z/OS: a smarter operating system for a Smarter Planet

Amy Bildzok
z/OS Product Development Team (PDT) Leader, Project Executive
Amy Bildzok knows that a Smarter Planet is essential to the challenges we face today. As Project Executive leader of the IBM® Power® and z System organizations, Amy knows all about the power of the z/OS® system to help make your programming work harder and smarter for you.

In our lead feature for this issue “A smarter operating system for a Smarter Planet,” Amy points out the cornerstones of the z/OS operating system that has been evolving for over thirty years: investment protection for your workloads, a hybrid technology that is ready to tackle the demands and complexities of today’s distributed environment, a commitment to simplify the way you interact with z System, and of course reliability, availability, and serviceability (RAS), hallmarks of z/OS from the beginning.

For example, in terms of simplification, we have three excellent articles on IBM z/OS System Management (z/OSMF). Introduced as a way to help you more easily manage and administer a mainframe system by simplifying day to day operations and administration of z/OS, z/OSMF is featured in “For immediate deployment z/OSMF software,” a comprehensive “How to” article “How to perform z/OSMF Software Deployment,” and “What’s new in z/OSMF V1R13” that provides an update on the latest enhancements.

And speaking of z/OS V1R13, we have focused on the functions and features of that release offering the best in performance and workload management. “zBX performance data at a glance” looks at how to keep track of your cross platform monitoring through Resource Measurement Facility Cross Platform (RMF™ XP). As Amy mentions in her article about the nature of hybrid technology in today’s operating environments, you can use DB2® on z/OS for data serving your SAP applications on IBM zEnterprise BladeCenter® Extension (zBX), and RMF XP can help with the end-to-end workload monitoring and management.

Performance improvements that V1R13 offers are featured in a number of other articles as are those on improved recovery for channel paths and resolving unresponsive name queries through Communications Server.

But what about managing your workloads, which are, after all, the heart of your business? “How to move file system management to the users” is another one of our popular “How to” articles. It describes how you can allow users who control the data to mount and unmount a z/OS UNIX System Service file system without the assistance of the system administrator, putting the data in the hands of the people who own it.

In “Calling COBOL: Java on the line” the authors introduce you to z/OS Batch Runtime, a new product in z/OS V1R13 that enables shared access to a DB2 for z/OS subsystem by both COBOL and Java programs, and allows updates to that database to be committed or rolled back across Java and COBOL language boundaries.

Finally, we tell you how to get to all of these new functions and features with Marna Walle’s article about migrating to z/OS V1R13. In that article, Marna anticipates the migration questions for moving to the latest z/OS release and provides the answers to make the going easier. And our own Professor Kimura offers additional migration insights for z/OS in his article.

These are only some brief highlights of another outstanding issue of z/OS Hot Topics, and we are certain you’ll find a lot of help addressing your system programming needs.

Drop us a line at newsletr@us.ibm.com.

The Editors
## Contents: Featured articles

<table>
<thead>
<tr>
<th>Page</th>
<th>Article</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A smarter operating system for a Smarter Planet</td>
<td>AMY BILDZOK</td>
</tr>
<tr>
<td>6</td>
<td>For immediate deployment: z/OSMF software</td>
<td>GREG DAYNES AND KURT QUACKENBUSH</td>
</tr>
<tr>
<td>8</td>
<td>perform z/OSMF Software Deployment</td>
<td>GREG DAYNES AND KURT QUACKENBUSH</td>
</tr>
<tr>
<td>11</td>
<td>Wake up to the new z/OS Java SDK 6.0.1!</td>
<td>CLARK GOODRICH, MARCEL MITRAN, AND THERESA TAI</td>
</tr>
<tr>
<td>13</td>
<td>Now performing! RMF XP</td>
<td>KARIN GENTHER, HARALD BENDER, AND JÜRGEN BAUMANN</td>
</tr>
<tr>
<td>16</td>
<td>Lucky 13: It’s more about good planning when you migrate to z/OS V1R13</td>
<td>Marna Walle</td>
</tr>
<tr>
<td>21</td>
<td>A grab bag of great new stuff! What’s new in z/OSMF V1R13</td>
<td>ANUJA DEEDWANIYA</td>
</tr>
<tr>
<td>28</td>
<td>Calling COBOL: Java on the line</td>
<td>STEPHEN HENKELS, RUTH RAY, AND JODI EVERDON</td>
</tr>
<tr>
<td>34</td>
<td>Rated z: The hottest DVD release from IBM</td>
<td>ROBERT GENSLER, JOE CONNIFF, AND JEFF SUAREZ</td>
</tr>
<tr>
<td>38</td>
<td>Mark your calendar: Upcoming IBM systems education events</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>move file system management to the users</td>
<td>XIAOYAN XIU, AHILAN RAJADEVA, AND DANIEL ACEVEDO</td>
</tr>
<tr>
<td>41</td>
<td>DFSMS on-demand migration</td>
<td>SEAN McMILLEN AND GLENN WILCOCK</td>
</tr>
<tr>
<td>44</td>
<td>The path to recovery: Realizing improved channel path recovery</td>
<td>COLIN D. CHEN, ILENE A. GOLDMAN, AND THOMAS H. ROSE</td>
</tr>
<tr>
<td>48</td>
<td>Houston, we have a solution: Recovering and monitoring name server failures</td>
<td>MICHAEL GIERLACH AND MICHAEL FITZPATRICK</td>
</tr>
<tr>
<td>51</td>
<td>When keeping up with the (z/OS) Joneses is a good thing</td>
<td>ED MEKEEL, MARIANNE HAMMER, AND MARK WISNIEWSKI</td>
</tr>
<tr>
<td>52</td>
<td>Don’t judge a server by its name! IBM Tivoli Directory Server and LDAP Version 3 protocol</td>
<td>JAY BROTTFUEHRER, JONATHAN COTTRELL, AND SAHEEM GRANADOS</td>
</tr>
<tr>
<td>55</td>
<td>ICSF has been working out</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Our contributors</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Corrections and clarifications</td>
<td></td>
</tr>
</tbody>
</table>

**Back cover:**
- Looking to fill a mainframe job? Check z job board!
- Where’s the Mainframe? See how many mainframes you can spot in this issue
A smarter operating system for a Smarter Planet

BY AMY BILDZOK

This year, IBM celebrates its 100th anniversary. Throughout these years, IBM has transformed and reinvented itself in order to meet our customers’ changing demands. IBM started as the Computing-Tabulating and Recording Company in Endicott, NY, and has since become a globally integrated enterprise. Our vision is to enable technology that can transform the way we live. We call this the Smarter Planet.

Just like IBM, z/OS has evolved and reinvented itself. IBM reinvented itself with S/360™ and born with it was the operating system 360, which became z/OS. It started with batch processing, added time sharing, and continued to add along the way to what it has become today—a highly secure, scalable, high-performance enterprise operating system.

Always evolving to meet customer demands

Through this evolution, z/OS continues to provide investment protection for your workload. We have been working to simplify z/OS by adding and extending enabling technologies that you can use for new workloads while improving performance for your existing workloads.

Some of these areas were the subjects of articles in earlier z/OS Hot Topics Newsletters, but many have not. If you take some time to see what we have added over the past several releases, you might see how many of these things can help you make your IT business more efficient and responsive to your business units.

A good place to begin is with the last few versions of z/OS Introduction and Release Guide, GA22-7502. Take a look at them to see what you can leverage to reduce cost, simplify management, and streamline administration and operations.

Hybrid technology for a Smarter Planet

The combination of z/OS and zEnterprise transforms the way we look at computing environments. System z® delivered a unique hybrid technology with the versatility of multiple architectures. It was combined with a single HMC-based set of management functions that span the entire ensemble. This system (of systems) consists of an IBM zEnterprise 196 (z196). The IBM zEnterprise System delivers choice—it lets you choose the best platform for each workload tier with policy-based workload management. Running z/OS on zEnterprise servers, you can move your applications closer to core data.

For example, you can use DB2 on z/OS for data serving with Software Application Programming SAP applications on IBM zEnterprise BladeCenter Extension (zBX) with end-to-end workload monitoring and management. We plan to enhance the z/OS environment to allow you to optimize elements of your workloads using accelerators and appliances where it makes sense. This can enable workloads, such as Business Analytics, to run in new ways on z/OS providing the ability to host both the operational and analytic data. Along with the support for new workloads, z/OS continues to improve performance of key existing workloads, having provided release-to-release performance improvements in many releases.

Performance data is based on IBM Large System Performance Reference (LSPR) data.

Note: All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Check out “Now performing: RMF XP” on page 13.

Simplification

As our IT environments become more complex and difficult to manage, we look for ways to simplify them. IBM announced the z/OS Management Facility (z/OSMF) to provide a modern, web browser-based management console to help system programmers more easily manage z/OS by simplifying many day-to-day operations. z/OSMF simplifies task such as:

- Gathering data for diagnosis and problem determination
- Configuring TCP/IP network and security
- Monitoring performance and workload management
- Handling capacity provisioning policy management
- Managing disk storage volume
- Cloning z/OS images.
These improvements can help simplify systems management, improve system programmer and operator productivity, and make the functions easier to understand and use. This is especially useful when training new systems programmers and operation staff.

See “For immediate deployment: z/OSMF software” on page 6 and “What’s new in z/OSMF V1R13” on page 21.

RAS and more!
IBM continues to enhance and extend z/OS capabilities to deliver the foundational requirement for reliability, availability, security, performance, and scaling that you need to support your workloads and application environments. The continued release to release improvements in z/OS are extensive across the entire system including I/O, networking, Parallel Sysplex®, scheduler, storage management, and many others too numerous to name. Some of these improvements require no effort or even awareness to receive the potential new benefit. In other cases, you might notice an increase in overall system performance.

Other operating system improvements are accomplished through the delivery of entirely new technologies like z/OS FICON® discovery and auto configuration (zDAC). IBM Health Checker for z/OS delivers new checks and also migration checks, Predictive Failure Analysis (PFA) technologies, and Runtime Diagnostics. The ability for the z/OS environment to provide self monitoring and adjustment for overall system performance and predict system behaviors, and take preventive action is critical going forward. This provides your business the ability to reduce the cost of managing your systems.

Be sure to see References at the end of this article where you can find more information about these z/OS enhancements.

Application enablement and z/OS
z/OS is also being enhanced to extend the value of your existing applications. z/OS V1R13 and z/OSMF V1R13 are planned to include the foundations for batch modernization. Our goal is to dramatically reduce, or even remove, the need for batch windows while simplifying batch programming and adding flexibility. z/OS Batch Runtime is planned to be the foundation for a powerful, integrated, and modern batch application development, deployment, and runtime environment.

While IBM celebrates its 100th anniversary through reinvention, take the time to look at the new things z/OS has to offer and reinvent your world using these new z/OS capabilities.

References
• z/OS Problem Management, G325-2564-08
• “Classified: Highly effective system programmers” in z/OS Hot Topics Newsletter Issue 24, February 2011, GA22-7501-20
• “Predict to prevent: Let PFA change your destiny” in z/OS Hot Topics Newsletter Issue 23, August 2010, GA22-7501-19
• z/OS Hot Topics Newsletter Issue 21, August 2009, GA22-7501-17 for articles about z/OS simplification
• z/OS Hot Topics Newsletter Issue 18, February 2008, GA22-7501-14 for articles about RAS.
Software deployment of z/OS can seem like a daunting task. But with IBM z/OS System Management (z/OSMF) and the new z/OSMF Software Deployment function available in z/OSMF V1R13 and z/OS V1R13 it’s just gotten simpler.

This article explains the z/OS software deployment process and how z/OSMF Software Deployment provides ways to manage the process. In “How to perform z/OSMF Software Deployment” on page 8 you can find step by step instructions about how to implement z/OSMF Software Deployment.

Current z/OS software deployment environment

One of the strengths of the z/OS platform is the ability to share the installed software, even configuration files, among multiple running instances of the operating system. The complete software installation process for z/OS consists of several steps, from planning, acquiring and installing software, to configuring, activating and running the software. Often, you install software once, but configure, activate, and run the software multiple times as the software progresses through its life cycle in different environments, from test and sandbox systems, development and quality assurance environments, and then finally into production.

While you can share an installed instance of software across the different life cycle environments, it is more typical to have unique copies of the software for each of the different environments. Each of these copies might be set up to be maintained by SMP/E. Alternatively, you might have one copy that is maintained using SMP/E, and then copy the software to the different environments.

Over the years, many people have developed their own procedures and tools to copy instances of software from one environment to another. While most of these procedures are successful, occasionally you might run into trouble caused by copying incomplete instances of installed software. In addition, sometimes the ability to exploit new technologies is not possible because those copy procedures do not support the technologies.

z/OSMF to the rescue

IBM z/OSMF V1R13 and z/OS V1R13 are introducing the z/OSMF Software Deployment function to allow you to copy an instance of SMP/E installed software. The deployable unit of software, called a software instance, consists of an SMP/E global zone, one or more target zones, the associated target libraries and target z/OS UNIX System Services file systems, and optionally the associated SMP/E DLIB zones and distribution libraries. When the DLIB zone and associated DLIB data sets are included as part of a software deployment you can maintain the software instance using SMP/E.

A software instance can also include non-SMP/E managed data sets. For example, you might include data sets that contain non-SMP/E installed software or configuration data sets associated with the SMP/E installed software when you deploy a software instance.

During a deployment operation, a source software instance is copied to a target software instance. (Figure 1.) Deployment scenarios: copy or clone

It is never a good idea to install software maintenance or new product releases directly on the software instance for a running system. Instead, it’s better to perform install operations on a copy, or a clone, of the software instance. You can use z/OSMF Software Deployment to make that copy of a software instance in preparation to upgrade one or more of the installed products in that instance either to a new product release level or a higher maintenance level. For this type of deployment, it’s important to have z/OSMF copy the DLIB zone and libraries for the software instance.

You can also use z/OSMF Software Deployment to copy a software instance to run in a different environment, like test, development, QA, production, and so forth. In these types of deployment, copying the DLIBs is optional. If you want the ability to perform SMP/E.
install operations on the new target software instance, you must deploy the DLIB zones and distribution libraries. If you want to create an execution instance only, don’t include the DLIB zones and distribution libraries in the deployment. If you don’t include the DLIBs, you must maintain the source software instance and redeploy it after you install any service, USERMODs, or product upgrades.

Requisites and regressions
Whenever you activate a new software level, it is always best to satisfy requisites and ensure nothing will be regressed. Unfortunately this kind of checking can be complicated and therefore is not always performed. The good news is that z/OSMF Software Deployment performs this kind of checking and helps you to identify missing requisite SYSMODs and SYSMODs that will be regressed.

For example, verify that any software instance sharing resources with the new target software instance has all required coexistence, migration, and fallback maintenance installed. This includes any preconditioning PTFs needed in a sysplex configuration. When you deploy software you also need to verify that conditional requisites are satisfied across software instances. ++IF REQ statements in the source software instance might identify PTFs required in other instances because of SMP/E FUNCTIONs installed in those other instances. These requisite PTFs can be for instances that are to share resources with the target instance, or instances form earlier levels.

Product dependencies
Similarly, when you upgrade to a new software release, verify that functional dependencies among different products are satisfied. Sometimes when you upgrade a product (for example, z/OS) you need to install PTFs on other products that need to run with the upgraded product for them to continue to run successfully. Many of these requisites are identified by SMP/E Fix Category HOLDDATA; however, other dependencies are identified by ++IF REQ statements. Therefore, you need to verify that both the fix category HOLDS and ++IF REQs are satisfied across all software instances that contain products that are to interact with and run on the same target system with the target software instance.

Environment dependencies
When you deploy software to a new environment, you want to install and test the required PTFs before you deploy the software instance. For example, if your test environment runs on an IBM System z10™ Enterprise Class (z10 EC™) system but your production system is an IBM zEnterprise 196 (z196), you ensure all PTFs required for a z196 server are installed before you deploy the software instance. Similarly, if you don’t use extended addressing volumes (EAV) on your source system, but plan to use them on your target system, you ensure all the required EAV maintenance is installed prior to deployment. z/OSMF Software Deployment can assist with all this requisite checking.

Potential service regressions
Finally, when you are deploying a maintenance upgrade or a new software release, verify that any service you apply (particularly PTFs and APARs to fix a problem) is installed or superseded in the software instance. Again, z/OS Software Deployment can help by comparing the source software instance with the software instance that reflects the earlier level of software that is running. z/OS Software Deployment identifies any SYSMODs that are installed in the earlier level and currently running software instances that are applicable to the source software instance.

In a similar way, z/OS Software Deployment is able to analyze any ++IF REQs that were used to identify PTFs needed on a later release if you installed a PTF on an earlier release. z/OSMF Software Deployment identifies any of the later level PTFs that are not installed in the source software instance. Before deployment, you can then install the appropriate PTFs in the source software instance.

In all these cases, the requisite and regression checking performed by z/OSMF Software Deployment helps make it safer for you to deploy software and reduces your exposure to problems because of missing requisite or regressed SYSMODs.

z/OSMF Software Deployment steps
For all deployment operations, z/OSMF Software Deployment leads you through the following checklist of steps:

1. Select the software instance to deploy.
2. Select the objectives for the deployment.
3. Check for missing requisite SYSMODs and regressed SYSMODs.
4. Configure the target software instance.
5. Validate the deployment configuration against the target system, summarize the deployment actions, and generate the deployment jobs.
6. Run the generated jobs to copy the source (this step is performed outside of z/OSMF Software Deployment).

The deployment checklist on page 8 is straightforward and codifies successful deployment practices. Now it’s time to check out the How to article “How to perform z/OSMF Software Deployment” on page 8 where each of these steps will be discussed in more detail.
perform z/OSMF software deployment

BY GREG DAYNES AND KURT QUACKENBUSH

As described in “For immediate deployment: z/OSMF software” on page 6, the z/OSMF Software Deployment task leads you through a guided checklist of steps:

1. Select the software instance to deploy.
2. Select the objectives for the deployment.
3. Check for missing requisite SYSMODs and regressed SYSMODs.
4. Configure the target software instance.
5. Validate the deployment configuration against the target system, summarize the deployment actions, and generate the deployment jobs.
6. Run the generated jobs to copy the source (this step is performed outside of z/OSMF Software Deployment).

