Smarter Computing with z/OS V2R1
If you’ve read any technology literature today, there’s a good chance that you came across information about the four major forces driving growth opportunities in the IT industry. Cloud computing, data analytics, mobile technology, and social networking are hot topics! It is only fitting that this issue of z/OS Hot Topics delivers articles from IBM experts about how z/OS is leveraging cloud, analytics, mobile, and social to drive the System z® platform into the future.

We haven’t forgotten that the mainframe’s world-class security is behind all of these exciting advancements, either. “New Encryption Facility goodies” and “The z-Files: The Suite B transcript,” explain several recent z/OS security enhancements, while “The Cryptographic beast release” details the newest z/OS cryptographic safeguards. Wrongdoers: beware of this beast!

“Cloud REST APIs in z/OSMF” and “Back to the future: IBM z/OS Cloud Services” outline some exciting, currently available cloud resources for z/OS and possible directions in cloud technology. “zAware and SCA-LA: Let analytics do the heavy lifting” focuses on helping you use all the tools at your disposal to detect and analyze potential issues as early as possible, using analytics to keep your systems humming.

You might consider pulling up “System z in a mobile world” on your smartphone from our website: ibm.com/systems/z/os/zos/library/hot-topics/hot-topics.html. Or simply scan the QR code on the back cover to get the full mobile effect. To help you get social with thousands of other people who are brushing up on their enterprise computing skills, check out “Got MOOC?” for details on some exciting new online (and free!) mainframe learning opportunities. As you can tell, the issue spans many areas of z interest!

Like us to win!

Following the social theme, check out the new IBM z/OS Hot Topics Facebook page: facebook.com/zosHotTopics. We post information of interest to you throughout the year. If you “like” the page before September 1, 2014, you are automatically in the running to be among the ten lucky readers, chosen at random, to receive a T-shirt from our most recent Master the Mainframe Contest in the US and Canada. You can read all about the exciting growth of the contest in “Master the Mainframe’s first world champions.”

We hope you enjoy this issue, and we look forward to seeing our ten lucky readers decked out in the hottest new mainframe fashions.

See you in the future!

The Editors
Contents: Featured articles

4 zAware and SCA-LA
   Let analytics do the heavy lifting
   BY ANUJA DEEDWANIYA, PAUL SMITH,
   AND GARTH GODFREY

48 Need flexibility? Managing capacity with CPM
   BY ADRIAN ALVAREZ DIEZ, MICHAEL
   GROETZNER, AND HORST SINRAM

51 The z-Files
   The Suite B transcript
   BY ROBERT FAIRLIE

54 The Cryptographic beast release
   BY STEVEN R. HART, BOB PETTI AND
   RITA BEISEL

57 New Encryption Facility goodies
   BY STEVEN R. HART, PAUL TAUKATCH,
   AND RITA BEISEL

64 Introducing IBM Knowledge Center
   BY JODI EVERDON AND GEOFF SMITH

6 Master the Mainframe’s first world champions
   BY MIKE TODD AND DAVE WILSON

12 Cloud REST APIs in z/OSMF
   BY MIKE YOUNG

14 z/OSMF Resource Monitoring—Reloaded
   Turn on the History Channel!
   BY JÜRGEN BAUMANN AND
   HARALD BENDER

16 Not all quiet on the consoles front
   BY PETER FATZINGER

18 Save BIG with QSAM/BSAM compression by using zEDC
   BY DAVID BETTEN, BOB ST JOHN, AND
   ANTHONY SOFIA

20 Show some resolve
   Optimizing host name resolution on z/OS
   BY MICHAEL GIERLACH AND
   GUS KASSIMIS

23 HOW TO detect and eliminate catalog contention
   BY ERIC HARRIS

25 Play it again, VSAM
   Record-level sharing
   BY NEAL BOHLING

26 Welcome to the block party
   z/OS Fixed Block Architecture services
   BY STEPHEN M. KOCIK AND
   SCOTT COMPTON

29 z/OS V2R1 Migration is going with the flow
   BY MARNAA WALLE, KIRSTEN
   MCDONALD, AND MIKE TODD

31 Valuable hints and tips for migrating to z/OS V2R1
   Advice from Professor Kimura
   BY SHIGEKI KIMURA

34 Get started with SMC-R on z/OS V2R1 for fast
   and efficient network communication
   BY MIKE FOX AND GUS KASSIMIS

37 PCI Express
   A new I/O paradigm
   BY MICHAEL GROETZNER AND
   BERTOLD REDDEMANN

39 Accelerating Java applications with new
   JZOS features
   BY GINA YUAN, MICHAEL WANG, AND
   CLARK GOODRICH

41 All aboard with zEDC
   BY AMAR DEVEGOWDA, CLARK
   GOODRICH, AND ANTHONY SOFIA

44 Using REXX to maintain the system log in real time
   BY ZHAN PENG HUO

59 Quick service for z/OS Management Facility
   BY MIKE YOUNG

60 Back to the future
   IBM z/OS Cloud Services
   BY FRANK DEGILIO

61 Peeling back batch
   IBM System z Batch Network Analyzer (zBNA)
   BY SAMANTHA HEGARTY

66 Our contributors

68 Back cover
   Got MOOC?
   Three new courses to build your enterprise systems skills
   BY KATHY PFEIFFER
Are you interested in optimizing the availability and resiliency of your z/OS systems? Two new tools can help you harness the power of data analytics to catch problems early and minimize their impact:

- IBM zAware offers faster problem diagnosis using message anomaly detection.
- IBM SmartCloud Analytics Log Analysis (SCA-LA) enables faster search, indexing, and analysis of System z logs during problem determination.

