIX—Automagically!  
   BY E. OZAN BARAN AND LISA NOVAL
26  Don’t panic!  Dump them all — zFS, z/OS UNIX, and DFS  
   BY TJ MORRISSEY
28  Refurbishing the z/OS foundation  The new real storage management in z/OS V1R8  
   BY ELPIDA TZORTZATOS AND DIETER WELLERDIEK
33  When you need a timeout  Managing WebSphere Application Server for z/OS timeouts  
   BY DAVID FOLLIS
35  Save the cables: an update  Flapping link threshold recovery  
   BY JOHN BETZ AND BOB FUGA
43  Need help fixing IP storage problems?  
44  GRS at your command  SETGRS enhancements in z/OS V1R8  
   BY CHRIS BROOKER AND BRYAN CHILDS
45  Safety first  zCDP for DB2  
   BY GLENN WILCOCK AND LAURA KUNIOKA-WEIS
48  Opening the window to z/OS  SMB opens a new window  
   BY EYSHA POWERS, JODI EVERDON, AND RENATA RAND MCFADDEN
50  Are you a z/OS master yet?  Take the IBM System z Entry Level for z/OS System Programmer Mastery Test  
   BY KATHY PFEIFFER
52  Send in the clones  New HCM copy wizards can enhance your productivity  
   BY FRIEDRICH BEICHTER
54  Making unforgettable changes  Automatic documentation for IODF updates  
   BY REGINE REICHERT AND FRIEDRICH BEICHTER
56  Gateway to your performance data  RMF Monitor III Data Portal  
   BY HARALD BENDER

Back cover

Need a little help? Try the IBM Education Assistant

Departments

2  A letter from the editors
31  Level 2 with me  The only thing to do with good advice is to pass it on  
   BY EVAN HARUTA AND KATHY PFEIFFER
27  zFavorites CD
58  Our contributors
Spring ahead without falling behind

What Daylight Saving Time changes mean to you

BY KERSHAW MEHTA AND NICHOLAS CARBONE

Would you believe that a simple stroke of the president’s pen could cause IBM to take an APAR? Well that’s exactly what happened when the Energy Policy Act of 2005 was signed into United States law. One of the measures in the bill extends the Daylight Saving Time (DST) by four weeks beginning in 2007.

Currently, DST begins at 2 a.m. on the first Sunday in April, and ends at 2 a.m. on the last Sunday in October. Starting in March 2007, DST will:

- **Spring ahead** at 2 a.m. on the second Sunday in March (March 11, 2007) by moving ahead by one hour.
- **Fall back** at 2 a.m. on the first Sunday in November (November 4, 2007) by moving back by one hour.

The main purpose of Daylight Saving Time, also called Summer Time in many parts of the world, is to make better use of daylight. We change our clocks during the summer months to move an hour of daylight from the morning to the evening. Countries have different change dates based on their latitude.

Is my system affected?

It’s safe to assume that all systems are affected by the DST extension, even if they are operating outside the United States. In fact, several countries besides the United States, like Canada and Bermuda, are also implementing the same DST change; others may follow. Even systems and applications in countries not implementing these DST changes are affected if users, transactions, or applications interact with systems or applications that have implemented these changes.

At this point, you’re probably thinking, “Wait a minute, today on z/OS I use the Sysplex Timer and manually schedule the time change to occur on the fly at the appropriate time or schedule a maintenance window and change the time manually for daylight saving time shift. With the extended DST, it just means that I need to schedule my Sysplex Timer change or maintenance window earlier (or later) in the year.”

This is true; existing processes to change the system time manually still work. However, there’s a function within the z/OS operating system that calculates the US DST algorithmically.

To enable C/C++ applications to specify time zone information, the Language Environment® element of z/OS provides support in the TZ (and _TZ) environment variable. The time zone information allows for time zone names—both standard and daylight saving, offsets from Coordinated Universal Time (UTC) for each, and the details of when daylight saving time starts and ends. When specifying a daylight saving time zone without start and end dates, the US DST rules are applied. z/OS XL C/C++ Programming Guide, SC09-4765, explains the TZ (and _TZ) environment variable and how Language Environment processes time zone information.

---

**TZ=CET-1CEST,M3.5.0/02:00:00,M10.5.0/02:00:00**

*Explanation:*

- **CET** = Central European Time
- **-1** = One hour ahead of UTC
- **CEST** = Central European Summer Time

*Start date/time for DST*

- **M3** = Third Month (March)
- **5** = Represents the last week. Given that a month can have four or five weeks, 5 represents the last week.
- **0** = Sunday
- **02:00:00** = 2 a.m. in Sweden is 1 a.m. UTC.

*End date/time for DST*

- **M10** = Tenth Month (October)
- **5** = Represents the last week.
- **0** = Sunday
- **02:00:00** = 2 a.m. in Sweden is 1 a.m. UTC.

---

**Figure 1. Setting the time zone**

needs to be performed, but on a different schedule. However, there’s a function within the z/OS® operating system that calculates the US DST algorithmically.

To enable C/C++ applications to specify time zone information, the Language Environment® element of z/OS provides support in the TZ (and _TZ) environment variable. The time zone information allows for time zone names—both standard and daylight saving, offsets from Coordinated Universal Time (UTC) for each, and the details of when daylight saving time starts and ends. When specifying a daylight saving time zone without start and end dates, the US DST rules are applied. z/OS XL C/C++ Programming Guide, SC09-4765, explains the TZ (and _TZ) environment variable and how Language Environment processes time zone information.

---

**Keep up with DST rules**

Given that the Energy Policy Act of 2005 changed the US DST rules, the algorithm used in Language Environment is changed. APAR PK24076 provides the new support. After installing the update, the simplest way to activate the new DST rules is to reinitialize the application that was using the TZ (or _TZ) environment variable. For many customers, it may be simpler to relPL the system. This allows
every application that relies on the TZ (and _TZ) environment variable to pick up the new DST rules.

The TZ (and _TZ) environment variable has a very flexible syntax that allows the setting of DST for any location in the world. For example, in the European Union (EU) where all time zones change at the same time, Summer Time:

• Begins at 1 a.m. (UTC) on the last Sunday in March by moving ahead by one hour
• Ends at 1 a.m. (UTC) on the last Sunday in October by moving back by one hour.

Figure 2 shows an example of the TZ setting for a customer in Sweden, which is one hour ahead of UTC normally.

You can use this same method for practicing the US DST rules, especially when your system will not be changing to adopt the extended DST. For locations in the Western Hemisphere that currently follow the US DST rules, no adjustments were necessary, but after the installation of the update, you will now have to set the TZ (or _TZ) environment variable by specifying the explicit start and end dates.

For example, if you were in Guadalajara, Mexico (which is six hours behind UTC normally), you would have previously specified TZ (or _TZ) as:

TZ=CST6CDT

Now, in order to override the default DST, you need to specify the TZ (or _TZ) environment variable as shown in Figure 2.

At the time of writing this article, Mexico has not decided to adopt the extended DST rules. This could change when the article is printed.

One thing to remember
The TZ (and _TZ) environment variable can be set in multiple locations on the system, including configuration files, profiles and shell scripts. Any explicit setting of these environment variables must be performed in all these places.

Finally, if you do not have the opportunity to install APAR PK24076, you can get the same behavior of the extended DST by changing the TZ variable as in the following examples. Assuming you previously used:

TZ=EST5EDT

Now, specify the TZ environment variable as:

TZ=EST5EDT,M3.2.0,M11.1.0

There’s help
Another element of z/OS, Distributed Computing Environment (DCE) Base Services also provides an update with APAR OA18692. As you might guess, the extended DST change affects more than just z/OS. This change has the potential to affect all software and firmware, on all platforms. To assist in this matter, IBM has created the following to help you identify the affected IBM products.

First, a special keyword, DST2007, has been created in RETAIN® to allow you to search for APARs that deliver function related to this change. This keyword is being used to flag APARs on all platforms supported by RETAIN.

Second, a centralized Web site to document all the affected and unaffected IBM products is available at

ibm.com/support/alerts/daylightingsavingstimealert.html.

Finally, a provision in the Energy Policy Act of 2005 states that Congress reserves the right to revert back to the old schedule if reports do not show improvement in energy use. If this happens one day, watch out for new APARs!

Trivia

• The official spelling is Daylight Saving Time, not Daylight Savings Time. Saving is used here as a verbal adjective (a participle). It modifies time and tells us more about its nature; namely, that it is characterized by the activity of saving daylight. Think of nursery rhymes like needle pulling thread. However, many people feel the word savings (with an ‘s’) flows more naturally and hence Daylight Savings Time is also in common usage, and can be found in dictionaries.

• The idea of daylight saving was first conceived by Benjamin Franklin in 1784, but wasn’t practiced until the early 1900s.

• For most practical and legal-trade purposes, the fractional difference between UTC and Greenwich Mean Time (GMT) is inconsequentially small, and for this reason UTC is colloquially called GMT sometimes, even if this is not technically correct.

• Locations near the equator, like the State of Hawaii, generally do not practice Daylight Saving Time. Because day and night are nearly the same length (12 hours), shifting the time by an hour is not very helpful in the tropics.

• A direct result of Daylight Saving Time is the savings in energy. A large percentage of energy consumed by lighting and appliances occurs in the evening when families are home. By moving the clock ahead one hour, the amount of electricity consumed each day decreases.

More trivia can be found on pages 43 and 49
Remember the story about the thirteen-foot python that was found in the Florida Everglades? It burst while attempting to eat an alligator that was more than half its length. Well, until recently, z/OS had a similar problem with console messages (and its handling of them was just about as pretty).

Before console restructure in z/OS V1R4, z/OS and its predecessors were like that python. When z/OS received a normal stream of messages, things went in one end and out the other, and there were no problems. Once in awhile, an alligator, elephant or camel’s worth of messages would happen along, and z/OS (like that unsuspecting python) would attempt to process all of them — with similar, unfortunate results.

A cast-iron stomach for z/OS
With the z/OS V1R4 console restructure, we fixed the problem — we gave z/OS console support a really resilient “stomach” in the form of the console message cache, a data space that contains a circular queue of all of the messages that are waiting to be queued to consoles.

Quite simply, it is no longer possible to “burst” z/OS’s stomach because a circular queue has infinite capacity. It is possible for the queue to “wrap” before all of the messages have been removed and queued to consoles, but this wrapping takes days on even the fastest machines. And yes, it is still possible to get into trouble with the 24-bit below-the-line storage that multiple consoles support (MCS) and subsystem MCS (SMCS) consoles use, although this is much less likely.

So the good news is that z/OS can now swallow a gargantuan number of messages without bursting itself. The bad news is that z/OS now needs to “digest” that big bulge of messages — and process all of them — then somehow pass them out the other end without overwhelming the consoles, operators and automation programs in messages they don’t want to see.

Controlling an appetite for messages
Enter message flood automation, new, specialized, built-in automation that attempts to determine when a message flood is occurring, and attempts to prevent z/OS from ever swallowing all those messages in the first place.

In the past, most automation that dealt with message floods was “after the fact” — dealing with the consequences of the flood (buffer exhaustion and excessively long console queues) rather than with the cause. By dealing with the cause, message flood automation obviates the need to deal with the consequences. With message flood automation, there should be no residual buffers or queues of messages that have to be “worked down” to return to normal processing. Furthermore, there is no need to take action on every console to “flush” unwanted messages (often discarding important messages along with those you don’t want to keep). Message flood automation attempts to minimize disruption by targeting only the offending message or offending address space.

Message classes
Message flood automation divides messages into three classes:
• SPECIFIC messages: a list of message IDs that the installation can specify
• ACTION messages: messages that require the operator to do something
• REGULAR messages: all other messages.

Message flood automation has independent controls and policies for each of these message classes.

Message flood automation modes
Message flood automation has two modes of operation: “normal mode” and “intensive mode.” In normal mode, message flood automation monitors the rate at which messages are being produced within each message class. Using an installation-defined policy, it attempts to identify when a message flooding situation is underway. If the rate at which messages are being produced in a class indicates that a message flooding situation is underway, message flood automation switches into intensive mode for that message class. Once in intensive mode, message flood automation attempts to identify the specific message ID or address space that is the source of the message flood. Message flood automation can track up to 30 SPECIFIC message IDs and up to 10 address spaces each for ACTION and REGULAR messages.
Once it has determined that a message flood is underway and that it needs to take action, message flood automation uses an installation policy to determine what it should do with the unwanted messages. Message flood automation can suppress a message from display, prevent the message from being automated, and prevent the message from being logged. If it is an ACTION message, message flood automation can prevent the message from being retained by the action message retention facility (AMRF). It can also delete messages and prevent them from being propagated to other systems in a sysplex and even has the ability to issue a command against the job that is causing the message flood, allowing the job to be automatically cancelled.

While it is critical to ascertain when a message flood is beginning, it is equally critical to ascertain when the message flood has ended — and to cease taking action against messages and jobs as quickly as possible to minimize the loss of involved messages. To do so, message flood automation monitors the time between successive flood messages and stops taking action when there is a sudden increase in the time between messages.

You can enable message flood automation very early during z/OS initialization, which will allow it to handle things like IOS message floods resulting from a Geographically Dispersed Parallel Sysplex™ (GDPS®) HyperSwap™. During early IPL, the system services needed to read PARMLIB do not exist, so message flood automation has a built-in policy for handling these early message floods. Later, when the IPL is complete and the system services needed for reading PARMLIB exist, an installation-unique policy can be read.

Message flood automation provides a very flexible policy override mechanism. In the absence of any installation-specified policy, message flood automation uses its own built-in policy. When reading the installation-specific policy from PARMLIB, it can override any of the built-in policy parameters. Within the installation-specific policy, policy can be established at the message class level and can then be overridden for a specific set of message IDs and jobs. While an MSGFLDxx PARMLIB member is the principal source of the message flood automation policy, you can amend the policy or establish it dynamically through a complete set of operator commands.

Message flood automation has a built-in “message rate monitoring facility” that can be turned on to gather information about the message rates on a system. You can produce a graph that shows the characteristics of the message traffic and the recommended threshold values for a policy.

Message flood automation has been available to z/OS GDPS customers for more than three years and used by GDPS customers to stem the message floods that can occur when performing a DASD volume HyperSwap. It is now part of the z/OS consoles environment and is available to all z/OS customers through Small Programming Enhancement (SPE) OA17514, which is available for z/OS V1R6, V1R7, and V1R8.
The latest thrilling celebrity hook-up

**DB2 and DFSORT!**

**BY R. DAVID BOENIG II**

There’s a hot new couple on the z/OS scene, and it isn’t Brad and Angelina. Among the many new features and improvements in DB2 Universal Database™ (UDB) for z/OS Version 8 is its exclusive pairing with DFSORT™, for use with the DB2® utilities suite of products.

Previous versions of DB2 required your installation to license a sort product to handle sorting and merging by the DB2 utilities. In Version 8, however, DB2 uses DFSORT—regardless of whether you are licensed for DFSORT.

This new interface was incorporated into the base for z/OS V1R5. If you are running z/OS V1R4 or earlier, you must apply DFSORT PTF UQ90054.

**Why this is good news**

The exclusive use of DFSORT by DB2 utilities helps you to leverage the strengths of both products. The DB2 and DFSORT development teams are working together on enhancements to improve mutual reliability and performance. We now use 32K pages for sorting data, for example.

IBM is committed to further improving the interoperability of these two products. For example, IBM has already provided the following PTFs for DB2 and DFSORT:

- PK13092 for DB2, which allows DB2 to recognize a DSA value greater than 99 megabytes
- PK14477 for DB2, which increases the degree of parallelism possible in the SORTBLD phase
- PK24768 for DB2, which allows DB2 to calculate average row length more accurately than before
- PK01155 for DFSORT, which improves coordination of parameters with DB2
- PK28879 for DFSORT, which corrects a LOOP or ABEND S0C4 when a large main storage value is specified
- PK25047 for DFSORT, which improves the DFSORT algorithm for using multiple hsuperspaces.

**Which DB2 utilities use DFSORT?**

To perform various sort and merge functions, the following DB2 Version 8 utilities use DFSORT exclusively:

- LOAD
- REORG TABLESPACE and REORG INDEX
- REBUILD INDEX
- RUNSTATS
- CHECK DATA, CHECK INDEX, and CHECK LOB.

**What are the setup considerations?**

DFSORT is included with z/OS, so you do not need to order it or install it. If DFSORT is already defined as your primary sort product, you have nothing more to do to make it available to DB2 utilities. If not, you will need to do some setup work to allow DB2 utilities to execute DFSORT while your other applications can continue to use your primary sort product. For example, you will need to make the DFSORT libraries available in the order of program call by adding them to the link list or LPA list, or both, as needed.

For more information about setup, see informational APAR I114047, which is available at [ibm.com/support/docview.wss?uid=isg1I114047](http://ibm.com/support/docview.wss?uid=isg1I114047).

The APAR covers these topics and more:

- Allowing DB2 utilities to provide accurate information to DFSORT
- Installing DFSORT if it is not your primary sort product
- Tailoring DFSORT installation defaults to your environment
- Verifying that ACS routines will not direct sort data sets to VIO
- Excluding sort data sets from any OEM products that reduce space allocations.

Also, if you are setting up DFSORT to run in a 64-bit environment, have a look at DFSORT informational APAR I113495, which describes how to take advantage of 64-bit z/Architecture™, and provides additional DFSORT migration and tuning recommendations. You can find this APAR at [ibm.com/support/docview.wss?uid=isg1I13495](http://ibm.com/support/docview.wss?uid=isg1I13495).
I didn’t know you could do that?

Little known secrets of ShopzSeries

Q: How can I view my licensed products?
A: First, log into ShopzSeries. Once you’re logged in, you can find the information in two ways:
  • Click My licensed software in the left navigation pane.
  • If you are in the process of placing an order, click Show licensed products to display only licensed products.

Q: I don’t want any materials physically shipped to me. How do I ensure 100% Internet delivery?
A: Be sure to click Do not ship any physical materials, I require 100% Internet delivery on the Specify Delivery Options page.

Q: How do I get started with ShopzSeries?
A: On the ShopzSeries home page, click a link in the sentence: “You can learn more about ShopzSeries by reading the online users’ guide, or better yet, relax and watch the instructional video clips.”

Q: How can I get a head start on my ServerPac order?
A: When placing your ServerPac order, click Upload a new report for this order on the page Step 3 of 8 Report installed software. By clicking the link, you create an installed inventory report. This report is used to populate your new ServerPac order selections based on your existing products.

Q: How do I find out what the FMIDs are for this product?
A: Click FMIDS for a product in the product catalog. For example:

```
[5655-N01] WebSphere for z/OS - DVD [FMIDs] 6.01.00 English (US)
```

Q: How can I find out when service will be discontinued for a product?
A: Click the product ID link for a product in the product catalog. For example:

```
[5655-N01] WebSphere for z/OS - DVD [FMIDs] 6.01.00 English (US)
```

Q: Can I download my order directly to my mainframe?
A: Yes! Go to Help > Internet Delivery to find out.

Q: How much space do I need to download this order?
A: The required download space estimates are included with the related installation materials that are displayed on the ShopzSeries download page for the offerings.