1. Select the software instance to deploy
   In this step you select the software instance to deploy. This software instance that will be copied is known as the source software instance. The source software instance might be an instance that was created as the result of SMP/E installation activities, or it might be an instance that has been running and was previously deployed.

2. Select the objectives for the deployment
   The objectives for a deployment define certain characteristics for the deployment operation, such as selecting which system the source software instance should be copied to, whether a new target software instance should be created or an existing software instance should be physically deleted and replaced, which global zone to use for the target software instance, and whether the DLIB zones and distribution libraries should be included in the deployment operation.

   If you want to reuse the DASD space that an existing software instance has used either because that instance contains obsolete software, or as part of your existing procedures, you rotate through a set of software instances, and choose to delete and replace that instance during the deployment.

   Remember to verify that the software instance being replaced is not in use! If you want to manage acquisition and cleanup of SMP/E SYSMODs in the new target software instance separately from other software instances, choose to create a global zone. Otherwise, you can choose to connect the new target and DLIB zones to an existing global zone that you select.

   Finally, if you want the ability to perform SMP/E installation operations on the new target software instance, choose to deploy the DLIB zones and distribution libraries. If, instead, you want to create an execution instance only, do not include the DLIB zones and distribution libraries in the deployment.

3. Check for missing requisite SYSMODs and regressed SYSMODs
   In this step, you select other software instances that share resources, run and interact on the same system, or are being replaced. z/OSMF Software Deployment then analyzes Fix Category HOLDDATA and ++IF REQs to identify potential missing requisites or possible regressions. In addition, z/OSMF Software Deployment compares SYSMODS with SYSTEM or USER HOLDDATA that were installed in the source software instance to the SYSMODs in the software instance being replaced to produce the HOLDDATA report that shows the deltas. This report identifies SYSTEM HOLDS that now might need to be resolved in a different environment.

   For a description of the types of requisite and regression testing that z/OSMF Software Deployment helps with, see “For immediate deployment: z/OSMF software” on page 6.

Six easy steps to z/OSMF Software Deployment
4. Configure the target software instance.

When you configure the target software instance, you define the layout for the data sets of that instance. That is, you define the data set names, the volumes where data sets will reside, and if, how, and where data sets are to be cataloged in the target software instance. With z/OSMF Software Deployment, you can easily model the data set names for the target software instance and the placement of the names after some other software instance, or you can take the default to pre-populate the configuration with the same names and placement as the source software instance.

For example, if your test source software instance uses 3390-9 volumes, but your production system uses 3390-27 volumes, you can model the configuration of the target software instance after the existing production software instance to use the same volume placement strategy. Of course, you can also adjust and fine-tune the data set names and placement as you see fit.

It is common when deploying a software instance that contains a release of z/OS to copy data sets from one volume to another and keep the same data set names, with minor exceptions for VSAM and zFS data sets. For subsystems or other licensed program software you might want to rename all of the data sets, perhaps adding or changing a middle level qualifier, or even a high-level qualifier. The data sets in the target software instance can be targeted to specific volumes or even assigned to an SMS Storage group. In both cases, defining the configuration for the target software instance can be easily accomplished by using z/OSMF Software Deployment.

After the data set names and placement are set, define if and how you want the data sets to be cataloged on the target system. z/OSMF Software Deployment analyzes the target system to determine the default catalog for all data sets that are to be deployed in the target software instance. You can choose to accept the default catalog, or for those data sets that allow it, you can choose for them to be uncataloged. In addition, for data sets with a high-level qualifier that is new on the target system, you can even choose to catalog them in a new or preexisting user catalog.

For those data sets that are to be cataloged, z/OSMF Software Deployment can help you exploit the successful practice of using indirect catalog entries. That is, you can specify a system symbol value, for example &SYSR1 or &SYSR2, and Software Deployment ensures that the catalog entries use the specified symbol for the indirect volume reference.

5. Validate the deployment configuration against the target system, summarize the deployment actions, and generate the deployment jobs.

After you configure the target software instance, z/OSMF Software Deployment compares the configuration to the actual target system to ensure that the deployment operation can succeed and that there are no unintended collisions of data sets on volumes and entries in catalogs. In addition, the deployment jobs are created and summary reports are produced that identify all of the impacts planned for the target system.

The summary reports show you what will be done when the generated deployment jobs are run. Specifically, the reports show the following information:

- SMP/E zones and their SMPCSI data sets to be created, the target volumes to be affected, and how the volumes will be affected
- Data sets to be added, deleted, and replaced on the target system
- Target catalogs that will be affected
- New catalog aliases to be created, if any
- Catalog entries to be added, deleted, and replaced.

By using these summary reports, you can see the anticipated results of the deployment operation and, if necessary, change the configuration. After a deployment is complete, the summary reports can be reviewed to provide a history of the deployment, which you might want for audit or problem determination purposes.

6. Run the generated jobs to copy the source.

The deployment operation (copying data sets, creating catalog entries and updating SMP/E zone entries) is performed not by z/OSMF itself, but rather by running a set of jobs generated by z/OSMF Software Deployment. The jobs are written to a partitioned data set, where each job is a separate member. The member name identifies the sequence in which the jobs should be run; there is also a “read me” file member that reinforces the job sequence, provides a description of what each job does, and documents the anticipated return codes. After you run each job, you might want to save or archive the output to keep with the deployment history for audit or problem determination purposes.

Cross system deployment

For those of you that are familiar with z/OSMF, you’ll know that there can be only one instance of z/OSMF active on one system in a sysplex at a time. Assuming that you share DASD that is accessible from that z/OSMF system with all other members in the sysplex, it’s easy to see how, from that system, you can deploy software to DASD that any system in that sysplex can use.

z/OSMF V1R13 introduces cross-sysplex z/OSMF to z/OSMF communication for software deployment. You can have one z/OSMF instance running in each sysplex, and one of those z/OSMF instances will be the system that you connect your browser to for all software deployment activities. This is the primary z/OSMF instance. Other z/OSMF instances running in other sysplexes are called secondary z/OSMF instances.

By using HTTPS communication between the z/OSMF primary and secondary instances, you can deploy software from one sysplex to another, throughout your enterprise. See Figure 1.
Figure 1 shows three different sysplexes with three different instances of z/OSMF. The primary z/OSMF instance controls the deployment operations; that is where your browser connects. The secondary z/OSMF instance in the sysplex where the source software instance resides is known as the source system, and the secondary z/OSMF instance in the sysplex where the target software instance will be created is known as the target system.

Deployment topologies
Deploy a source software instance that is accessible from the primary z/OSMF instance to one of the following target software instances:

1. A target software instance that is accessible from the primary z/OSMF software instance. For example, source software instance A is accessible to target software instance X. This is a local software deployment.

2. A target software instance that is accessible from the secondary z/OSMF software instance. For example, source software instance A is accessible to target software instance Y in Sysplex B, or to target software instance Z in Sysplex C. These are remote software deployments.

3. Deploy a source software instance that is accessible from a secondary z/OSMF instance to a target software instance that is accessible from the same secondary z/OSMF instance. For example, source software instance B in Sysplex B is accessible to target software instance Y in Sysplex B, or source software instance C to target software instance Z in Sysplex C. These are local software deployments.

4. Deploy a source software instance that is accessible from a secondary z/OSMF instance to a target software instance that is accessible from the primary z/OSMF instance. For example, source software instance B in Sysplex B is accessible to target software instance X in Sysplex A, or source software instance C in Sysplex C is accessible to target software instance X in sysplex A. These are remote software deployments.

Six easy steps to z/OS software deployment
As you can see z/OSMF V1R13 Software Deployment is a powerful tool that can help you deploy SMP/E managed software throughout your enterprise. It should integrate well with your existing procedures, providing data needed for audit trails or debugging. Moreover, it should make it safer for you to deploy software, reducing your exposure to software errors.
Wake up to the new z/OS Java SDK 6.0.1!

Workload optimization with zEnterprise z196 technology innovations

BY CLARK GOODRICH, MARCEL MITRAN, AND THERESA TAI

On March 15, 2011, the announcement of IBM for z/OS Java software development kit (SDK) Technology Edition Version 6, Release 0, Modification 1 (z/OS Java SDK 6.0.1) unveiled a comprehensive 31-bit and 64-bit runtime for the z/OS platform. This new delivery provides an SDK for z/OS, designed to be compliant with the Java Standard Edition (SE) 6 application programming interface (API) and comes packaged with new z/OS integration features, as well as significant performance enhancements.

z/OS Java SDK  6.0.1 includes:

• J9 Java Runtime Environment (JRE)
• Enhancements to JZOS
• Support for AES secure keys of the IBMJCECCA hardware crypto provider
• Enhanced Integrated Cryptographic Service Facility (ICSF) exception handling
• Reliability, availability, and serviceability (RAS) features
• New signal handling capability.

The J9 2.6 JRE includes a new balanced garbage collection (GC) policy for large heap-intensive workloads and significant just-in-time (JIT) compiler performance enhancements. The JZOS batch toolkit improvements include support for Workload Manager (WLM) services.

Other times we boast

The z196 boasts an impressive list of new features that are designed to be beneficial to Java performance. More specifically, for Java programs that typically demonstrate heavy data-cache dependencies, the out-of-order pipeline, larger caches, and improved large pages are features that represent an important advantage.

Beyond improvements to the micro architecture, z196 includes a significant set of new architectural facilities. These new facilities include:
• Non-destructive facility that is used by the JIT to improve path-length of JITED code
• High-word facility that is used to extend the effective register set of the core
• Condition-load and store facility that is used to mitigate the cost of poorly predicted and expensive branches.

Just in time to show off

z/OS Java SDK  6.0.1 packages the J9 2.6 runtime environment. J9 2.6 includes a new JIT compiler and GC. In addition to z196 instruction exploitation and scheduling, the JIT optimizer also includes a large set of enhancements to help improve data locality and instruction path-length. These optimizations include more aggressive inlining, pre-fetching and escape analysis. Beyond this, additional improvements to loop alignment and other typical code sequences were included in the new JIT.

and responsiveness for Java workloads on IBM workloads on IBM zEnterprise 196 (z196) in combination with specialty processors, System z Application Assist Processor (zAAP) and System z Integrated Information Processor (zIIP), which might further improve the economics of running mission critical workloads on System z.

At times we brag

Internal IBM laboratory tests have shown that z/OS Java SDK 6.0.1 provides outstanding performance improvements with significant throughput improvements compared to Java 6 SR7 running on IBM System z10. These internals tests also show unmatched optimized throughput
The GC has a new policy that is designed for enhanced responsiveness for large heap applications. The balanced policy is designed to achieve more consistent application behavior by reducing maximum pause times. The GC uses incremental result-based collection and targets best return-on-investment areas of the heap.

The balanced policy is recommended for applications that:

- Require heaps that are greater than 4G
- See frequent collections
- Spend an excessive amount of time in global compaction
- Allocate many large objects.

The GC policy was changed from OPTHRUPUT to GENCON in z/OS Java SDK 6.0.1.

Time is on our side
Figure 1 shows the multi-threaded workload from internal testing that demonstrates an impressive 2.1x aggregate throughput improvement running a multi-threaded 64-bit Java workload on a 12-way System z, using large pages (LP) and compressed references (CR) options.

Moving from the bottom line in the graph toward the top, we first see a 17% improvement measured between System z10, running z/OS V1R11 with Java 6 SR7 and System z10 running z/OS V1R12 Java 6 SR8. This 17% improvement was the result of a new z/OS V1R12 IEASYxx TIMESLICE feature.

Moving higher on the graph we see a 56% hardware improvement between System z10 and z196 both running z/OS V1R12 Java 6 SR8.

At the top of the graph we see an additional 16% software improvement between Java 6 SR8 and z/OS Java SDK 6.0.1, both running on z196 z/OS V1R12. At 12 threads this Java workload is using well over 90% of the processor available on a 12-way z196.

Figure 1. z196 and System z 64 bit throughput with SDK6

Figure 2, also from internal testing, shows an outstanding 7x aggregate improvement over two generations of System z hardware and IBM Java for the same multi-threaded 64-bit Java. This graph shows different levels of Java and z/OS running on a 16-way z9, 16-way z10, and 16-way z196.

The bottom line in the graph is System z9 with z/OS V1R9 and Java 5 SR5. The top line shows Java SDK 6.0.1 on z196. In this graph, both Java SDK 6.0.1 and Java 6 SR4 exploited System z10 and z196 LP hardware and take advantage of the CR option that was not supported in z/OS Java 5.

Figure 2. z196 and System z 64 bit throughput with J9 JRE

Step forward
z/OS Java SDK 6.0.1 represents another exciting step forward. To find out more about z/OS Java SDK 6.0.1 and the latest JZOS enhancements, see:

- [ibm.com/systems/z/os/zos/tools/java/products/sdk601_64.html](http://ibm.com/systems/z/os/zos/tools/java/products/sdk601_64.html)
- “Launch into JZOS! Migrating from JRIO to JZOS” on page 30.
Do you want to keep track of one or more IBM zEnterprise BladeCenter Extension (zBX) and performance? Then Resource Measurement Facility Cross Platform (RMF XP) is your choice for cross platform monitoring!

RMF XP provides an integrated performance monitoring solution for heterogeneous environments by currently supporting the operating systems:

- AIX®
- Linux on System x®
- Linux on System z.

Hence, with RMF XP, you can monitor all operating systems which can run on an IBM zEnterprise System, including the zEnterprise BladeCenter Extension.

RMF XP exploits the existing Common Information Model (CIM) instrumentation for the AIX and the Linux operating systems. The CIM server and the metrics providers are already an integral part of the supported AIX and Linux distributions; therefore no additional proprietary software needs to be installed. You have to ensure that the CIM environment is properly set up and running on the monitored endpoints. RMF XP can communicate with two different CIM server implementations: The Open Pegasus CIM Server and the Small Footprint CIM Broker (SFCB).

Performance data at a glance!
The core component of RMF XP is the GPM4CIM server. Similar to the existing Distributed Data Server (DDS) for z/OS, the GPM4CIM server receives HTTP requests and sends back responses as structured XML documents. Because the GPM4CIM started task runs in the z/OS UNIX System Services environment, at least one z/OS system is necessary to run the RMF XP component.

No rehearsal
To start the GPM4CIM server from the console, RMF provides the procedure GPM4CIM as a member in SYS1.PROCLIB, as the JCL example in the GPM4CIM PROC shows:

- The GPM4CIM environment variables for BPXBATCH are kept in the file gpm4cim.env (STDENV DD card).
- The log and trace output is written to the files specified with the STDOUT and STDERR DD cards.
- The ‘cfgs’ program parameter in the PARM statement points to the GPM4CIM configuration file.
- Different platforms are distinguished by the variable added to the OS statement:
  - OS=A (AIX on System p®)
  - OS=X (Linux on System x)
  - OS=Z (Linux on System z).

Many of the configuration parameters are well-known from the GPMSERVE procedure, which are called options in that environment.

Here’s the JCL for procedure GPM4CIM:

```sql
//GPM4CIM PROC OS=X
/***********************************************************
* STEP 1 - Execute GPM4CIM                             *
/***********************************************************
//STEP1 EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=0M,
//       PARM="PGM /usr/lpp/gpm/bin/gpm4cim
//       cfg=/etc/gpm/gpm4&OS..cfg"
//STDENV DD PATH='/etc/gpm/gpm4cim.env'
//STDOUT DD PATH='/var/gpm/logs/gpm4cim&OS..out',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP)
//STDERR DD PATH='/var/gpm/logs/gpm4cim&OS..trc',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP)
//...
```
Starring role
The most important parameter unique to GPM4CIM is the name of the system complex, shown in the following configuration file as LNX_COMPLEX. This keyword specifies the top level resource of GPM4CIM's resource model. Comparable to the notion of a sysplex in z/OS, each system complex consists of the systems forming your distributed environment. Furthermore, the frequency of RMF XP retrieving data from the monitored endpoints can be defined with the INTERVAL configuration parameter as in the following example:

```plaintext
/**********************************************/
/* NAME: /etc/gpm/gpm4X.cfg                */
/**********************************************/
... INTERVAL(300) /* Monitoring interval (seconds) */
LNX_COMPLEX(SAPPLEX) /* User defined name of LNX complex */
LNX_IMAGE(xbrmf1.us.ibm.com:5988)
LNX_IMAGE(xbrmf2.us.ibm.com:5988)
LNX_IMAGE(xbrmf3.us.ibm.com:5988)
```

Rate the show
The systems’ configuration data and performance values are returned by GPM4CIM in the same manner as from GPMSERVE. For example, Figure 1 displays the utilization of all logical processors in a XLINUX_SYSTEM_COMPLEX.

Casting call
Beyond the capabilities of an open and standardized API for zBX performance data, as a matter of course, RMF XP can also serve as data source for a modern user interface (UI) with the following functions:

- Users of the Resource Monitoring plug-in for the IBM z/OS Management Facility (z/OSMF) can now display performance data from Linux or AIX systems in the same manner as z/OS performance data.
- The GPM4CIM server supplies a style sheet that allows you to format the XML documents instantly in any Web browser using the RMF Performance Data Portal.

The z/OSMF Resource Monitoring plug-in seamlessly integrates the usage of RMF XP. With the system status task, you maintain an overview list of all z/OS sysplexes and AIX or Linux system complexes that you want to monitor. The Resource Monitoring task

![Figure 1. Linux on System x logical processor utilization](image)
lets you create performance dashboards that provide a custom view of the metrics and display real-time data as bar charts.

z/OSMF gives you the freedom of choice about how to combine metrics in dashboards. For instance, you can create a dashboard that visualizes various performance aspects of a single system. Or you can create a dashboard that provides an overview across all platforms in one browser window. Figure 2 shows an example of process metrics for z/OS and AIX in a dashboard view.

Keep track of your zEnterprise BladeCenter extensions and their performance—with only a few mouse clicks!

Get your tickets now
RMF XP is included in z/OS RMF in z/OS V1R13. For z/OS V1R12, you need the following PTFs:
• APAR OA36030 and OA36486 for RMF
• APAR PM36726 for z/OSMF.

For detailed information about RMF XP and the new GPM4CIM server, see:

For detailed information about z/OSMF, see:
• IBM z/OS Management Facility Configuration Guide, SA38-0652.