Using the capabilities provided by zAware and SCA-LA, you can analyze massive volumes of log data to surface and resolve problems quickly and ensure that your enterprise continues to run smoothly.

**Why use zAware?**

Enterprise systems are more complex and integrated than ever. To keep our lives interesting, errors can occur anywhere. Some problems are difficult to detect and diagnose because anomalies build up over time. Even if you can find the system or component where the error has occurred, the huge volume of data is often not humanly consumable, especially when seconds count. Therefore, we need information and insight quickly—ideally before the problem causes a business impact.

IBM zAware, first introduced in 2012, is a solution based on machine learning that uses cutting-edge pattern recognition techniques to look at the health of a system, pinpointing deviations from normal operations. zAware can help identify and diagnose problems earlier on z/OS systems. It can even identify changes after you apply maintenance, or when you bring new workloads online.

**How fast can problems go?**

With high-speed analytics, zAware is able to consume large quantities of z/OS message logs and provide problem diagnosis across a set of System z servers. zAware takes a proactive approach that can “point the finger” at a component and can speed up the determination of appropriate corrective actions before problems surface—problems that could potentially affect critical workloads. Proactive problem detection also allows for establishment of procedures to help prevent reoccurrence of a problem, such as identifying new z/OS messages for automated actions.

zAware can monitor multiple z/OS systems concurrently, identifying unusual behavior quickly by analyzing and scoring the z/OS messages in real time. For example, zAware can identify which z/OS image is having unusual message patterns, or which subsystem or component is behaving abnormally. It determines when the behavior started, which messages are unusual, how often the message occurred, and if the system issued similar messages previously.

**Seeing is believing**

Often, after you apply maintenance to the system or make some configuration changes, there can be abnormal behavior that is difficult and laborious to resolve. zAware can help! In addition to flagging unusual behavior, it can easily pinpoint changes caused by new software levels, configuration settings, or workloads. zAware helps you to distinguish between changes that are normal for the new workload and potential problems that you need to address. This early diagnosis can shorten the impact time and help you avoid similar problems in the future. It also gives an edge in your ability to respond to events.

zAware runs in a special-purpose firmware partition on zEnterprise EC12 or zEnterprise BC12 servers, and can monitor the OPERLOG from one or more System z servers running z/OS V1R13 (with additional service) or later.

zAware features an easy-to-use graphical interface that you can use to specify and watch the systems you want to monitor. Other system management solutions (including NetView® and OMEGAMON®) can consume the anomaly information from zAware using one or more application programming interface (API). These solutions can use the data from zAware to provide better alerts, or to examine the zAware scores within the context of other metrics.
Model behavior

zAware models the message patterns of each z/OS system it monitors based on past message traffic, when the systems are running normally. Then, it scores the messages in real time by comparing them to the model to determine if there is an anomaly. It presents the anomalies in a graphical view, with bars of varying colors and heights. Each bar represents a 10-minute interval. The current bar updates in context every two minutes (as shown in Figure 1). You can click and drill down on the details of each interval, which tell you the particular messages, frequency, score and how each message contributed to the overall score.

zAware updates the model every 30 days by default; based on the past 90 days of activity (you can change the interval value).

If you have a day with unusual activity and you do not want zAware to include it in the model, you can identify and remove that day from future modeling for the system. Similarly, if zAware identifies any particular message as an anomaly, but you want it treated as a normal message, you can remove it from future analysis scoring.

IBM zAware is a self-learning, integrated solution that analyzes messages in near real time to provide insight into the behavior of your z/OS system.

Why use SCA-LA?

Proactive analytics tools such as zAware, coupled with monitoring tools such as NetView and OMEGAMON, help to surface potential problems. After those problems surface, you need the proper tools in your toolbox to troubleshoot the problems quickly. One such tool is SCA-LA.

Logs are a major source of information on all systems, and even more so on System z. After zAware surfaces a z/OS log anomaly, you most likely need to perform a deeper dive into the logs.

A typical approach to problem determination is to search log contents just prior to the time that the anomaly was detected. Since the System z components are usually part of a more complex workload, it makes sense to analyze the logs from all of the components of the workload. SCA-LA does just that. It consumes, consolidates, indexes, and annotates log messages from multiple sources, enabling you to perform searches across multiple logs. SCA-LA is increasingly valuable if your workload spans multiple heterogeneous systems.

SCA-LA Insight Packs annotate and index the logs to enable quick searches across huge volumes of data from a single search bar. With the recent availability of the SCA-LA z/OS Insight Packs, we can now support log analysis of the entire workload, end-to-end. As SCA-LA builds the search results, it also analyzes the logs, looking for common errors and surfacing the error diagnostics to help speed you through the problem determination process. This analysis helps pinpoint problems in one component that can contribute to problems in another.

Expert advice

In addition, SCA-LA provides graphical application views to help visualize problems in one or more logs. It’s flexible, too, so you can build your own application views and save your favorite string searches. When you focus in on the problem area, SCA-LA provides an expert advice tool that enables a quick search of the IBM support site to help find a resolution to the problem.

SCA-LA consists of two components:

- SCA-LA engine
- Log Forwarders.

The SCA-LA search engine resides on an x86 Linux system (today) and is capable of indexing and annotating messages from multiple log sources to enable fast search and analysis. The Log Forwarder installs on the z/OS LPAR where the logs reside. Log messages stream from a Log Forwarder to the SCA-LA search engine every 10 seconds. SCA-LA also provides a simple user interface to search and display search results and insights.

Bad news for problems

zAware and SCA-LA provide a complementary set of capabilities to keep your enterprise running smoothly. Let zAware proactively surface the anomalies in the z/OS logs, then use SCA-LA to complete the problem determination effort. With these data analytics tools on your side, your systems can be even more available and resilient.
Earlier this year, IBM brought together 43 students from 23 countries and 6 continents to compete in the first-ever IBM Master the Mainframe World Championship. The six winners were recognized onstage at the Mainframe50 celebration in New York City.