Q: My order hasn’t arrived yet! How do I find out where my order is?
A: Check your order status on the in process or completed tabs on the My Orders page. Still no luck? Contact ShopzSeries support by clicking CUSTOMER SERVICE in the left navigation pane.

DB2 uses DFSORT—regardless of whether you are licensed for DFSORT.

What are the service considerations?
These changes mean that you need to include DFSORT in your usual service procedures. IBM recommends that you specify DFSORT in the group of FMIDs that you monitor for service. If you use Enhanced HOLDDATA, DFSORT HIPERs will show up in the REPORT ERRSYSMODS listing. If you have tailored notifications at the FMID level, such as ASAP in ServiceLink, you should add notifications for DFSORT HIPERs.

Also, if one of your DB2 utility runs encounters a DFSORT error message (for example, ICE039A or ICE046A) or an abend, you should open a PMR with DB2 Support. If necessary, they can contact DFSORT for assistance. If you can reproduce the problem, add SORTDIAG and DFSPARM DD statements to the failing job, as shown below, and rerun it:

```
//SORTDIAG DD DUMMY
//DFSPARM DD *
DEBUG ABEND
/*
```

Doing so causes DFSORT to display additional diagnostic (ICE8xxx and ICE9xxx) messages, and to abend if a DFSORT error message is issued. Send the complete JOBLOG and dump (if an error occurs) to the appropriate IBM service representative.

Together at last
Let’s hear it for this dynamic duo — DFSORT and DB2, a perfect match!
Perform amazing new data processing tricks with DFSORT

BY FRANK YAEGER

Although DFSORT has been part of z/OS and OS/390® (…and MVS®) for decades, we continue to add amazing new data processing tricks to this popular act. You already knew, for example, that DFSORT is quite adept at helping you to sort, merge, and copy data (either directly using batch jobs, or indirectly from programs). You probably also knew that DFSORT can include, reformat, and sum records, and produce reports.

But did you know that DFSORT has many new tricks up its sleeve, such as the ability to perform these feats, and much more:

- Parse delimited fields
- Justify data
- Test for numeric values
- Reformat different records in different ways
- Overlay specific parts of records
- Handle numeric values that have separators and decimal points.

In this article, I share with you some of my favorite DFSORT tricks, using simple examples of DFSORT control statements. For complete details, see the DFSORT product publications at [ibm.com/storage/dfsort].

Abracadabra! Overlay specific fields

Previously, if you wanted to change just one value in a record, you had to specify all of the following data:

- The bytes before the record
- The bytes you actually wanted to change
- The bytes after the record.

For multiple fields, this meant having to figure out all the unchanged bytes between the changed bytes. No more. With the DFSORT OVERLAY function, you can specify just the changes to the fields of interest and you can overlay multiple fields in any order.

You can use OVERLAY to specify any reformatting item allowed with the INREC, OUTREC or OUTFIL control statements. Earlier overlays are “seen” by later overlays, which means that you can change the same field multiple times in succession.

Figure 1 shows an INREC statement that uses OVERLAY to write ‘Betty’ in positions 21 - 25, and a zero packed decimal (PD) value in positions 11 - 13 of every record, without changing anything else in the record.

For multiple fields, this meant having to figure out all the unchanged bytes between the changed bytes. No more. With the DFSORT OVERLAY function, you can specify just the changes to the fields of interest and you can overlay multiple fields in any order.

You can use OVERLAY to specify any reformatting item allowed with the INREC, OUTREC or OUTFIL control statements. Earlier overlays are “seen” by later overlays, which means that you can change the same field multiple times in succession.

Figure 1 shows an INREC statement that uses OVERLAY to write ‘Betty’ in positions 21 - 25, and a zero packed decimal (PD) value in positions 11 - 13 of every record, without changing anything else in the record.

Figure 1. OVERLAY makes it easy for you to change only specific fields.

INREC OVERLAY=(21:C’Betty’, 11:+0,TO=PD,LENGTH=3)
Poof! Reformat different records different ways

Wouldn’t it be great if you could change different types of records in different ways? Well, now you can. Use the new DFSORT IFTHEN clauses, which you can specify on the INREC, OUTREC, or OUTFIL control statements.

On each IFTHEN clause, you specify an action to be taken (BUILD or OVERLAY) whenever a certain condition is met (WHEN). DFSORT offers you a choice of several types of IFTHEN clauses, including the following types:

- **WHEN=INIT** clause, which applies the action to all of the records.
- **WHEN=(logexp)** clause, which applies the action to only records that meet specific criteria. Specify criteria using the same logical expressions you use for the INCLUDE statement.
- **WHEN=NONE** clause, which applies the action to only records that do not meet the criteria.

In Figure 2, note the use of OVERLAY to overlay specific fields, and the use of BUILD to build a record field by field.

By the way, you can use BUILD as an alias for these parameters:
- **FIELDS** in an INREC or OUTREC statement
- **OUTREC** in an OUTFIL statement.

By allowing you to employ conditional logic, IFTHEN helps you avoid having to make multiple passes over the same data.

Voila! Test for numerics or non-numerics

Want to identify those annoying non-numeric values in numeric fields? Nothing to it. Just use the DFSORT NUM function, which lets you check character, zoned decimal or packed decimal numerics (EQ) or non-numerics (NE), respectively.

Figure 3 shows two OUTFIL statements that use NUM to examine zoned decimal (ZD) values in positions 31 - 36. The first OUTFIL statement writes records with non-numeric ZD data to the BAD file. The second OUTFIL statement writes records with numeric ZD data to the GOOD file. You can use FS, ZD, or PD format with NUM to check for character, zoned decimal or packed decimal numerics (EQ) or non-numerics (NE), respectively.

Presto! Justify and squeeze data

Sometimes data is just not right (or left) when it should be. You can use the DFSORT JFY (justify) and SQZ (squeeze) functions in INREC, OUTREC and OUTFIL statements to align your data as needed:

- Use JFY to shift your data left or right, remove blanks or other characters at the beginning or end of your data, and add leading or trailing strings.
Formats with strict rules, such as zoned decimal, cannot handle these values properly for sorting, merging, comparing, editing, arithmetic, and so on. However, the DFSORT unsigned free form (UFF) and signed free form (SFF) formats have no trouble with fancy numeric data. You can use them to ignore everything except the digits (UFF), or the sign and digits (SFF), so that values can be handled properly.

Figure 5 shows how you might use SFF in SORT and OUTFIL statements to sort and total a fancy numeric field in positions 17-25.

Hocus-pocus! Parse delimited fields
Fixed fields in records are no longer the only game in town for z/OS data. These days, we also see many delimited fields, such as comma-separated values (CSV), tab separated values, keyword fields, null terminated strings, and so on, in files exported from other platforms, and from applications like spreadsheets and financial programs.

DFSORT has many functions for dealing with fixed fields, but what about those delimited fields? No worries. You can use the DFSORT PARSE function in INREC, OUTREC and OUTFIL statements to extract delimited fields into fixed fields. You can then use all of DFSORT’s fixed field functions for the extracted fields.

Using PARSE, you can define the rules for extracting up to 100 delimited fields into parsed fields named %00 - %99. The syntax for defining the rules is quite flexible. For example, you can use PARSE to tell DFSORT:

• To start a delimited field at or after a specific string
• To end a delimited field before or at a specific string
• Whether to look for the delimiter between apostrophes or quotes
• The length of the fixed field to contain the extracted field.

You can even use % instead of %nn to tell DFSORT to ignore a field of no interest.
Figure 6 shows an INREC statement that uses PARSE to extract the first, third and fourth fields of four CSV fields and reformats them in various ways as fixed fields. You can use each %nn parsed field in the same way that you would use a p,m fixed field in BUILD and OVERLAY. You can even use PARSE in IFTHEN clauses.

```
INREC PARSE=(%01=(ENDBEFR=C',',FIXLEN=6),
            %=(ENDBEFR=C','),
            %03=(ENDBEFR=C',',FIXLEN=8),
            %04=(FIXLEN=10)),
BUILD=(%04,2X,%03,UFF,EDIT=(IIIIT.TT),
       2X,%01,TRAN=LTOU)
```

If you are not familiar with DFSORT, I suggest reading through z/OS DFSORT: Getting Started. It is an excellent tutorial for all of the DFSORT functions, as well as the DFSORT ICETOOL utility and DFSORT symbols, with lots of examples. You can access it online, along with the other DFSORT books at ibm.com/servers/storage/support/software/sort/mvs/srtmpub.html.

Also, see my Smart DFSORT Tricks paper at ibm.com/servers/storage/support/software/sort/mvs/tricks for clever ways to use the DFSORT functions described here (and many others) to solve common problems.

Figure 6. PARSE extends DFSORT’s fixed field functions to delimited fields.

Sim sala bim! Split a file
Breaking up is not hard to do with DFSORT. You can use the DFSORT SPLIT1R=n function on the OUTFIL statement to write n consecutive records to different output files. Here, you specify the number of records (n) for each output file, and the number of output files. The first n records go to output file1; the second n records go to output file2, and so on. Any remaining records go to the last output file, so that all of the output files contain consecutive input records.

Figure 7 shows an OUTFIL statement that uses SPLIT1R to write input records 1 - 1000 to OUT1, input records 1001 - 2000 to OUT2, and input records 2001 - last to OUT3.

```
OUTFIL SPLIT1R=1000,
        FNAMES=(OUT1,OUT2,OUT3)
```

Figure 7. SPLIT1R makes it easy to break up a file.

Learn the other tricks of the trade
For complete information on DFSORT, see the DFSORT Web site at ibm.com/servers/storage/dsort (click “Product details”).

Here are just a few of the many tricks you will find there:
- Split a file to n output files dynamically
- Extract and justify delimited fields
- Create files with matching and non-matching records
- Deconstruct and reconstruct CSV records
- Include or omit groups of records
- Find and extract values from different positions
- Include records using relative dates
- Sort records between a header and trailer
- Change all zeros in your records to spaces
- Join fields from two files on a key.
Moving right along

z/OS V1R8 migration checker

BY MARNA WALLE AND JIM BECKER

If you think that the z/OS Migration book was a big step in the right direction when we introduced it back in z/OS V1R4, but it’s time we took another step, do we have something for you! It’s a tool called the IBM Migration Checker for z/OS.

Exactly what is the IBM Migration Checker for z/OS?
The IBM Migration Checker for z/OS is an “as is” download tool available from the z/OS download Web page at ibm.com/servers/eserver/zseries/zos/downloads/. It is composed of several batch programs (which can run either independently or serially in one job) that check the applicability of certain migration actions on the currently running system.

Under no circumstances will the IBM Migration Checker for z/OS perform migration actions on your system. The tool will only report on which migration actions you have to perform. The value of the tool lies in helping you determine the applicability of some migration actions on your system, and verifying that those migration actions are done correctly.

The tool is best used in conjunction with the z/OS Migration book, to assist you in creating your migration plan.

Which migration actions does the IBM Migration Checker for z/OS help me with?
We’ve selected a set of migration actions for this tool that will assist you in your migration from z/OS V1R7 to z/OS V1R8. Because of the timing of the release of the tool, we expected that many customers who were migrating from z/OS V1R4 to z/OS V1R7 would be well through their migration plans. Helping with a subsequent migration (from z/OS V1R7 onward) was where we could introduce this tool with the maximum usage potential.

However, don’t fret if you’re not on the z/OS V1R7 to z/OS V1R8 migration path! Most of the programs in the tool provide value on other migration paths (such as z/OS V1R6 to z/OS V1R8, and z/OS V1R4 to z/OS V1R7). The output reports you get from running the tool will tell you whether the tool was able to do any analysis on the currently running system, so running it outside the intended migration path does no harm, and in many cases is helpful.

An early (alpha) version of the IBM Migration Checker for z/OS performs the following checks. Note that the names of the checks are the same as the corresponding titles in the z/OS Migration book.

• Run the TN3270E Telnet server as a separate address space.
• Migrate from HFS file systems to zFS file systems.
• Discontinue use of multi-file system aggregates.
• Use the new default for BLOCKTOKENSIZE in IGDSMSxx.

By the time you read this, other checks have no doubt been added to the latest version of the tool. Some other checks we are looking to include:

• Add references to new data sets and paths.
• Remove deleted data sets, paths, and references.
• Accommodate the removal of 1-byte console IDs.
• Update /etc z/OS UNIX® and Communications Server configuration files changed by IBM.
• Ensure the integrity of SMS control data sets.
• Redefine the existing VSAM data set that contains the IMBED, REPLICATE, and KEYRANGE attributes.
• Ensure that UDP port 514 is available to syslogd if not started with the -i option.

Of the many migration actions in the z/OS Migration book, how did we choose which ones to include in the tool? We surveyed a number of clients, settled on a priority list, and determined which migration actions on the list could be checked programmatically. Some of the migration checks have several parts to them — some pieces that are programmatically possible and others that aren’t. In the latter case, the output report for the check indicates what was checked and states which manual steps are still required by you.
What are the requirements for running the IBM Migration Checker for z/OS?

There is very little in the way of requirements to run the tool! The only requirement is that the user who runs the programs has to have the authority to issue the TSO/E CONSOLE command.

How do I set up and run the IBM Migration Checker for z/OS?

Getting the IBM Migration Checker for z/OS up and running is very simple. Here’s an outline of the steps:

1. Download the three files that compose the tool from the z/OS download Web page. Upload them to your z/OS system as binary.
2. On z/OS, enter TSO RECEIVE INDATASET for each of the three data sets. This restores the data sets to their original format. You must keep the middle- and low-level qualifiers of MIGRATE.CHECKER.xxx. However, you can make the high-level qualifiers whatever you want.
3. Edit hlq.MIGRATE.CHECKER.CLIST($JOBCARD) to change the default job statement to be appropriate for your environment.
4. Enter TSO EXEC or EX 'HLQ.MIGRATE.CHECKER.CLIST($SETUP)' to customize the JCL for your particular system, create system-specific JCL and OUTPUT data sets, and add a new member in your CLIST data set called $sysname.
5. Edit hlq.MIGRATE.CHECKER.CLIST($sysname) to provide any additional information you would like the tool to use. For instance, you can provide the z/OS SMP/E CSI and zone names for your currently running system, and do additional verification in the CSI. Also, you can provide the names of up to eight TCP/IP and eight Telnet profiles, for either active or inactive stacks. You don’t have to provide the IBM Migration Checker for z/OS any information at all, but if you choose to provide this input, more analysis can be done for migration actions.
6. Submit the job hlq.MIGRATE.CHECKER.JCL($MIGALL) to run each of the programs serially.
7. Review the output in data set hlq.MIGRATE.CHECKER.OUToutput. There is one member for each migration program’s output, and a consolidated member ($MIGALL) that contains a one-line result for each program. Correct any exceptions that are identified and rerun at will!

When should I run the IBM Migration Checker for z/OS?

You can run the tool as often as you like — on z/OS V1R7 before you’ve even ordered z/OS V1R8, after installing z/OS V1R8, and after deploying each z/OS V1R8 image on your enterprise. Because the tool does not make any changes to your system and because it is aware of the level of the system on which it is running, it can:

• Alert you to migration actions that you should plan for in the future
• Detect migration actions that you can do now to clear the way to z/OS V1R8
• Verify that you’ve done the migration actions correctly when you’re running z/OS V1R8.

The value of the tool lies in helping you determine the applicability of some migration actions on your system, and verifying that those migration actions are done correctly.

Many of the items that the tool checks for can be changed dynamically on your currently running system. As configurations change, you should rerun the tool. For instance, if your enterprise has multiple TCP/IP stacks, and activates and deactivates those stacks at different times, it is beneficial to run the tool multiple times to analyze each active TCP/IP stack.

You can set up your automation to submit IBM Migration Checker for z/OS jobs on a regular schedule that fits your needs, or you can manually submit the jobs. The frequency is up to you. Just keep in mind that only the latest report will be in the output data set. If you want to save the output from previous runs (if you even see a need to save them!), you can include that in your automation steps too.

Why are the programs in the IBM Migration Checker for z/OS individual batch jobs?

Why not just provide checking programs using the IBM Health Checker for z/OS?

Because our intended migration path for this tool was z/OS V1R7 to z/OS V1R8, we reviewed the IBM Health Checker for z/OS V1R7 for our purposes. Unfortunately, the function we needed in IBM Health Checker for z/OS — specifically the remote checking capability — was not available, and would not be made available, on z/OS V1R7.

What should I do if I find any problems running IBM Migration Checker for z/OS?

This tool is not supported by the IBM Service organization, but rather on a best-can-do basis. Please report any bugs, suggestions, or comments to the “IBM Migration Checker for z/OS” forums on IBM Customer Connect. For access to the forums, send an e-mail to zosmig@us.ibm.com.

What about all the migration actions in the z/OS Migration book that the IBM Migration Checker for z/OS doesn’t help with? What do I do about them?

Migration actions in the z/OS Migration book that are not covered by the programs in the IBM Migration Checker for z/OS tool still must be reviewed by you. Unfortunately, our resources didn’t allow us to write a program for every migration action in the book. However, we feel this tool is a step in the right direction and hope you share our sentiment.

What are you waiting for? The IBM Migration Checker for z/OS can help drive away some of your migration blues. Climb on board!
Is there ESP in your future?

The z/OS Early Support Program

BY KATHY KULCHOCK AND LAUREL MORSCHHAUSER

What exactly is ESP? Does it stand for Extra Special Person? Can it mean Extra Sensory Perception? It could, but within IBM, it stands for Early Support Program. If you have been involved with z/OS products, you have probably heard about ESP and you might have even participated in one. If not or if it has been awhile, this article will provide an overview of the ESP. IBM offers an ESP for products across all its brands, but this discussion is limited to the z/OS ESP.

What is ESP?

Back to the original question, “What exactly is an ESP?” ESP is an opportunity for customers to install, test, use and comment on IBM products and solutions before they are generally available. ESP participants receive an advance copy of the newest release of z/OS to install and run the product in their current environment. The intent is to validate that the current z/OS functions continue to run as expected and to gain user experience on new functions in the z/OS release. The ESP validation is not limited to the new z/OS code; it also includes ordering, delivery processes, documentation, and IBM Support. ESP participants use the new z/OS release as if it was already available in the marketplace, but in a monitored environment.

Developing ESP

The z/OS ESP process starts approximately nine months prior to the general availability of a new release. The first step is to collect nominations from IBM customer reps and select prospective ESP participants based on z/OS prerequisites for the new release. Other factors for selecting customers for an ESP are a willingness to meet an aggressive installation and implementation schedule, willingness to act as a customer reference and past participation in an ESP. The next step is to present the details of the program to an ESP candidate and gain commitment. If you want to be an ESP participant and agree to the terms, an ESP representative makes sure that you receive access to various online tools, support structure, and ESP education. A z/OS education session is held for ESP participants in Poughkeepsie, NY. This weeklong session is given at least one month in advance of receiving the ESP code. The education provides details on what’s new or changed in the z/OS release. It’s intended for experienced z/OS users.

ESP participants have the option of scheduling an Installation Plan Review (IPR) with an IBM Advanced Technical Services representative. This session is set up as a conference call and reviews migration information, previously provided by the participant, to identify concerns or special considerations for the installation of the new z/OS release.