Credits
Special thanks go to Peter Mailand for his contributions to GPM4CIM and this article.
Lucky 13

It’s more about good planning when you migrate to z/OS V1R13

BY MARNIA WALLÉ

You might think that thirteen is an unlucky number. Luck won’t play any role, though, if you are prepared for z/OS V1R13. Knowing about some of the most important migration actions coming in z/OS V1R13 will help you avoid any problems. These important migration actions briefly described in this article occur in z/OS V1R13. If you are migrating from z/OS V1R11, remember to also plan for the migration actions introduced in z/OS V1R12.

Reference

As you know by now, all migration actions can be found in z/OS V1R13 Migration, GA22-7499-19. See this book for the complete details on what your migration plans should entail. Also, don’t forget to refer to the ZOSV1R13 ZOSGEN PSP bucket for late-breaking information that didn’t make the book.

Action: Prepare for DCE and z/OS UNIX System Services Connection Scaling removals

z/OS V1R13 marks the removal of the Distributed Computing Environment (DCE), DCE Security Server, and the z/OS UNIX System Services Connection Scaling components (Connection Manager and Process Manager).

What you can do now:

Ensure that you have no applications that are dependent upon the removed functions. For a replacement of DCE, see the DCE Replacement Strategies at the following Redbooks site:


Many target, distribution libraries, and paths are now deleted with the removal of these components that might require clean up later.

Action: Recognize the console operating mode default change to DISTRIBUTED

Before z/OS V1R13, if you did not specify a console-operating mode, you would be using SHARED mode. DISTRIBUTED mode is the new default in z/OS V1R13, and is the strategic mode. If you do not have any mode specified on your CON= parameter in the IEASYSxx parmlib member, DISTRIBUTED mode is used in z/OS V1R13.

What you can do now:

Use migration health check ZOSMIGV1R13_CNZ_CONS_OPER_MODE (from PTF for APAR OA32930) to know if you are using the default mode, and thereby know if the default mode change affects your system.

Action: Accommodate the HIPERDISPATCH default change for z196 on z/OS V1R13

Before z/OS V1R13, the HIPERDISPATCH default setting in the parmlib member IEAOPTxx was NO; however, with more than 64 CPs, the system always uses HIPERDISPATCH. z/OS V1R13 has changed so that when your system is running on a z196, the HIPERDISPATCH default setting is YES.

What you can do now:

If you plan to run z/OS V1R13 on a z196 and you did not specify the HIPERDISPATCH setting in IEAOPTxx, the default will be YES. Verify that you are using the default for HIPERDISPATCH and whether your system is affected by this default change.

Action: Prepare for the VTAM internal trace table (VIT) moving from ECSA to HVCOMMON

As of z/OS V1R13, the VIT data space is eliminated. What you specified on the SPACE specification of the TRACE start option and on the MODIFY TRACE command is what will be used as the size in megabytes of HVCOMMON for the VIT. (Previously your SPACE specification meant the number of pages of ECSA.) The DSPSIZE parameter, which previously specified the size of the VIT, is no longer valid.

What you can do now:

If DSPSIZE is specified on the VTAM start list, it is ignored. If DSPSIZE is specified on the MODIFY TRACE command, the command is ignored. Review what you specified in your VTAM start list and on any command specifications (especially in automation), and make any changes knowing how the VIT will be established in z/OS V1R13. In addition, review the other tasks associated with this migration action in z/OS V1R13 Migration, GA22-7499-19.

Action: Accommodate that zFS file systems might need more DASD space

Before z/OS V1R13, z/OS File System (zFS) would both read and write in 1 K and 8 K blocks. Now in z/OS V1R13, zFS will no longer write in 1 K blocks (but can still read 1K blocks). When zFS updates data read in from 1 K blocks, it converts the data into 8 K blocks. Therefore, it is possible that you need additional DASD space...
Migration action, the exist on the volume. To help with this extend if enough extents and space it to allow the aggregate to dynamically IOEFSPRM today, you can start using OA32925, to verify that all systems are available in the PTF for APAR ZOSMIGV1R13_ZFS_FILESYS, IPL. Use migration health check it active on all systems with a rolling specify all systems, update IOEFSPRM to V1R13. After the PTF is installed on all systems that will coexist with z/OS Install the PTF for APAR OA32925 on

What you can do now:
If you do not use the aggrgrow option in IOEFSPRM today, you can start using it to allow the aggregate to dynamically extend if enough extents and space exist on the volume. To help with this migration action, the aggrgrow default is ON in z/OS V1R13.

To anticipate which file systems might be affected by this change, you can examine existing zFS file systems. The downloadable zfsspace utility reports on small (1K or less) files, and whether the aggrgrow option is on. Download the zfsspace utility from the following site:


This shell command can also be helpful for finding how many files are less than 1k in size:

find <mountpoint> -size -3 -type f -xdev | wc -l

Action: For shared file system environments using zFS, make sure you have sysplex-filesys

If your system is running in a shared file-system environment with zFS, z/OS V1R13 requires that all z/OS V1R11 and z/OS V1R12 systems are running sysplex-filesys before joining the sysplex. sysplex-filesys support is provided in the PTF for APAR OA29619.

What you can do now:
Install the PTF for APAR OA32925 on all systems that will coexist with z/OS V1R13. After the PTF is installed on all systems, update IOEFSPRM to specify sysplex-filesys and make it active on all systems with a rolling IPL. Use migration health check ZOSMIGV1R13_ZFS_FILESYS, available in the PTF for APAR OA32925, to verify that all systems are at sysplex-filesys before introducing z/OS V1R13 into the sysplex.

Action: Prepare for symbolic links automatically provided to make it easier for a read-only version root

We’ve always recommended that you mount the version root file system as read-only. However, to do that required that you add your own symbolic links to three utilities (ucpq, cron, and mail) each time you received a new z/OS release level. As of z/OS V1R13, we provide the necessary symbolic links for these three utilities under the /var directory, so that no extra customization is required by you.

What you can do now:
Use migration health check ZOSMIGV1R13_RO_SYMLINKS, available with the PTFs for APAR OA35605 and OA35636, to assist you in determining if the symbolic links added to z/OS V1R13 will affect your system.

You can start using the symbolic links that we provide in z/OS V1R13 today. We previously described how to customize the three utilities using the /etc directory, but in z/OS V1R13 we describe the customization procedure (and providing the symbolic links for) using the /var directory. If you use the /var directory symbolic links today, you do not need to make any change when you install z/OS V1R13. For a complete list of the symbolic links for these three utilities, provided in z/OS V1R13, see z/OS V1R13 Migration, GA22-7499-19.

Action: Use SMP/E FIXCATs for PTF verification

This task should be business-as-usual by now, but ensure you that you use SMP/E FIXCATs to verify that you have all the PTFs you need in preparation for z/OS V1R13. (There are similar SMP/E FIXCATs for z/OSMF as well, if you use that product.)

What you can do now:
After you receive the latest enhanced HOLDDATA using the SMP/E RECEIVE command, run the SMP/E REPORT MISSINGFIX command with these z/OS V1R13 important FIXCATs:

• IBM.Coexistence.z/OS.V1R13:
  Ensure that you run the command against your z/OS V1R11 or z/OS V1R12 zones to verify you have all the necessary PTFs for z/OS V1R13 coexistence installed.

• IBM.TargetSystem-RequiredService.z/OS.V1R13:
  Ensure that you run the command against all the zones where your other IBM products and subsystems are installed to verify you have all the necessary PTFs for those products to work with z/OS V1R13.

• IBM.Function.HealthChecker:
  Run the command against your z/OS V1R11 or z/OS V1R12 zones to see if you have any missing IBM Health Checker for z/OS health checks. Remember, migration health checks are initially set to INACTIVE, so you must activate them to see what they show.

A helpful enhancement for z/OS V1R13!
Some great enhancements in z/OS V1R13 (or z/OS V1R12) require that you update the IEASYSxx member. Enhancements include the new z/OS V1R13 IXGCNF= and IGGCAT= support. To assist you in allowing those updated IEASYSxx members to be shared between z/OS V1R13 and pre-z/OS V1R13 systems, we added a new WARNUND statement to IEASYSxx. If you put this statement at the top of the IEASYS00 member, you receive a new warning message IEA660f for undefined statements on the lower levels, and the IPL process will continue rather than prompt you for a correct statement. You need to install the PTF for APAR OA35929 on pre-z/OS V1R13 systems to use the WARNUND statement properly on those systems. This makes it so much easier to exploit newer functions, because separate IEASYSxx members do not have to be maintained.

Looking for more on migration? See “Professor Kimura answers migration questions” on page 18.
Professor Kimura answers migration questions

BY SHIGEKI KIMURA

Profesor Kimura has extensive knowledge of z/OS migration, so we’ve asked him to answer some typical questions from system programmers like you. Here are his hints and tips.

Suppressing DELETE/RENAME messages

From the customer:
After we migrated to z/OS V1R11, we started to receive the message IGD17054I (see Figure 1) when an attempt was made to delete a data set that did not exist. In this case, we’re scratching a data set that sometimes spans multiple volumes (although it usually fits on only one volume). On some of our systems, we would like to suppress this message. We set the MPFLSTxx parmlib member to suppress the console display, but is there any way to suppress the messages from being written to the JOBLOG and SYSLOG?

JCL: (There is no extent allocated in the volume WZX002.)

```
//STEP1   EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=*
//SEQOUT   DD DSN=BEANS.TEST.DS1,DISP=(NEW,CATLG),
//         SPACE=(TRK,(15,1),RLSE),
//         DCB=(DSORG=PS,LRECL=133,RECFM=FB,BLKSIZE=0),
//         VOL=SER=(WZX001,WZX002)
//SYSIN    DD *
DSD OUTPUT=(SEQOUT)
FD  NAME=FIELD1,LENGTH=10,FORMAT=ZD,INDEX=1
CREATE QUANTITY=1,NAME=(FIELD1),FILL=X'FF'
/*
//STEP2    EXEC PGM=IEFBR14
//DD1      DD  DISP=(OLD,DELETE),DSN=BEANS.TEST.DS1
```

JOBLOG(JESMSGLG):

```
23.25.21 JOB04713 IGD17054I DATA SET NOT FOUND FOR DELETE/RENAME ON
VOLUME WZX002 888
888         DATA SET IS BEANS.TEST.DS1
```

SYSLOG:

```
JOB04713 00000290 IGD17054I DATA SET NOT FOUND FOR DELETE/RENAME ON
VOLUME WZX002 888
888 00000290 DATA SET IS BEANS.TEST.DS1
```

Figure 1. Message IGD17054I displayed in the JOBLOG and SYSLOG

Professor’s advice:

I have good news for you. Starting in z/OS V1R13, a new parameter SUPPRESS_DRMSGS(Y|N) for the IGDSMSxx parmlib member determines whether SMS suppresses DELETE/RENAME messages. The default is N (NO). It will be processed at IPL or on the issuance of a SET SMS=xx command (where xx is appended to IGDSMS to determine the specific IGDSMSxx member to be used). Also, a new operator command, SETSMS SUPPRESS_DRMSGS(Y|N), lets you modify the value of this parameter between IPLs.

You can specify the SUPPRESS_DRMSGS(Y) parameter explicitly in your IGDSMSxx parmlib member to suppress the IGD17054I message. The change you mentioned was introduced in z/OS V1R10, and it applies to SMS-managed data sets as well as non-SMS-managed data sets. So, the new feature of z/OS V1R13 also applies to both data sets. Good luck.
**Enhanced LLA restart processing**

**From the customer:**
When we omit the SUB=MSTR parameter from an S LLA command, the system issues message CSV209I indicating that the request to start LLA is ended and the system reissues the command with SUB=MSTR. (This was a new feature in z/OS V1R8.) However, during the automatic restart of LLA to include SUB=MSTR, LLA fails to include the LLA=xx specified on the original start of LLA. What can I do?

**Professor’s advice:**
I’m pleased to say that this has been improved in z/OS V1R12 and the LLA= specification is no longer omitted during the automatic restart of LLA. See Figure 2.

**Message Flood Automation (MFA) facility constraint relief**

**From the customer:**
In z/OS V1R11 (and in z/OS V1R10 with the PTF for APAR OA25602 applied), MFA is now fully integrated into the z/OS BCP and no longer requires the use of customer exits. With this new functionality, the MSG statement in the MSGFLDxx parmlib member is now able to define up to 50 messages. That’s great, but we need to define more SPECIFIC messages in the MSG statement.

**Professor’s advice:**
In z/OS V1R13, we allow you to specify up to 1024 SPECIFIC messages on the MSG statement. This is an important enhancement, because customers who might want to prevent MFA from taking action against messages that are critical to their automation might not have been able to do so using the IGNORE option because of the SPECIFIC message limit.

**New RMF Monitor II Report “OPT Settings”**

**From the customer:**
A new employee in our shop asked me if the IEAOPTxx parmlib member would work the same with or without commas. There is no documented restriction regarding the use of commas, but we have no command interface to display the current setting of IEAOPTxx parmlib member to ensure that the intended OPT parameters are in effect. We are a z/OS V1R10 shop, and I can find no such capability in V1R10. Do you know if such an enhancement could be made in a later release? We would find it very helpful.

**Professor’s advice:**
Beginning in z/OS V1R11, RMF provides a new Monitor II Report “OPT Settings” (which is a new selection under “Library Lists”) to display information about the active OPT parmlib member and the settings of all OPT parameters. Note that you need the PTF for APAR OA34872 to display the IEAOPTxx parmlib parameters CCCSIGUR and TIMESLICES (which were added in z/OS V1R12).
A tool called WLM OPT parameter viewer (WLMOPT) can be downloaded to assist you in displaying the OPT settings. However, this tool was stabilized at the z/OS V1R10 level. This means that no OPT parameters introduced after z/OS V1R10 can be displayed with it, and all future OPT parameters will only be made available through the new RMF report.

New authorization to run the SMF dump program

From the customer:
After migrating to z/OS V1R12, the output from the D SMF,O command includes some new information. (See Figure 3.) What does this mean? Is there any migration action for us?

You do not need to explicitly define the exits used by the RACF® SMF unload utility, because they are registered by default (as is shown in the D SMF,O command output).

Note that in z/OS V1R13 (or with the PTF for SMF APAR OA33696 applied to V1R10, V1R11, or V1R12) when IFASMFDP is executed in an unauthorized environment, the exits specified with USER1, USER2, and USER3 are not verified against what is in the SMFPRMxx parmlib member:

Your future coupling facility…
Just an upgrade away!

Are you getting ready to upgrade your coupling facility to a new z196? Perhaps you are adding additional capacity to a central processor complex (CPC) that contains a coupling facility? Or maybe you are upgrading a coupling facility on a z10, which requires a reactivation of the coupling facility image?

Whatever the change you’ll be making — whether it involves reactivating a coupling facility, POR of a CPC with a coupling facility on it, or moving a coupling facility to a new CPC — go to ibm.com/support/techdocs/atmsmar.nl/WeblIndex/WP101905 to obtain and follow the guidance in the “Best Practices for Upgrading a Coupling Facility,” Document ID WP101905, to ensure a simple, fast, and smooth upgrade.

You do not need to explicitly define the exits used by the RACF® SMF unload utility, because they are registered by default (as is shown in the D SMF,O command output).

Note that in z/OS V1R13 (or with the PTF for SMF APAR OA33696 applied to V1R10, V1R11, or V1R12) when IFASMFDP is executed in an unauthorized environment, the exits specified with USER1, USER2, and USER3 are not verified against what is in the SMFPRMxx parmlib member:

Figure 3. Output message IEE967I from D SMF,O command

Professor’s advice:

Beginning with z/OS V1R12 (and in V1R10 and V1R11 with the PTFs for APAR OA29894 applied), in order to allow user exits to be called by the SMF dump programs using the USERx (x=1, 2, or 3) parameter, the exits must now be predefined to the system using the following new keywords included in the SMFPRMxx parmlib member:

• The SMFDPEXIT keyword allows exits to be specified for the IFASMFDP program.
• The SMFDLEXIT keyword allows exits to be specified for the IFASMFDL program.

If you fail to register an exit specified for IFASMFDP(L), you will receive the following error message, and IFASMFDP(L) will stop the processing:

IFASMFDP(L) USER EXIT xxxxxxxxxx NOT REGISTERED WITH SYSTEM.
A grab bag of great new stuff!

What’s new in z/OSMF V1R13

BY ANUJA DEEDWANIYA

z/OSMF V1R13 brings another exciting z/OSMF release, and it’s loaded with new goodies. Look what’s new:

• The Storage category includes a DASD Management task to make it simpler for you to add storage capacity to an SMS pool storage group.
• The Software category includes a Deployment task that lets you clone installed software using a best practices approach.
• The Performance category includes a Capacity Provisioning task allowing you to view status of the Capacity Provisioning Manager (CPM).

Storage made easy

Until now, when you needed to add storage to a group, the storage administrator had to identify volumes available from a list maintained manually, using multiple interfaces with multiple context switches between ICKDSF, ISMF, and other elements. Now, with the V1R13, we introduce Reserve Storage Pools in the DASD Management task, a pool of volumes available for future use. We also introduce a new policy for storage group attributes that makes it easier to add storage to an SMS pool storage group with a single user interface in the z/OSMF Add Storage wizard. The wizard guides you through the steps to add the appropriate amount of storage, update the SCDS, initialize volumes, and optionally vary them online and activate the SCDS.

An eye on performance

CPM keeps an eye on your own installation-defined performance objectives, looking for capacity bottlenecks, helping control processor capacity, and even providing advice on balancing capacity to handle workload fluctuations. Using the Capacity Provisioning task, you can manage connections to multiple CPMs in a central repository. The Capacity Provisioning task also makes analysis of the domain status, active configuration and active policy reports quick and easy by letting you optimize report views using the customizable filter and sort functions.

Rigorous software deployment

The new Deployment task under the Software category provides rigor in deploying SMP/E installed software. See “For immediate deployment: z/OSMF software” on page 6 in this issue for more details.
The new Deployment task . . . provides rigor in deploying SMP/E installed software.

Rest your APIs
z/OSMF V1R13 provides further end-to-end task flow simplification by enabling linking function and context-sensitive launching between z/OSMF tasks and to web applications and products outside z/OSMF. You can link using REST APIs programmatic interfaces or by using the Application Linking Manager task. There’s also a new REST APIs for z/OS Jobs management that let you submit and access job information using the HTTPS interface.

And so much more!
We implemented tighter integration with z/OS System Authorization Facility (SAF) with the introduction of the new resource class ZMFAPLA for z/OSMF task-based resources.