The regional winners had to demonstrate serious technical prowess. Master the Mainframe contests require students to logon remotely to a zEnterprise mainframe system and complete increasingly difficult tasks. As they progress through the contest, students can win increasingly valuable prizes, from custom T-shirts and sweatshirts to tablets and laptops.

The contest owes its popularity in large part to its accessibility—it is advertised as “No experience necessary,”

BY MIKE TODD AND DAVE WILSON

Figure 1. The top six winners on stage with Pat Toole, General Manager, System z.
and students pick up mainframe navigation and programming skills as they go. In the US and Canada contest, which drew a record-breaking 5,601 students in 2013 (including over 2,000 high school students), 75% of contestants reported having no prior mainframe experience.

The contest aims to take students who may have no knowledge of the mainframe and teach them skills and concepts that will prepare them for further exploration of the platform, both academically and professionally.

To win a T-shirt, a student needs to spend perhaps an hour on the system, learning basic commands and navigation. To advance to the later stages, they tackle extremely difficult challenges, programming in various languages, interfacing with several middleware products, and creating applications that tackle real-world problems.

In the World Championship, students competed remotely from their home countries for three weeks before flying to New York City for the final rounds. They visited the IBM plant in Poughkeepsie, NY, for the first in-person round of the competition and then returned to New York City for the finals at the IBM facility at 590 Madison Avenue.

From the field of forty-three initial competitors, six separated themselves from the pack. They received their awards at the Mainframe50 event in New York City in front of hundreds in the audience. Thousands watched the live-stream of the presentations.

The winners of the first IBM Master the Mainframe World Championship are:

1. Yong-Siang Shih, Taiwan
2. Rijnard van Tonder, South Africa
3. Philipp Egli, United Kingdom
4. Mugdha Kadam, United States
5. Shahini Sengupta, India
6. Aaron Call Barreiro, Spain

The top three students also won a top-of-the-line Lenovo laptop with Rational® Developer for System z installed to aid them in their future mainframe endeavors.

Interested in recruiting from the contest or participating in it? The Master the Mainframe Contest has a new worldwide web portal to help students, educators, employers, and parents find everything they need to know about becoming involved with the program. Visit the portal at ibm.co/mtm, where you can also see video highlights from the World Championship, view profiles on all 43 students, and sign up for contests in your country.

If you are a student or if you know a student who could benefit from learning marketable mainframe skills in a fun and challenging environment, please share the home page with them and encourage them to compete. We look forward to seeing the next generation of students who will rise to the challenge and Master the Mainframe.
The independent IT industry analyst, International Data Corporation (IDC), predicts that by 2015 more people will access the internet using a mobile device than a personal computer. IDC identified the following five mobile trends with significant implications for the enterprise:

- **Mobile is primary:** 91 percent of mobile users keep their device within arm’s reach 100 percent of the time.
- **Insights provide new opportunities:** 75 percent of mobile shoppers act after they receive a location-based message.
- **Mobile is about transacting:** 96 percent year-to-year increase (2011 - 2012) in mobile sales on cyber Monday.
- **Continuous brand experience:** 90 percent of users use multiple screens as channels that come together to create integrated experiences.
- **Enables the Internet of Things:** 2 billion global machine-to-machine connections in 2011 are expected to increase to 18 billion by the end of 2022.

See the “System z trends for mobile” on page 11 for more information.

**System z view of mobile**

Back in 1998, when the web was revolutionizing how people accessed enterprise data, many of the same concerns you hear about mobile were voiced then, especially about security and performance. IBM spent many years building the hardware and software infrastructure on System z® to make web access for z/OS transactions and data simple and secure.

From the System point-of-view, mobile is simply another channel to get to those same enterprise transactions and data. But now customers have the advantage of all the infrastructure that has been built since 1998 to handle web access to System z.

System z has always had this dual role—as a well-integrated, trusted repository of data and business logic, and as an environment where application programs use that data and logic to interact with customers. Our new mobile world has fancy terms for these roles.

Today’s mobile world uses contemporary language for these roles:

- **System of Record:** The trusted repository of your key business data with unmatched availability, scalability, and security: z/OS.
- **Systems of Engagement:** The agile, flexible, easy-to-deploy execution environment for applications that interact with your customers: Linux on System z.

System z bridges Systems of Record and Systems of Engagement to provide essential services for mobile applications.

z/OS is the System of Record environment, providing the rock-solid access to your key data and transactions. Mobile applications can easily access all the z/OS subsystems, such as CICS®, IMS®, DB2®, and WebSphere® MQ (more information later). z/OS scalability helps you avoid lengthy delays when you must scale up to meet the demand of new mobile applications.

Linux on System z is an agile, cloud-based environment, where new mobile application infrastructures are provisioned quickly and mobile applications that are deployed and tested and put into production. Using collocation with z/OS gives these applications the advantage of low latency access to z/OS transactions, which means faster response times for your customers.

**IBM MobileFirst**

IBM MobileFirst is a family of products that can help you “put mobile first.” The family includes products that are used for:

- Securing the mobile device, applications, and network
- Improving the mobile user experience
- Creating, managing, and serving mobile applications
- Connecting to back-end data, and services to help you build your entire mobile infrastructure including your mobile app.

The IBM MobileFirst family of products covers the entire mobile application lifecycle. Everything from designing and developing with Rational® Developer for z or Worklight® Studio to testing with Rational Test Workbench or IBM Mobile Quality Assurance (MQA). As shown in figure 1, there are multiple multiple products for every step of the way.
IBM Worklight

IBM Worklight is a key component for building mobile applications on System z.