Approximately three months before planned general availability, ESP participants get the new z/OS code and documentation. This code is built to the customer’s request, as provided in the z/OS order checklist. At this point, ESP participants follow their individual implementation and test plans for exercising the new z/OS release. During the implementation and test period, weekly status calls with an ESP representative are conducted to collect feedback, discuss open problems and other issues the customer might have.
The ESP ends one to three months after the new z/OS release is GA, depending on the participant’s schedule. At this point, we distribute an exit questionnaire to gather feedback on the z/OS release, the supporting infrastructure and the ESP as a program. Total duration for an ESP program is around ten months. For ESP participants, actually hands-on usage time with the code is about four months.

The program provides customers with an early look at the release and an opportunity to influence development’s actions, while z/OS developers get an early view of how customers will use the new release.

Who benefits from ESP?
ESP is good for the participating customer and for IBM. The program provides customers with an early look at the release and an opportunity to influence development’s actions, while z/OS developers get an early view of how customers will use the new release. Participants can comment on how things worked out in their environments, while IBM can better prepare to support the new release. Because ESP environments provide different systems configurations, problems might be found that z/OS testing wouldn’t normally expose, and we can jointly make sure the new release works in these environments. Because these scenarios provide earlier knowledge about the new product, customers have a chance to provide input while IBM has the opportunity to change the product before general availability.

In addition to providing early product to participants, ESP is also a relationship program. For this timeframe, participants have multiple and frequent opportunities to speak with the IBM development team. Some of these opportunities are through:

- ESP forum for Q&A and broadcast communications
- Weekly status calls with an ESP representative
- Formal feedback reports
- Direct contact with z/OS developers.

ESP participants also regularly submit requirements. All of these vehicles give IBM an opportunity to respond to customer feedback and affect customer satisfaction.

Do you have ESP?
ESP participation has increased since the first z/OS program. ESP participants see value in the program because early experience with the product can provide them with competitive edge by leveraging new function and enhancements on or before general availability. The ESP education provides a faster learning curve to the new release. Early availability of z/OS documentation also contributes to shortening the learning curve.

During the ESP, customers report problems through the normal IBM Support structure. However, ESP problems have special handling to provide additional focus for quicker resolution.

ESP provides each participant with a dedicated representative. The ESP representative can arrange for appropriate resources to assist with any concerns that can arise. The ESP also offers some coordination with Independent Software Vendors (ISVs) to assist their products’ preparedness for running on the new z/OS release as soon as possible.

As previously mentioned, IBM offers multiple feedback vehicles during ESP. Customers appreciate the opportunity to give IBM an opinion on the new z/OS release and view it as a chance to influence future product directions.

These are just some of the features that have brought customers to repeat the ESP experience.

Get ESP
The IBM Systems and Technology Group Product Introduction team in Poughkeepsie, NY manages the z/OS ESP. If you would like more information on how you can participate, please contact Sarah Angel at sangel@us.ibm.com.
Overlooking something?

Seven tips for improving your system tests — and your luck

BY JOEL MASSE AND NANCY FINN-DENAPOLI

System maintenance upgrades, application maintenance upgrades, hardware upgrades, system parameter changes … the list goes on and on. If you are the Test or Quality Assurance person at your company, the changes never end.

As with inspecting your car before a long trip, you need to check the test system often and give it regular “tune-ups.” This is especially important prior to major system changes, such as migrating to a new processor or a new release of the operating system.

If, as we know, the primary purpose of a test system is to prevent problems in the production system, we need to be sure that our testing methods are as effective as they can be. When viewed as a “lean, mean, problem-finding machine,” how well does your company’s test system compare to its production system?

Here are seven tips for you, the test person at a large z/OS installation, from two experienced z/OS system testers.

Tip #1: Keep it real
While it’s advisable to keep your test sysplex separate from your production sysplex, you should configure Test to match Production as much as (reasonably) possible. Include things like processors, channels, DASD, sysplex organization, and so on. Pay close attention to anything having to do with concurrency and sharing, such as data sharing and catalog sharing. If your production sysplex shares a catalog, for example, you should duplicate that sharing in the test sysplex.

We do not, however, recommend sharing a catalog between Test and Production.

If your test sysplex has processor or DASD models that differ from the production sysplex, consider those differences when you tune or adjust your test workloads.

Also, include any relevant product and application installation procedures in your test cases. To help ensure a successful production launch, document the tasks you used to install and set up the test sysplex.

By the way, did you know that z/OS Health Checker can help you ensure that various system settings used in Test mimic those used in Production? If you start z/OS Health Checker in both environments with the same check parameters, you can compare the Test and Production exception reports. Any difference in check exceptions can help you identify settings that might be different between the two environments. Eliminating the differences can help you ensure that the systems match.

Tip #2: Sing a different tune
When it comes to creating workloads for the test system, don’t just clone your production workloads. To enable improved portability and tunability, you need to simulate real workloads.

After all, the baseline is just the beginning. Simulation gives you an opportunity to broaden the scope of your testing. For example, by varying the number and types of tasks throughout a test run, you help to simulate changing transaction patterns.

If your test workloads have enough dials and knobs, you can adjust them to simulate production in ways that cover all the bases.

Through simulation, you have the ability to define parameters for many aspects of the production workload. As part of your testing, you might try parameterizing certain data set attributes, like block size or control interval (CI) size.

If you are testing a VSAM application, for example, try using a small CI size, which tends to create more CI splits and can expose problems more quickly.
Also, by substituting subtasks in a test batch program, you help to simulate production transactions coming from remote terminals.

Testing a new application with poorly tuned files can help you uncover problems that would occur in production under stress. In the case of CICS/VSAM, experiment with the parameters for buffer sizes. By intentionally running with inefficient buffer pools, you might expose potential problems early. Similarly, for an IMS or DB2 application, parameterizing the data set definitions will allow you to test under a wider variety of conditions and thus, increase your chances of finding bugs.

Tip #3: Ain’t broke? Break it!

In real estate sales, the three most important factors are “location, location, location.” In testing, it’s probably “variation, variation, variation.” While stability might be the most important factor in a production system, a successful test system relies on you to push the envelope. Make use of randomness, asymmetry and error injection. For instance, ensure that your direct access key test pattern is truly random. As an example of asymmetry, try having two different release levels of the system in the same sysplex, sharing a catalog.

There are always certain predictable errors. Could they cascade and bring down the whole application or system? To catch more errors, include error-testing scenarios, like canceling a data set listing in ISPF 3.4.

Also, try doing some physical tests, like removing channels or links to key devices. Or, create coupling facility structure errors with the INJERROR tool, which you can find on the z/OS Integration Test Web site at www.ibm.com/servers/eserver/zseries/zos/integtest/injerror.html.

On the other hand, do not waste time and money testing the same thing repeatedly. If it works, move on to look for problems somewhere else. And, if it does fail, change it to get different failures while the problems are being fixed.

If it ain’t fixed, don’t test it.

Tip #4: Get stressed out, totally

Drive the test system’s activity level off the chart. Exceed production transaction and throughput rates to provide a “margin” in production. Push at least to the point of performance degradation. Experience gained here can help with capacity planning later. Stress the system by exceeding the production contention levels and by adding resource constraints.

This information can help you determine where to spend more effort in improving your testing. In particular, it will help you to ensure that you are testing the error paths and paths for unusual conditions.

To enable improved portability and tunability, you need to simulate real workloads.

For example, drive up ENQ and coupling facility activity. Constrain the system by including “background noise” that consumes resources such as virtual storage, real storage, CPU, or DASD.

Here is something else you might try: vary offline all CPUs, except for one on a single image. Does the system back up? What damage might this cause in a production environment?

Also, how about setting up small coupling facility structure sizes and ramping up your workload to force auto-alter situations? Setting up smaller log steams sizes will slow performance as well. Does the application-under-test handle the added stress?

Tip #5: Hope for the worst

Do not assume optimal performance from your applications. Besides testing production-like settings for optimal performance, use different settings to cause bottlenecks. Doing so helps you see what might happen in Production under unusual load.

It is in the test system — not production — that you should learn what will happen, for example, if the transaction arrival pattern causes degradation. Will your applications just slow down, or will they fail altogether?

Tip #6: Don’t let sleeping dogs lie

Do code coverage analysis. Several commercially available tools are available to help you see which sections of the application code are being exercised insufficiently—or not at all.

Tip #7: Go looking for trouble

Recreate known problems. If your test system cannot recreate problems found on the production system, it cannot give you the protection you need.

If your test workloads have enough dials and knobs, you can adjust them to simulate production in ways that cover all the bases. Try recreating problems described in APARs before applying the PTFs as a trial of the test system’s effectiveness. Add problem recreations to your regression bucket.

The point is to continually learn from production experiences and enhance the test environment based on them. You can’t always start off with a large, robust test environment, but you can use what you’ve got, and strive to get better every day!

Testers of the world — “unite!”

We hope these tips improve your luck during your next test experience. Always, review your workloads and your environment to find glaring problems. Do your older applications still run the way you expect them to? Be aware of any glitches in your setup before adding those outside changes.

Some day, perhaps, testers everywhere will unite under the rallying cry, “Shake it ‘till you break it!”

One final thought:

Look before you leap
So you won’t wind up knee deep.
Remember these tips for a great payback,
You’ll be able to just lay back,
And your pager won’t go beep, beep, beep.
It’s a lock

Evolution of byte range lock manager

BY MICHAEL COX AND KERSHAW MEHTA

Have you ever heard of multiple applications opening and writing to the same file simultaneously? If you are familiar with the concept of VSAM RLS processing, this might not sound so unusual. But, if you aren’t, this might sound totally alien and raise the question “Why do you want this function?” Let us explain…

Byte range locking is a UNIX method of serializing data at both the file and sub-file level. A program can lock all or any part of a file, as the name suggests, based on a range of bytes in the file. Unlike MVS data set disposition processing (for example, DISP=OLD), which is required during data set allocation, byte range locking is optional and runs as a system call after the file is opened, although an authorized server need not necessarily have the file open. Also, because the locking is advisory only, all programs accessing that file must participate in this scheme in order for proper data serialization to occur. As you might surmise, byte range locking is a bit more flexible and freewheeling than MVS data set disposition.

Back story

Even though the interfaces have not changed much, the management of byte range locks has evolved significantly since the introduction of z/OS UNIX shared file system support (the ability to share z/OS UNIX files in a sysplex environment). Just about every other release of z/OS has an additional enhancement for management of these locks.

It is worth noting which programs actually perform byte range locking. Examples of application programs include:

- inet daemon (inetd)
- cron daemon (crond)
- Lotus Notes® Domino®
- Server Message Block (SMB) server
- TSO OMVS login.

All these programs perform byte range locking to ensure integrity. The inet and cron daemons lock their process id (PID) files. This means that many UNIX applications during initialization create a file and place their PID number in that file. Next, these applications obtain a byte range lock for the length of the entire PID file (known as whole file locks) to prevent other instances of the same application from starting on the same system. This method allows for only one instance of that application running on a system. Lotus Notes Domino and SMB use byte range locking to manage their configuration files. TSO OMVS locks the /etc/utmpx user accounting file. This list shows that byte range locking has an important data integrity function in the z/OS UNIX environment.

Within z/OS, byte range lock management is handled by two components:

- The DFSMS component contains the byte range lock manager (BRLM), which stores the active locks, and provides deadlock detection.
- The Logical File System (LFS) component of z/OS UNIX provides the system call interfaces and sysplex management facilities, which ensures locking at the correct byte range lock manager in the sysplex.

In a single system environment, there is only one BRLM, so all locks are stored in the same place. If the system fails, both the application and the locks are lost, and there are no application-level recovery issues. When the system is re-IPLed and the applications restarted, the byte range locks are reestablished by the applications and whole file locks for PID files are recreated. It is in the shared file system environment where things get interesting.

First take

Initially for OS/390 V2R9, the release that introduced z/OS UNIX shared file system support (known at that time as shared HFS support); there existed one BRLM for the entire sysplex. The name of the system running this central BRLM server was logged in the z/OS UNIX couple data set and all byte range requests for any file were routed to this one central BRLM server. The main problem with this scheme was the single point of failure that occurred when the system running the central BRLM server went down. And, of course, a system could be brought down for normal maintenance purposes as well as for system failures. When this happened, a new BRLM was started elsewhere in the sysplex, but the active locks would be lost. To maintain data integrity, z/OS UNIX prevented existing applications, which lost their locks, from issuing I/O commands as well as issuing locking commands. Although this allowed new applications to access safely the previously locked files, this scheme had to be improved.

For OS/390 V2R10, an interesting defect was reported. When a system running the BRLM server went down and a new BRLM was initialized elsewhere in the sysplex, the locking application, inetd in this case, never knew it had lost its lock. This was because inetd, as well as some other utilities that lock PID files, never
accessed the locked file after the lock was initially obtained, so no error ever got reported. Moreover, the loss of the lock allowed other instances of inetd to start, thus nullifying the intention of the PID file in the first place.

APAR OW42742 addressed this concern by having the LFS send a termination signal (SIGTERMD) to the application. The intent was to notify the application of a potential data integrity problem because of the lost lock. However, as a result, some customers then noticed that inetd and crond were getting terminated and had to be restarted manually or through automation scripts. Not long after, we introduced distributed BRLM for the shared file system.

Distributed BRLM
z/OS UNIX started a BRLM in every member system in the sysplex. All locking requests for a file were routed to the system that owned the file system containing the locked file. As a result, the active locks were distributed across all systems in the sysplex. The main benefit was that most locking applications no longer sending lock requests to a remote BRLM server because most locking applications lock only files mounted and owned locally. Data integrity was maintained because any system going down would eliminate both the lock and the locking application at the same time. Another side benefit was improved performance because there was less sysplex message traffic for locking. Distributed BRLM was built into z/OS V1R2.

Distributed BRLM had a restriction: moving a file system from one sysplex member to another was not allowed if any active locks were held in that file system. Because of this restriction, z/OS UNIX provided an option to allow customers to choose between the older central BRLM scheme and the newer distributed BRLM scheme. The z/OS UNIX couple data set could be updated to identify which scheme would be used. But again, more improvements were in order.

Moveable BRLM
z/OS V1R6 introduced moveable (distributed) BRLM, which allowed for the movement of file systems that contain locked files to another member of the sysplex during planned outages. With this main restriction removed, distributed BRLM became the new default scheme in the sysplex.

Central BRLM became obsolete. In fact, when all the systems in a sysplex were at z/OS V1R6, the z/OS UNIX couple data set setting for BRLM type was ignored and z/OS UNIX switched all sysplex members to using moveable (distributed) BRLM. This was a transparent change and no action was required by the customer for this switchover to occur.

There remained a final issue — addressing the unplanned system outage issue with distributed BRLM. As mentioned earlier, it was possible for a locking application to stay up but lose its locks, because the locking application and the lock could be on different systems (this is called remote locking). During normal shutdown procedures, moving the file system and the locks solved this problem, as provided in z/OS V1R6. However, remote locks were lost during unplanned system outages.

Recoverable BRLM
This brings us to the latest stage in the evolution of BRLM, which occurs in z/OS V1R8: recoverable (distributed) BRLM. The LFS redundantly stores locks in the local BRLM server as well as in the remote BRLM servers. This way the recoverable (distributed) BRLM allows applications that perform byte range locking of remote files to stay active in the sysplex—even when the local BRLM or the individual system fails.

As mentioned earlier, it was possible for a locking application to stay up but lose its locks, because the locking application and the lock could be on different systems (this is called remote locking). During normal shutdown procedures, moving the file system and the locks solved this problem, as provided in z/OS V1R6. However, remote locks were lost during unplanned system outages.

Find your locks
A new utility, rangelks2, is available for z/OS V1R6 and later releases. The rangelks2 utility is a replacement for the rangelks utility, which lists the currently active and backup locks. The rangelks2 utility also provides more information and captures some locking applications that rangelks might have missed.

Find rangelks2 on the z/OS UNIX Tools and Toys page at ibm.com/servers/ eserver/zseries/zos/unix/bpxa1toy.html.
HFS to zFS file system migration

BY AHILAN RAJADEVA AND BILL SCHÖEN

In z/OS V1R7, IBM announced that the strategic file system is the zFS file system. If you are currently using HFS, then you should start planning the migration to zFS.

Before you begin, be sure to follow the steps in z/OS Migration, GA22-7499, to prepare yourself. Once you’re ready, the BPXWH2ZZ tool can assist you with the migration. BPXWH2ZZ is an ISPF dialog-based utility that you can use in the foreground or run as an ISPF batch job. Use the tool to migrate one HFS at a time or build a list of HFS file systems and migrate them all at one time.

You can preallocate your zFS file systems or let this tool allocate them for you. If a zFS is not pre-allocated, it will be allocated based on the attributes of the HFS. You can modify these attributes through the Change Allocation Attributes panel.

Invocation methods

Use the BPXWH2ZZ tool with arguments or without arguments. If you invoke the tool without arguments, it will guide you with prompts. Otherwise, you can invoke the tool with arguments.

No arguments

Invoking BPXWH2ZZ without arguments is the default behavior. It will prompt you for an HFS file system name and volume. The names can be simple names or patterns like those you would use on the ISPF data set list panel. All names are effectively treated as patterns.

Once you enter an HFS file system name, the next panel will prompt you to enter the volume and SMS class defaults. These defaults will be used to allocate the new zFS file system. If defaults are not provided, attributes will be set based on the HFS file system allocation attributes. The next panel, Data Set List, will contain a table of the HFS file systems that are ready to be migrated. You can scroll through the migration list. Each table entry shows:

- File system names
- HFS space utilization
- Space allocation
- Allocation parameters
- Volume or SMS classes that will be used to create the zFS file system.

This panel is primed to rename your HFS with a .SAV suffix and name the zFS with the original HFS name.

For example, if your original HFS file system is OMVS.HFS.ETC, it is renamed and saved as OMVS.HFS.ETC.SAV, and your new zFS file system is renamed to OMVS.HFS.ETC. Your zFS is then mounted at the same mount point as the HFS.

A is for Alter Allocation Attributes

Place an A in front of the HFS name in the Data Set List panel to alter the allocation attributes of the zFS including its final name and the saved HFS name.

D is for Delete

Specify a D in front of the HFS name in the Data Set List panel to delete the HFS from the migration list.

After you have verified the migration list, specify FG (foreground) or BG (background) at the command prompt to begin the migration. Up to this point, there are no changes to the file systems and you are able to exit from the tool using the END command.

Option flags

You can specify option flags as the first argument and an optional substitution string as the second argument, followed by one or more file system names or patterns separated by spaces. This lets you change the final zFS file system name using the substitution string at invocation.

The option flags are preceded by hyphens. The following flags are valid:

- V is for Verbose
  Use the -V flag to have additional messages issued.

- C is for Console
  Use the -C flag to create a summary log file for background processing. If not specified, the summary information goes to the console.

- E is for Exact
  Use the -E flag to match the input HFS file system name exactly and not consider the HFS name as pattern string. The default for the input HFS file system name is pattern string.