ISP/ enhancements galore
We made ISPF available from within the web-based interface, so you can perform most ISPF tasks from your browser. Although our ISPF interface is still much like the 3270 version, the huge benefit of having ISPF in z/OSMF is that your ISPF functions become URL-addressable. This means you can launch ISPF functions from any other browser-based application, with both z/OSMF tasks and non-z/OSMF web applications. The Incident Log task is the first to exploit this function, allowing you to start directly from the View Diagnostics Details panel into the ISPF Browse data set function for viewing one of the log snapshots.

Another new ISPF feature lets you display up to four panes side by side horizontally or vertically! And the panels can be individually sized with multiple split screen tabs. You can also log into your 3270 session and the z/OSMF based ISPF session simultaneously with profile sharing enabled, and of course multiple z/OSMF logins continue to be supported.

Support in workload management, support for additional endpoints in resource monitoring, application linking exploitation in incident log, and a bunch of Configuration Assistant enhancements for z/OS Communications Server, including the ability to configure multiple z/OS release level policies. (See “Let the Configuration Assistant help you manage multiple stacks” on page 23.)

Grab bag assist
Thanks to Benjamin Kaeckenmeister, Bertold Reddemann, and Cecilia Carranza Lewis for their input.

And so much more!
We implemented tighter integration with z/OS System Authorization Facility (SAF) with the introduction of the new resource class ZMFAPLA for z/OSMF task-based resources.

We’ve made lots of enhancements to existing tasks too, including more granular authorization support in workload management, support for additional endpoints in resource monitoring, application linking exploitation in incident log, and a bunch of Configuration Assistant enhancements for z/OS Communications Server, including the ability to configure multiple z/OS release level policies. (See “Let the Configuration Assistant help you manage multiple stacks” on page 23.)

Grab bag assist
Thanks to Benjamin Kaeckenmeister, Bertold Reddemann, and Cecilia Carranza Lewis for their input.
CA stacks up

Let the Configuration Assistant help you manage multiple stacks!

BY KIMBERLY BAILEY AND SCOTT MOONE

In z/OS V1R13, the Configuration Assistant for z/OS Communications Server introduces new support to make it easier to configure IP Security for multiple stacks. In this article, we’ll show you how to take advantage of this new function.

Reusable connectivity rules
Sometimes it’s important to replicate connectivity rules between multiple stacks. For example, if you have a connectivity rule that protects a distributed Dynamic Virtual IP Address (DVIPA), you might want to repeat it on every stack sharing that DVIPA. Until now there was no easy way to duplicate rules from one stack to another, and rule updates had to be repeated on each stack.

But now, beginning in V1R13, the Configuration Assistant supports a new type of reusable object: reusable connectivity rules. You can share reusable rules across multiple stacks, and they remain synchronized when you make changes.

Create a reusable rule
There are two ways to create a reusable rule: either create a new one, or else convert an existing stack-specific connectivity rule to a reusable one.

To create a new reusable connectivity rule, do the following:

1. In the navigation tree of the IPSec perspective, click Rules under Reusable Objects, as highlighted in Figure 1.
2. Click the Add button to create a new reusable rule.
3. Follow the normal procedure for configuring a connectivity rule.

To convert an existing stack-specific connectivity rule into a reusable rule, do the following:

1. In the IPSec perspective, navigate to the TCP/IP stack.
2. Select the connectivity rule that you want to convert into a reusable rule.
3. From the action menu, choose Make Reusable. You’ll be prompted to choose a unique name for the reusable rule.

Add a reusable rule
When you’ve created a reusable rule, add it to a TCP/IP stack using the following steps:

1. In the IPSec perspective, navigate to the TCP/IP stack.
2. Click Add to add a new connectivity rule.
3. On the Welcome screen for your new connectivity rule, select the Reusable rule radio button.
4. Choose the reusable rule from the list of rules.

Figure 1. Reusable connectivity rules
Identify reusable rules

To identify reusable rules in the TCP/IP stack view, look for the rules with the (R) icon beside their name, as highlighted in Figure 1:

(R) RR_4

You can see all your reusable rules by selecting Reusable Objects in the navigation tree.

Make changes to a reusable rule

Because any change you make to a reusable rule can impact multiple stacks, you can only edit reusable rule attributes from the Reusable Objects view.

Change reusable rules back to stack-specific rules.

Sometimes, you might need to make temporary changes to a reusable rule for just one stack. To do this, you can convert a reusable rule to a stack-specific connectivity rule using the following steps:

1. In the IPsec perspective, navigate to the TCP/IP stack.
2. Select the reusable rule that you want to convert into a stack-specific rule.
3. From the action menu, choose Make Stack Specific.
4. You can now edit the rule attributes, such as filter logging settings, without affecting other stacks. When you are done, you can remove the stack-specific rule and re-add the original reusable rule.

Variable substitution

When you create a reusable rule for a distributed DVIPA, the rule uses the same local IP address on all TCP/IP stacks. But what if you want to configure a reusable rule that applies to distinct home IP addresses for each TCP/IP stack?

Beginning in V1R13, the Configuration Assistant allows you to configure variable names for local IP addresses and for local IKE identities. This gives you the ability to create a reusable rule that applies to different home IP addresses.

To configure and use named local IP addresses, perform the following steps:

1. In the IPsec perspective, navigate to a TCP/IP stack.
2. Choose the Local Addresses tab, which shows the named addresses configured for a stack.
3. Click Add to add a new named local IP address.
4. Create a new connectivity rule within Reusable Objects or within the TCP/IP stack view.
5. On the Data Endpoints screen for the connectivity rule, select the Local IP address name radio button.
6. Choose the IP address name from the list of names.

When you create a named IP address on one TCP/IP stack, the name is available to all TCP/IP stacks but is not yet filled in with a value. If you create a reusable connectivity rule that references a named IP address, then all stacks using that reusable rule must have an IP address assigned to that name.

If you need to create a reusable rule that applies to multiple named IP addresses, include named IP addresses in an address group using the following steps:

1. In the IPsec perspective’s navigation tree, click Address Groups under Reusable Objects.
2. Click Add to create a new address group.
3. Click Select Local Address Name.

Choose the local IP address name that you want to add to the address group. Repeat as needed.

This address group can then function as the local data endpoint for your reusable connectivity rule.

Named local IKE identities work the same as named local IP addresses—you can configure a named local IKE identity on the IKE symbols tab for a TCP/IP stack, and select a named identity on the Local Security Endpoint screen when you are creating or editing a reusable connectivity rule.
**Dynamic discovery**

In support for variable substitution of IP addresses, V1R13 Configuration Assistant also includes dynamic discovery of IP addresses for a TCP/IP stack. This allows you to configure IP addresses in reusable rules without resorting to manually entering each address. But note that this function is only supported in the z/OSMF version of the Configuration Assistant.

To use dynamic discovery of local IP addresses, you must enable the ServicesConnection between Policy Agent and the Configuration Assistant. For information on enabling this connection, see the “Policy-based networking” chapter in z/OS Communications Server IP Configuration Guide, SC31-8775-17. If you have enabled the ServicesConnection, you can discover the local IP addresses of your stacks using the following steps:

1. In the IPsec perspective, navigate to the TCP/IP stack.
2. Select the Local Addresses tab.
3. Click Discover.
4. Enter the connection information for your Policy Agent ServicesConnection if it is not already filled in.
5. Click Go.

If the connection is successful and you are authorized to import TCP/IP configuration information, this process populates the Local Addresses tab with names and IP addresses for your TCP/IP stack, as shown in Figure 2. You can then reference these names in address groups and in reusable connectivity rules.

**Multiple releases**

With this improved support for managing multiple stacks, it is more important than ever that the Configuration Assistant support multiple releases of z/OS. Beginning in V1R13, the Configuration Assistant now supports the configuration of multiple z/OS releases in a single instance of the Configuration Assistant.

At present, the Configuration Assistant supports configuration for V1R12 and V1R13 systems. When you create a new z/OS image, the Configuration Assistant prompts you to choose the release level for the image. You can change the release level by selecting the image in the navigation tree. The Configuration Assistant ignores policy options that do not apply to a particular release level and issues warnings to explain the problem.

**Acknowledgments**

We think you'll find that the new function available in the V1R13 Configuration Assistant simplifies the task of managing IPsec configuration for many TCP/IP stacks. We'd like to thank Diane Shannon for her technical and editorial reviews of our article.

---

**Figure 2. Imported local IP addresses**
Since its inception way back in version 2 release 2 of the Resource Access Control Facility (RACF), the RACF Remote Sharing Facility (RRSF) has used Advanced Program-to-Program Communication (APPC) as the transport mechanism to synchronize the RACF database between remote z/OS systems. Over time, however, installations have shifted their networks to TCP/IP. For this reason, RRSF has added support for TCP/IP in z/OS V1R13.

When implementing TCP/IP for RRSF node connections, security of the RRSF network traffic is improved by the required use of the Application Transparent Transport Layer Security (AT-TLS) function of the z/OS Communication Server. Configuration of AT-TLS, and its associated certificates and SERVAUTH profiles, is beyond the scope of this short article. Although not covered here, AT-TLS configuration is required before RRSF communications can be converted to use TCP/IP.

Converting a node from APPC to TCP/IP

Existing installations use APPC as the RRSF network protocol. If you want to implement TCP/IP for RRSF node connections, you will need to perform a few simple conversion steps. Note that you can use the same steps to convert to TCP/IP or to convert back to APPC. The main tool used for conversion is the TARGET command.

To show you this procedure, we’ll step you through the conversion of a sample configuration:

1. First, we issue the TARGET LIST command to display our sample current configuration of a LOCAL node (NODE1) and a single APPC remote node (NODE2).

```plaintext
TARGET LIST
IRRM009I (?) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE STATE.
IRRM091I (?) - LOCAL NODE APPC LISTENER IS ACTIVE.
IRRM009I (?) REMOTE RRSF NODE NODE2 IS IN THE OPERATIVE ACTIVE STATE.
```

We’ll use the TARGET LIST command after each step in the procedure to display the current state of the configuration.

2. Next, we do some preparation for the conversion. Before we change any protocol information for the LOCAL node, we have to make sure it is dormant. And we need to tell the LOCAL node that it should understand TCP as well as APPC:

```plaintext
TARGET NODE(NODE1) DORMANT
TARGET NODE(NODE1) PROTOCOL(TCP) OPERATIVE
TARGET LIST
IRRM009I (?) LOCAL RRSF NODE NODE1 IS IN THE OPERATIVE ACTIVE STATE.
IRRM091I (?) - LOCAL NODE APPC LISTENER IS ACTIVE.
IRRM091I (?) - LOCAL NODE TCP LISTENER IS ACTIVE.
IRRM009I (?) REMOTE RRSF NODE NODE2 IS IN THE OPERATIVE ACTIVE STATE.
```

The TARGET LIST command output shows that the LOCAL node is now listening for communications using both the TCP and APPC protocols.
3. We now define a new version of the remote node that speaks TCP instead of APPC. The remote node can be identified by its host name or IP address.

```plaintext
TARGET NODE(NODE2) PROTOCOL(TCP) WORKSPACE(workspace info) PREFIX(prefix)
```

The TARGET LIST command output shows that a second instance of the remote node (NODE2) has been defined, but is not yet operative.

4. Next, we initiate TCP communications to NODE2 over TCP by issuing the TARGET OPERATIVE command:

```plaintext
TARGET NODE(NODE2) PROTOCOL(TCP) OPERATIVE
```

The APPC communications continue to function until the corresponding configuration is performed on NODE2 and TCP communications are established. Then, when the NODE2 TCP connection enters the OPERATIVE-ACTIVE state, APPC communications shut down. Any requests queued in the workspace files are processed by TCP in the expected order until the files are empty. When the files are empty, they will be automatically deallocated and deleted. Normal operations continue using the new workspace files (with their new naming convention) for the TCP protocol, into which pending requests have been queued.

The following message indicates that the protocol conversion is complete:

```plaintext
IRRC058I (?) RSF PROTOCOL CONVERSION FROM APPC TO TCP FOR NODE NODE2 IS COMPLETE.
```

At this point, TARGET list output shows only the TCP instance of the remote node and that the APPC instance has been deleted.

```plaintext
TARGET LIST
```

In this example, we no longer have any need for the APPC listener, so we can eliminate that port of entry as follows:

```plaintext
TARGET DORMANT NODE(NODE1) 
TARGET DELETE PROTOCOL(APPC) NODE(NODE1) 
TARGET OPERATIVE NODE(NODE1) 
```

We extended this example to show some of the nuances of the conversion process. But think of it this way: You simply enter the TARGET command as though you are defining the remote node from scratch using the TCP protocol, and RRSF takes care of the rest. Your final responsibility is to save the necessary commands in your RACF parameter library.

**Want more information?**

For more information on RRSF and on implementing TCP/IP as its transport mechanism, refer to the following z/OS V1R13 publications:

Calling COBOL: Java on the line

Hello, it’s z/OS Batch Runtime

BY STEPHEN HENKELS, RUTH RAY, AND JODI EVERDON

The z/OS customers who want to modernize their traditional JES Batch environment now have some help. The z/OS Batch Runtime is a new z/OS component that provides Java COBOL interoperability with DB2 Version 9.1 for z/OS and DB2 10 for z/OS. This managed batch environment for z/OS provides the facility and framework to enable shared access to a DB2 for z/OS subsystem by both COBOL and Java programs and allows updates to that database to be committed or rolled back across the Java and COBOL language boundaries.

Batch Runtime enables selective enhancement, replacement, or even direct use of existing components of a traditional COBOL embedded Structured Query Language (SQL) DB2 application inventory with new Java code using the same local DB2 attachment. This, in turn, enables updates to the local DB2 database resources from both the legacy and modern application components to be committed under a single transaction. This support allows you to take an incremental approach as you attempt to modernize your application suites while preserving your investment in the existing set of functions. The bonus here is that you can add new capabilities in whatever development environment you choose.

Evolution not revolution

Instead of an “all-or-nothing—multi-year-and-total-replacement-modernization-project,” the idea is to provide an evolutionary approach to batch modernization. Additionally, this preserves the performance, scalability, and reliability of any traditional COBOL code, while allowing any new Java components to inherit the dispatch capability of the z/OS offload engines—System z Integrated Information Processor (zIIP) and IBM System z Application Assist Processor (zAAP).

Here are some of the support highlights, programming requirements, and restrictions in this pure JES-driven batch environment:

• Supports the following cyclic scenarios: COBOL calls Java, which then can call COBOL, as well as Java calls COBOL, which then can call Java. This recursion has been verified to considerable depth.
• Commits updates of DB2 for z/OS resources from both COBOL and Java components in a single transactional scope using a local RRS attachment.
• Supports COBOL and Java running under the same TCB and connecting to a single local DB2 for z/OS subsystem.
• Gives access to all the features of object-only COBOL and its intrinsic Java interoperability to the programmer in addition to traditional Java Native Interface (JNI) services.
• Requires no changes or re-compile when given the correct program structure, which means many existing COBOL procedural routines can remain “as is”.

In addition, this solution does not depend on external run-time environments like WebSphere® Application Server, DB2 Stored Procedures, CICS®, or IMS®. It’s not necessary to establish a pool of any pre-started Java Virtual Machine (JVM). A JVM is started and stopped with each job step. This cost is amortized over the life of the job step, implying suitability for longer job steps. The one caveat to keep in mind is that Batch Runtime does not support multithreaded applications.

The environment

The initial z/OS V1R13 implementation focuses on a scoped environment with the following assumptions regarding job control:

• There can be multiple steps in a batch JCL; however, each job step starts fresh with a clean environment, having its own instantiation of the JVM, and is transaction independent from any previous program step.
• JZOS, a z/OS JDK-provided Java batch launcher, launches the batch container for each job step using a tailored catalogued procedure.

Figure 1 describes a high-level architecture of the z/OS Batch Runtime environment.

The batch container is essentially a job step launcher that performs the initialization that sets up the environment for COBOL, Java, and DB2 interoperability. This includes:

• Setting up the proper Language Environment® for the COBOL programs to run
• Setting up the job step under the umbrella of a Resource Recovery Services (RRS)-managed global transaction
• Initiating the DB2 JDBC driver in this special “BatchContainer” mode
• Invoking the DB2 JDBC driver to create a DB2 connection and attachment thread
• Invoking the primary COBOL or Java application after the environment is properly initialized.

After the user application is launched, it runs almost transparently under the auspices of the z/OS Batch Runtime with the following characteristics:

• User application code does not have to change because explicit Java JDBC driver initialization to obtain a local type 2 connection is simply returned to the already initialized environment.
• COBOL embedded SQL does not have to change with the exception of special COMMIT and ROLLBACK callbacks provided by the z/OS Batch Runtime.
• To keep the z/OS for DB2 database engine and client JDBC state synchronized, RRS-based global transaction semantics block, at runtime, the use of explicit embedded SQL or JDBC commit/rollback invocations.
• A new JVM option allows error percolation of a subset of hardware generated exceptions (for example, divide by 0). These are transformed to "catchable" Java exceptions when they are detected in COBOL code, which does not itself recover them. Before this implementation, these exceptions would result in a user abend condition.

Using z/OS Batch Runtime

The batch container is invoked through JCL that invokes the z/OS Batch Runtime supplied procedure BCDPROC. This procedure invokes the JZOS launcher to initialize the batch container Java environment.

A sample batch job (BCDBATCH) is supplied to invoke the batch container using the BCDPROC procedure JCL. You must tailor BCDBATCH to configure the application environment.

The batch container itself is a Java application, so it is necessary to configure the CLASSPATH and LIBPATH with the list of JARs and DLLs necessary to run both the batch container and the user application. The JZOS launcher provides a facility to run an inline shell script that you can use to configure these (and any other) environment variables. To simplify this process, the inline environment script provided by the JZOS launcher primarily consists of exported variables. These variables reference the home paths of the batch container, the JDBC driver, and any application paths.