IBM Worklight® is a solution for creating and hosting mobile applications. Just as IBM introduced WebSphere Application Server to host emerging Java web apps in the early 2000s, Worklight server plays the same role for mobile applications. Worklight provides a runtime environment for the parts of the mobile app that runs on the server. IBM Worklight contains the following four main components:

Worklight Studio

Worklight Studio is a component that runs on a workstation and installs as an Eclipse plug-in to Rational® Developer for System z®. Developers can continue to use the familiar Rational Developer interface, but incorporate new Worklight Studio capabilities to create applications for mobile devices.

Device runtime components

The device runtime components give IBM Worklight one of its main advantages — the ability to write code once and deploy it to several mobile platforms, such as iOS, Android, Windows, and BlackBerry. The device runtime components are similar to a JVM, providing a device-independent language layer.

Worklight Server

The server component of Worklight provides key services for mobile applications.

- Push notifications: Push notifications are those messages that pop-up on your mobile device. They are sent by the server components of apps you are running. In order to send one, a mobile app needs to know how to invoke the proper push server for each device. iOS, Android, and Windows all have separate push servers for their devices. Again, Worklight helps by providing a common service that your mobile app can call to send a push notification. Worklight also figures out what type of device each user is using and sends the push notification to the proper servers.

For example, your mobile application wants to convert speech to text. You can use a cloud service from AT&T to do that. You open an account at AT&T and sign up to use the API. Now, in your mobile app, you need to write code to record a sound file and send it to the AT&T API. The method for doing so (open the microphone, record sound, get the file back into your app) is different for every mobile device. To support the app on all of the devices, you have to write separate versions of the app for each platform.

The device runtime components solve this problem by creating Worklight APIs for the tasks. You code the app to use the Worklight APIs to open the microphone and retrieve a sound file, and Worklight translates your request into the appropriate device APIs.

Figure 1. Worklight overview.

Figure 2. System z bridges Systems of Record and Systems of Engagement.
• **JSON conversion:** JavaScript Object Notation (JSON) is the preferred data format for communicating with mobile devices. It is a standard because it uses little bandwidth, and doesn’t take much battery power to parse. All communication between the Worklight server and the mobile app uses JSON. Many back-end systems do not support JSON, so Worklight includes a conversion service.

• **Analytics:** Similar to web server statistics, you need statistics for the use of applications that are being hosted by the Worklight Server.

• **Authentication:** Worklight provides several ways to call the authentication provider you choose.

• **Adapters:** These bits of “helper code” offload some of the work that a mobile application has to do when it is communicating with back-end systems. Worklight ships several adapters, including HTTP communication, SOAP communication, and SQL communication. Adapters are an extensible feature of Worklight, and new ones can be written by app developers and vendors. Adapters are written in JavaScript (JavaScript can run within Worklight on Linux on System z). It is common for a mobile application to also contain some adapters to catch events from System z or communicate to CICS or IMS.

• **Enterprise App Store:** Worklight Server hosts a private app store that your company can use to distribute apps to employees. Worklight Server runs in Linux on System z as a Java app inside WebSphere Application Server or Liberty.

**Worklight Console**

Worklight Console is the browser-based console to deploy and control applications that are being served by Worklight. Worklight Console runs in the Worklight Server.

**Making System z transactions and data easy to use for mobile developers**

MobileFirst and Worklight are only part of the story for mobile on System z. The most important focus is to make it easy for mobile applications to access and use the transactions and data you already have, whether that data is coming from CICS, IMS, WebSphere Application Server, or WebSphere MQ. Whatever mobile font-end you choose to use (whether it is Worklight or some other mobile framework), it is easy to use the transactions and data you already own. IBM continues to add new mobile-friendly APIs to the APIs you are already know.

**Closer look: CICS**

For many years, CICS created CICS Web Services APIs and talked to those APIs using SOAP/XML. Many System z customers are already running mobile apps using those APIs. SOAP and XML are often not the protocols mobile developers want to use, though. That’s why CICS shipped “CICS TS Feature Pack for Mobile Extensions” that provides APIs that use HTTP/JSON. Now, mobile developers can send JSON data to CICS and receive JSON data in return.

In Figure 3, you can see how to:

• Add a mobile front-end to easily modernize a CICS application.
• Create a new mobile application from existing CICS transactions.

**DB2 for z/OS**

It’s also important to be able to store JSON data in a database, where complex joins or SQL queries aren’t needed. This need gave rise to noSQL cloud databases such as MongoDB™, which offers a simple set of APIs to create a database, store data, select data, and delete data. DB2 for z/OS was recently enhanced with the ability to store and retrieve JSON-format data, and expose new APIs that simplify the process. This gives application developers on System z the best of both worlds: NoSQL agility and flexibility that is built on the trusted foundation of DB2.

**Figure 3. Worklight provides a mobile interface to CICS services**
z/OS Connect
Even with all the mobile-friendly enhancements, IBM wants to make discovering and using z/OS transactions as easy as possible. IBM shipped IBM WebSphere Liberty z/OS Connect, which is an application that provides a central set of REST APIs for any CICS, IMS, and Batch transaction in z/OS. It provides APIs for application developers to discover which z/OS transactions are available to them, and the API syntax. It also provides security, auditing, and chargeback features.

Next
Whether it’s time to add a mobile face to an existing z/OS application or create a new mobile application using existing z/OS transactions or data, the resources you need are only a click or two away.

Check out these mobile resources:
• IBM Redbooks Point-of-View publication: System z in a Mobile World highlights the secure and stable base that you need to extend existing System z enterprise data and transactions to mobile users.
  www.redbooks.ibm.com/redpieces/abstracts/sg248215.html

System z trends for mobile
Do you remember the five mobile trends we mentioned earlier? System z has the following unique characteristics that support mobile applications and the trends.