Substitution string

The substitution string is specified as/fromstring/tostring/. Note that all strings matching fromstring in the file system name are replaced with tostring. When you specify the substitution string, the Data Set List panel will be primed to save the HFS as it is and your final zFS name will be changed using the substitution string. You have the option to change the save HFS name, final zFS name, and allocation attributes by entering an A in front of the HFS name in the Data Set List panel and modifying it in the Change Allocation Attributes panel (same method as described above in the “no arguments” section).

Here’s an example of how to use the “substitution string” option: If your HFS file system is OMVS.HFS.ETC and you want your zFS file system to be named OMVS.ZFS.ETC, you can specify the substitution string “/HFS/ZFS/” when invoking the tool.

The following example shows how to invoke the tool with three arguments:

BPXWH2ZZ -VC /HFS/ZFS/ OMVS.HFS.ETC
ISPF batch job processing will end with one of the following completion codes:
- Completion code 0 when all migrations are successful
- Completion code 4 when one or more are successful and one or more fail
- Completion code 8 when all fail.

Mounted file systems switch to read-only during the migration.

What’s new and how to take advantage of it
APAR OA13154 adds the following enhancements to BPXWH2Z:
- ISPF batch job support with no user interaction.
- Ability to set allocation units for zFS in the Change Allocation Attributes panel (you must specify allocation units for zFS in cylinders or tracks).
- Detailed zFS file system allocation errors displayed or saved in the log file.

This enhancement also added a time stamped “migration complete message” issued for successful migration that the system handles as follows:
- Saves the message in the summary log file when background processing with the -C option
- Sends the message to the operator console when running as background processing without the -C option
- Sends the message to the terminal when running in the foreground.

In background processing, the system also issues a TSO/E notification when the migration is successful.

SMS-managed file systems
APAR OA18196, BPXWH2Z adds support for migration from multivolume SMS-managed HFS file system to multivolume SMS-managed zFS file systems. In addition, the tool ensures zFS file system allocations are no less than the minimum allocation supported by the zFS file system.
Restart z/OS UNIX—Automagically!

BY E. OZAN BARAN AND LISA NOVAL

Everyone needs a little automagic, so z/OS V1R8 introduces the z/OS UNIX System Services /etc/inittab file, which allows you to start daemons and system processes, and execute shell scripts or commands during z/OS UNIX initialization. The /etc/inittab file is processed only once during initialization (by /etc/init). However, you can assign the command entries an attribute that will allow them to restart automatically when they end. Otherwise, you need to restart them manually.

But don't go looking for /etc/inittab in your installation just yet. This file is optional, not a default, which means the z/OS distribution you receive does not have a file named inittab in your /etc directory. To create it, you need to go through a simple configuration step first.

A magic wand
z/OS V1R8 comes with a sample /etc/inittab file in the /samples directory. To start using /etc/inittab:

1. Copy the sample file into your /etc directory.
2. Customize it according to your needs.

First, look at the last line of Figure 1. The colon (:) is both a delimiter and a comment character, so the last line is just a comment. Can you see the similarities with the first three lines?

The /etc/inittab file is composed of position dependent entries that use the following format:

Identifier:RunLevel:Action:Command

The first line in the example identifies:
• The process to be started: etcrc
• The action to be taken: wait
• The command to be run: /etc/rc > /dev/console 2>&1

Something is missing! What happened to RunLevel?

The RunLevel entry field isn't supported in z/OS UNIX. It's in the inittab entry to be compatible with other UNIX implementations.

“Customizing /etc/inittab” in z/OS UNIX System Services Planning, GA22-7800, describes the different values you can assign to the actions entry field. Here's a quick summary:

Figure 1. Example of etc/inittab
**once**: Starts the process, continues scanning the /etc/inittab file and does not restart the process when it ends.

**respawn**: Starts the process, continues scanning the /etc/inittab file and restarts the process when it ends.

**respfrk**: Starts the process, continues scanning the /etc/inittab file. If the process issues a fork, the respawn attribute is transferred to the forked child process and the original process is respawned when the child process ends. This transfer takes place only for the first fork. If the process does not fork at all, then respfrk will behave the same way respawn does.

**wait**: Starts the process, waits for it to end before continuing to scan the /etc/inittab file. The process is not restarted when it ends.

### Starting
Initially, you can copy the /samples/inittab file into your /etc directory, comment out the inetd entry in your /etc/rc file and /etc/inittab is implemented. The next time OMVS is initialized, the /etc/rc file along with the command to start the inetd daemon is started through /etc/inittab. If you keep the defaults, inetd is also started with the respfrk attribute.

Of course, you can move the start of all of your daemons from your /etc/rc file to your /etc/inittab file, too. That way, you can take advantage of all that /etc/inittab provides. For example, the respfrk action is a perfect match for the cron and inetd daemons.

You can even go all out and completely replace your /etc/rc file with the /etc/inittab file! However, you might still want to keep all their commands and scripts in /etc/rc and keep /etc/inittab focused on the daemons, which is also perfectly fine to do.

Sometimes, you might start a process with /etc/inittab and the respawn attribute, but then want to stop it without restarting. If a respawnable process ends and then ends again after being restarted within 15 minutes of its first ending, the system displays message BPXI083D as shown in Figure 2. The message identifies the process entry and asks whether to retry the action or ignore the error.

![Figure 2. Example of BPXI083D](image)

**Respawn**

z/OS V1R8 introduces BPX_INITTAB_RESPAWN environment variable that complements /etc/inittab. This environment variable enables you to start processes with the respawn attribute even after IPL when /etc/inittab has already been processed. From the shell, you can set this environment variable to YES or NO, as follows:

- **YES** enables any processes that are spawned (non-local), in this shell, to run with the respawn attribute and behave as if they were started from /etc/inittab with the respawn attribute.
- **NO** specifies that future processes that are spawned (non-local), in this shell will not start with the respawn attribute. This setting has no effect on processes that are already up and running in the system.

The respawn option is ignored when the process is also SHAREAS=YES because the two options are mutually exclusive. For more information on the _BPX_INITTAB_RESPAWN environment variable, see the topic on "BPXK Environment Variables" in z/OS UNIX System Services Planning, GA22-7800.

In z/OS V1R8, there are a couple of ways to tell if a running process has the respawn attribute. The first way is by using the **PS** shell utility. The **PS** command has a new format specification, attr. Use attr through the -o option to display process attributes. Processes are not required to have attributes, but these are allowed:

- **R** respawnable process
- **P** permanent process
- **B** shutdown blocking process.

A subset of the **PS -EF -O ATTR, COMM** command can look like this:

<table>
<thead>
<tr>
<th>ATTR</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>IRRSSM00</td>
</tr>
<tr>
<td>R</td>
<td>/usr/sbin/cron</td>
</tr>
<tr>
<td>B</td>
<td>HZSTKSCH</td>
</tr>
<tr>
<td>-</td>
<td>CSQXJST</td>
</tr>
</tbody>
</table>

The second way of determining whether a process has the respawn attribute is by using the **DISPLAY OMVS** system command. A new indicator on the command output indicates whether a process is started with the respawn attribute. The new attribute is on the STATE column of the **D OMVS,ASID** and **D OMVS,PID** displaying as the fifth indicator. This means that if you want to look at all z/OS UNIX address spaces to see which processes are assigned to the respawn attribute, enter the **D OMVS,ASID=ALL** command. Figure 3 shows an example of a subset of the output:

![Figure 3. Example of a subset from D OMVS,ASID=ALL](image)
Don’t panic!
Dump them all — zFS, z/OS UNIX, and DFS

BY TJ MORRISSEY

Ever had one of those days when everything seemed to be going just right? The last round of maintenance you applied on Sunday went on without problems. All of your key applications are up and running and all of your end users are productive. Your recent migration to the IBM System z™ File System (zFS) went smoothly. Further, you’ve expanded the use of the IBM System z Server by implementing the z/OS Distributed File Service SMB (DFSKERN) server to act as your primary file sharing and serving source for your numerous Microsoft® Windows® clients. Ah, life is good!

Then it happens — that dreaded phone call from your end users indicating that there is a problem… somewhere. After talking to several people, you realize there’s a hang or slowdown that seems to involve your z/OS UNIX-based applications. You’re aware that you have DFSKERN active on your system, and you’re using at least one zFS file system.

After doing some initial review of the symptoms, you determine that some, or even all, of the following conditions might exist on your system:

• z/OS UNIX users are hanging when they try to enter into the z/OS UNIX shell.
• Applications are hanging when they try to access the UNIX file systems.
• System log is indicating there is latch contention (from the displayed output of D GRS.C and D OMVS.WAITERS, or the presence of BPXM056E messages).
• Users are having trouble accessing DFS™/SMB exported file systems residing on z/OS, from their Windows clients.
• SYSLOG is showing messages IOEZ00547I or IOEZ00524I as persistent.

You aren’t sure what or where the problem is. You do know that you are using the z/OS DFS/SMB server, but unless there’s a specific indication, it can be difficult to determine what documentation to gather. The same is true for what, if any, involvement that zFS might have. Unless you see the IOEZ00547I or IOEZ00524I messages that might indicate a potentially hanging thread in zFS, you can’t really be sure whether zFS is involved.

The iron’s hot
It’s at this point that you have an opportunity to get the most accurate and complete set of documentation you can get for the problem. Before deciding how to proceed, consider the close interaction between the z/OS UNIX System Services environment, the z/OS DFS/SMB (DFSKERN) server, and zFS. Each has a unique dependency on the others, in conjunction with the user/application that are accessing data that resides in zFS file systems on z/OS. In addition, DFSKERN might or might not export these same file systems for access from Windows clients, through the SMB protocol. The result is a close interaction between any or all three of the components.

It’s important to be aware of this close interaction and gather the correct documentation right away.

Dump them all!
Yes, “dump them all.” At first, this might seem like an overreaction, but it’s not. In order for IBM support to have the best chance at diagnosing the cause of the problem, it is extremely helpful, and in many cases absolutely necessary, to determine from the dump what all three components are doing in response...
to the user or application requests for data residing on z/OS. A failure in one component can have far-reaching, adverse affects on how the other components are functioning. You need to capture a dump of all components, even if it is not obvious that DFSKERN is involved with a suspect hang involving z/OS UNIX System Services, or if you feel the problem is isolated to z/OS UNIX System Services and not DFSKERN or zFS.

It’s important to be aware of this close interaction and gather the correct documentation right away. If not, you might be left with insufficient or missing documentation that results in the inability to fully diagnose the cause of the problem, which can require a recreation of the problem. This is something that neither you nor IBM support wants to see happen.

**Before recovery**

When you think you have a hang condition that involves z/OS UNIX, DFSKERN, and zFS, make sure you perform the dump before attempting any type of recovery actions, including entering a **STOP** or **CANCEL** command for one of the address spaces. Figure 1 shows an example of the dump command:

```
DUMP COMM=(title)
 R x,JOBNAME=(OMVS,DFSKERN,ZFS),CONT
 R x,SDATA=(RGN,LPA,SQA,LSQA,PSA,CSA,GRSQ,TRT,SUM),CONT
 R x,DSPNAME=('OMVS'.*) END
```

*Figure 1. Dump command example*

Next, send the dump to IBM, along with the several hours of SYSLOG and LOGREC data. Finally, open a PMR with IBM support.

Before long, everything will be going right again.

**It’s the zFavorites for zSeries® credit card CD!** You’re gonna love this! It has all sorts of helpful Web links, like those for:

- Hardcopy
- Operating systems
- Software
- Language and tools
- ISV development and applications
- Product documentation
- Marketing information
- Education
- Support
- Links to FREE downloads
- IBM Redbook™ sampler
- WebSphere® Application Server
- XML

To use the CD: Insert it in any standard CD-ROM and it should start automatically. If it does not, click **Start > Run**, and then type `x:\index.htm` (where x is your CD-ROM drive letter) and press **Enter**.

Additional copies of zFavorites CD-ROM (GK3T-4331-10) are separately orderable.
Refurbishing the z/OS foundation

The new real storage management in z/OS V1R8

BY ELPIDA TZORTZATOS AND DIETER WELLERDIEK

In September 2006, z/OS V1R8 introduced a new real storage management system that is able to manage single z/OS images with up to 4 terabytes (TB) of real storage. Changing the infrastructure of z/OS storage management to support this extra storage is like trying to convert the Guggenheim Museum into a skyscraper! This article explains the remodel of the z/OS foundation and its new features.

Building plans
Before we go into more detail about the new foundation, let’s briefly review the old foundation. The basic foundation was laid in the early 1960s when central storage was measured in kilobytes (a modest cottage by today’s standards). Over the next three decades, z/OS witnessed many improvements and additions, including the introduction of “expanded storage.”

Think of the storage management component of z/OS as an apartment building manager. The storage management in z/OS manages real storage in units, known as frames. Like an apartment building manager who is responsible for the maintenance of apartment units in a building, z/OS is responsible for maintaining storage frames. Each incremental improvement to z/OS storage management, such as expanded storage, enabled “the building manager” to manage more “apartments.”

Managing changes
In 2000, the amount of real storage supported by a single MVS image increased from 2 GB to 128 GB. This increase also required some changes in the z/OS storage management algorithms. Another important change was the frequency with which the building manager checked on the status of each apartment. With 2 GB or less, the building manager verified each apartment’s occupancy status every second. With the increase of real storage up to 128 GB, the z/OS building manager started verifying apartment status once every 10 seconds. This check is what is called an unreferenced interval count (UIC). Because the frequency of this check changed, the UIC maximum value also changed from 254 for 2 GB to 2540 for up to 128 GB. These minor adjustments to existing algorithms were all that was needed to support up to 128 GB; the base (the foundation of z/OS) didn’t need to be modified.

Changing the foundation for the future
With z/OS V1R8, we have enabled the operating system to handle 4096 GB (4 TB) of storage. For this support, refurbishing the foundation has become necessary. The building manager can no longer spend any time at all verifying apartment occupancy status; there are simply too many of them, and this activity would interrupt other more important management tasks. Additionally, in such a large structure, the availability of apartments for new tenants is rarely an issue, so the value of spending time checking for changes in status is minimal.

In z/OS terms, because UIC update processing was designed to assist in paging decisions in a real-storage constrained system, the availability of up to 4 TB of real storage minimizes the need for stealing and paging, thus, making UIC update processing less critical for overall system functions. In other words, because the operating system has more than enough real storage for the normal processing, spending time looking for the oldest frame is just unnecessary overhead.

Often, companies rent whole floors or suites, not just single apartments. In the new world of large real systems, it is not uncommon for applications to work with storage in terms of arrays of frames, not just single 4 KB chunks. And like the structure we have built with z/OS V1R8, this can be upwards of a thousand apartments. In small buildings, you can easily move (physically swap) tenants into external storage (auxiliary storage). In huge buildings like the skyscraper where companies occupy thousands of apartments, such a move (physical swap) is no longer feasible; it is only possible to move apartments (stealing) to the external storage (paging device), but not the tenants occupying large areas of the building.
The new UIC calculation in z/OS V1R8

Earlier, we described why the UIC processing had to change. Let’s look at how the UIC calculation changed. Before z/OS V1R8, the operating system had a service request block (SRB) running, which analyzed all frames of active “IN” address spaces every 10 seconds. The frame analysis was done by address space and the oldest frame of an address space represented the system high UIC. (You see the same UIC value in the RMF™ monitor reports.)

As we described above, such a process is very time consuming. Actually, the SRB did even more! The SRB optimized some queues to find the best frame even faster during stealing, which required an additional amount of time.

The z/OS V1R8 storage management is optimized for large storage configurations, so you will see the best throughput when the system has a need for only moderate stealing.

The new storage management no longer has an SRB analyzing frames every 10 seconds and no longer performs queue optimization. Actually, the replenishment algorithm has changed from a least recently used replenishment algorithm to a modified clock replenishment algorithm.

Figure 1 shows how the replenishment algorithm moves forward when frames are needed. We also redefined the UIC so that it now represents the time in seconds (t) that a steal cycle needs to complete.

A complete steal cycle is the time the stealing routine needs to check all frames in the system. In the figure, a purple box represents an unreferenced frame; a green box, represents a referenced frame. When there is a demand for storage, the stealing routine does the following:

• Tests the reference bit of a frame
• Decides whether to steal the frame
• Schedules the page-out of the frame.

Stealing only occurs when there is a demand for storage.

The UIC algorithm forecasts the UIC based on the current stealing rate. The UIC can vary between zero and 65535 and is calculated every second. When there is no demand for storage in the system (that is, when there is no need for stealing), the system has a UIC of 65535. If there is a very high demand for storage in the system (that is, when stealing might be necessary), the system has a UIC close to zero. Stealing takes place strictly on a demand basis; that is, there is no periodic stealing of long-unreferenced frames. In a large real system, a steal cycle can take days to complete.

The storage management stealing routine honors storage targets set for WLM managed address spaces and for storage critical workload.

How z/OS V1R8 resolves pageable storage shortages

Before z/OS V1R8, pageable storage shortages were resolved by swapping the address spaces with the most fixed frames to auxiliary storage (AUX). Because z/OS V1R8 no longer supports physical swaps of address spaces, this technique is not used to handle page shortages.

Think of the storage management component of z/OS as an apartment building manager. The storage management in z/OS manages real storage in units, known as frames.

Instead, z/OS V1R8 only logically swaps the address space and then moves the fixed frames above 2 GB. This technique has the advantage of not requiring the swap of the whole address space ( pageable and fixed frames ) to AUX. But it only moves the fixed frames of the address space; the pageable frames stay where they are.

This also resolves the pageable storage shortages below 2 GB. If the system runs into a pageable storage shortage between 2 GB and 4 TB, the operator must cancel the user that fixed the most frames in this area. Message IRA404I informs the operator which address spaces have allocated the most fixed frames. If canceling this workload is undesirable, consider installing additional real storage for this image to enable higher workload demands on the system.

What if there is moderate physical swapping on my pre z/OS V1R8 system?

Before you migrate to z/OS V1R8, analyze the RMF reports (especially the CPU activity report). Check the RMF report to see if your system does physical swaps of address spaces with standard workload. Figure 2 shows an example of the CPU activity report.
If your RMF reports show address spaces on the **OUT READY** or **OUT WAIT**, or both queues, consider increasing the amount of real storage for the system or logical partition. Otherwise, you might see a performance degradation of the system.

**Other nice storage enhancements in z/OS V1R8**

Here are some other changes that ensure a smoother running system:

- New informational message IRA205I
  In z/OS V1R8, the system issues IRA205I when the system has allocated more than 50% of the auxiliary storage. This allows adding additional page data sets when the system is not already in an auxiliary storage shortage.
  The message gets reissued every 2 hours, as long the system has allocated more than 50% auxiliary storage.

- Available frame queue (AFQ) management
  The minimum available frame queue target is now managed by the operating system. Because today’s systems have so much storage, it is time to reserve some percentage of storage for system use. In the past, this reservation was done through the **MCCAFCTH** keyword in the IEAOPTrn member in the SYS1.PARMLIB. This is still possible, but now if the specified value is too small, it gets overridden.
  The new AFQ adjusts the minimum AFQ target to 0.4% of pageable frames in the system, but with a minimum of 600 frames. If your installation specifies the **MCCAFCTH** in the IEAOPTrn member, the system honors this value, as long as the value is above 600 and 0.4% of pageable frames are in the system.
  This function is also available through APAR OA14409 on systems before z/OS V1R8.