Samples are provided with the z/OS Batch Runtime that show the tight integration with the JZOS java batch launcher. A section of a sample BCDBATCH job, //BCDIN file, is shown here with key user input items as read from the z/OS Batch Runtime:

//BCDIN DD *
#
# Update: Uncomment the option corresponding to the language of
# the application being launched
# bcd.applicationLanguage=COBOL
# bcd.applicationLanguage=Java
#
# Update: The program name or fully qualified Java class name
# of the application to be launched
# bcd.applicationName=your.application.name
#
# Update: Arguments to be passed to the launched application.
# For Java applications, any number of arguments can be used.
# For COBOL applications, a single argument with a maximum
# length of 100 characters can be used.
# bcd.applicationArgs.1=java arg1
# bcd.applicationArgs.2=java arg2

Only the beginning

The z/OS Batch Runtime extends the current interoperability of Java and COBOL to include DB2 for z/OS attachment sharing. This produces new opportunities for legacy transformation when DB2 for z/OS is the data repository of choice. It allows heritage investment in procedural COBOL with DB2 for z/OS application suites to be preserved even as new Java based DB2 for z/OS functionality is developed.

For complete details, see z/OS Batch Runtime Planning and User’s Guide, SA23-7270, at the following web site:

ibm.com/systems/z/os/zos/bkserv
Launch into JZOS!

*Migrating from JRIO to JZOS*

BY GINA YUAN AND MICHAEL WANG

Included in the z/OS Java software development kit (SDK) Technology Edition Version 6, Release 0, Modification 1 (z/OS Java SDK 6.0.1) products, IBM Java Record I/O (JRIO) provides record-oriented access to mainframe data sets on z/OS. But especially with z/OS Java SDK 6.0.1, JRIO is no longer a best way to access mainframe data sets. We’re strongly recommending that customers and ISVs both migrate existing Java applications to and develop new applications with IBM JZOS Batch Toolkit (JZOS).

JZOS is now the z/OS Java direction for accessing mainframe data sets

JZOS has been an integral part of z/OS Java SDK products for many years, but over the last few years, we’ve enhanced it a lot. Not only does JZOS include equivalent function to JRIO, it also offers additional benefits not available in JRIO. Some examples of great JZOS function that you can’t get in JRIO include:

- Improved I/O error and exception reporting.
- Support for all record formats supported by the C I/O Library.
- Dynamic allocation of both non-VSAM and VSAM data sets. (JRIO is limited to non-VSAM data sets.)
- The ability to specify all parameters that are accepted by the C I/O Library.
- Access method services (IDCAMS) commands to create or delete VSAM data sets, list catalog entries with LISTCAT, and more.
- Functions like LOCATE/OBTAIN to find data set information, search a catalog, read PDS directories, and for log stream data support.

Because none of these great JZOS functions will be included in JRIO, migration to JZOS is definitely the way to go.

Using JZOS or migrating from JRIO to JZOS is straightforward. We provide lots of information and sample code on the JRIO and JZOS web pages. Finding the JRIO code you need to update in your applications

In the in z/OS Java SDK 6.0.1 Javadoc, the Java JRIO classes are marked with `@Deprecated` tags. When you compile your application using z/OS Java SDK 6.0.1, you’ll get compile warnings that identify places in the Java source code referencing JRIO deprecated classes. Using this method, you can easily update your applications. Another way to find out whether an application is using JRIO is to search for the following JRIO import statement that identifies source code that might contain references to JRIO classes:

```java
import com.ibm.recordio
```

How to use JZOS

Go to the following website to download the IBM Java Record I/O (JRIO) to IBM JZOS Batch Toolkit migration and sample code:

ibm.com/systems/z/os/zos/tools/java/products/jzos/sdk601_jrio2jzos_mig.html

This sample code package contains information on how to create, read, write, copy, and delete mainframe data sets using JZOS. For example, the following JZOS code segment reads a data set—you can easily convert an equivalent JRIO application to a JZOS application. See Figure 1.

Using JZOS or migrating from JRIO to JZOS is straightforward.
import com.ibm.jzos.ZFile; // ZFile is the main class in JZOS
// for accessing datasets

class ZFileExample {
    public static void main(String[] args) {
        System.out.println("Reading JZOS sample code package");

        // Create a new ZFile object
        ZFile zfile = new ZFile("example.dataset", "rb, type=record, noseek");

        // Open the dataset using C I/O library options
        System.out.println("Open dataset:");
        try {
            zfile.open("rb, type=record, noseek");
        } catch (Exception e) {
            System.out.println("Unable to open dataset");
        }

        // Read data one record at a time
        System.out.println("Reading data:");
        while (zfile.read() != -1) {
            System.out.println("Record data:");
        }

        // Close the dataset
        System.out.println("Closing dataset");
        try {
            zfile.close();
        } catch (Exception e) {
            System.out.println("Unable to close dataset");
        }
    }
}

For more information on JZOS

Go to the JZOS website to learn more:
- JZOS Java Launcher and Toolkit Overview
  ibm.com/systems/z/os/zos/tools/java/products/jzos/overview.html

- For API specifics, see IBM JZOS Batch Toolkit API:
  ibm.com/developerworks/java/zos/javadoc/jzos/jzos240/index.html

- To download additional sample code from the Sample Java Programs for IBM JZOS Batch Toolkit for z/OS Java SDK, Version 6.0.1 web page:
  ibm.com/systems/z/os/zos/tools/java/products/jzos/sdk601_jzos_samples.html

Figure 1: JZOS sample code package

Go to the zFavorites website where you will find the latest information for all of your System z needs including product documentation, software, ISV development marketing info, education, links to FREE downloads, and much much more!
Now you can have the Object Access Method (OAM) store your primary object data (as files) in a zSeries File System (zFS) or a Network Attached File System (NFS).

**Overview: prepare for takeoff**
Because OAM is a hidden gem, we thought we'd start by introducing ourselves. OAM is part of DFSMSdfp™, a base element of z/OS, and is typically used for storing archive-type data. Technically speaking, OAM is an access method, but it offers so much more. OAM not only stores your object data, but it also manages the life cycle of your object data from creation to expiration within a storage hierarchy that consists of multiple levels (disk, optical, and tape).

So what is an object and how does the data differ from that of your traditional data set being managed by DFSMSHsm®? An object (also referred to as unstructured data) is a named stream of bytes and, unlike traditional data set data (also referred to as structured data), objects have no record orientation. To OAM, an object is simply a data stream, and we don’t really care what it contains—it could be a medical image, billing statement, financial record, your company’s email - whatever content you’d like and each object can be up to 2000 MB.

With the OAM (OSREQ) application programming interface (API), you can use OAM to store, retrieve, query, delete, and change the way an object is managed. You can write your own application program or use one of several available solutions (for example, IBM DB2 Content Manager and IBM DB2 Content Manager OnDemand). We’re very flexible and use the policy-based storage management capability in SMS and the SMS ACS routines. Using the SMS ACS routines, you establish policies for your object data, and OAM manages the objects throughout their life cycle, which includes the movement of objects within the OAM storage hierarchy.

---

**Expanded storage hierarchy**
With z/OS V1R13, the OAM storage hierarchy is expanded to include support for a new file system sublevel in the disk level of the OAM storage hierarchy. With this support, OAM can store object data (as files) in a z/OS UNIX file system directory. The directories might be within a zSeries™ File System (zFS) or Network File System (NFS). When the zFS or exported NFS file system is mounted in the z/OS UNIX System Services (z/OS UNIX) file-system hierarchy, OAM can use the file system as part of the OAM storage hierarchy; it can manage the life cycle of the objects that are in that file system. This support provides you with new and exciting opportunities for storing your object data. Figure 1 provides an overview of the storage hierarchy.

---

**OAM is an access method, but it offers so much more.**

In releases earlier than z/OS V1R13, the disk level of the OAM storage hierarchy is supported through DB2 tables (4 KB, 32 KB, and large objects or LOBs), and when storing objects to the disk level, OAM stores the actual object data in DB2 tables. Because unstructured data continues to be one of the fastest growing data segments, we knew you were looking for new disk storage options with OAM, and with V1R13 we’ve finally delivered.
For zFS, you create the file system itself on disk storage that is attached to z/OS (for example, IBM System Storage DS8700). To use a zFS for OAM, first create the file system as described in z/OS Distributed File Service zSeries File System Administration, SC24-5989. A zFS file system is created by defining a VSAM linear data set and then formatting it as a zFS by using the IOEAGFMT program.

Then for NFS, the file system exists on disk storage within an NFS file server. The NFS file server is typically a separate hardware device (for example, IBM N Series) and is connected through a network attachment to z/OS. When the file system on the NFS file server is mounted in the z/OS UNIX file system hierarchy, requests to access files within the mounted file system are processed through the NFS client on z/OS to the NFS file server on the hardware device and use TCP/IP. The z/OS NFS client protocol support is documented in the z/OS Network File System Guide and Reference, SC26-7417-12.

OAM then uses basic file I/O functions to create, open, read, write, close, and delete files in the file system based on policies you've established through the SMS ACS routines.

**File system planning**

We'll start by describing the planning steps. First, associate a directory location within the z/OS UNIX file system hierarchy with each object storage group in which the OAM file system support will be implemented. (A storage group is a way for OAM to segregate and manage like data.) A unique directory location is identified for each object storage group, and this directory location contains a mounted file system associated with the storage group. This process is similar to the unique DB2 object storage tables for the DB2 sublevel of the OAM disk storage hierarchy.

To allow the OAM address space access to directories and files in the z/OS UNIX file system hierarchy, your external security manager (such as RACF) must associate a z/OS UNIX UID and GID with the OAM address space. For example, with RACF, a definition for a user name must have an OMVS segment with an assigned UID, a group name must have an OMVS segment with an assigned GID, and the STARTED class must be used to assign these identities to the OAM startup procedure. With these settings in place, when OAM is started, the OAM procedure name is used to find a matching resource profile to be associated with the OAM address space.

Second, you must mount the physical file system in the z/OS UNIX file system hierarchy and must specify the location within the z/OS UNIX file system hierarchy (for each object storage group) by using additional OAM configuration steps. By using the user ID and group ID that is associated with the OAM address space, you indicate that the mounted file system will be modified to allow access only by OAM.

**OAM configuration steps**

Now we'll provide a brief overview of the configuration steps needed in OAM.

**SMS storage class construct**

The file system support expands on the OAM use of the SMS storage class construct to control where in the OAM storage hierarchy an object should be placed. Use the storage class construct and the initial access response seconds (IARS), the sustained data rate (SDR), and the OAM sublevel (OSL) parameters to determine where in the OAM storage hierarchy an object should reside. For the file system support, an OAM sublevel value of “2” associates an object with the new file system sublevel of the disk level of the OAM storage hierarchy. You specify the IARS value as 0, similar to the existing DB2 disk sublevel. The SDR value does not apply to OAM objects written to the disk level of the OAM storage hierarchy.

![Figure 1. Overview of OAM storage hierarchy](image-url)
CBROAMxx parmlib member

You might be asking: Then how does OAM know in which file system an object should be stored? At the object storage group level, you can use a new SETDISK statement in the CBROAMxx member of parmlib to specify the directory location (path) in the z/OS UNIX file system hierarchy where the objects might be stored and identify the type of file system to OAM. See Figure 2.

Object storage database

OAM maintains metadata about each object in a DB2 table that provides an object directory (that is, where in the OAM storage hierarchy an object resides and other tracking information). The existing OAM object directory location field (ODLOCFL) has two new values:

- E for disk sublevel 2 (file system)
- 2 for recalled to disk sublevel 2 (file system).

The existing OAM object directory instance ID field (ODINSTID) can now contain a nonzero value for objects that:

- Reside in the file system sublevel (ODLOCFL value of E) or
- Have been recalled to the file system sublevel (ODLOCFL value of 2), which identifies a unique instance of an object in the z/OS UNIX file system hierarchy.

OAM configuration database

A new table, the file system delete table, is added to the OAM configuration database (residing in DB2) in support of the new file system sublevel. The file system delete table is used to identify objects that are to be physically deleted from the z/OS UNIX file system hierarchy. The objects that are being deleted can result from normal deletion activity (for example, life cycle expiration of the object) or can be for objects written to the file system that have not been committed (for example, application rollback).

OAMplex considerations

When OAM is to use a file system in an OAMplex, the file system must be defined through z/OS UNIX as a shared file system. In a shared environment, a file system mounted by one system in the OAMplex is accessible to all systems that participate in the shared file system. To learn more about how to establish a shared file system, see z/OS UNIX System Services Planning, GA22-7800.

Next steps

We've provided a brief summary of our new support; you can read more about OAM and the detailed configuration steps in the z/OS DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support, SC35-0426-11. □
Let your aliases grow

In V1R13, many more catalog aliases are supported

BY STEPHEN M. BRANCH AND TAN Q. NGUYEN

Have you ever had to split a catalog only because you needed to add more aliases and the alias limit had been reached? We understand how frustrating it can be if your catalog is large enough to contain your entries, but must be split just to add more aliases. When this is the only reason for splitting a catalog, the extra administrative headache of maintaining the RACF profiles for the new user catalog seems all the more annoying. If you think so too, we’ve got some good news for you. In z/OS V1R13, the 3500-alias limit has been lifted!

Why the 3500-alias limit anyway?
While it is great that the 3500-alias limit has been lifted, you might be wondering why this limit existed in the first place. Here’s a quick overview. Catalog alias records are defined in the master catalog. They point to a user catalog connector record also called a UCON. A UCON is used by catalog management to find a user catalog. The UCON has an alias pointer for every alias record that is associated with that UCON. Primarily, the alias pointers are used to remove all alias entries associated with a UCON when you delete a user catalog or when you export disconnect a UCON. A UCON resides in the master catalog, so its record size can grow up to the logical record length of the master catalog. The default maximum record size for the master catalog is 32768 bytes long. Given this maximum record size, and considering the information already in the UCON without aliases, the average number of alias names that can be added to a UCON is approximately 3500.

So, what’s new?
In z/OS V1R13, the 3500-alias record constraint is removed. You can now grow the number of aliases that are referencing a single user catalog to 3500 and beyond. How is this accomplished? A new record type called a user catalog connector extension record, or UCONX, is created by catalog management. The record key of a UCONX contains its related UCON name and an extension pad character. Storage management uses the UCONX to contain more alias names that are referencing the related user catalog. The UCON record and UCONX records are chained together using the pad character. The maximum number of UCONX entries for one UCON is 255. Potentially, you can define over 500,000 aliases per user catalog.

Sounds Great! Is there anything I need to do?
This new ability to create extension records for user catalog aliases is disabled by default. The reason it is disabled is so you can first migrate all of the systems in your sysplex to z/OS V1R13. Install z/OS V1R13 or later on all systems before using this function. Toleration support at lower releases is limited, and the new UCONX record type is not available for all Catalog Management functions.

Enable the support by specifying the EXTENDEDALIAS feature on the MODIFY CATALOG command:

```
P CATALOG,ENABLE (EXTENDEDALIAS)
```

You can also enable the support more permanently (between IPLs) by setting the parameter for “Alias Number Constraint Relief” in the new catalog PARMLIB member.

Toleration support
Toleration support is provided back to z/OS V1R10. On your pre-V1R13 systems, the toleration support provides the ability to access and delete aliases in UCONXs. You can also import z/OS V1R13 portable data sets as long as you have not exceeded the 3500-alias limit. However, with the toleration support, you cannot delete UCONXs. Also, on your pre-V1R13 systems, you still have the 3500-alias limit, which means you cannot define more aliases even if the user catalog has UCONXs.

Add as many aliases as you like
There you have it. The 3500-alias limit has been lifted to a new theoretical “limit” of over 500,000 aliases (depending on the alias name length). So let the number of aliases grow without worry. For more information, refer to z/OS DFSMS Managing Catalogs, SC26-7409-10.
Microwave the popcorn, grab the theater-size box of candy, the super-size soft drink, and head to the raised floor. Don't forget the jacket and steel toe shoes.

The DVD is about to begin! What movie is playing? It's an indie flick named...

“The COD”

For anyone not familiar with the Customized Offerings Driver (COD), it is a stand-alone z/OS driving system available at no additional charge to new or currently licensed z/OS customers. You can use the COD if you do not have an existing system to use as a driving system, or your existing system does not meet driving-system requirements and you do not want to upgrade it to meet those requirements. Other uses of the COD are not supported by IBM.

The level of z/OS running on the COD system depends on the minimum driving system requirements of the current release of z/OS. As of today, the COD is at z/OS V1R10. The COD includes the full z/OS UNIX environment, Java runtime, Hardware Configuration Definition (HCD), the System Modification Program/Extended (SMP/E), and the standard z/OS utilities a system programmer needs to complete a z/OS software installation.

The COD is included in DFSMSdss™ full volume dump/restore format and supports the 3390 DASD device geometry. It ships with an I/O definition file (IODF) that has device-address ranges defined so that you can IPL it after the stand-alone DFSMSdss restore is complete.

Up to this point, the COD has been available only on tape media. Late last year, 34xx media support was withdrawn, leaving you with a choice of 3590 or 3592 tapes. This change, combined with the growth of tapeless installations, automated tape libraries, tape importation restrictions, and virtual tape solutions, left many of you without stand-alone 3590 or 3592 tape drives capable of supporting the COD restore processing. IBM had to come up with a new plan for delivering the COD.

A DVD sensation is born

How easy is it to get a DVD COD? The web ordering tool Shopz, as shown in Figure 1 and Figure 2, currently supports COD.

The COD arrives on three separate DVD’s:

- D9ESY1 (system residence volume 1),
- D9ESY2 (system residence volume 2),
- D9ECAT (system catalog volume).

Each DVD contains one folder with all the files necessary to restore the volume. The DVD contains two DFSMSdss core images including the DFSMSdss physical dump image of one of the 3390 DASD devices that comprise the COD system. Use one core image if you want to use the Hardware Management Console (HMC) operating system message area as your interface for the DFSMSdss stand-alone environment. Use the other core image if you want to use a channel-attached console for your master console.

Figure 1. Shopz step 1
The **COD Installation Guide**, also included in the package, contains all the information required to use the DFSMSdss stand-alone environment to restore the COD volumes from DVD. It also contains the information required to IPL the actual COD system including the address ranges for consoles, TSO/E terminals, and DASD devices that are pre-defined in the Hardware Configuration Definition (HCD) IODF shipped with the COD.

**After the popcorn is ready**

Put the DVD in the drive and then:

Use the Recovery view in the tasks area of the HMC to load the DFSMSdss stand-alone environment from DVD. Be sure to select the CPC image on which to perform the install of the COD.

On the HMC, from the server tasks, select “Recovery” from the “Task List Work Area” for the classic interface. When using the tree view, select the tasks pad drop-down menu or context menu.