Mobile is primary
Expect a growing percentage of your transactions to come from mobile devices, and expect your overall transaction rates to rise as your mobile channels become more popular. z/OS enables massive yet simple scalability in a single footprint to handle the workload of millions of mobile devices. z/OS Workload Management (WLM) can help ensure that your crucial applications remain responsive during sharp spikes in demand. Reduced pricing for mobile workloads can help ensure that your IT investments scale with your growth and enhance your business returns.

Insights provide new opportunities
Advanced analytics against the transactional data you have on System z today can be combined with unique data from mobile devices to yield unprecedented insight into customer behavior and attitudes.

Mobile is about transacting
Easy-to-consume APIs from CICS, DB2, and IMS allow you to leverage your investment in z/OS transactions to quickly add a mobile channel. System z provides low-latency access to critical enterprise data. Mobile usage patterns favor short, read-only data requests (think users who are checking account balances). Fast access to operational data with low latency is the key. The mainframe offers exceptional I/O with dedicated hardware I/O processors. This reduces latency, which increases mobile app response times.

Continuous brand experience
With System z, you can aggregate data from many channels into common System of Record databases, so you can build applications that use an always-current single copy of user data. Worklight security integrates with z/OS security, which can help you provide end-to-end security and data privacy for mobile apps and customer data.

Enables the Internet of Things
z/OS enables massive and simple scalability for transactions and data in a single footprint to gather and analyze the data from millions of devices and sensors.
When the z/OS Management Facility (z/OSMF) was first released, it was billed as the “new face of z/OS” and promised a host of new graphical user interfaces for z/OS tasks. Today, many system programmers use the great features found in z/OSMF like the Communications Server Configuration Assistant, z/OS Software Management, Workload Management, and the new SDSF UI. But, did you know that z/OSMF offers more than just those snazzy web interfaces? Behind the scenes, it also hosts some of the first z/OS Cloud REST services for your developers and off-platform applications!

REST stands for “Representational State Transfer” and is the architecture that underlies web technologies like HTTP. REST interfaces use everyday web URLs and standard HTTP methods and links to get and put data. By transmitting over normal HTTP, the network traffic can more easily transit through proxies, firewalls, and load balancers. With REST interfaces, an application can be written quickly and easily without needing to worry about specialized libraries or the underlying implementation details of the platform or programming language involved.

The z/OSMF REST Services

In z/OSMF version 2.1, IBM offers three major REST services. These can be used by application developers both on and off the mainframe, since they are accessible via the same URLs as z/OSMF. As a plus, because z/OSMF is secured with HTTPS/TLS, all of the REST services offer secure communications automatically. Several IBM products are already using these REST interfaces to replace the legacy communication methods they used in the past. The three primary services z/OSMF offers today are:

- Jobs Interface
- Dataset and File interface
- TSO/E address space services

This article will describe each of these services and what they offer to application developers. For more details on how to use these services, check out the IBM z/OS Management Facility Programming Guide SA32-1066. In the examples that follow, replace “your-zosmf-hostname” with system name (and port if necessary) where you have z/OSMF installed.

The z/OSMF Jobs Interface (aka SubmitAPI)

The z/OSMF REST Jobs API is the oldest of the interfaces available in z/OSMF. It allows your users to submit and manage batch jobs and their output. In the past this kind of operation needed to occur either at a terminal, or using special libraries or applications. Many customers today use FTP or other unsecured communication methods to submit batch jobs and transfer the resulting output.

With the z/OSMF Jobs API, all communication is secure and uses standard HTTP requests, so integration with existing applications is simple. You can even try it out yourself using your web browser! Try typing the following URL in your browser to start: https://your-zosmf-hostname/zosmf/restjobs/jobs. After authenticating, you should see the jobs (if any) for your userid. If you’d like to see a different user’s jobs (and you are authorized to do so) you can add the query option ?owner=userid to the end.

Seeing a list of jobs is barely scratching the surface of the REST Jobs API. Perhaps you’d like to view the output from one of those jobs you found? Simply take the jobname and jobid from your list and enter them into your browser like this: https://your-zosmf-hostname/zosmf/restjobs/jobs/jobname/jobid/files. This will return a list of output files for your job along with some...
information about the step that generated them. You can even go a step further (no pun intended) by retrieving the contents of that output directly in your browser. The file-list output provides a special “escaped” link that applications can use to download the appropriate spooled data. That relevant link will work just as well when typed manually too; substitute “nn” with the job file id number that was returned in the list:


The z/OSMF Jobs API also has the ability to submit, cancel, and modify a job’s execution class. Those operations use the same URL you used in your web browser example but require special HTTP methods and headers to be passed on the request. It is very easy for an off-platform or web-based application to submit these extra headers, thereby allowing developers and users without System z experience to submit and manage jobs without special training or software.

The z/OSMF Data Set and File Interface

New in z/OSMF V2.1 with APAR PM98630 is another REST API designed to let your applications access data sets and z/OS UNIX System Services files. In its first release the API offers two primary functions: the ability to list z/OS data sets that match a specific filter; and the ability to list z/OS UNIX directories. This interface is undergoing active development, so be on the lookout for new function designed to allow you to access data stored on z/OS.

Like the Jobs interface, you can try out part of the File interface directly in your web browser. Type the following address into your browser to see all of the datasets under your High Level Qualifier:

https://your-zosmf-hostname/zosmf/restfiles/ds/?dslevel=hlq

where “hlq” is the starting qualifier of the data sets you’d like to list.

Similarly, for z/OS UNIX files, you can use the URL: https://your-zosmf-hostname/zosmf/restfiles/fs/u/username to list the contents of the home directory for “username”.