**Is this the end of the story?**

As you may have guessed, this is not the end of the story. As in the world of construction, improvements with z/OS always come fast and furious. So stay tuned and see what the future holds for the z/OS skyline. ■
In our latest Level 2 with me about recognizing potential problems before they become real ones, we look at what some hypothetical users say about installation and migration that can impact your Language Environment or Integrated Cryptographic Service Facility (ICSF) environment.

If you plan to apply Language Environment FMIDs or PTFs (z/OS V1R6 or higher) or install ICSF (FMID HCR7730, HCR7731, or higher), see the following advice for some tips on how to install and migrate these components successfully to prevent undesirable results.

Thanks to Ritu Bhargava and Cheri Prendergast from the Level 2 test team for their help with this version of "Level 2 with me".

---

Errors received during Language Environment FMID (and PTF) installation

Dear Level 2,

I recently tried to install Language Environment FMID HLE7709 (z/OS V1R6) and received the following error messages. When I reviewed the messages, they appear to be impacting multiple components other than Language Environment, for example, Binder, High Level Assembler and SMP/E. Are all of these messages related to the Language Environment FMID installation?

IEW2690E  3530 ONE OR MORE FIELD DESCRIPTORS IN GOFF RECORD 3 WITHIN MEMBER CELQxxxx IDENTIFIED BY DDNAME SMPyyy ARE NOT VALID. ERRORID = 40.
IEW2307E 1032 CURRENT INPUT MODULE NOT INCLUDED BECAUSE OF INVALID DATA
IEW2509S 3602 MODULE CELQxxxx IDENTIFIED BY DDNAME SCEERUN2 IS AN UNSUPPORTED VERSION AND CANNOT BE PROCESSED
IEW2307E 1032 CURRENT INPUT MODULE NOT INCLUDED BECAUSE OF INVALID DATA
IEW2322I 1220 9 NAME CELQxx(R) MAX ACCEPTABLE RC=00
IEW2230S 0414 MODULE HAS NO TEXT
IEW2648E 5111 ENTRY CELQyy IS NOT A CSECT OR AN EXTERNAL NAME IN THE MODULE
IEW2677S 5130 A VALID ENTRY POINT COULD NOT BE DETERMINED
IEW2638S 5384 AN EXECUTABLE VERSION OF MODULE CELQxxx EXISTS AND CANNOT BE REPLACED BY THE NON-EXECUTABLE MODULE JUST CREATED

BPXF150I MVS DATA SET WITH DDNAME SYSUT1 SUCCESSFULLY COPIED INTO BINARY HFS FILE /SERVICE/usr/lib/nls/locale/IBM/CEH$xxxx
BPXF173E RETURN CODE 00000079 REASON CODE 05E80241.
PROGCTL ATTRIBUTE CANNOT BE SET FOR FILE /SERVICE/usr/lib/nls/locale/ibm/CEH$xxxx

GIM9168E ** HFSCOPY PROCESSING TO THE SFSUMLCL LIBRARY FAILED FOR HFS CEH$xxx IN SYSMOD UKnnnn. THE RETURN CODE (12) EXCEEDED THE ALLOWABLE VALUE DATE 06.095.
GIM3021E APPLY PROCESSING FAILED FOR SYSMOD UKnnnn. SYSTEM UTILITY PROCESSING FAILED FOR AN ELEMENT IN UKnnnn.
GIM30219E ** APPLY PROCESSING FAILED FOR SYSMOD UKnnnn. PROCESSING FAILED FOR SYSMOD UKnnnn, WHICH WAS SPECIFIED ON THE ++VER PRE OPERAND.

Signed: Unfulfilled Requirements
Dear Unfulfilled Requirements,
When installing Language Environment FMID and PTFs (z/OS V1R6 and above), be sure to use the correct levels of the following components during the apply process: the Program Management Binder, High Level Assembler (HLASM), and SMP/E. The error messages that you received are related to one or more of these components being at the incorrect level. If these messages are ignored or overlooked, you might see ABEND0Cx or U4038, U4039, U4087 during the initialization of the run-time.

Advice: To satisfy the Program Management Binder, HLASM, and SMP/E requirements, after the wave 0 installation, you can use a STEPLIB DD statement to access the target system’s Binder, HLASM, and SMP/E programs, which reside in SYS1.MIGLIB and ASM.SASMMOD1 data sets. Ensure that the target system’s SYS1.MIGLIB and ASM.SASMMOD1 data sets are APF authorized.

Reference:  
- z/OS Program Directory, GI10-0670  
- Search for TECHDOC article #T102551 at ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs.

Sincerely,  
Level 2

Installing ICSF z/OS V1R8 and above

Dear Level 2,
I recently tried to dynamically install ICSF FMID HCR7731 and received varied and unpredictable external symptoms, all associated with x’20’ byte overlays of ESQA common storage.

Signed: Byte Overlay

Dear Byte Overlay,
An installation of or migration to ICSF FMID HCR7730 (available as a Web deliverable) and HCR7731 (available in the base of z/OS V1R8 and as a Web deliverable) requires an IPL. Simply stopping and restarting the ICSF started task is not sufficient.

Before ICSF FMID HCR7730, the CSFDACC control block, which remains resident is ESQA, was x’130’ bytes. Beginning with ICSF FMID HCR7730, the CSFDACC control block size has grown to x’150’ bytes. You must IPL the system to update the control blocks in storage. If you install ICSF without an IPL, it overlays the data (x’20’ bytes) in ESQA common storage when it accesses the expanded control block areas. When this happens, you must perform an IPL to correct the situation. The RSM ASID block (RAB) is one control block that has been affected by the overlay. See Figure 1.

Sincerely,  
Level 2

Advice: An installation of or migration to ICSF FMID HCR7730 or higher requires an IPL.

Reference:  
- z/OS Program Directory, GI10-0670  
- HCR7730 and HCR7731 PSP buckets.

Figure 1. CSFDACC control block

<table>
<thead>
<tr>
<th>&amp;...ENCSDES DACC02</th>
</tr>
</thead>
<tbody>
<tr>
<td>...ENCTDES</td>
</tr>
<tr>
<td>...DECSDES</td>
</tr>
<tr>
<td>...DECTDES</td>
</tr>
<tr>
<td>...MACGEN</td>
</tr>
<tr>
<td>...MACVER</td>
</tr>
<tr>
<td>...OWH</td>
</tr>
<tr>
<td>...PTR</td>
</tr>
<tr>
<td>...PVR</td>
</tr>
<tr>
<td>...OWH256</td>
</tr>
<tr>
<td>&quot;0..0&quot;.</td>
</tr>
</tbody>
</table>

Sincerely,  
Level 2
When you need a timeout
Managing WebSphere Application Server for z/OS timeouts

BY DAVID FOLLIS

Sometimes when a WebSphere Application Server for z/OS servant region processes an application work request, the request encounters problems or just plain behaves badly. If an application has a problem, and multiple requests are using that application, those other requests are stuck, too. Thus, one errant request can tie up multiple threads.

In WebSphere Application Server for z/OS, you can use dispatch timeouts to deal with a request that goes on too long. These timeouts, and some recent enhancements, can help you deal with misbehaving work requests without “punishing the innocent.”

The problem: guilt by association
On z/OS, a WebSphere Application Server consists of a controller region and one or more servant regions in which application requests can be dispatched. While the Java™ virtual machine (JVM) does not allow you to destroy a single thread without affecting the entire process, the architecture of WebSphere Application Server on z/OS does allow you to stop and restart a single servant region without having to take down the entire server. These actions, however, can have unfortunate consequences:

• Other, “innocent” work requests dispatched on other threads in the stopped servant region are ended, too.
• Because a servant region might take some time to restart, your throughput is affected. Without enough servant regions available to run work, requests in the queue between the controller region and the servant regions pile up, waiting for a place to run.

Furthermore, because traditional timeout values cover both the queue time and the dispatch time, it is possible for work requests to be significantly delayed on the queue while waiting for a servant region to start. By the time the servant region starts running again, the work might not have enough time left to execute.

Is it possible to avoid terminating normal work requests along with the culprits? Yes, through judicious use of timeouts, along with some new features you might not have heard of.

How timeouts work
Let’s review how you set timeouts for WebSphere work through configuration variables. For each of the ways in which work requests are dispatched in a servant region, you can set a specific timeout (in seconds), as follows:

For example, reason code 04130007 identifies an HTTP (or HTTPS) dispatch timeout. If this was the last (or only) active servant for the server, new work will just pile up until another servant is started.

Using the auto-pause function
However, application request timeouts have existed since the dawn of WebSphere Application Server for z/OS. What about those recent enhancements I promised to tell you about?

To address the problem of work requests piling up in the queue, we added the auto-pause function in WebSphere Version 5 (maintenance level 510212) and Version 6 (maintenance level 6.0.2.1). This function causes the controller region to stop its listeners and not accept new work after the last or only servant region for this server goes away. Auto-pause remains in effect until the minimum number of servants (which you set through the configuration variable wlm_minimumSRCount) is back up and ready for work.
To use the auto-pause function, you specify the configuration variable `control_region_dreg_on_no_srs=1`. Also, if you’d like the application server to delay accepting new work until you are certain that the problem is resolved, specify the configuration variable `control_region_confirm_recovery_on_no_srs=1`. This setting causes WebSphere Application Server to issue WTOR message BBOO0297A to which you must respond before the controller region can restart the listeners.

When the timeout delay expires, an abend occurs and any requests still in progress or queued with affinity to the abended servant are cleaned up.

### Controlling outbound requests

Sometimes applications make outbound IIOP requests to other servers. If the servers do not respond before the locally dispatched request times out, the application is abended.

Starting in WebSphere Version 5 (maintenance level 510235) and Version 6 (maintenance level 6.0.2.13), you can use the variable `protocol_iiop_local_timeout` (the “Request Timeout” under “ORB Service” in the administrative console) to set a timeout for a response to an outbound request. If a response is not received in time, the controller returns an exception to the application, which can continue processing or give up as needed. That is, the application now has a choice.

By the way, this timeout causes warning message BBOO0325W to be issued … but if a response ever does turn up, no one is waiting for it anymore.

Therefore, we now also issue informational message BBOO0328I to tell you how long the request actually took, so you can adjust the timeout values, if appropriate.

### Speaking of messages

The controller issues message BBOO0232W whenever a timeout causes a servant region to be abended. This message contains some useful information about the expired request, but it was originally designed for IIOP requests. Its fill-in values for “class” and “method name” are not very useful if you are analyzing an MDB or HTTP request that expired.

For this reason, we added informational message BBOO0303I to provide you with more information for those types of requests. Also, starting with WebSphere Version 5 (maintenance level 510236) and coming soon in a Version 6.0 PTF, new message BBOO0327I will provide lots of additional information about the timed out request. The message includes the ASID and TCB address for the task that was running the request, and the time at which the request was:

- Received by the controller
- Placed on the work queue
- Dispatched in a servant region.

### All in a day’s work

Being able to timeout requests and keep the server rolling along is an important feature of WebSphere Application Server for z/OS. IBM continues to make improvements to this processing over time to make it even more responsive to your needs.

---

You can use dispatch timeouts to deal with a request that goes on too long.

You can pause a server dynamically, too, through the console command, `MODIFY PAUSELISTENERS`. To resume, enter the `MODIFY RESUMELISTENERS` command.

**Delaying an abend**

When an abend occurs, you might wonder what happens to all of those other requests running on the other threads in the servant region that abended. They are, after all, just innocent bystanders who are taken out along with the guilty thread.

Starting in WebSphere Version 5 (maintenance level 510224) and Version 6 (maintenance level 6.0.2.5), you can specify the configuration variable `control_region_timeout_delay=nn`. Here, if the controller region determines that a request has timed out, it will wait the specified number of seconds before abending the servant. Threads in the servant that finish their current requests do not take on any new work. If another servant is available, the controller diverts work there without affinity to the abended servant.
Save the cables: an update

Flapping link threshold recovery

BY JOHN BETZ AND BOB FUGA

In the article “Save the Cables” in z/OS Hot Topics Newsletter Issue 13, August 2005, GA22-7501-09, IBM introduced you to a condition called “flapping links,” which is a potential problem with intermittent link failures that can occur on FICON® single mode fiber. In this article, we discuss the design that we implemented to help avoid costly recovery from flapping links and the actions you can take for successful recovery.

On the IBM System z9™ Business Class (BC) GA1 and the IBM System z9 Enterprise Class (EC) GA2 levels of microcode, there is new support for removing a logical path when a flapping link occurs. This design helps prevent z/OS from spending significant time in I/O recovery that could potentially affect your operations.

Flapping link threshold
A flapping link threshold is reached when a logical path is re-established five or more times within a five-minute period. The count of how many times a logical path is re-established is kept in the channel subsystem microcode. The current implementation has a fixed five-minute window. If a link fails and is re-established only four times within that window, the threshold count is reset when the timer window closes and a new set of five link failures must occur before the system declares a threshold event. If the number of failures does not reach a count of five within any five-minute timer window period, the system does not declare a flapping link threshold.

When the channel has determined that a flapping link has occurred, the channel sets the threshold event for the physical path, that is, the channel or destination link for all logical partitions (LPARs). Active I/O operations are either returned with an interface control check (IFCC), an IFCC time-out, or a “path-not-operational” condition. Subsequent I/O operations that target the threshold path receive path-not-operational errors.

How z/OS responds
During a failure, if an I/O operation gets a path-not-operational condition, z/OS issues messages that indicate the failing event and a reason message that indicates that a threshold event was exceeded. z/OS removes the failing path from the logical path mask of the device.

However, z/OS does not attempt to minimize the number of I/O operations down the failing path for other devices on that same logical control unit, and it does not attempt to remove the path from any of the other devices on the failing logical control unit unless an I/O operation tries to occur down that path. As a result, some devices on some LPARs have the path logically removed from the device by z/OS while other devices continue to have the path in the device’s path mask. It’s possible that on some LPARs the system will not issue any messages and will not remove the logical path from any devices if no I/O requests were attempted on the failing path from that LPAR.
SYSLOG messages

From z/OS, you should see in the SYSLOG the messages shown in Figure 1 and Figure 2.

Figure 1. Message for devices with multiple paths defined

Figure 2. Message for devices with single path defined or where the last path is lost

A threshold condition can also be recognized by entering command 

D M=DEV(DEV#, (CHP#)), which provides the output shown in Figure 3.

Figure 3. Recognizing the threshold condition

Threshold reset on control unit port

If a flapping link threshold is encountered on a control unit port, the system removes only those paths associated with that CHPID port.

Recovery Actions

A threshold condition is reset by entering a VARY PATH command. You only need to enter the VARY PATH command to a single device path in order to reset the threshold condition.

Suggested practice for resetting a threshold condition

Before you enter the VARY PATH command, ensure that you have performed the necessary service to repair the condition that caused the flapping link. When you have completed service, enter the VARY PATH command by specifying a single device. This approach allows you to verify if the repair action is successful. If not, the only impact is a flapping link threshold for that one path and the device that the system brings online. If the path successfully comes online, and I/O to the device is successful, you can vary the path online to the remaining devices that are associated with any of the logical control units. You must enter the VARY PATH command to all devices for all logical control units that are associated with the control unit port that has reached the flapping link threshold.

Threshold reset on CHPID port

If a flapping link threshold is associated with a CHPID port, the system removes the path from all devices associated with the control units that are connected through the CHPID.

Recovery actions

The same rules apply for resetting the threshold condition and recovering the path for CHPID threshold as they do for a control unit threshold.
Suggested practice for bringing a path online

Use the VARY PATH command to bring a path online to a single device. You must enter the VARY PATH command for all the devices associated with the control units that are defined to the CHPID on all LPARs that have the CHPID defined.

Because a CHPID can have many control units associated with it and be spanned or shared between many LPARs, it can become cumbersome to enter all of the required commands to fully recover any of the paths. Although the suggested practice is to use VARY PATH command for recovery, you can enter CF CHP(CHP#),OFFLINE, then CF CHP(CHP#),ONLINE on a logical partition basis to reset the threshold and recover the path to all associated control units. You must ensure that the repair for the flapping link threshold condition has occurred before you use this procedure in order to avoid the many potential I/O error messages that the system generates when it encounters a flapping link condition.

Threshold reset on a FCTC CHPID

If you reach a flapping link threshold on a FICON channel-to-channel (FCTC) CHPID, you might be required to perform additional steps if the connections are between System z9 servers, or are internal to a System z9 server that has flapping links support installed. In these cases, you might need to enter the VARY PATH command on both sides of the FCTC.

The CONFIG CHPID(cc), NOVARY recovery option

After a link failure, if the user has configured the failing CHPID offline and causes the link threshold condition to be reset, expect some slight behavioral differences.

For the D M=DEV(dddd,(pp)) command, the status line does not show that a link threshold event has been exceeded, but shows that the CHPID has been configured offline, as shown in Figure 4.

Here is a summary of the commands:

VARY PATH(DEV#,CHP#),ONLINE—vary path online to a single device

VARY PATH(DEV#-DEV#,CHP#),ONLINE—vary path online to a range of devices

D M=DEV(dddd,(pp))—display MATRIX command for path status of a single path (pp) to a device (dddd)

APAR support

Flapping link threshold recovery is only supported on FICON channels on the System z9 servers. Software support for this new function is available for z/OS V1R4 and later releases by applying the PTFs for the following APARs:

- APAR OA10906 provides enhanced problem determination for non-operational paths on the z9 EC.
- APAR OA13644 provides support for the reset of the link recovery threshold condition. It also fixes a problem in OA10906 that improperly displays the control unit number on the D M=DEV operator command for CTC devices.
- APAR OA14156 provides a fix for ABEND378 RSN1C that occurs when entering D M=DEV(DEV,(CHP)) command on a system prior to the z9 EC.

New support for removing a logical path when a flapping link occurs…helps prevent z/OS from spending significant time in I/O recovery.

IEE174I 16.37.29 DISPLAY M 248
DEVICE 6000 STATUS=ONLINE
CHP 95
ENTRY LINK ADDRESS 6B41
DEST LINK ADDRESS 6F62
PATH ONLINE N
CHP PHYSICALLY ONLINE N
PATH OPERATIONAL N
MANAGED N
CU NUMBER 6000

Figure 4. Recognizing that a CHPID is offline

If the operator issues the CONFIG CHPID command to take the path offline, the system removes the path from the logical path mask for all devices that are accessible on that CHPID. When the repair action is complete, the suggested practice is to use the CONFIG CHPID command with the NOVARY option to configure the CHPID back online. This action allows the channel path to come back online but does not allow the devices to begin to use the failed path until the operator issues the VARY PATH command for each of the devices that are affected by the original CONFIG CHPID offline command.
Every picture tells a story

Best practices for stand-alone dump

BY ED MEKEEL AND GLEN GARRISON

In photography, every picture tells a story... if everything is set up correctly. That is, after you’ve chosen the best exposure setting, the best lighting and the best composition.

When the right considerations go into the photograph, the result is something that is eye-catching, interesting and relays information to the viewer.