Select “Load from Removable Media, or Server” and specify the directory name in the optional file location. Do not select “Load from Removable Media, or Server and Assign for Operating System use”.

After the HMC loads the contents of the DVD, you must select which core image you want to use. Recall there are two images in the directory. Select the image based on the type of master console you intend to use:

- If you have a channel-attached master console, select the first radio button, and use the dfsmsdss.ins.
- If you do not have a channel-attached master console and want use the HMC Operating System Message area, select the second radio button for dfsmsdss _ opercnsl.ins.

Click **OK**

You are now running in the DFSMSdss stand-alone environment. You might have used this environment before during disaster recovery. During disaster recovery, this environment performs restore processing of the system packs without having to rely on a z/OS environment. In the case of the COD however, you will be using it to restore your new driving system.

**Bonus footage**

In the past, there was a requirement to initialize DASD volumes before restoring data to them. That requirement is gone! Yes, it's gone! Whether you are restoring a COD from TAPE or DVD, or performing a disaster recovery, it is no longer necessary to start the IBM Device Support Facilities (ICKDSF) stand-alone program to perform DASD initialization. Now, you simply specify the NOVERIFY keyword on the DFSMSdss RESTORE command and the DFSMSdss stand-alone service uses the uninitialized volume. Find out all the details in the **COD Installation Guide**.
When to call the usher

Unlike a traditional COD tape image, the DVD image is broken down into 120 MB pieces called file segments. When restoring the COD images, the DFSMSds Stand-alone environment guarantees the integrity of the restoration data by using an MD5 hash, or checksum of the data. As COD builds the image, it creates a checksum for each file segment. During the stand-alone restore from the DVD, the program calculates a checksum for each file segment. When COD gets to the last file segment, it compares the two values. If the values are equal, the system issues an ADRY4306I message indicating the file segment processing was successful. If the values are different, the system issues an ADRY4307I message to the console indicating which file segment is corrupt. If you encounter this error, contact the IBM Support Center.

For the initial IPL, you must use a local non-System Network Architecture (SNA) terminal to log on to the COD. You can satisfy this requirement using the Open Systems Adapter-Express Integrated Console Controller (OSA-ICC) function. After that, however, you can configure TCP/IP to reach the COD system through a Telnet 3270 session.
Mounting and unmounting a z/OS UNIX System Service file system was always a privileged operation managed by system administrators. When users want to have their own file system mounted or unmounted, they must submit a request to the system administrator, and then wait for their request to be performed. This affects the efficiency and the productivity of the applications and users in today’s ever-changing environment.

Wouldn’t it be nice to move the management of user data away from the system administrators and move the management into the hands of the users who own the data? We thought so, too. Now, in z/OS V1R13, with the new nonprivileged user mount and unmount file system capability, application developers and other users will be able to manage their own data without requesting system administrator services.

Getting to the magic
z/OS V1R13 contains two new BPXPRMxx parmlib keywords to control nonprivileged user mounts.

MAXUSERMOUNTSYS
MAXUSERMOUNTSYS is the maximum number of nonprivileged user mounts in the entire system.

MAXUSERMOUNTUSER
MAXUSERMOUNTUSER is the maximum number of nonprivileged user mounts for each nonprivileged user.

Set or change the values of MAXUSERMOUNTSYS and MAXUSERMOUNTUSER by simply issuing the following SETOMVS commands:

- `SETOMVS MAXUSERMOUNTSYS`  
- `SETOMVS MAXUSERMOUNTUSER`  

The default value for both keywords is zero. When the values change to non zero, that’s when the magic happens! You can configure BPXPRMxx parmlib member to set the values at system IPL time. In a shared file system environment, the most recent specifications of MAXUSERMOUNTSYS and MAXUSERMOUNTUSER prevail for all of the systems participating in a shared file system configuration.

Managing the RACF profile
The ability to use the nonprivileged user mount and user unmount are only for those users who you permit to the new RACF profile SUPERUSER. FILESYS.USERMOUNT, which is initiated in z/OS V1R13 in the UNIXPRIV class.

The following information summarizes the requirements for the non-privileged user mount to ensure that users can only mount and unmount their own file systems.

- User must have read-write-execute (rwx) access to the directory (that is, the mount point) that the file system will be mounted on.
- User must have read-write-execute (rwx) access to the file system’s root.
- If the directory that the file system is to be mounted on has the sticky bit on, the user must be the owner of that directory.
- If the file system’s root has the sticky bit on, the user must be the owner of the root.
- The file system to be unmounted must have been mounted by that nonprivileged user.

Performing the mount
The mount operations of nonprivileged user mount are implemented by TSO MOUNT, /usr/sbin/mount in z/OS UNIX System Services shell, Ishell function and BPX2MNT callable service. Other mount methods, such as BPXPRMxx parmlib mount and BPX1MNT callable service, don’t have the capability of the nonprivileged user mount.
Restrictions

Be aware of the following restrictions that are applied to the nonprivileged user mount:

- File system type must be zFS, HFS, or NFS.
- The mount must specify NOSETUID and cannot specify NOSECURITY.
- Remount function isn’t allowed for the nonprivileged users.
- Placeholder /// cannot be in the file system name.
- Mount point must be an empty directory.

Monitoring the mount

MVS™ commands can assist system administrators to monitor the nonprivileged user mounts and diagnose any problems.

DISPLAY command

Use the DISPLAY OMVS system command to display the set up of MAXUSERMOUNTSYS and MAXUSERMOUNTUSER, current usage, and high water marks of the nonprivileged user mounts and UID value of the nonprivileged user:

- DISPLAY OMVS,OPTIONS
- DISPLAY OMVS,LIMITS
- DISPLAY OMVS,FILE

Two new commands designed for nonprivileged user mount are available for users in V1R13:

DISPLAY OMVS,FILE,UID=

Using the new UID= filter, a user can choose to display file systems:

- Mounted by nonprivileged users
- Mounted by privileged users
- Mounted with a specific effective UID.

Figure 1 and Figure 2 show two variations of the DISPLAY OMVS,FILE,UID=<euid|USER|PRIV> filter options.

Find out more

Beginning with z/OS V1R13, the system administrators can stop mounting and unmounting end user data. Place the ownership and the responsibility of managing application data in the hands of the application user.

For deeper insight about nonprivileged user mounts, the command options, as well as the entire list of restrictions, see the following publications in the z/OS V1R13 library:

- z/OS UNIX System Services Planning, GA22-7800
- z/OS MVS Initialization and Tuning Reference, SA22-7592
- z/OS MVS System Commands, SA22-7627.
Tired of waiting for the top-of-the-hour interval migration process to move data from important primary volumes that have exceeded their threshold?

Fed up with the spikes in DFSMShsm processor usage at the beginning of each interval migration window?

Need a solution that proactively monitors volume thresholds and efficiently space manages your level-0 primary data?

If you answered YES to any one of these questions, DFSMShsm on-demand migration might be for you!

In the z/OS space management area, everyone knows how important it is to cost effectively manage storage so that there is always adequate space to meet the demands of daily operations for your business. Historically, installations have done this using the policy-based function of DFSMShsm interval migration. During interval migration, DFSMShsm performs a top-of-the-hour space check on every volume it manages to determine whether it should move data from the volume to meet the specified threshold occupancy. Because DFSMShsm checks every volume it manages during this process, interval migration typically causes a spike in DFSMShsm processor usage and can consume a lot of wall-clock time.

Beginning in V1R13, installations have a new option in DFSMShsm space management processing. DFSMShsm on-demand migration queues a volume for space management processing immediately after the volume exceeds its high threshold, rather than waiting for the top-of-the-hour space management window. This breakthrough concept in DFSMS space management is sure to bring smiles to the faces of storage administrators and users.

How does on-demand migration work?

DFSMShsm collaborated with the z/OS Core Component team and the Storage Management Subsystem (SMS) team to create a new Event Notification Facility code 72 (ENF72). When a volume exceeds its specified threshold because of a new data set allocation or extension, DFSMShsm issues ENF72 on each system within the SMSplex. In on-demand migration mode, the new DFSMShsm listening exit captures each ENF72 volume over-threshold event and relays the information to DFSMShsm. If the volume resides within a storage group that is defined with auto migrate option set to Yes (AM=Y), DFSMShsm immediately queues the volume for space management. The volume is queued for processing on each DFSMShsm system that has on-demand migration enabled, but is processed by the first host to select the volume. Space management
continues on the volume until the volume reaches its low threshold or there are no more data sets to migrate, expire, or delete.

That sounds interesting, but is it difficult to set up and manage the on-demand migration function?

The on-demand migration function is very simple to set up. Enable on-demand migration is to issue the SETSYS ONDEMANDMIGRATION(Y) command. This command prompts on-demand migration to process any volumes:

- In a storage group that is defined with AM=Y that exceeds the threshold
- That are already over threshold and have another allocation or extend occur.

Follow these guidelines before setting up on-demand migration:

- After you enable on-demand migration, each DFSMShsm host at V1R13 or later, will process primary SMS volumes on demand, so you must ensure that there are enough hosts available to perform other critical DFSMShsm activities too. This also holds true in a mixed-HSMplex environment (pre-V1R13 and V1R13 systems) where the V1R13 hosts might be busy performing migrations. Remember that your V1R12 hosts can only perform interval migration.
- Analyze the SMS storage group settings for your installation. After on-demand migration is in effect, all SMS volumes with the storage group setting of AM=Y are processed on demand.
- Keep in mind that after enabling on-demand migration, interval migration still runs on volumes specified with AM=I and eligible non-SMS volumes.

- If your installation uses tape mount management (TMM), consider changing AM=I to AM=Y to enable processing of TMM volumes on demand.
- On-demand migration does not run concurrently when interval migration (IM) or primary space management (PSM) runs on a host. When IM or PSM starts, on-demand migrations pause and begins again after PSM or IM are finished.

What if a lot of volumes go over threshold at the same time?

To notify you of exception cases when a large number of volumes go over threshold at the same time, DFSMShsm issues the following highlighted message to the console:

```
ARC1901E NUMBER OF VOLUMES ELIGIBLE FOR ON DEMAND MIGRATION HAS REACHED nnnnn.
```

DFSMSShm tracks the number of volumes that are concurrently over threshold, removing this message from the console after the number of volumes drops below the specified number (100 is the notification limit default). You can modify this value for your installation by using the SETSYS ODMNOTIFICATIONLIMIT (nnnnn) command.

How do I know what my SETSYS values are for on-demand migration?

To display the value of ONDEMANDMIGRATION and ODMNOTIFICATIONLIMIT parameters, issue the QUERY SETSYS command. Message ARC0153I describes what the values are for the on-demand migration settings for this host:

```
ARC0153I SCRATCHFREQ=days, SYSSOUT(CLASS=class, COPIES=number, SPECIAL FORMS={form | NONE}), SWAP= {YES | NO}, PERMISSION= {YES | NO}, EXITS={NONE | exits}, UNLOAD={YES | NO}, DATASETSEQUENCING={USER | DFSM}, USECMS={YES | NO}, ONDEMANDMIGRATION={YES | NO}, ODMNOTIFICATIONLIMIT=limit
```

Can you show me the benefits of using the new DFSMShsm on-demand migration?

Using DFSMShsm on-demand migration means that your primary SMS volumes are space managed immediately after going over threshold rather than having to wait for the top-of-the-hour interval migration processing and accompanying top-of-the-hour processing spikes. That means that when you use on-demand migration, your installation's applications and users are more likely to have plenty of available primary DASD space.

Figure 1 shows how interval migration works. It shows volumes filling up over time before they are processed by interval migration at the top of the hour.

With on-demand migration implemented, as shown in Figure 2, volumes are processed immediately after each volume's occupancy exceeds the volume threshold. DFSMShsm processes these volumes immediately, one by one, so they don’t build up and cause a processing spike.
What happens if DFSMSshm misses an ENF72 signal?

We recommend that you have two or more DFSMSshm hosts that are processing on-demand migration. This distributes the workload across multiple systems and minimizes the chance of DFSMSshm missing an ENF72 signal for a volume because DFSMSshm is stopped for any reason. If DFSMSshm does miss an ENF72 signal, don’t worry! In addition to issuing an ENF72 when a volume first goes over threshold, SMS issues ENF72 signals at specific intervals, up to the point where an ENF72 is issued for each allocation and new extent on the volume.

What if the volume is over threshold but there is not enough eligible data to move?

We’ve planned for this too. If your volume is over threshold but there are not enough eligible data sets to migrate to drop the volume below its high threshold, DFSMSshm does not select this volume, or subsequent processed volumes remaining over their high threshold, for on-demand migration again for 24 hours. This prevents DFSMSshm from trying repetitively to process a volume for which there are no eligible data sets to process. You can customize the length of the delay. See z/OS DFSMSshm Implementation and Customization Guide, SC35-0418.

The advantages of this new function are tremendous and your users will applaud you for using on-demand migration processing to provide adequate space for data set allocations or extensions. Give it a shot and let us know how you like it!
The path to recovery
Realizing improved channel path recovery

By Colin D. Chen, Ilene A. Goldman, and Thomas H. Rose

No matter how well you plan and implement your I/O configuration, at some point you’ll end up with path errors. Until now, path errors were handled on a device-by-device basis, but as the size of I/O configurations increase and become more complex, this approach no longer works well. We saw room for improvement in this area, and with the release of z/OS V1R13 we introduced improved channel path recovery to increase system resilience and ease diagnosis of the path-related errors that result from some I/O hardware failures. When you enable this improved channel path recovery, z/OS becomes more proactive, removing a failing path from all DASD attached to FICON or ESCON and tape devices in the Logical Control Unit (LCU). This can significantly reduce recovery time and improve application I/O performance because it helps avoid subsequent path errors.

Broadening the scope of recovery
When path-related errors occur, it usually affects multiple devices. Until now, recovery from path-related problems was done at the device level—recovery processing would remove the path from one device at a time. However, in a path error, each I/O to a different device on the same LCU and using the same path encounters the same error. Until now, recovery processing had to remove the path for each device. Subsequent I/Os down the failing path causes recovery processing to remove the path from each device as the error is encountered. This type of recovery processing takes time and resources away from the system and your applications. Improved channel path recovery performs this processing at the LCU level rather than at the device level. This means that after the system initiates a recovery for a path problem, the path is removed immediately from all devices in the LCU. Subsequent I/O operations to that device or others on the same LCU will not use the affected paths, and applications can proceed as usual. Figures 1 and 2 show the benefits of the new control-unit scope recovery.

Figure 1 shows that, before enabling improved channel path recovery, path failure and recovery repeats for each I/O down the failing path to a different device. Figure 2 shows how, with improved channel path recovery enabled, removing the failing path from all devices in the LCU greatly reduces the recovery impact to the system.

You can enable improved LCU-level channel path recovery by specifying the RECOVERY PATH_SCOPE=CU statement in either the IECIOSxx parmlib member or the SETIOS RECOVERY command. Then, all devices in an LCU are recovered when any device encounters an error.

If you want to keep the scope of channel path recovery to a device level, you can specify or default to PATH_SCOPE=DEVICE, and any

---

Figure 1. I/O recovery for failing path - before

Figure 2. I/O recovery for failing path - after
devices encountering path-related errors undergo recovery individually, just as they always have.

You can use the RECOVERY statement subparameters, PATH_THRESHOLD and PATH_INTERVAL, to specify that the system proactively remove paths for errors that occur intermittently. By using the PATH_SCOPE=CU option, along with the PATH_THRESHOLD and PATH_INTERVAL options, you can reduce the elapsed time it takes for the system to recover from channel-path-related errors, and help prevent system performance problems that can occur when a significant amount of time is spent in repetitive channel-path error recovery. Specifying a PATH_INTERVAL of 10 (minutes) and a PATH_THRESHOLD of 20 (errors per minute) means that at least 20 errors must occur every minute for 10 consecutive minutes before the path is removed from all the affected devices.

We recommend against setting both the interval and threshold to very low values (for example, setting both to values to 1), because this might cause z/OS to remove paths unnecessarily.

When the problem that caused the path-related errors has been corrected, the path that was taken offline can be restored by using one of the following commands:

- VARY CU
- VARY PATH
- CONFIG CHP

Before issuing the VARY CU command, we recommend that you first issue the VARY PATH command for a single device or path to verify that the problem is resolved before issuing the VARY CU command for all devices or paths.

New and Changed Messages

The existing DISPLAY IOS,RECOVERY command now displays the following new information in message IOS103I to display the path recovery settings:

- PATH RECOVERY SCOPE IS BY CU
- PATH RECOVERY INTERVAL IS nn MINUTES
- PATH RECOVERY THRESHOLD is nnn ERRORS

- or -
- PATH RECOVERY SCOPE IS BY DEVICE

The existing DISPLAY M=DEV(dddd,(chp)) command now displays the reasons why a path was taken offline (new text highlighted):

```
    iee174i hh.mm.ss DISPLAY M idr
    device nnnn status=status
    Chp nn
    entryi link address la
    dest link address la
    path online y|n
    Chp physically online y|n
    path operational y|n
    managed y|n
    cu number cccc
    destination cu logical address=da
    . . .
    not operational reason text
    pAv base and aliases pp
    [paths not validated]
    [path offline due to the following reason(s)]
    [path recovery error]
    [by operator]
    [control unit initiated reconfiguration]
    [configuration manager]
```

The existing DISPLY M=DEV(dddd,(chp)) command now displays the reasons why a path was taken offline (new text highlighted):

```
    ios210i path recovery initiated for path pp on cu cccc, reason=rnstxt
```

where rnstxt is one of the following reasons:

- LINK RECOVERY THRESHOLD REACHED
- PATH ERROR THRESHOLD REACHED
- DYNAMIC PATHING ERROR
- REQUESTED BY DEVICE ERP ROUTINE

Regardless of the setting of the PATH_SCOPE parameter, if at least one detecting component was identified for a path-related error, new message IOS054I is issued with either message IOS050I or IOS051I and identifies the components that detected the error:

```
    ios054i ddddd,pp errors detected by comp, comp,
```

where comp is one or more of the following components:

- CHANNEL
- CHAN SWITCH PORT
- CU SWITCH PORT
- CONTROL UNIT

For information about the syntax of the SETIOSxx parmlib member, see z/OS MVS Initialization and Tuning Reference, SA22-7592.