The access controls for viewing these data set and file listings are generally the same as the access controls placed on a TSO/E user. If your user ID has access to z/OSMF and authority to list this information in TSO/E, it should also be able to use the data set and file interface as well.

z/OSMF TSO/E Address Space Services

The final REST API described in this article may be the most exciting. Users of the z/OSMF ISPF or SDSF tasks may already be familiar with the way z/OSMF creates a TSO/E session behind the scenes to process your work. This same functionality is available for your applications as well. Unlike the other two REST APIs, it isn’t possible to experiment with this one in your web browser, but developers of web and remote applications can use this API to start TSO/E sessions, launch applications, and send/receive messages.

Starting a TSO/E session and sending data to it are fairly straightforward tasks. The session is started by passing some data like the TSO/E Proc name and the desired size of your “virtual screen” via POST to the URL: https://your-zosmf-hostname/zosmf/tsoApp/tso. The server will return a “servletKey” value which you can use for further communications with that TSO/E session. Messages to and from the TSO session occur via PUT and GET requests to the URL:


Starting and communicating with applications inside of the TSO/E address space is a little more complicated. Communication to and from applications is handled using z/OS UNIX IPC Message Queues. The z/OSMF server takes care of creating and managing these message queues for you, but any TSO/E applications you launch must be enabled to read/write to these queues. More details about how these queues work can be found in the IBM z/OS Management Facility Programming Guide SA32-1066 and z/OS MVS Programming: Callable Services for High-Level Languages SA23-1377.

Hopefully this introduction to the z/OSMF Cloud Services has inspired you to investigate how these REST APIs might be useful to you. If you have a specific service or API you’d like to see added to z/OSMF, please feel free to use the new Request for Enhancement (RFE) site to send us your request! RFE is available here:

ibm.com/developerworks/rfe/?BRAND_ID=352
visualize your enterprise-wide performance data graphically using the z/OSMF Resource Monitoring (RM) task. z/OSMF RM delivers a set of predefined dashboards that cover the most common performance metrics. In addition, z/OSMF RM lets you define your own dashboard, to combine measurement data into the views that most interest you.

As soon as you open a dashboard, it starts to collect the current performance information, interval by interval, as provided by the RMF™ Distributed Data Server (DDS). This is well-suited for when you use z/OSMF RM as a performance monitor with all dashboards of interest open and running continuously. If an incident happens, you can use the sliders in the dashboard to look at the data in earlier iterations to analyze the performance problem. Up to now, it was not possible to go back in time to view performance data from the past when the dashboard was not yet running.

Since March 2014, the Resource Monitoring task of z/OSMF V2R1 has provided this feature to populate your dashboards with data from the past. Click the Retrieve Historical Data item of the Actions menu to open the Retrieve Historical Data window, where you can specify the time frame for which you want to pull data, as shown in Figure 1.

The Metric groups list allows you to specify for which metric group data should be retrieved. There are three different ways to specify the time frame: Past, Dates from, and Dates between. By default, the data is retrieved using the original sample range with which the data was gathered (Monitor III MINTIME). You can specify a larger range, for instance, to reduce the number of retrieved intervals when you want to investigate a long time frame.

When you press OK, the dashboard switches to the Retrieving historical data state. Interval by interval is now loaded into the dashboard. When done, the dashboard switches back to the previous state (Running or Paused).
The time range of available data is determined by the setup of the RMF Monitor III Data Gatherer. When you define virtual storage access method (VSAM) data sets to the gatherer, you can extend the reporting range up to several days.

With this easy access to performance data from the past, you can make direct hits when investigating performance issues that happened hours or days ago.

Another smart enhancement of the Resource Monitoring task is the Export action. It allows you to export all values to comma-separated values (CSV) format for later use in spreadsheet applications.

Clicking the Export item in the Actions menu shows the Export wizard. On the first page, specify whether you want to export individual metrics, a complete metric group, or the entire dashboard, as shown in Figure 2. In addition, specify the start and end intervals to export. All intervals available in the dashboard are eligible to be exported.

If you export a metric group or the entire dashboard, you can click the Finish button immediately. If you want to limit the export to a subset of metrics or resources of one metric group, the second wizard page can help you do that.

After clicking Finish, the data is transformed to CSV format and offered for download by the browser. When you export the entire dashboard, a CSV file is generated for each metric group and packed into a ZIP file, which is then downloaded. Use any spreadsheet application of your choice to load the CSV files.

You can leverage the full power of both new features by combining them: retrieve performance data from the past and export this data to a spreadsheet application for further evaluation, investigation, and analysis.

The March 2014 Enhancements for z/OSMF V2R1 include these new functions.

For more details about the March 2014 enhancements, see:

ibm.com/systems/z/os/zos/features/zosmf/enhancements_032014.html

For more information about the enhancements to z/OSMF RM (PTF number UI16026), see:

ibm.com/support/docview.wss?uid=isg1PI08825

Figure 2. The z/OSMF RM Export wizard
Not all quiet on the consoles front

BY PETER FATZINGER

Beginning with a feature available with z/OS V1R4 and continuing through the creation of distributed mode in z/OS V1R10, the consoles component went through a metamorphosis. A complete redesign of the message production and queuing, as well as a new paradigm for sysplex communication, has led to a leaner, meaner, more reliable component. Even though the dust on this major overhaul has settled, the consoles team has continued to deliver enhancements, some of which weren’t even possible before the restructure.

One enhancement that is only available in distributed mode (and let’s face it—there’s no reason you shouldn’t be running distributed mode by now) is the ability to dynamically add and delete consoles. It has been possible to delete consoles through samplib programs IEARELCN and IEARELEC for quite a while, but many customers are unwilling to use them, and it’s never been possible to dynamically add new console definitions. In z/OS V2R1, processing has been added to allow consoles to be dynamically added and deleted!