Similarly, when we take a stand-alone dump, we are taking a snap shot. This picture, however, is a snap shot of processor memory, which contains the details of a system problem that somehow affected system and workload health.

The service folks who will diagnose the problem and provide a fix use this picture. These folks depend on complete and accurate information in the dump. To ensure we’re successful in capturing the necessary picture, there’s a set of steps to follow.

Just like a “How To” guide for taking good pictures, we’ll cover the current set of “best practices” to consider when taking and handling large stand-alone dumps for z/OS.

Getting in focus

When submitting a problem to IBM Support for analysis, you need to focus on optimizing stand-alone dump data capture and optimizing problem analysis time. Consider these specific factors:

- Stand-alone dump data set definition and placement
- IPCS performance
- Providing documentation to IBM Support.

This information assumes your installation is operating at least at the z/OS V1R6 level. We note additional enhancements available in z/OS V1R7 where appropriate.

Setting up the shot

When you define the stand-alone dump data set correctly, you can reduce capture time. To achieve the best dump performance, write the dump to a “striped” DASD stand-alone dump data set, also called a multi-volume dump group. Because z/OS is not running when taking a stand-alone dump, striping is simulated by defining a dump data set composed of two or more DASD volumes. The more DASD stripes dedicated to stand-alone dump, up to the maximum of 32 volumes, the better the overall performance. In general, the stand-alone dump needs to be placed on a volume behind a modern control unit like the DS8000™ or IBM Enterprise Storage Subsystem, or dumping to a fast tape subsystem.

Striping is the use of spanned volumes for the stand-alone dump data set. You do this by specifying a volume list (VOLLIST) in AMDSADDD to designate a list of VOLSERS corresponding to each DASD volume making up the data set. Allocate a multi-volume data set using the specified list of volumes. Use the device number of the first volume to specify the data set to stand-alone dump. Each volume should be on a different LSS to ensure parallelism when writing the dump.

When a problem occurs, there are several ways for you to help IBM Support get quick access to the data they need.

When you define a multiple-volume stand-alone dump data set, ensure that each volume resides in a separate Logical Subsystem (LSS) to ensure maximum parallel operation. Use the AMDSADDD utility to define the stand-alone dump data set. For more information, see z/OS MVS Diagnosis: Tools and Service Aids, GA22-7589.

When dumping to a DASD dump data set, we have observed significant performance improvements with striping the data to a multi-volume stand-alone dump data set. For more information, see Flash 10143 at www.ibm.com/support/docview.wss?uid=tss1flash10143.

Reviewing the picture: Analysis of large stand-alone dumps and large SVC dumps

The IPCS dump directory is a VSAM data set that you can tune for optimal performance. One optimization shown to improve IPCS performance is to reduce the number of Control Interval (CI) splits during initialization and analysis of dumps. To do this, use the RECORDSIZE parameter in BLSCDDIR (shipped in SYS1.SBLSCLI0) and then observe the number of CI splits using standard VSAM data set analysis techniques, such as the LISTCAT command.
In z/OS V1R7, the RECORDSIZE parameter in the shipped BLSCDDIR is ‘RECORDSIZE (2560 3072),’ which has proven to yield well performing CISIZEs for the data portion of the data set. Obtain additional IPCS performance by using a DFSMS striped input dump data set. Placing the dump into a data set with at least eight stripes has shown marked improvement in IPCS response.

Use the COPYDUMP command to extract a list of ASIDs from the stand-alone dump, producing a reduced size dump data set. The smaller dump will result in faster FTP time and will speed up initial IPCS analysis of the dump. (We’ll come back to COPYDUMP shortly.) For IPCS on z/OS V1R7, be sure to install PTF UA26080 to fix a problem with unnecessarily long IPCS initialization time.

More than a thousand words

Very large dumps have the potential to cause considerable delays in the initiation of problem analysis and diagnosis. The time to process and FTP very large stand-alone dumps can take hours to complete.

When a problem occurs, there are several ways for you to help IBM Support get quick access to the data they need.

Viewfinder

Remote Screen Viewing Facility (RSVSF), Virtual OnSite (VOS) and Assist On Site (AOS) allow IBM Support to immediately log into an IPCS session and view available documentation with no initial data transfer. This should always be the first option in a time-critical situation. Rapid viewing of the documentation has the advantage of allowing IBM Support to itemize or customize any additional documentation they might want to FTP to IBM for the given situation. In fact, if documentation must be sent through FTP, the analysis of the current documentation can continue while the requested documentation is in transit.

Exposure settings

In many cases, we don’t need the complete stand-alone dump for diagnosis of a given problem. Sending a subset of the stand-alone dump to IBM might prove sufficient.

Using the IPCS COPYDUMP command, you can extract a subset of ASIDs from the stand-alone dump into a separate data set and send it to IBM. The use of COPYDUMP has been shown to reduce the data transferred by roughly 30 to 40% compared to transmitting a full stand-alone dump.

IBM recommends that you include the first 20 ASIDs in the system in the COPYDUMP operation, as well as the CATALOG and JES ASIDs, if known. This allows extraction of the base system address spaces from the dump. The COPYDUMP should also include any other ASIDs known to be involved in the problem, or that IBM Support requested.

An example of the syntax of the COPYDUMP command is:

```
COPYDUMP ODS('OUTPUT DATASET NAME')
    ASIDLIST(1:20,JES ASID,CATALOG ASID,PROBLEM ASID)
    IDS('INPUT DATASET NAME')
    NOCONFIRM
```

If you don’t know the JES, CATALOG, and problem ASIDs, an example of the syntax is:

```
COPYDUMP ODS('OUTPUT DATASET NAME')
    ASIDLIST(1:20)
    IDS('INPUT DATASET NAME')
    NOCONFIRM
```

Pictures in the mail

You must compress your dumps with the TRSMAIN utility before sending them to IBM. Download TRSMAIN from the IBM Support Web site at [techsupport.services.ibm.com/390/trsmain.html](http://techsupport.services.ibm.com/390/trsmain.html).

To send large dumps to IBM, use the PUTDOC utility. PUTDOC allows the dump to be split into multiple files and then transmitted in parallel. See the IBM Support web site for complete instructions on the PUTDOC utility at [techsupport.services.ibm.com/server/nav/zSeries/putdoc/putdoc.htm](http://techsupport.services.ibm.com/server/nav/zSeries/putdoc/putdoc.htm).

Snapshots of the future

As memory sizes continue to increase, the performance of our system dump utilities are a major consideration for z/OS. Areas that we’re looking at include:

- Reduced stand-alone dump duration
- Reduced IPCS dump initialization time
- Guidance to ensure that you’re using the appropriate dump data set characteristics
- Limit the amount of data that needs to be extracted from a stand-alone dump
- Reduced time to write the SVCDUMP from data space to data set.

Photo finish

In situations where dump services are required, it’s important that you get the correct system snapshot in the hands of the IBM Support folks quickly to allow them to diagnose problems efficiently. We recognize that taking dumps is a bit more than dealing with snapshots in the family album. The process of getting the dump to the IBM service folks should not get in the way of you getting a quick resolution to your reported problem.
Overcoming obstacles

IMS DLIModel utility

BY KEVIN WANG AND DANNY NGUYEN

Are you interested in developing Java application programs to access your IMS data and even store and retrieve XML documents into and out of IMS database? If your answer is yes, then you need the IMS DLIModel utility.

The way we were: before DLIModel utility

When IMS Connector for JDBC (formerly known as IMS Java application support) was first introduced in IMS Version 7.1, users were able to write Java application programs to access IMS databases from various environments including:

- IMS dependent regions
- IBM WebSphere Application Server for z/OS
- IBM WebSphere Application Server on distributed platforms
- IBM CICS® Transaction Server for z/OS
- IBM DB2 Universal Database for z/OS stored procedures.

It brought all the benefits including rapid application development, leveraging industry standards, leveraging new and abundant skills in the marketplace to mitigate the loss of mainframe programming skills for customers, and integrating with other products.

However, it also required a Java metadata class that describes the logical data view of the application’s program specification block (PSB). The name of this class is passed to the IMS Java library by the application program when the JDBC connection to the database is established during runtime. IMS Java library uses the class to map the JDBC SQL query statements issued by the application to IMS database calls, and to map the segments and fields returned by IMS to the result set returned by IMS Java to the application program.

A manual procedure

Unfortunately, the manual procedure for defining this Java metadata class presents numerous difficulties and challenges to IMS Java users. First of all, it is not trivial creating this class from scratch. The outer class is derived from a provided superclass, DLIDatabaseView. The user must instantiate objects from other provided classes to describe the segment hierarchy and the fields that make up the PSB’s logical view. If the PSB contains multiple database program-communication blocks (PCBs), then multiple hierarchies, one for each PCB, must be described within the outer class. Moreover, the segment layouts must correctly allow for the presence of concatenated segments, field-level sensitivity, and XDFLD fields of secondary indexes.

Having to merge information from multiple application sources

Even then, the segment layout information from the FIELD macros in DBDs might be incomplete, and the user might need to merge information from other sources about additional fields, their lengths, positions in the segment, and data types. This information is typically extracted, again, manually from fragments of existing high-level language applications such as COBOL or PL/I applications.

Overall impacts

Overall, this manual procedure has an impact on the usability of the IMS Java facility and specifically affects the following areas:

- Availability of IMS Java applications, because the manual creation is error-prone and certain errors might not show up until runtime
- Maintainability, because changes to data views might require manual rework of the class definition.

The manual class creation procedure seriously reduces the potential productivity benefits of Java for IMS application development. To address the difficulties and obstacles involved in manually creating the Java metadata class, IMS developed a tooling called the DLIModel utility.

Why the IMS DLIModel utility?

The IMS DLIModel utility is designed to assist IMS DBAs and application programmers in eliminating the need to manually create the Java metadata class required by the IMS JDBC connector. First introduced in IMS Version 8.1, the utility generates various outputs based on input that was specified in the user-supplied control statements. PSB, DBD source members and, optionally, COBOL XMI files are read from their PDS or PDSE data sets. Then they are parsed by the utility to build an in-memory object model of the database structure and the PSB’s view of that structure.
You can specify that an IMS Java metadata class should be generated together with a corresponding easy-to-read IMS Java report. Optionally, you can also specify that an XMI description of the PSB and all the referred DBDs processed in the entire run of the utility should be generated.

**Evolution of the IMS DLIModel utility**

Ever since its debut in IMS Version 8.1, the DLIModel utility has been enhanced with each new release of IMS to provide corresponding tooling support for new IMS functions.

**IMS XML database support**

When the XML database was introduced in IMS Version 9.1, it enabled users to store and retrieve XML documents from both existing and new IMS databases. To support this function, the DLIModel utility had been extended to generate the hierarchically aligned XML schemas in IMS Version 9.1.

**XMI enhancements and IMS XQuery support**

The DLIModel utility has been further extended to provide more enhancements since IMS Version 9.1 became generally available. Most notably, it is now capable of generating XML schemas with new annotations to support the latest IMS XQuery function, as well as generating individual PSB and DBD XMIs.

**DLIModel utility with GUI support**

The DLIModel utility graphical user interface (GUI) was introduced, as a technology-preview in IMS Version 9.1, as an Eclipse plug-in. It provides a user-friendly interface and, therefore, eliminates the need to write the control statements. With the latest DLIModel utility GUI enhancements, users can use their existing control statements as input, provide a printable UML graphical view of the metadata, and provide the ability to directly import COBOL copybooks without first converting them to COBOL XMIs. The UML graphical view of the PSB is very useful even though users might not have IMS Java and do not need the DLIDatabaseView classes. Figure 1 shows the IMS perspective of the DLIModel utility GUI plug-in.

All IMS Version 9.1 customers can download the DLIModel utility GUI. For download and installation information, see [ibm.com/software/data/ims/toolkit/dlimodelutility](http://ibm.com/software/data/ims/toolkit/dlimodelutility). Figure 2 shows the DLIModel utility with the latest enhancements.
OK… how do I use it?
The DLIModel utility is a Java application, so you can run it from the host z/OS system or as an Eclipse plug-in.

Running the DLIModel utility on z/OS
You can run the DLIModel utility from the z/OS UNIX prompt, or you can run it using the z/OS-provided BPXBATCH utility.

1. Create the Control Statements. Before you can run the DLIModel utility, you need to write the control statements to tell the utility where to locate the input files, what outputs to generate and where the outputs should go.

2. Run the utility as a z/OS job or from UNIX System Services. To run the DLIModel utility as a z/OS job, you need to invoke the BPXBATCH utility, a utility provided by z/OS to run any z/OS UNIX shell command or executable file. The JCL procedure, DLIMODEL, shown in Figure 3, runs the DLIModel utility using BPXBATCH.

To run the DLIModel utility from z/OS UNIX, enter the GO command from the /imsjava91/dlimodel directory. The go script file contains the Java command to run the DLIModel application class. Use the following syntax, as shown in Figure 4.

Running the DLIModel utility GUI using Eclipse
To run the DLIModel utility GUI, using Eclipse, follow these steps:
1. Create a new DLIModel utility project and create IMS DB metadata. The DLIModel utility New Project wizard guides you through the tasks of creating metadata. Using the wizard, you can browse the local file system for input files and select the output you want to generate.
2. Modify IMS DB metadata. Using the Multi-page UML editor, which contains multiple PCB pages, you can update PCBs, segment and field information, import COBOL copybooks, and add fields that are not in the DBD source or the COBOL copybooks. You can also use the Properties view to update the segment and field information of the metadata. The graphical editor has all capabilities of a standard editor such as undo, redo, zoom and print.

So what are you waiting for? Consider using the DLIModel utility with the IMS JDBC connector to help develop your IMS Java application programs.
Need help fixing IP storage problems?

Then the new z/OS V1R8 Communications Server IP Diagnosis Guide, GC31-8782-07 is for you! It contains:

- A new chapter devoted to getting started diagnosing storage abends and storage growth
- Common and not-so-common storage terms
- Simple techniques for monitoring storage utilization.

Controlling storage is essential for optimum performance, so you will also benefit from information on how to limit the use of common, private, and Communication Storage Manager (CSM) storage.

Along with decoding common storage messages, sysplex problem determination and recovery messages related to storage, this chapter describes the abends associated with each of the three kinds of storage (common, private, and CSM), and starts you down the path of what to do about them.

Would it help you to know how to determine the largest users of CSM ECSA and dataspace pools? You will after reading this chapter. We’ve even included the steps to take when diagnosing a storage problem.

Here’s what we mean: you receive message

```
EZ7840I SENDTO() ERROR,
ERRNO=1122:EDC8122I
NO BUFFER SPACE AVAILABLE., ERRNO2=74420324
```

and you don’t know what "no buffer space available" means.

This message is an indication that a TCP/IP application has encountered a storage shortage condition. Before reading this chapter, you might not have taken a dump for this condition because you didn’t know whether the "NO BUFFER SPACE AVAILABLE" was due to a temporary spike in storage use or due to an out of storage condition.

After reading this chapter, you’ll know that monitoring your ECSA and CSM storage utilization can help solve the problem. This chapter not only describes methods you can use to start to monitor storage, it also contains information on how to capture documentation for a storage problem.

We sent early versions of this storage diagnosis information to customers, and incorporated their feedback, so we are confident that you will find this new information helpful. Check it out the next time you have a storage problem. You’ll be glad you did.

To access the book, go to publibz.boulder.ibm.com/epubs/pdf/f1a1c560.pdf.

More DST Trivia

Daylight Saving Time (DST) inspired the musical standard “(There Ought To Be A) Moonlight Saving Time” by Irving Kahal and Harry Richman. The song was made famous by Guy Lombardo, Maurice Chevalier, and other popular performers.

The standard calendar in most parts of the world is the Gregorian calendar. It adds a 29th day to February in all years that are evenly divisible by four, except for centennial years (those ending in 00), which receive the extra day only if they are evenly divisible by 400. As a result, the Year 2000 was a leap year, but the Year 2100 won’t be.

In 1942, during World War II, U.S. President Franklin Delano Roosevelt ordered a year-round Daylight Saving Time, called “War Time.” It lasted for three years until 1945.

Studies conducted by the U.S. Department of Transportation show that daylight saving time reduces the country’s electricity usage by approximately one percent, thanks to reduced consumption for lighting and appliances. Similarly, in New Zealand, utility companies have found that power usage decreases 3.5 percent when daylight saving time starts. In the first week, peak evening use commonly drops by about five percent.
GRS at your command

SETGRS enhancements in z/OS V1R8

BY CHRIS BROOKER AND BRYAN CHILDS

There haven’t been any new keywords for the global resource serialization (GRS) SETGRS system command since SYNCHRES was added in OS/390 R7. Yes, that’s OS/390! With z/OS V1R8, GRS is adding dynamic functions to assist your management of a GRS complex with corresponding DISPLAY GRS support to check their status. The new keywords are:

• CNS
• ENQMAXA
• ENQMAXU
• GRSQ.

Who says there’s contention?
In a GRS star complex, one system is the designated contention notifying system (CNS). GRS contention notification processing can consume many resources, so choosing the right CNS is important. Until now, the location of the CNS was random and painful to move.

With the new SETGRS CNS command, you can select which system will be the CNS. With automation in mind, we added the NOPROMPT keyword to allow you to suppress the confirmation WTOR. Additionally, we added a new message, ISG364I, to alert you of a change in location of the CNS.

Who needs more ENQs?
GRS issues ABEND 538 when an address space issues too many concurrent ENQs. The authorized maximum is 250,000 and the unauthorized is 4,096. However, increased workloads can exceed these.

In z/OS V1R8, keywords ENQMAXA and ENQMAXU are available both in GRSCNFxx and SETGRS to configure these maximums appropriately. A new authorized service, ISGADMIN, is available to allow applications to specify address space specific maximums. This helps maintain ABEND538 protection against accidental runaway ENQ users while enabling heavy ENQ users to scale better. ISGQUERY REQINFO=ENQSTATS as well as monitoring messages ISG368E and ISG369I can help track these counts and diagnose problems before an ABEND538 occurs.

What’s in your dump?
In a GRS star complex, SDUMPs that specify the GRSQ SDATA option generate an internal GQSCAN, which gathers all global request data in the complex. This process can result in very long dump times. We added a GRSQ keyword to the GRSCNFxx parmlib member through PTF UA13824 through UA13826 for z/OS V1R4 and later releases. This allows you to specify how much ENQ data you would like to see in a dump.

In z/OS V1R8, the GRSQ keyword becomes dynamic through the SETGRS GRSQ system command. You can now change the amount of ENQ data gathered in dumps without the need for an IPL.

Can’t wait?
Can’t wait to take advantage of all these new GRS functions? You’re in luck! All of the functions described in this article are going into z/OS V1R7 through a small programming enhancement APAR OA11382 with planned availability in the first quarter of 2007.

Find out more
Learn more about all the new SETGRS commands, GRSCNFxx keywords, and system messages in:

• z/OS MVS System Commands, SA22-7267
• z/OS MVS Initialization and Tuning Reference, SA22-7592
• z/OS MVS System Messages Vol. 9, SA22-7639.
key characteristic of data protection solutions is their ability to achieve the Recovery Time Objective (RTO) and Recovery Point Objective (RPO) of the data that is being protected. RTO characterizes how quickly data can be recovered and RPO characterizes how closely to an arbitrary point-in-time the data can be returned. Continuous Data Protection (CDP) is a backup technology that provides an excellent RPO and, in many cases, an excellent RTO.