For information about the SETIOS RECOVERY command, see z/OS MVS System Commands, SA22-7627.

z/OS MVS System Messages, Volumes 7, SA22-7637, and z/OS MVS System Messages, Volumes 9, SA22-7639, contain the new and changed messages.
Currently, applications use the z/OS Communications Server system resolver to map human-readable host names such as ibm.com to numeric IP addresses that can be used for communication within a TCP/IP network. Typically, the resolver queries one or more domain name system (DNS) name servers to perform this mapping. Most enterprises define a list of multiple DNS name servers to be used by the resolver. For each host name resolution attempt, the resolver contacts the name servers serially, in the order defined by the list.

This system works well while the first, or preferred, name server in the list is active and responsive to resolver queries. However, if the preferred name server is down for an extended period of time, each resolution attempt incurs a delay while the resolver waits for a response from the preferred name server. The delay is multiplied if network conditions impact more than one name server in the list, as shown in Figure 1.

In this example, the two most preferred name servers in the list are not responding to resolver queries, so each host name resolution incurs a delay of two times the resolver timeout value. By default, the resolver time out is five seconds, and using that value for our example, we’re looking at a delay of 10 seconds for every resolution attempt. Ouch!

Here comes the solution!

In z/OS V1R12, Communications Server provided a way to detect name servers that are unresponsive for an extended period of time. Communications Server alerts the operator to the condition, but continues to send resolution queries to the unresponsive name servers.

In z/OS V1R13, Communications Server extends the name server monitoring function to optionally quiesce sending all resolution queries to an unresponsive name server. The resolver considers a name server unresponsive when it fails to respond to a significant percentage of resolver queries within a 30- or 60-second window. The user defines the significant percentage value for their environment as part of the system resolver configuration.

After the resolver has stopped sending queries to an unresponsive name server, the resolver periodically sends polling queries to that name server. The resolver continually monitors the name server’s failure rate in a sliding 60-second window. The name server remains quiesced until the name server’s failure rate is less than the significant percentage value configured for the resolver. Once the name server is responsive, the resolver will resume sending application resolution queries to that name server.

---

**Figure 1. Resolution attempts before quiescing function**
Figure 2 shows how resolution attempts are processed if the quiescing of unresponsive name servers function is enabled in the same sample configuration:

As in the earlier example, the two most preferred name servers are unresponsive, but now the resolver does not send resolution queries to these name servers. Instead, the resolver immediately sends the resolution query to the one name server that is responding in an acceptable manner, which is the third name server in the list. This eliminates the 10-second delay and improves overall response time for resolution processing.

As soon as the preferred name server becomes responsive to the resolver polling queries, the resolver sends resolution queries to the preferred name server. This eliminates the need to coordinate resolver configuration changes with scheduled outages of the name servers in the list.

When you take advantage of the z/OS Communications Server name server monitoring enhancements, your system temporarily removes unresponsive name servers from the list, re-adding them later as they become available. This solution allows host name resolution to continue to be accomplished without delays!

For further information, see z/OS V1R13 Communications Server Configuration Guide, SC31-8808.
Protect and detect

Intrusion Detection meets IPv6… and more!

BY CHRIS MEYER AND JOYCE ANNE PORTER

It’s no secret that the exhaustion of IPv4 addresses brings new urgency to the adoption of IPv6. Given that urgency, the delivery of z/OS Communications Server IPv6 support for Intrusion Detection Services (IDS) arrives just in time. However, the new IDS features aren’t limited to monitoring IPv6 traffic. Several new attack types that apply to both IPv4 and IPv6 have also been added.

In this article, we will review the existing z/OS Communications Server IDS capabilities, explore the new support that’s been added in z/OS V1R13, and point you in the right direction to begin deploying IDS on your z/OS systems.

Existing IDS categories

Before we get into the new features, let’s quickly review what the existing intrusion detection support brings to the table.

There are three categories of z/OS intrusion detection services for IPv4 traffic:

- Scan detection reports when a remote node attempts to gather information about the z/OS node. The TCP/IP stack counts the number of times the remote node attempts to access unique IP resources (ports or interfaces) during a configured period of time called a scan interval. If the number of access attempts exceeds the configured threshold for the scan interval, the stack reports the activity. You can configure a list of IP addresses to be excluded from scan detection in order to avoid false positives. For example, if your security team has installed a compliance monitor on your network, you might want to exclude it from scan detection.

- Attack protection looks for a variety of single-packet and multi-packet attacks designed to disrupt the TCP/IP stack or consume its resources. When an attack is detected, attack protection can report on the activity and can also take defensive actions, such as dropping the suspect packet. You can enable protection against eight different attack types in z/OS V1R12, ranging from detecting a single malformed IP packet to detecting a TCP SYN flood.

- Traffic regulation monitors the use of TCP and UDP resources. For TCP, traffic regulation monitors the number of TCP connections that each client establishes with a server and for UDP, it monitors the length of the UDP receive queues. You can configure traffic regulation to take defensive actions such as limiting new TCP connections from a client, based on a configured percentage of available (that is, unused) connections, and discarding new UDP messages.

z/OS V1R13 Communications Server Enhancements

In z/OS V1R13, Communication Server enhances intrusion detection services in a number of ways. Let’s take a look at each of these enhancements.

Extending IDS protections to IPv6

With V1R13, scan detection, traffic regulation, and the vast majority of attack protections extend to IPv6 traffic.

Scan detection and traffic regulation now perform essentially the same function for IPv6 as they do for IPv4.

Many of the existing attack types are extended “as-is” to IPv6 traffic while others apply conceptually to IPv6, but are specific to the IPv4 header format. These attack types are implemented differently for IPv6. Table 1 illustrates how existing IDS attack types are supported for IPv6 traffic.

Now let’s explore the new attack types introduced with V1R13.

<table>
<thead>
<tr>
<th>Existing IPv4 attack type</th>
<th>IPv6 implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Malformed packet</td>
<td>Existing IPv4 support extended to IPv6. No new configuration needed.</td>
</tr>
<tr>
<td>• ICMP redirect restrictions</td>
<td>IPv6 next header restrictions (specifies a list of restricted IPv6 next header values, which may include IP protocol values)</td>
</tr>
<tr>
<td>• UDP perpetual echo</td>
<td>• IPv6 destination option restrictions (specifies a list of restricted IPv6 destination options)</td>
</tr>
<tr>
<td>• Flood (both interface flood and TCP SYN flood)</td>
<td>• IPv6 hop-by-hop option restrictions (specifies a list of restricted IPv6 hop-by-hop options)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP protocol restrictions (specifies a list of restricted IP protocols)</th>
<th>IPv6 option restrictions (specifies a list of restricted IPv4 options)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound RAW (specifies a list of restricted IP protocols for IPv4 and imposes other restrictions)</td>
<td>IPv6 outbound RAW (specifies a list of restricted IP protocols for IPv6 and imposes other restrictions)</td>
</tr>
</tbody>
</table>

Table 1. IPv6 implementation of existing types
Data hiding attack
This attack type detects an attempt to hide data in specific fields (such as padding bytes) of IPv4 and IPv6 packets. The data hiding attack type can be configured to generate notification of the attack, drop the packet, or both.

TCP queue size attack
This attack type protects against excessive consumption of storage by the various data queues of a TCP connection. In previous releases, the TCP queues were monitored and automatic storage relief measures were invoked when built-in thresholds were reached. However, the threshold limits and reporting options were not configurable and no automated defensive action was available. In V1R13, the receive, send and out-of-order queues for each TCP connection are monitored according to IDS policy. Notifications are generated when the queue lengths reach configured thresholds. Additionally, you can configure a defensive action to reset the associated TCP connection.

Global TCP stall attack
Whereas the TCP queue size attack type focuses on individual TCP connections, the global TCP stall attack type looks for a pattern of behavior where many TCP connections stall and are unable to send data. When this condition is detected, IDS can generate notifications, defensively reset stalled connections, or both, according to what you configure in the IDS policy.

EE attacks
A set of Enterprise Extender (EE) attack types are introduced:
• EE malformed packet events
• EE LDLC check events
• EE port check events
• EE XID flood events

Each of these attack types monitors EE packets for specific nonstandard usage of the related SNA protocol messages. For each attack type, you can configure reporting options and, with the exception of the EE XID flood attack type, whether the packet should be discarded as a defensive action. For each attack type, you can also configure a list of remote IP addresses that are exempt from monitoring in order to accommodate EE implementations that depend on nonstandard behavior.

Figure 1. Working with IDS attacks in the Configuration
Configuring IDS

Use the IBM Configuration Assistant for z/OS Communications Server to configure z/OS intrusion detection services. Based on your selections, the Configuration Assistant (CA) generates IDS policies that the Policy Agent can read and install on the target z/OS system. See Figure 1 for an example of the CA user interface.

The CA provides a default requirement map (a set of rules) that you can use as a starting point for a TCP/IP stack IDS policy. This default IDS requirement map enables all attack types in a mode that provides reporting, but takes no defensive actions.

Default settings are provided for each attack type. For example, a list of option values that are typically not used in IPv6 networks is provided as the default for the IPv6 Destination Options restrictions attack type.

The default IDS requirement map does not enable scan or traffic regulation protection because these must be customized for your installation. You can copy the default requirement map and then enable scan and traffic regulation protection in the copy, based on your requirements.

You can also customize the attack protection settings in a copy of the default requirement map. The reporting options enabled in the default requirement map are:

- Log to syslogd.
- Generate statistics.

You can modify those options or enable logging to the system console or tracing of suspicious packets. You might also want to change one or more of the attack types to take a defensive action such as dropping suspicious packets or resetting connections.

Consider using the default requirement map provided by the Configuration Assistant as a starting point for crafting an IDS policy that meets your installation needs.

Summary

z/OS intrusion detection services can be a powerful element in your overall z/OS network security strategy. V1R13 extends IDS to cover your IPv6 traffic.

In z/OS V1R13, Communication Server enhances intrusion detection services in a number of ways.

defaults provided for all attack types, basic scans that you can enable, and a large number of traffic descriptors that you can use to define your traffic regulation requirements, CA can take you a long way toward deploying IDS to protect your z/OS system.

Analyzing IDS messages in your syslogd file

So once you've deployed your IDS policy, how do you know what it's doing? If you configured IDS to report events to syslogd, then you can use the z/OS UNIX trmdstat command to generate reports from a syslogd file.

Begin with the overall summary report (trmdstat -I) to determine what IDS activity is recorded in the syslogd file. Based on what you learn from the overall summary report, you can generate more specific reports. For example, if the overall report shows that attacks were detected, you can run an attack summary report (trmdstat -A) to see the types of attacks detected. Then, you might want to run an attack detail report (trmdstat -A -D) filtering on a specific remote IP address. Instead of searching through a syslogd file, let the z/OS UNIX trmdstat command pull the data you need from the file and format it in an easy-to-use report.

and also adds several new attack types. The Configuration Assistant is a great place to start exploring, experimenting with, and deploying IDS. The trmdstat command provides a rich set of reports to help you analyze your IDS results.

Learn more!

IBM publications provide more information about the following topics:

- Intrusion Detection Services: z/OS Communications Server IP Configuration Guide, SC31-8775
- Configuration Assistant: Help panels and tutorials. You can download the Windows version of the Configuration Assistant at the following web site:

  ibm.com/support/docview.wss?uid=swg24013160

Acknowledgements

The authors would like to thank Alan Packett of the z/OS TCP/IP development team in Raleigh, North Carolina, for his help and expertise in preparing this article.
When keeping up with the (z/OS) Joneses is a good thing

BY ED MEKEEL, MARIANNE HAMMER, AND MARK WISNIEWSKI

When you hear the phrase “keeping up”, one’s mind conjures up several possible endings to that idea. The first one that jumps out for me is “keeping up with technology”. There are all kinds of technical, digital gadgets and devices available today. Trying to keep up can be a stressful venture. Each day it seems we’re introduced to a new wireless phone, digital cameras, digital tablets and computers. How do we make sense of it all?

Probably the next ending for “keeping up” that occurs to me is “keeping up with the Joneses”. It was an idea that stemmed from an infamous comic strip that debuted in 1913, and that concept has never gone away. You still hear the phrase today. It has a somewhat negative connotation and typically describes folks who measure their affluence by comparing themselves to their neighbors. The preceding examples are two examples of “keeping up” and both generate questions about how much weight we should give them in governing our daily lives.

Keep on keeping up

Another example of “keeping up” is something we’d like all our customers and IBM Business Partners to consider—it’s something we believe has a positive outcome. That’s keeping up with the latest z/OS releases. We can call it “keeping up with the z/OS Joneses”. For z/OS, we constantly strive to improve performance. The prior release comparisons, as shown in Figure 1, because the average percent improvement based on Large Systems Performance Reference (LSPR) workloads, come from measuring four workloads on a 32-way single image. The z/OS component represents approximately 25% of the total processor usage for the average of the LSPR workloads. Since z/OS V1R8, this translates to approximately a 1.25% overall system improvement release to release, on average.

Proud yet modest

As always, IBM intends to continue focusing on improving performance. This includes z/OS and the measurements mentioned earlier in this article. While we can’t forecast that we’ll continue at the same rate and pace with the success of releases to this point, the goal continues to be equivalent to the 1.25% improvement over several releases for z/OS. We also want to improve performance across the entire z/OS product stack.

Benchmarks of success

Another focus area for release-to-release measurements is virtual storage constraint relief (VSCR). The goal is to consistently reduce 31-bit common storage for a release (compared to the previous release). Reductions in z/OS 31-bit common storage free up more 31-bit private virtual storage for your use.

The performance observations mentioned are based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual benefits that you experience can vary depending upon considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, performance results that you see can be different from the performance ratios in our controlled environment.

Hopefully it’s clear that comparing things like what you’re driving to your neighbor’s “ride” has questionable value. However, keeping up with the (z/OS) Joneses can have a very positive effect on your workloads.

For more information about LSPR for IBM System z, see:

Don’t judge a server by its name!
IBM Tivoli Directory Server and LDAP Version 3 protocol

BY JAY BRODFUEHRER, JONATHAN COTTRELL, AND SAHEEM GRANADOS

Don’t let the name “IBM Tivoli® Directory Server” mislead you! IBM Tivoli Directory Server x for z/OS is a base z/OS component, fully optimized to run on z/OS and support the LDAP Version 3 protocol. Moreover, the z/OS and non-z/OS versions of the Tivoli Directory Server are distinct products that have different features and capabilities.

Each IBM directory server implementation depends on platform specific technology. In the case of z/OS, the directory server can be set up to allow access to RACF data, cut SMF security audit records, and encrypt or hash sensitive data (such as user’s passwords) with AES, DES, or SHA-2 hashing support provided by Integrated Cryptographic Service Facility (ICSF). Further, on z/OS, the directory server leverages many z/OS specific technologies such as Workload Manager (WLM), Parallel Sysplex, Automatic Restart Manager (ARM), and DB2.

Clearly, the directory server is not just a straight port from a distributed platform, such as Windows, Linux, or AIX, to z/OS! It is designed to meet the reliability, availability, scalability, serviceability, and security expectations that come with the System z and the z/OS operating system, while still being designed to be fully compatible with the LDAPv3 protocol and providing new LDAP related function as the protocol evolves.

System z Reliability, Availability, Serviceability, Scalability, and Security
To highlight the Tivoli Directory Server for z/OS’s strength, we must first focus on the essential traits of anything z. Figure 1 provides an overview of the z/OS platform:

Figure 1. Strengths of the z/OS operating system
Reliability and Serviceability

Tivoli Directory Server for z/OS undergoes the same extensive test cycle as do all other base z/OS components. In addition, it follows the same service process as all other z/OS base components; that is, code and documentation defects are addressed through APARs and PTFs, where PTFs are SMP/E-installable packages of code fixes that can be applied to your z/OS system. To facilitate service and problem determination, Tivoli Directory Server for z/OS provides different levels of dynamic tracing and leverages z/OS Component Trace (CTRACE). CTRACE provides first failure data capture and allows the use of IPCS CTRACE format, merge, and locate routines to process the trace data that Tivoli Directory Server generates.

For serviceability, IBM Tivoli Directory Server for z/OS provides activity logging and cn=monitor searches. The activity log keeps a record of the LDAP operations that have been handled by the LDAP server in a z/OS UNIX System Services file or a partitioned or sequential data set. Also, cn=monitor search support provides useful serviceability information such as the number of LDAP operations handled by the server on a back end basis and the hit rates for the internal caches.

Availability

Tivoli Directory Server for z/OS was designed to be consistently available. One key element to this design is complete Parallel Sysplex Clustering Technology support. In this configuration, many Tivoli Directory Server for z/OS instances can be clustered to help achieve performance and availability requirements. VIPA and DVIPA are especially useful when there are multiple Tivoli Directory Server for z/OS instances configured in a Parallel Sysplex. In this environment, TCP/IP sends LDAP client requests to an instance that is not currently busy handling other requests.

Parallel Sysplex provides a great means for ensuring availability. However, it requires shared access to common data from all participating images. An important method for ensuring IT availability is data redundancy at the physical data level. To provide this level of assurance, Tivoli Directory Server for z/OS can be configured to enable replication. Replication maintains consistent duplicate copies across multiple LDAP servers.

In addition to server instance and data redundancy, another Tivoli Directory Server for z/OS configuration option is supported to address availability. Tivoli Directory Server for z/OS can be configured to work with Workload Manager for z/OS (WLM). WLM helps ensure installation availability and performance requirements are met across all System z elements. In particular, Tivoli Directory Server for z/OS is compatible with WLM health weighting services to help avoid storm drain problems that might occur when servers are overly-taxed.

One final key element to the Tivoli Directory Server for z/OS design is support for z/OS Automatic Restart Manager (ARM). Tivoli Directory Server for z/OS can be configured to use ARM to help ensure that, even in the event of a software failure that results in abnormal ending, Tivoli Directory Server for z/OS will restart and continue to be available.