In the past, an IPL was required to define a device as a console, even though all the other definition requirements could be done dynamically (for example, using the ACTIVATE command). Now, you can define a new console on the fly by using one of the following methods:

• Create a CONSOLxx parmlib member containing the definition and issue the SET CON=xx command:

```
CONSOLE
DEVNUM(3E7)
ROUTCODE(2,4-10,12)
NAME(MFRAME50) /* CONSOLE NAME MFRAME50 */
AUTH(MASTER) /* MASTER AUTHORITY */
LEVEL(ALL,B) /* EVERYTHING BUT BROADCAST */
DEL(RD) /* ROLL-DELETABLE MODE */
RTME(1/4) /* SCREEN ROLLTIME OF 0.25S */
MSCOPE(*) /* RECEIVE MESSAGES FROM THIS SYSTEM ONLY */
```

• Add the new console to an existing CONSOLxx member and issue the SET CON=xx command.

The second method provides additional benefit and power, but it must be used wisely. By adding the new console definition directly to your in-use CONSOLxx member, you won’t need to remember to add it later, but be aware that the entire CONSOLxx member is processed. No changes are made to any existing console definition, and no console is deleted, even if there is no definition for it in the CONSOLxx member.

However, all new consoles are defined and all the settings on the DEFAULT, INIT, and HARDCOPY statements are applied to the system. (including some that could not previously be updated dynamically).

In addition to the existing samplib programs, a command-driven method for removing unneeded, inactive consoles is now provided by using the system command SETCON DELETE,CN=consname. The command removes all relevant information about the specified inactive MCS, SMCS, System, Subsystem, EMCS, or HMCS console from the system.

Another recent area of focus was aiding in the reply to rare or unexpected WTORs. Because installations have become more complex and experienced system operators have become scarce, an event such as an unusual WTOR can bring a system to a halt. We’ve been told it can take 30 to 45 minutes before the correct response is determined and agreed upon, and in the meantime, critical processing might hang and affect other tasks in the sysplex.

Beginning in z/OS V1R12, a facility to automatically reply to WTORs was created. IBM provides an expandable list of WTORs along with the preferred reply in a new parmlib member, AUTORxx. In this parmlib member, you can specify a policy for how you would like to reply to a specific WTOR by supplying the message identifier, reply text, and wait time. IBM provides an initial policy in the AUTOR00 member of parmlib. One example from that member is for WTOR IXC394A, and allows the operator 60 seconds to reply to the WTOR. If there is no response in that time, the system is told to reply ‘N’.
You can customize your own AUTHORxx parmlib member and activate it by using the system parameter AUTHOR=(xx,00) or dynamically by using the system command SET AUTHOR=(xx,00). While none of the IBM-specified messages take advantage of this feature, you can also specify that there should be no delay in responding. For example:

```
/* IBM1234A REPLY 'YES' IF YOU LIKE MAINFRAMES */
/* */
Magid(IBM1234A) Delay(0S) Reply(YES)
```

Take care when specifying a delay of 0 because you might override existing automation. Also, this setting does not give the operator an opportunity to reply.

You can see the Auto Reply policy by issuing the DISPLAY AUTOR, POLICY command:

```
CNZ2603I 15.05.19 AUTOR POLICY
POLICY ACTIVATED AT 12.27.53 ON 04/21/2014
NOTIFYMSGS(HC)
FROM PARMLIB MEMBERS 00
---MSG ID--- DELAY MEM RATE ----REPLY TEXT-----
IEE800D  60s  00 --- NO
```

Any outstanding WTORs that are acted upon by Auto Reply can be displayed by using the DISPLAY AUTOR, WTORS command:

```
CNZ2604I 12.46.02 AUTOR WTORS
0007 STATUS=12.46.50 SYS=SY1
MSG=IEE800D CONFIRM VARY FORCE FOR ZZZ - REPLY NO OR YES
REPLY=NO
```

This command response shows that IEE800D is outstanding on system SY1, but shortly after 12.46.50 the system will reply NO to the WTOR.

In z/OS V2R1, the Auto Reply function was extended to synchronous WTORs. It is critical that these WTORs be responded to quickly. While they are outstanding, little or no other processing can occur on the system, and systems in this state can often appear to be unresponsive.

Due to the rarity of these WTORs and the unfamiliarity with how to deal with them, our next enhancement improves how they are presented to the operator. By default, these WTORs are displayed on the system console (the Operating System Messages Panel) located on the Hardware Management Console (HMC), also known as the console of last resort. The system console is a difficult interface to master for those who haven’t spent much time using it. To make matters worse, when the synchronous WTOR is routed to the system console, all the other consoles on the system appear to be “frozen.” Often, a decision is made to re-IPL the system, even though a simple reply would allow the system to continue.

The process is made more manageable by using a SYNCHDEST group (defined in parmlib member CNGRPxx). The system chooses the first eligible console in the group on which to display the WTOR. Because getting a reply to the WTOR is critical, if no response is received within approximately two minutes, the WTOR is displayed at the next eligible console. The reply can only come from the console that is displaying the WTOR. This process continues until a reply is sent or the WTOR reaches its final destination at the system console on the Hardware Management Console. In z/OS V2R1, this processing has been enhanced to help the operator determine the WTOR in question and to locate the console that is displaying it.