CDP overview
The basic concept of CDP is to capture every update that is performed against the source data. Every update is captured using a journaling method. Journaling performed at the hardware level is commonly referred to as block-based CDP. Journaling performed at the software level is commonly referred to as application-based CDP. Point-in-time (PIT) backup copies are often performed in conjunction with this journaling.

The recovery process involves first recovering the most recent PIT backup copy that precedes the desired recovery point. Then the appropriate journal records are applied to return the data to the actual recovery point. The recovery time for this process depends on the storage medium that the PIT backup copy and journal records are stored on, the number of records that apply, and the amount of journal to read. Recovery of the PIT backup copy from disk, especially when done with an advanced copy feature such as FlashCopy®, performs much faster than recovery from tape.

zCDP for DB2
A joint solution between IBM DFSMS and IBM DB2 provides an application-based CDP solution for large and complex DB2 systems that are utilized by a single application on System z. This solution was introduced with z/OS DFSMS V1R5 and DB2 V8, and was significantly enhanced with z/OS DFSMS V1R8 and DB2 9. This solution is comprised of the DB2 BACKUP SYSTEM and RESTORE SYSTEM utilities that invoke DFSMShsm™ to create PIT backup copies and use the existing logging features of DB2. This solution manages backup copies on both disk and tape.

zCDP for DB2 supports any vendor disk that utilizes either the FlashCopy or SnapShot interfaces. As such, the generic term fast replication is used when referring to the function that creates a PIT backup copy.

SMS copy pools
z/OS DFSMS V1R5 introduced a copy-pool SMS construct and a copy-pool backup storage-group type. A copy pool is a named set of storage groups that are processed collectively for fast replication functions. The copy pool construct defines the characteristics of the fast replication processing, such as:

- The number of backup copies that should be maintained on disk. You can maintain 1 to 85 copies. An option for the disk copies allows them to be temporary. A temporary disk copy is only maintained until a tape copy is successfully created from the disk copy. After the tape copy is created, the disk copy is withdrawn and the volumes are made available for use by other copy pools.
- The number of parallel tape copies to create. You can generate up to five tape copies from each disk copy. The copy pool uses the existing DFSMShsm dump-class structure to define the attributes for each tape copy. These attributes include frequency, expiration, volume stacking, and encryption.
A copy pool backup storage group is a container for the volumes from which DFSMShsm selects target volumes for the fast replication disk copies. DFSMShsm dynamically selects target volumes during copy pool processing. This feature ensures that any volumes added to a storage group will always be included in all subsequent PIT backup copies.

When you use the **BACKUP SYSTEM** and **RESTORE SYSTEM** utilities, all DB2 data sets must be SMS managed, and organized into SMS storage groups. You define the database copy pool, which includes all of the SMS storage groups that contain DB2 catalog data sets, directory data sets, the data sets for all user created objects, and the associated integrated catalog facility (ICF) catalogs. Optionally, you can define the log copy pool, which includes all of the SMS storage groups that contain the DB2 boot strap data sets (BSDS), active log data sets, and the associated ICF catalogs.

**DFSMShsm Fast Replication**

DFSMShsm uses volume-level fast replication to create point-in-time backup copies of copy pool data. When creating a fast replication disk copy, the application outage only persists for the duration of time that it takes to establish a fast replication source-to-target relationship for each volume in the copy pool. The backup copy is only marked successful if every volume that is defined to the copy pool has a fast replication relationship successfully established. When the creation of a disk copy fails, DFSMShsm retargets the same volumes so that there is at most one failed disk copy.

You can generate a tape copy from a disk copy any time after the disk copy has been successfully established. A tape copy can be created by command or during a predefined automation window. The tape copies are created to appear as full-volume dump copies of the source volumes. As such, recovery from a tape copy is done to the original source volume, not to the intermediate target volume. See Figure 1.

DFSMShsm guarantees that a disk copy that is being dumped to a tape will not be overlaid with a subsequent fast-replication copy until all the disk volumes that comprise that copy have been successfully dumped to tape. This guarantee holds true while the volumes are actually being dumped and also when one or more of the disk copies fails to be copied to tape.

You can perform recovery at the data set, volume or copy pool level. When recovering data from a disk copy, the system uses fast replication for the recovery. An exception to this is data-set level recovery that occurs before the background copy has completed, in which case the system uses standard I/O.

**DB2 backup**

The **BACKUP SYSTEM** utility takes system-level backups of the database copy pool, and optionally, the log copy pool with minimal disruption using DFSMShsm fast replication services. The system-level backup taken is a fuzzy copy, and updates are allowed during backup because consistency is achieved during the restore or recovery process. Data set create, extend, rename, and delete processes are suspended by DB2 while the system-level backup is being taken.

DFSMShsm guarantees that a disk copy that is being dumped to a tape will not be overlaid with a subsequent fast-replication copy until all the disk volumes that comprise that copy have been successfully dumped to tape. This guarantee holds true while the volumes are actually being dumped and also when one or more of the disk copies fails to be copied to tape.

The system-level backup is recorded in the BSDS of the DB2 member where the **BACKUP SYSTEM** utility was run. The unique token that identifies the system-level backup to DB2 and DFSMShsm is stored in the BSDS along with the recovery-based log point, which is the starting point for the log-apply process used by the **RESTORE SYSTEM** utility.

**DB2 recovery**

Restoring the entire DB2 system to a prior point in time requires a conditional restart of DB2 with a log point value (SYSPITR) that identifies the recovery point that you want (refer to Identify Recovery Point, 1, in Figure 2). When DB2 is restarted, the restart process corrects for the fuzziness of the system-level backup by generating compensation log records to back out any committed work. At the completion of DB2 restart, you can run the **RESTORE SYSTEM** utility to restore the database copy pool and to apply the logs. The system is able to restore the database copy pool using DFSMShsm fast-replication services, from the system-level backup that was taken prior to the SYSPITR point (refer to Recover appropriate PIT copy, 2, in Figure 2). The logs are applied using the fast log-apply feature, if enabled, to bring all objects
The RECOVER utility provides the ability to use a system-level backup as the recovery base for a single object or a list of objects. This ability is part of the significant step of connecting system-level backups with object level recoveries.

The RECOVER utility also provides the ability to recover an object or a list of objects to a point in time with consistency.

With these features, the system-level backup and recovery solution are no longer limited to customers interested only in restoring or recovering their entire system to a prior point in time. Even with system-level backups, object level image copies are still needed for recoverability after LOG NO utility events.

Tip: The RECOVER utility is able to restore objects from a system-level backup if its underlying data sets are currently cataloged on the same volumes that they resided on at the time of the backup.

More information
Want to find out more? See the following publications:
- IBM Redbook DFSMShsm Fast Replication Technical Guide, SG24-7069
- DFSMShsm Storage Administration Guide, SC35-0421
- DB2 for z/OS Utility Guide and Reference, SC18-7427

You can generate a tape copy from a disk copy any time after the disk copy has been successfully established.
Opening the window to z/OS

SMB opens a new window

BY EYSHA POWERS, JODI EVERDON, AND RENATA RAND MCFADDEN

In many ways, being a new hire at IBM is just like being a new hire at any other company. With that in mind, I went into the z/OS lab and interviewed new hire, Eysha Powers, to find out what product was most helpful getting her started on IBM z/OS. Eysha graduated from the University of Illinois at Urbana-Champaign in 2003, and came to work for IBM as a software engineer in IBM z/OS Distributed Middleware. Eysha’s answers might surprise you.

Q: What’s one of the most useful tools that you’ve found as a new hire at IBM?
A: SMB

Q: What is SMB?
A: SMB is the acronym for Server Message Block.

Q: Hmm, that isn’t very descriptive. What is Server Message Block?
A: SMB is actually a network protocol. The SMB Server is a product in z/OS that connects your SMB Remote Clients (Microsoft Windows PC or Linux®) to z/OS, so that you can access the z/OS file systems.

Q: Why would you want to access your z/OS files from Windows?
A: I’m a Java programmer who learned to develop quite a bit of code on Windows, so I’m accustomed to the tools and technologies that make coding more efficient. For instance, I took for granted how easy it was to use an editor and a mouse to select a line and then just start typing. I never had a need to use arrow keys or worry about line wrapping. That is … until I starting working on z/OS.

Q: So it’s more difficult to work on z/OS?
A: It wasn’t difficult — it was inconvenient. Not all of the files that need to run on the mainframe are on the mainframe. I was familiar with IBM Rational® Application Developer and other Windows-based development tools. It was faster and easier to create and test applications in an environment that I was familiar with using.

Q: Many system programmers have been working on z/OS for years and they might not understand your frustration. Can you explain?
A: z/OS has built in text editors such as oedit and ISPF editor. These editors are the equivalent to Windows Notepad and work well for basic editing. You can use the function keys to add more features to the document and there are commands to navigate the document and make global changes. But for more complex operations these tools are not sufficient. The biggest advantage of SMB is that all of your files that need to run on the mainframe are on the mainframe.

Q: How did you get those files between Windows and z/OS?
A: I would use File Transfer Protocol (FTP) to transfer files between systems.

Q: Is that straightforward?
A: Setting up FTP is interesting, depending on the z/OS configuration. I’ve actually done this a couple of ways. One particular test system is shared between many users. On this system, an FTP server was not set up to allow access to the files directly. I had to transfer my files to the intermediate server. After uploading the files, I had to use the test system to transmit them to the intermediate system and download the files. This was always a two-step process for transferring files from Windows to the test system. The second approach was a bit easier. If your mainframe is set up with an FTP server running, then you can use FTP to copy the file to it directly.
Q: Is that where the SMB server came in?
A: Yes, the SMB Server connects the Windows PC and Linux directly to the mainframe, which means you can access the z/OS file systems transparently.

Q: Is it a standard component of z/OS?
A: Yes.

Q: Was it difficult to use?
A: No, it was very straightforward. From Windows there are two ways to connect to the SMB Server. You can use the net use command or you can map a Windows network drive to your file system. I prefer to map the drive. I just open Windows Explorer and select the mapped network drive for my z/OS system. On a Linux system you issue a MOUNT command.

Q: How do you access your z/OS files?
A: I just open, modify, and view all files in my file system just as if they were on any other local drive on my PC or Linux.

Q: How does it work when you are developing code?
A: When I need to develop code for z/OS, I create a file and save it to my networked Windows drive. When I’m working on the file, I’m directly manipulating the file on z/OS using Windows applications. Therefore there is no need to transfer files between systems.

Q: What happens when you want to work on the files that are on the Windows PC?
A: Of course, I can still use SMB for file transfer. Just drag and drop files from Windows to the mapped network drive.

Q: Do you have to specify a transfer and translation mode?
A: You never need to specify binary or ASCII because the SMB Server handles that for you.

Q: It sounds like SMB can save new users a lot of time. Is it difficult to configure?
A: There are a number of configurations files that need to be setup for the z/OS system. These files specify the file systems to be exported (name of file system, what type of file systems, and the name that will be used across the shared network), as well as what translation needs to happen. For development work, add the file system in which the code resides and runs.

After the configuration is complete, there are a few options for exporting the file systems such that you can access all files from Windows. If the SMB server is already running on your system, then you can use the dfsshare and dfsexport commands to pick up the update from the modified configurations files. You can also restart the SMB server, which will use the updated configurations files. If the SMB server is not already running, then you can start it to export the file systems.

Messages on the console indicate when the SMB server starts sharing files and the progress. The messages show the share names of every exported file system, along with status of whether the share was successful or unsuccessful. If a share was unsuccessful, the job log is useful in determining the cause of any problems.

Q: How does the file sharing work?
A: The SMB server monitors all files in exported file systems. To prevent conflicts and ensure data integrity, the SMB Server is aware of what file accesses and operations are occurring from both SMB Remote Clients and local z/OS applications. This is important because you might try to write data to a file from Windows that is currently in use on z/OS.

For more information
The z/OS Distributed File Service SMB Administration, SC24-5918, can help you learn more about the full features of z/OS SMB server and show you how to help your employees make the transition to z/OS a little easier.

Q: Is that where the SMB server came in?
Are you a z/OS master yet?

Take the IBM System z Entry Level for z/OS System Programmer Mastery Test

BY KATHY PFEIFFER

Are you a z/OS professional looking for official recognition or proof of your technical knowledge? Or, are you a new hire or student who’s been studying mainframe technology and are looking for an opportunity to validate your z/OS knowledge to the world? Look no further; the IBM System z Entry Level for z/OS System Programmer Mastery test is now available!

What is the Entry Level for z/OS System Programmer Mastery test?
A mastery test is an examination that’s used to verify mastery of subject matter covered in a course or a defined set of learning materials. The IBM System z Entry Level for z/OS System Programmer Mastery Test verifies knowledge that is contained in the IBM Redbook Introduction to the New Mainframe: z/OS Basics. This textbook provides students of information systems technology with the background, information, and skills necessary to begin using the basic facilities of a mainframe computer. The textbook is available through the IBM Academic Initiative System z program and is also available for download at www.redbooks.ibm.com/abstracts/sg246366.html.

Who else can benefit from this textbook?
- Experienced professionals who have worked with non-mainframe platforms
- Other IT professionals who are familiar with some aspects of the mainframe but want to become more knowledgeable about the mainframe environment and its value.

The IBM System z Entry Level for z/OS System Programmer Mastery Test provides an avenue for both students and professionals to distinguish their z/OS knowledge and expertise from others.

Is a mastery test the same as a certification test?
Mastery tests are not certification tests, but they are similar. Certification tests are designed to validate skills that are needed in a specific job role. Typically, several years of experience is required before you can take a certification test.

A mastery test, however, is knowledge-based and helps ensure that an individual has achieved a certain foundation of knowledge and understanding of a particular subject matter: in this case, the contents of the IBM Redbook Introduction to the New Mainframe: z/OS Basics.

How do you prepare to take the test?
You can prepare for the mastery test by following these steps and using these resources:
1. Go to ibm.com/certify and select Mastery tests.
2. Select System z from Select a Hardware mastery test category.
3. Select the test name IBM System z Entry Level for z/OS System Programmer Mastery Test, which is test Z01.
4. Read through the test objectives and complete the recommended education requirements.

Introduction to the New Mainframe: z/OS Basics...provides students of information systems technology with the background, information, and skills necessary to begin using the basic facilities of a mainframe computer.
Listed under Test detail, you find:
- Objectives
- Education requirements
- Assessment test.

The format of the test is multiple choice:
- Number of questions: 64
- Time allowed: 90 minutes
- Passing score: 66 – 75% depending on the test version.

**How do you know when you are ready to take the test?**

After you have completed the education requirements, you can choose to take an assessment test prior to taking the actual mastery test. Click Assessment test located under the Test detail section. You can complete the assessment test online. The test simulates the actual mastery test. It helps you determine whether you might benefit from further study prior to taking the actual mastery exam. The assessment test provides the following benefits:
- Experience taking a timed test that is full length and similar in design to the actual test
- A reasonable indication of test readiness
- A reasonable indication of areas of relative strength and weakness, to direct further review.

The test is available at any time through the Web at a cost of $10 compared to the $95 for the actual proctored mastery.

**How do you register to take the test?**
The IBM System z Entry Level for z/OS System Programmer Mastery test is a proctored exam and is available worldwide. To register for a test, or to locate the test center nearest you, contact Prometric or Pearson VUE from the Web site.

**Is there a fee for the assessment test and the mastery test?**
There is a separate fee for the assessment test and for the mastery test. The assessment test fee is $10. The IBM System z Entry Level for z/OS System Programmer Mastery Test is $95.

**What happens after you take the test?**
After you complete the test, you find out the results immediately. Your results are documented in a score report in the IBM professional skills database.

**What are the benefits of the IBM System z Entry Level for z/OS System Programmer Mastery Test?**
The Entry Level for z/OS System Programmer Mastery Test provides:
- A reliable, valid, and fair method of assessing z/OS knowledge
- A method of building and validating the knowledge of individuals and organizations
- A recognized tool for professional development
- A means for investing in your career and future.

Passing a mastery exam is a tremendous accomplishment. What are you waiting for? Check out our Web site today!

**Daylight Saving Time trivia quiz**

Now that you’re up on the latest developments in Daylight Saving Time (see “Spring ahead without falling behind” on page 4), let’s test your knowledge of this time-honored tradition. Here’s a brief quiz to see whether you earn a DST doctorate:

**Question 1:** At what time does Daylight Saving Time officially begin?
- Midnight
- 1 AM
- 2 AM
- Noon

**Question 2:** Who originally came up with the idea of Daylight Saving Time?
- Benjamin Franklin
- George W. Bush
- George Washington
- Richard Nixon

**Question 3:** Which one of these U.S. states or territories first recognized Daylight Saving Time in April 2006?
- Northern Mariana Islands
- Louisiana
- The Virgin Islands
- Indiana

**Question 4:** Which of these is a valid reason for observing DST?
- To benefit farmers
- To conserve energy
- To aid the poor
- To increase the lifespan of the Sun

**Question 5:** Which DST mnemonic is incorrect?
- Spring forward, fall back
- Spring up, fall back
- Spring ahead, fall behind
- Spring back, fall farther

**Question 6:** Which U.S. president instituted year-round Daylight Saving Time?
- Franklin Roosevelt
- George H. W. Bush
- Theodore Roosevelt
- James Garfield

Trivia quiz continues on page 55
Send in the clones

New HCM copy wizards can enhance your productivity

BY FRIEDRICH BEICHTER

As a user of Hardware Configuration Manager (HCM), do you ever find yourself missing the old Hardware Configuration Definition (HCD) "repeat" or "add-like" functions for cloning I/O configuration definitions for an operating system, partition, I/O subsystem, or processor? Do you find it troublesome to switch to HCD to do this work, as it requires resynchronizing your HCM configuration file with your IODF, and reworking the physical connectivity back on HCM?

Happily, those days are past. With z/OS V1R8, we added powerful new wizards to HCM to help you clone your existing configuration. You might never again need to switch back to HCD to get those functions for cloning your control units and devices.

A tale of two CPCs
Take the example of installing a new CPC. There are two different scenarios in which you might need to clone the I/O configuration definitions, depending on whether the CPC is a replacement of an existing CPC, or a new addition:

• **Scenario 1:** You are replacing an existing CPC with a new one. Here, you probably want to retain most or all of the existing CPC’s logical I/O definitions, and you might need to merge the I/O definitions of several of your CPCs.

• **Scenario 2:** You are adding a new CPC to your configuration. Here, you probably want to base the new CPC’s logical I/O definitions on the definitions of an existing CPC.

These different scenarios require different solutions. For Scenario 1, you need to move the existing channel-to-channel (CTC) links, coupling facility (CF) connections, and physical connections to the target processor definition. For Scenario 2, you can leave the connections where they are, and add more connections to the new CPC.

HCM provides wizards to help you with either scenario. We call them Upgrade and Repeat (Figure 1).