Scalability

Once again Tivoli Directory Server for z/OS strives to live up System z expectations by being a scalable server. Tivoli Directory Server for z/OS can use DB2 for its data repository. DB2 is a mature relational database technology that has stood the test of time in terms of availability and scalability. Tivoli Directory Server for z/OS can take advantage of DB2 data sharing and partitioning support. In addition to leveraging existing scalable z/OS technology like DB2, as of z/OS V1R13, Tivoli Directory Server for z/OS now fully supports 64-bit addressing. Unlike earlier releases where DB2 back ends could not run in 64-bit mode, Tivoli Directory Server for z/OS V1R13 supports DB2 based (TDBM and GDBM) back ends running in 64-bit addressing mode. One overall benefit of 64 bit support is to support larger amounts of directory data in the TDBM back end. For example, 64 bit addressing allows customers to migrate large directory data with a single ldif2ds (bulkload) utility command.
Tivoli Directory Server for z/OS strives to live up to the reliability, availability, serviceability, scalability, and security standards one comes to expect from System z.

through LDAP and have the same authorization to RACF data as they do when logging into TSO. A LDBM or TDBM user entry can be mapped to an existing RACF user with the native authentication support provided by Tivoli Directory Server for z/OS. Native authentication support allows the existing RACF user’s password to be used for LDAP authentication.

Another key System z security service involves cryptography. For password encryption and hashing, ICSF is used for AES or DES encryption, and, as of z/OS V1R13, SHA-2 or Salted SHA-2 hashing support. If Tivoli Directory Server for z/OS is configured to listen for client requests on a secure interface, z/OS System SSL provides the SSL handshaking support. Finally, if GSS-API (Kerberos) authentication is configured in Tivoli Directory Server for z/OS, Integrated Security Services Network Authentication Service is used to provide this support between the client application and the server.

We have described Tivoli Directory Server for z/OS tight integration with System z and z/OS. Now we would like to highlight new LDAP protocol related function available in Tivoli Directory Server for z/OS V1R13.

**z/OS V1R13 enhancements**
In z/OS V1R13, Tivoli Directory Server for z/OS continues to deliver the LDAP enhancements that you expect from a feature packed LDAP server.

Clearly, administrative access must be closely guarded, especially if a system allows only one set of credentials to be assigned to an administrator and if better password security. SHA-2 hashes consist of SHA-224, SHA-256, SHA-384, and SHA-512. Salted versions of these algorithms provide the final four choices that are available.

Server side paged and sorted search results support has been added to Tivoli Directory Server for z/OS. Paged search results provide paging capabilities for LDAP client applications that want to just receive a subset of search results at a time. Sorted search results enable LDAP client applications to receive to only request a subset of the entries that are to be returned sorted search results based on a list of criteria, where each criterion represents a sort key or attribute type.

Finally, group search limits can be configured to allow users in those groups higher limits than the configured default server limits. With this support, these users can have the authority they need to perform necessary searches.

**Conclusion**
Throughout this article we have highlighted many different Tivoli Directory Server for z/OS optimizations for the System z and z/OS platform. These optimizations are very important to any workload deployed on System z. Moreover, Tivoli Directory Server for z/OS is a robust LDAP server, ensuring many LDAP directory data related functions are available. Ultimately, Tivoli Directory Server for z/OS strives to live up to the reliability, availability, serviceability, scalability, and security standards one comes to expect from System z, while still maintaining LDAPv3 protocol support as new features and requirements are introduced.

See IBM Tivoli Directory Server for z/OS V1R13 Administration and Use, SC23-5191-06, for more information.
software programs, like the people who create them, tend to get fatter and slower as they mature. With Cryptographic Support for z/OS V1R11-R13 (FMID HCR7790), however, the Integrated Cryptographic Services Facility (ICSF) has managed to shed pounds and build muscle. Specifically, the Cryptographic Key Data Set (CKDS) symmetric key repository has been redesigned to be easier to administer and to run workloads faster!

What do I need?
To take advantage of these new features, you must have a z900 or newer server, all LPARs running ICSF must be running z/OS V1R11, z/OS V1R12, or z/OS V1R13, and all ICSF instances must be at the FMID HCR7790 level.

A new communication protocol trims message overhead
When all ICSF instances in a sysplex are at the new level, a new communication protocol is used for CKDS updates. This new protocol trims down ICSF message overhead to provide more efficient sysplex communication between LPARs. In a 2-way sysplex, with both ICSF instances sharing the same active CKDS, up to a 144% improvement in CKDS update throughput has been observed over the prior release. Additionally, when a third ICSF instance in the sysplex is sharing the same active CKDS, up to a 194% improvement in CKDS update throughput has been observed over the prior release.

If an ICSF instance running at a lower level joins the sysplex at any time, all ICSF instances in the sysplex using the new level transition to the old ICSF sysplex communication protocol. Dropping back to the old ICSF communication protocol supports mixed releases of ICSF in a sysplex at the cost of the performance improvement provided by the new protocol. When all ICSF instances running at a lower level leave the sysplex, the instances using the new level transition to the new communication protocol and gain the performance improvements.

Stronger CKDS administration capabilities
Cryptographic Support for z/OS V1R11-R13 introduces Coordinated KDS Administration, which simplifies administration of a CKDS (especially in a sysplex). Coordinated KDS Administration consists of:

• Coordinated CKDS change master key
• Coordinated CKDS refresh

With the coordinated CKDS change master key capability, changing symmetric master keys and reencrypting the active CKDS becomes easier than ever. This new capability combines and enhances the old, but still supported, local change master key and local reencrypt. When all ICSF instances in the sysplex are running at the new level, and all the new symmetric master key register(s) are loaded for each instance of ICSF sharing the same active CKDS, the administrator can set the new master key or keys on one LPAR and have that change reflected on all other LPARs sharing the same active CKDS.

The coordinated change master key capability also allows an administrator to archive the old CKDS, rename the new CKDS, and create a backup copy of the new re-encrypted CKDS after changing the master key or keys. This simplifies CKDS administration by eliminating the need to perform these steps manually or to change the ICSF options data set to point to the new CKDS.
In addition, the coordinated CKDS change master key is non-blocking. Dynamic CKDS updates can run during the entire process.

**Coordinated CKDS refresh**

The faster, more elegant coordinated CKDS refresh enhances the old, but still supported, local refresh. When all instances of ICSF in the sysplex are at the new level, an administrator can refresh the active CKDS or refresh to a new CKDS from a single LPAR and in doing so drive that update across all LPARs sharing the same active CKDS.

Optionally, coordinated CKDS refresh can archive the old CKDS and rename the new CKDS. During a coordinated CKDS refresh, dynamic CKDS updates are internally suspended until the operation completes. If this cannot be tolerated, IBM recommends disabling dynamic CKDS updates before performing this operation.

**ICSF does the heavy lifting**

The coordinated CKDS refresh and coordinated CKDS change master key operations may be performed in a single system environment and provide some benefit over the “local” refresh and change master key. However, maximum benefit will be observed when performing these operations across LPARs that share the same active CKDS.

The new toned physique of ICSF HCR7790 will run CKDS sysplex update workloads faster than ever before. So kick back and let ICSF do the heavy lifting for CKDS administration. The new sysplex communication protocol and coordinated CKDS administration functions make Cryptographic Support for z/OS V1R11-R13, pound for pound, the best release to date.

For more information, refer to z/OS Cryptographic Services ICSF Administrator’s Guide, SA22-7521-16.
**Our contributors**

**Daniel Acevedo** is a Software Developer for z/OS UNIX System Services at IBM Poughkeepsie, New York. His expertise includes the z/OS UNIX file system and the checks for IBM Health Checker for z/OS. Daniel graduated from the University of Puerto Rico and has been with IBM for eight years.

**Kimberly Bailey** is a software engineer for IBM in Raleigh, North Carolina. She is a designer for z/OS Communications Server and focuses on network management.

**Jürgen Baumann** is a software engineer for z/OSMF at the Boeblingen lab. His expertise includes UI design and development in various z/OS system management products. Jürgen has been with IBM for 18 years.

**Harald Bender** has worked with z/OS and earlier forms of the operating system for more than 20 years. He leads the architecture and ongoing development of Resource Measurement Facility (RMF).

**Amy Bildzok** is a Project Executive in IBM Systems and Technology Group, and is the z/OS Product Development Team (PDT) Leader, working in Poughkeepsie, New York.

**Stephen M. Branch** is an IBM Senior Software Engineer at Silicon Valley Lab whose 25 year career has dealt with all aspects of DFSMS. He is the author of the Catalog Search Interface (CSI). Stephen holds a Master of Science Degree from Louisiana State University.

**Jay Brodfuehrer** is a Senior Software Engineer at the IBM Poughkeepsie Development Lab. His expertise includes LDAP design and architecture. Jay holds a Bachelor’s Degree from SUNY University at Buffalo and has been with IBM for 26 years.

**Colin D. Chen** is a Software Engineer for the IOS component of z/OS. His expertise includes software testing and level 3 support. Colin holds a Bachelor of Computer Science from Pennsylvania State University and has been with IBM since 2007.

**Joe Conniff** is an I/T Specialist with IBM Global Services. He graduated from the University of Scranton in 1986 with a B.S. degree in Computer Science and has been with IBM for 24 years. He is currently working on Serverpacs, CBPDO and Shopz.

**Ross Cooper**, CISSP*, is an Advisory Software Engineer at IBM Poughkeepsie, New York. In his 12 years at IBM, he has served as a designer and developer for the RACF and PKI products.

**Jonathan Cottrell**, CISSP*, is an Advisory Software Engineer who has worked for the past eleven years on the IBM Tivoli Directory Server for z/OS Development and Level 3 teams. He joined IBM in 1998 after graduating from Clarkson University with a Bachelor of Science degree in Computer Engineering.

**Erika Dawson** is a Senior Technical Staff Member (STSM) at IBM in Tucson, Arizona. In her 24 years with IBM, Erika has spent her entire career in DFSMS Development. Most of her career has been associated with the support of DFSMSdfp OAM and the software support for IBM’s tape products.

**Greg Daynes** is a Senior Technical Staff Member and the z/OS Installation Deployment Architect in IBM at Poughkeepsie, New York. His current responsibilities include z/OS software acquisition, installation, migration, deployment, and maintenance strategies. Greg is a frequent speaker at SHARE and the IBM System z Technical University (formerly zExpo).

**Anuja Deedwaniya** is a Senior Technical Staff Member in the z/OS Development lab at IBM Poughkeepsie, and software architect for z/OS Simplification and System Management initiative. Anuja has over 27 years experience at IBM in various mainframe areas and currently leads the design and delivery for IBM z/OS Management Facility.

**Jodi Everdon** is an Advisory Software Engineer at IBM Poughkeepsie, New York. She is a managing editor for z/OS Hot Topics, the Information Planner for z/OS DFS zFS and SMB, and the author of z/OS Problem Management (z/OSMF).

**Robert Gensler** is a Software Engineer for DFSMSdss at IBM Tucson. His experience includes work on EAV, CA Reclaim and large block support for DFSMSdss. Robert holds a Master of Science degree in Computer Science from the University of Arizona.

**Karín Genther** has been with IBM at the Boeblingen lab for nine years. She holds a degree in Computer Science from the University in Darmstadt, Germany. During her IBM career, she has worked in system automation and RMF.

**Michael Gierlach** is an Advisory Programmer in z/OS Communications Server in Research Triangle Park. Mike has worked on VTAM, TCP/IP, and z/OS Communications Server for 20 years, most recently focusing on enhancements to the z/OS Communications Server system resolver.

**Ilene A. Goldman** is an Advisory Software Engineer for the MVS/IOS component in Poughkeepsie, New York. Her expertise includes Dynamic I/O and Flapping Links. Ilene holds a Bachelor of Computer Science from Queens College and has been with IBM for 26 years.

**Kevin Goldsmith** is a software engineer at IBM in Tucson, Arizona. In his 29 years with IBM, Kevin has worked in MVS development, Tivoli, storage hardware and is currently a technical lead in the Object Access Method (OAM) component of DFSMS.

**Clark Goodrich** is a Senior Software Engineer at the IBM STG, System z New Technology Center in Poughkeepsie, New York. Clark’s focus is performance analysis of Java on System z and System z customer benchmarking support.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional
Saheem Granados, CISSP*, is an Advisory Software Engineer at the IBM Poughkeepsie Development Lab and currently a member of the ITDS for z/OS development team. His expertise includes Encryption and Java Security. Saheem joined IBM in 1998 and holds a Master’s Degree from Binghamton University in Computer Science.

Marianne Hammer is a Senior Software Engineer in the System z performance organization IBM Poughkeepsie, New York. For the past 10 years she has been responsible for the overall performance of a new z/OS release with focus on the BCP components developed in Poughkeepsie. She has been with IBM for 25 years.

Steven Hart, CISSP*, is a Staff Software Engineer for z/OS Cryptographic Services in Poughkeepsie, New York. In his eight years with IBM, he has worked on Integrated Cryptographic Services Facility (ICSF), Trusted Key Entry (TKE), Encryption Facility for z/OS, and Tivoli Key Lifecycle Manager (TKLM) for z/OS. Currently he works on ICSF design, development, and service.

Stephen Henkels is a software engineer in z/OS Development with 37 years experience in many and varied aspects of the z platform.

Shigeki Kimura is the technical lead for z/OS migration at IBM Japan. He has participated in key reviews of z/OS Migration, GA22-7499, since 2006. Shigeki has been with IBM for 24 years and is an International Council on Networks (ICN) author.

James McClure is a Staff Software Engineer for z/OS Cryptographic Services in Poughkeepsie, New York. In his five years with IBM, he has worked on Integrated Cryptographic Services Facility (ICSF), Tivoli Directory Server, System Management Facility (SMF), and Resource Access Control Facility (RACF). Currently, he works on z/OS system integrity test.

Sean McMillen is a software engineer who has been with DFSMSshsm for 10 years. This is the 4th article he has written for Hot Topics. He resides in San Diego and enjoys spending time with his son Jacob and the beach.

Ed Mekeel has been with IBM for 43 years. His background includes 5 years in computer operations and 7 years involved with automated operations software development. He is currently working as a member of the System z Performance organization in Poughkeepsie, New York.

Chris Meyer, CISSP*, is a Senior Software Engineer in the z/OS Communications Server design team at IBM Research Triangle Park. He joined IBM in 1983 and has spent his entire career in software development and design, focusing on file systems, electronic payment systems, communications and most recently, network security. Chris holds a BS degree in Computer Science from East Tennessee State University in Johnson City, Tennessee.

Marcel Mitran is a technical manager with the TR-JIT compiler team at the IBM Toronto Lab.

Scott Moonen is a software engineer for IBM in Raleigh, North Carolina. He is a developer for z/OS Communications Server and focuses on network security.

Tan Q. Nguyen is an IBM Staff Software Engineer at Silicon Valley Lab whose more than 4 year career has dealt with all aspects of DFSMS and DASD Common Services. He specializes in ICF Catalog. Tan has a Bachelor of Computer Science Degree from San Jose State University.

Mike Onghena, CISSP*, is a Senior Software Engineer in the z/OS security area at IBM Poughkeepsie, New York and has been with IBM for 17 years.

Joyce Anne Porter is an Advisory Software Engineer in the z/OS Communications Server development team at IBM Research Triangle Park. She joined IBM in 1985 and has spent the last 10 years working with network security for z/OS Communications Server. Joyce Anne holds a BS degree in Computer Science from Indiana University.

Kurt Quackenbush is a Senior Software Engineer at IBM Poughkeepsie, New York. He is the architect and team lead for SMP/E and z/OSMF Software Deployment, and has been a member of that team since 1987.

Ahilan Rajadeva works in z/OS UNIX Kernel and File System Development at IBM Poughkeepsie, New York.

Ruth Ray, PMP, IBM Certified Project Manager, has been with IBM for 29 years. She has led solution teams for a variety of z/OS software development projects.

Thomas H. Rose is a Software Engineer for IOS System Test at Poughkeepsie, New York. His expertise includes z10 and z196 bring-up and test. Tom has been with IBM for 25 years as a mainframe technician, z/OS system programmer, builder and tester.

Jeff Suarez is a Software Engineer for DFSMSdss development in Tucson, Arizona. His focus area includes copy services. He holds a B.S. in Management Information Systems from the University of Arizona and has been with IBM for 12 years.

Theresa Tai is a Senior Software Engineer, Senior Certified IT Specialist at IBM STG, System z New Technology Center in Poughkeepsie, New York. Theresa’s focuses are on workloads and IT infrastructure for the System z Enterprise and zEnterprise BladeCenter Extension (zBX), including the WebSphere Application Server (WAS), Portal Server, IBM Specialty Engine, Java, and Information Management software workloads.

Marna Walle is an IBM Senior Technical Staff Member in the z/OS Build and Installation department at IBM Poughkeepsie, New York. 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 EXPO, SHARE, and other customer conferences.

Michael Wang is a software engineer for z/OS Java Development at IBM Systems & Technology Group at IBM Poughkeepsie, New York. His expertise includes IBM Tivoli Directory Server and Java batch on z/OS. Michael has been with IBM for seven years.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional
Bruce Wells, CISSP®, is a Senior Software Engineer in the z/OS security area at IBM Poughkeepsie, New York. He has 20 years of experience with both the z/OS and z/VM® versions of RACF.

Glenn Wilcock is a Senior Software Engineer and architect for DFSMShsm Development at IBM Tucson, Arizona. He has 20 years of experience in developing storage management software.

Mark Wisniewski has worked in Systems Performance for 30 years. He is responsible for assessing current and future systems growth constraints and improving performance across the systems z platforms. He also drives work efforts to improve high MIPS scalability for system z.

Xiaoyan Xu is a software engineer in z/OS UNIX System Services at IBM Poughkeepsie, New York, where she specializes in the logical file system component. Xiaoyan holds an M.S. degree in computer science from Marist College.

Gina Yuan is a software engineer for z/OS Java Development at IBM Systems & Technology Group at IBM Poughkeepsie, New York. Her expertise includes Java batch on z/OS systems, system management user interfaces and clustering technologies on eServer™ platforms.

* Certified Information Systems Security Professional
** Certified Secure Software Lifecycle Professional

Clarifications, corrections, and announcements

Visit the Hot Topics website for clarifications and corrections to articles from our past issues:
ibm.com/systems/z/os/zos/bkserv/hot_topics.html

Did you find all 16 mainframes?
Looking to fill a mainframe job? Check z job board!

SystemzJobs.com is a new no-fee service that connects IBM System z clients, partners, and businesses with students and experienced professionals seeking System z job opportunities.

The System z Job Board on SystemzJobs.com is an important addition to IBM’s Academic Initiative portfolio, connecting the future pipeline of skills with the mainframe community.

This new site enables employers to post job descriptions to attract the most qualified candidates from universities and industry. Job seekers can browse a list of the hottest new jobs available worldwide and create alerts so that job postings matching their criteria will be e-mailed directly to them.