Using the Upgrade wizard
The Upgrade wizard copies the source object to the target object, moves all physical definitions (like patch ports and cables), and copies the logical connections to the source to the target. This wizard also moves the CTC and CF connections from the source to the target. With the Upgrade wizard, you probably want to delete the source at some time. If you have enabled HCM for visual warnings, the configuration diagram identifies the source connections as “logical without physical” connections.

Using the Repeat wizard
The Repeat wizard copies the source object to the target object but does not move any of the physical connections. The wizard does not change the CTC and CF connections of the source, but it does allow you to select new port connections on ESCON® or FICON directors or controller interfaces. The Repeat wizard also helps you define more CTC and coupling facility connections using the corresponding HCM dialogs.
There is no longer any need for you to switch to HCD to get the “add-like” function for control units and devices.

What new wizardry is this?
Besides these functions, the wizards also allow you to selectively copy the channel paths from the source. You will find this function handy if the target processor does not support certain channel path types. Consider, for example, parallel or Open System Adapter (OSA) channel paths that are no longer supported by an IBM z990 or System z9 server.

The new HCM wizards also help you do the following:

• Span shared channel paths with, or map them to, existing shared channel paths on the target processor when performing a copy of a channel subsystem or a partition.
• Enforce renaming of the partition names to avoid duplicates on the target when copying a channel subsystem or a partition.
• View a summary of the configuration changes that are to be performed — before you press the “Finish” button. This gives you another opportunity to see whether you need to make additional changes. You can also save the summary, perhaps to help document your changes for future reference.

And that’s not all
HCM also supports the cloning of an operating system, eligible device table (EDT), or esoteric device group. The new HCM Copy I/O Subsystem wizard uses the selected controller as a model to define a new controller object alike. You can redefine control unit numbers and device numbers using the numbering scheme from the source object with the starting numbers of device groups and distances between adjacent device groups. The segment and interface layout and other physical attributes are inherited from the source.

As mentioned earlier, there is no longer any need for you to switch to HCD to get the “add-like” function for control units and devices. The new wizards help you clone I/O configurations without using HCD directly (and this goes without saying), avoid the resynchronization step and possible rework of the physical connectivity in HCM.

We think the new wizards rule — try them for yourself. You might find that they provide the productivity boost and ease-of-installation that you have been looking for.

Find out more

Early user feedback
Here is what one HCM user, Jeff Stokes of the IBM z/OS Consolidated Service Test (CST) Team, told us about the new wizards:

“I ‘upgraded’ my aging FR24 SMP machine by copying to CSS0 an imaginary ‘new’ z9 EC machine. All of the physical connection data and all of the logical connections, including CTCs and CF links, were moved into the new CSS on the new box! If this were all I wanted in the new box, all I would have to do is assign PCHIDS and go! This is what I have been looking for, we used to have to ‘drop’ the physical connections and reconnect them when we migrated. I think customers will love this if they do things the way we do here, or in a similar way.”
Making unforgettable changes

Automatic documentation for IODF updates

BY REGINE REICHERT AND FRIEDRICH BEICHTER

Suppose you need to make some changes to your I/O definition file (IODF). Then, several months later, someone asks you for specifics on what you changed. If you did not document your changes carefully, you might be lost for words.

Until now, that is.

As an HCD and HCM user in previous releases, you had to manually record your changes to the IODF in an activity log file. Those entries were free format and their accuracy and completeness depended mostly on your time and personal discipline in documenting the changes. Now, with z/OS V1R8, you can have your logical configuration data changes recorded automatically in a change log file associated with the IODF.

Why not let this log file serve as your “memory” for IODF changes in the future?

How changes are logged

To automatically log changes, HCD needs to have write access to your change log file, which is a VSAM data set with the IODF name followed by the qualifier CHLOG. HCD uses a subset of the records in the change log file to generate a sequence of configuration change entries in the activity log file.

When you exit an HCD session, or access a different IODF, HCD displays the activity log panel with this information automatically filled in. The entries describe updates made to HCD objects, such as adding, deleting, connecting, or disconnecting I/O definitions in the IODF.

These entries provide you with a base for documenting further changes, or you can just append them, unchanged, to your activity log file (see Figure 1).

Date & Time . . . . . : 2006-10-06 13:08:47
User . . . . . . . . . : BRRE
I/O definition file . . : 'BRRE.IODFC1.WORK'
Change reference number : 000001

****** ************************** Top of Data ******************************
000001 Partition SYS02.1.PART12 added
000002 ChPID SYS02.1.10 (2) added
000003 ChPID SYS02.1.10 (2) connected to
000004 ... Partition SYS02.1.PART12, access link
****** ************************** Bottom of Data ******************************

Figure 1. Activity log file.

Getting started

Allowing HCD to log changes automatically requires that you do two things:
1. When you create an IODF, specify the activity logging enabled option.
2. In your HCD profile, set the following option: CHANGE_LOG = YES. By default, this option is NO; no entries are created in the activity log panel.

Checking in periodically

The change log file itself is not in a readable format. To see the contents, you need to enter the HCD trace command: Trace on id=clog level=8. The output is written to the HCD trace data set; Figure 2 shows an example.
**Question 7:** What was Daylight Saving Time originally called when it was instituted year-round?
- Moonlight Time
- Summer Time
- War Time
- Conservative Time

**Question 8:** In which month does Daylight Saving Time begin in 2007?
- March
- April
- November
- Wednesday

**Question 9:** Which month marks the end of Daylight Saving Time in 2007?
- August
- September
- October
- November

**Question 10:** In April 2006, which of the following U.S. states and territories will not be observing Daylight Saving Time?
- Hawaii, American Samoa, Guam, Puerto Rico, the Virgin Islands, Arizona
- Indiana, Samoa, Puerto Rico, Washington D.C., Guam
- American Samoa, Alaska, The Virgin Islands, Maine
- Alaska, American Samoa, California, Brooklyn

---

**Figure 2. TRACE ON command output**

**Bonus use**
Given that the activity log file shows the sequence in which I/O definitions are done, you might also find this file useful for analyzing problems in the I/O configuration or the IODF.

**Thanks for the memories**
Let HCD and HCM remember your changes automatically. It might someday save you time and effort in remembering changes to the I/O definition.

For more information, see z/OS V1R8 Hardware Configuration Manager User’s Guide, SC33-7988.
Gateway to your performance data

RMF Monitor III Data Portal

BY HARALD BENDER

Wouldn’t it be nice to see z/OS performance data presented graphically, and in real-time, on your workstation? That is, without having to install any additional client software?

We thought so, too. That’s why we introduced RMF Monitor III Data Portal in z/OS V1R8.

Getting started
RMF Monitor III Data Portal allows you view RMF Monitor III performance data through your Web browser. To use this tool, you do not need to perform any special RMF setup or customization tasks.

The core component of RMF Monitor III Data Portal is the RMF Distributed Data Server (DDS), which runs as a single server address space with sysplex-wide scope. Once started, this server can retrieve performance data from any system within your sysplex.

To begin using RMF Monitor III Data Portal, just follow these steps:

1. Start the DDS on one system in your sysplex: From a z/OS console, enter the START GPMSERVE command.

2. Display the RMF Monitor III Data Portal welcome panel: In the address bar of your Web browser, enter the hostname or IP address of your sysplex, and port number 8803. DDS listens for incoming connections on this port.

Monitoring performance data
From the welcome panel, select the overview menu item to display a predefined performance desktop that we call PerfDesk. This view, as shown in Figure 1, provides you with a kind of high-level sysplex health check, presents data views for the following metrics:

- Percentage of processor utilization per system
- Percentage of total utilization (CP) by partition
- Performance index per WLM service class period
- Percentage of CSA utilization per system.

Besides these metrics, the RMF Monitor III Data Portal can graphically display more than 1200 other metrics for all kinds of resources in your z/OS sysplex. You can define customized views of your performance data and save them for periodic use. RMF Monitor III automatically refreshes active data views based on RMF intervals.

Define customized views of your performance data and save them for periodic use.

Viewing ISPF reports in the browser
If you work with RMF Monitor III ISPF reports, you can now view them in the RMF Monitor III Data Portal browser window as shown in Figure 2.

When you view an ISPF report this way, you find that traditional ISPF limitations no longer apply. For example, you can:

- View the entire report, including the “hidden” columns
- Scroll the window or resize it to suit your needs
- Send the report content to a printer, save it to disk, or even convert it to spreadsheet format.

While you’re at it, try the new RMF Monitor III Data Portal SORT function. Click on a column header of your choice to have the report sorted according to the values of the column. In Figure 2, we sorted the report by number of fixed frames used (the highlighted column).

RMF Monitor III Data Portal helps you analyze z/OS performance data on your workstation with minimal effort.
How to get it

RMF Monitor III Data Portal is included in RMF for z/OS V1R8. For earlier releases, you need one of these PTFs:

- UA90251 for z/OS V1R5
- UA90253 for z/OS V1R7.

For a z/OS V1R6 installation, use PTF UA90251; there was no RMF shipment for that release.

To use this tool, you do not need to perform any special RMF setup or customization tasks.

DST trivia quiz answer key

1. 2 AM
2. Benjamin Franklin
3. Indiana
4. To conserve energy
5. Spring back, fall farther
6. Franklin Roosevelt
7. War Time
8. March
9. November
10. For the U.S. and its territories, Daylight Saving Time is not observed in Hawaii, American Samoa, Guam, Puerto Rico, the Virgin Islands, and Arizona—with the exception of The Navajo Nation due to its large size and location in three states.
Our contributors

E. Ozan (Oz) Baran graduated from Clarkson University in 2003 and has been with IBM since January 2004. He is a software engineer in the z/OS Integration Test Team at IBM Poughkeepsie.

Jim Becker is an Advisory Software Engineer at IBM Poughkeepsie. He has been writing information for the mainframe for many years and currently writes the z/OS Migration book. Jim still has his “welcome to IBM” letter signed by Tom Watson Jr.

Friedrich Beichter is the design and development team leader for HCD and HCM. He joined IBM at the Boeblingen Lab in 1977 and has worked on z/OS, OS/390, z/VM™, and VSE.

Harald Bender has more than 15 years of experience with z/OS and earlier operating systems, with a focus on system performance. He is responsible for the continual development and improvement of Resource Measurement Facility (RMF).

John Betz is a Senior Software Engineer in z/OS System Test at IBM Poughkeepsie. Recently, he has been involved with testing z/OS software that exploits new processor functions, with a focus on I/O functions.

R. David Boenig is the service team lead for the DFSORT Development. Prior to his current assignment, he spent six years as a Level 2 representative for DFSMS products, specializing in DFSMSrmm™ and DFSORT.

Chris Brooker is a Staff Software Engineer in the z/OS Global Resource Serialization (GRS) Development/Service Team and is the service team lead. Chris holds an M.S. degree in Software Development from Marist College of Poughkeepsie, N.Y.

Nicholas Carbone is a Senior Software Engineer in Language Environment Development at IBM Poughkeepsie. His focus areas include the design and development of the Common Execution Library (CEL) and C/C++ Run-Time Library (CRTL). Nick has been with IBM for 20 years.

Bryan Childs is an Advisory Software Engineer in the z/OS Global Resource Serialization (GRS) Development/Service Team and is the development team lead. Bryan holds an M.S. degree in Software Development from Marist College of Poughkeepsie, N.Y.

Michael Cox is an Advisory Software Engineer in z/OS UNIX Development at IBM Poughkeepsie, where he specializes in the logical file system component.

Jodi Everdon is a Staff Software Engineer at IBM Poughkeepsie. Besides being a managing editor for z/OS Hot Topics Newsletter, she is the Information Planner for Distributed File Service and the Subcapacity Reporting Tool (SCRT), and is the author of z/OS Problem Management.

Nancy Finn-DeNapoli is an Advisory Software Engineer in z/OS Integration Test at IBM Poughkeepsie. Her focus areas include VSAM RLS and CICS TS, enhancing workloads, and building skills for multiple test areas. Nancy has been with IBM for 24 years.

David Follis is a Senior Software Engineer. He is the WebSphere Application Server for z/OS Runtime Architect, responsible for exploitation of z/OS qualities of service by the application server. David has worked for IBM for 18 years, starting in z/OS development and moving on to WebSphere.

Bob Fuga works in z/OS System Verification Test, specializing in the I/O subsystem, HCD, and new processor verification test. He previously worked in Engineering System Test on new processor delivery to IESP/ESP customers. Bob has been with IBM for more than 25 years.

Glen Garrison is a Senior Software Engineer. He has been with IBM for 21 years, starting in the PC division at Boca Raton, but spending the last 20 years in z/OS and MVS world-wide customer support at IBM Poughkeepsie. Glen supports the RSM and ASM components.

Evan Haruta is an Advisory Software Engineer in the Systems and Technology Group at IBM Poughkeepsie. He works in z/OS Level 2 for the DIV, ASM, and RSM components.

Kevin Kelley is a Senior Software Engineer in z/OS Core Component Development at IBM Poughkeepsie. He has worked in consoles design for almost 30 years. Among Kevin’s many interests is a life-long fascination with herpetology. He owns a Carpet Python named Percival.

Kathy Kulchock brings more than 20 years of mainframe experience to her role on the zSeries Product Information Team at IBM Poughkeepsie. She started her career as an IBM Systems Engineer (SE), and her later assignments included software services development, product marketing, marketing communications, and market research.

Laura Kunioka-Weis is a Senior Software Engineer in DB2 for z/OS Utilities at IBM Silicon Valley Lab in San Jose, CA. She specializes in backup and recovery, and has more than 20 years of experience in DB2.

Joel Masser works in DFSMS System Verification Test at IBM Silicon Valley Lab in San Jose, CA. In his ten years with IBM, he has worked in Test, Performance, Service, and Development. Joel’s current projects include developing tools and techniques for testing, problem detection, and problem determination.

Renata Rand McFadden is a Senior Software Engineer and team lead for the DFSMS/SMB Development and L3 Team. She has worked for IBM for 10 years in z/OS Development. Her projects include Shell and Utilities, Language Environment, FCT, and SMB. Renata holds an MBA, as well as Master’s degrees in computer science and psychology.

Kershaw Mehta is an IBM Senior Technical Staff Member (STSM) in the System z Strategy and Design Group at IBM Poughkeepsie. He has development experience in z/OS UNIX, Language
Environment, and Unicode Services and is now responsible for their future direction and strategy.

**Ed Mekeel** is the content manager for the BCP and Service Aids, and a member of the zSeries Software Portfolio Management department at IBM Poughkeepsie. His background includes five years in computer operations and seven years in automated operations software development. Ed has been with IBM for 37 years.

**TJ Morrissey** is a Software Engineer with IBM Systems and Technology Group at IBM Poughkeepsie. He specializes in software support for z/OS Distributed File Service (DFS/SMB/ZFS).

**Danny Nguyen** is a Staff Software Engineer at IBM Silicon Valley Lab in San Jose, CA. He is an IMS Fast Path and DLIModel Utility developer, and holds a B.S. degree in computer science from San Jose State University.

**Lisa Noval** works in z/OS UNIX System Services Kernel Development and Service at IBM Poughkeepsie.

**Kathy Pfeiffer** is an Advisory Software Engineer in Systems and Technology Group at IBM Poughkeepsie. She is the curriculum manager for the IBM Academic Initiative System z9 Program.

**Eysha Powers** graduated from the University of Illinois at Urbana-Champaign in 2003. She has been with IBM for 2.5 years as a software engineer in z/OS Distributed Middleware as a tester/developer for Java and C. Eysha is pursuing her M.S. degree in Software Design from Rensselaer Polytechnic Institute (RPI) of Troy, N.Y.  

**Ahilan Rajadeva** is a Staff Software Engineer, working in z/OS UNIX File System and Kernel Development at IBM Poughkeepsie. Ahilan holds an M.S. degree in Computer Science from Rensselaer Polytechnic Institute (RPI) of Troy, N.Y.

**Regine Reichert** is a member of the HCD/HCM development team. She has worked at the IBM Boeblingen Lab since 1991.

**Bill Schoen** is an IBM Senior Technical Staff Member (STSM). He works in z/OS UNIX File System and Kernel Development at IBM Poughkeepsie.

**Elpida Tzortzatos** is a Senior Software Engineer in the z/OS Core Technologies department at IBM Poughkeepsie. She leads the storage management teams, and was instrumental in the delivery of 64-bit support for the z900 processor. Elpida joined IBM in 1989 with a B.S. in Mathematics and Computer Science from New York University. In 1996, she obtained an M.S. in Software Engineering.

**Marna Walle** is an IBM Senior Technical Staff Member (STSM) in the z/OS Build and Installation department at IBM Poughkeepsie. Her current responsibilities include z/OS release migration and new installation technologies of the z/OS platform. Marna is a frequent speaker on z/OS installation topics at SHARE and other customer conferences.

**Kevin Wang** is an Advisory Software Engineer at IBM Silicon Valley Lab in San Jose, CA. He is the development team lead for the DLIModel Utility product in the IMS organization. Kevin holds B.S. degrees in Computer Science and Electrical and Computer Engineering from the University of Wisconsin-Madison.

**Dieter Wellerdiek** is a former system programmer. In 1999, he joined the WLM/ SRM Design and Development team at IBM Boeblingen, where he is responsible for the storage management related functions in SRM.

**Glenn Wilcock** is a Senior Software Engineer and an architect for DFSMSshsm Development at IBM Tucson. He has 16 years of experience in developing storage management products with an emphasis on Parallel Sysplex and continuous data-protection.

**Frank Yaeger** is a Senior Programmer with more than 20 years experience in developing functional enhancements to DFSORT and DFSORT ICETOOL. Frank provides DFSORT and ICETOOL solutions in response to how-to questions on online help boards and by direct e-mail. He also helps customers migrate, and is responsible for the content of the DFSORT Web site and publications.

---


International Business Machines Corporation
Department 55JAJ, Mail Station P181
2455 South Road Poughkeepsie, NY 12601-5400
United States of America

Produced in the United States of America
02-07
All Rights Reserved

The z/OS Hot Topics Newsletter is published twice yearly as a service to customers, providing articles of interest to the z/OS community.

The information in this Newsletter is presented as is and is based on the best knowledge of the authors. No warranty is provided (either expressed or implied).

IBM may not offer the products, features or services discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM representative for information on the products or services available in your area.

IBM, CICS, DB2, DB2 Universal Database, DFS, DFSMSdss, DFSMSrmm, DFSORT, Domino, DS8000, ESCON, FICON, FlashCopy, GDPS, Geographically Dispersed Parallel Sysplex, HyperSwap, ibm.com, IMS, Language Environment, Lotus Notes, MVS, OS/390, Parallel Sysplex, Rational, Redbooks, RETAIN, RMP, System z, System z9, WebSphere, z/Architecture, z/VM, z/OS, zSeries, and z/ are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Need a little help? Try the IBM Education Assistant

IBM Education Assistant provides a new way to obtain timely information on the latest releases of z/OS. To help make your job easier the site contains:

- Narrated presentations
- Show me demonstrations
- Tutorials
- Hints and tips
- More!

There are topics such as “How to get started with digital certificates,” or “Enhanced Preventive Service Planning.” There are also many presentations on installation and migration, availability, service, optimization, application development ease of use, and much more!