Now, when a synchronous WTOR is to be displayed, all locally attached, active, non-printer, MCS consoles (except the console that will display the WTOR) are cleared and message CNZ4215W is displayed on those consoles. The message indicates the message ID of the WTOR, where the WTOR is currently displayed, and whether the Auto Reply function will manage the response. To ensure that this isn’t too subtle, the display supports color, the entire screen turns red with black text.—Just try to ignore that! When the message is moved to the next console in the list, all the displays are updated with a new message indicating the new location. By paying attention to the screen, it should be much easier to know what the problem is, and to know which console is waiting for the reply.

The final enhancement involves a new type of console that can be used to alleviate some of the difficulty associated with using the system console on the HMC. As implemented, the system console does not look like other operator consoles and, due to constraints, cannot handle all of the system message traffic.

The enhancement uses the existing Integrated 3270 Console session that is available on the HMC to provide a new type of console known as Hardware Management Console Multiple Console Support (HMCS). The console looks identical to an MCS console, and can be used during NIP, or any time after. It is especially useful for installations that moved completely to SMCS consoles to provide an active console between the time NIP ends and when z/OS Communication Server is available.

To bring this discussion full-circle, the HMCS console can be dynamically defined (although there is a limit of one per system). Add a definition to a CONSOLxx parmlib member and issue the SET CON=xx command:

```
CONSOLE DEVNUM(HMCS) ROUTCODE(1-10,12-128) AREA(25) LOGON(OPTIONAL) AUTH(MASTER) NAME(HMCS&SYSCLONE.) RBUF(10) RTME(1/4) RNUM(25)
```

The consoles component continues to evolve. To take full advantage of the recent changes, migrate to distributed mode, create (dynamically!) an HMCS console, and implement an AUTOR policy…and ensure that you keep your eyes and ears open for the next updates because it’s definitely not quiet on the consoles front. ■
Using actual performance results, we’re about to show you how z Enterprise Data Compression (zEDC) can offer big savings when used with QSAM and BSAM, with a great compression ratio, low processor usage, and potentially a reduced elapsed time for batch workloads.

Workload overview

To demonstrate the performance benefits of zEDC for compression of QSAM and BSAM data sets, we ran a number of batch jobs concurrently. Each job consisted of a standard utility program that reads and writes MVS data sets.

After each run, we captured the processor time and elapsed time of all the batch jobs. We also recorded the number of inputs and outputs, as well as the size of the input and output data sets.

We ran this test for each of the following data set types:

- Large format sequential data sets (not compressed)
- Extended format sequential data sets (not compressed)
- Extended format compressed by using generic compression
- Extended format compressed by using tailored compression
- Extended format compressed by using zEDC compression.

We also selected sample data sets that we thought would make sense in the context of z/OS and could be easily recognized and understood.

Job characteristics

For these tests, we selected a set of standard utilities that are I/O intensive, with nominal processor usage. We did this to make it easier to measure the processor impacts of various compression options. This approach, however, also results in best-case elapsed time results. Applications that are more processor intensive can probably see less improvement in elapsed time than what we measured in our tests.

The following standard utilities were used:

- IEBGENER (BSAM access)
- IEBDG (QSAM access)
- DFSORT (BSAM access).
  - The OUTFIL option was used to write multiple outputs in the same job step.
  - The MERGE option was used to read multiple inputs in the same job step.
- IFASMFDP (QSAM access)
- IDCAMS REPRO (QSAM access)
- IEBCOMPR (BSAM access).

Data characteristics

The following data sets were used:

- SYS1.LINKLIB and SYS1.LPALIB, which were copied into sequential files. These data sets consist of mostly binary (executable program) data.
- SYS1.PARMLIB, SYS1.PROCLIB, and SYS1.MACLIB copied to sequential files. These data sets consist of mostly text data.
- An SVC dump and a stand-alone dump.
- SMF data, including various combinations of SMF record types.
- A directory of company personnel, including their contact and job information.
- An XML document that contains a profile of a running measurement system.
- A listing of the locations of various Java executable methods in memory. This is all text with significant printable hex.
- A DB2 archive that was generated by using IBM Optim z/OS.

Data set tuning

For tests with no compression, JCL buffer sizes were tuned for performance. Data sets were allocated with system-determined block size and we specified appropriate BUFNO and NCP values in the JCL.

Results

The compression ratios using zEDC compression were significantly better! Using zEDC, the combined size of our sample data sets was reduced by 5.5 times. Furthermore, zEDC consistently provided a better compression ratio than either generic or tailored compression. In our test, generic compression reduced the total data set size by 2.1 times, and tailored compression reduced the total data set size by 2.4 times. See Figure 1 for the compression ratios with generic, tailored, and zEDC compression.

Because our batch jobs were I/O intensive, with little processor consumption, we saw a significant reduction in elapsed time with zEDC compression. Our results showed an elapsed time reduction of up to 65% relative to non-compressed extended format data sets and up to 59% compared to generic and tailored compression. Your elapsed time savings will be less if your workload is more processor intensive or less I/O intensive.
Our testing began with a single zEDC Express device. Each zEDC Express device has a FIFO work queue, but because of the high performance of the device, it requires a significant workload to begin to see heavy queuing. From a software perspective, the zEDC requests are blocking processing, so queue time can contribute to an elongated elapsed time. With the parallel workload and the size of the system, the queue began to build to about two requests with the zEDC Express running 80% utilized.

The new RMF PCIE reports were useful in analyzing this. A new RMF postprocessor PCIE activity report is available in XML output format, and RMF also provides new overview conditions for the postprocessor based on SMF record 74 with the new subtype 9. This RMF data provides information such as percent busy, MB per second that is transferred for reads and writes, response and queue times, compression and decompression requests per second, compression and decompression ratios, and more. Based on this data, we decided to re-execute our tests with two zEDC Express devices, which resulted in further elapsed time reduction due to a lack of queueing delays, with both devices at 70% utilization. We also re-executed the tests with four zEDC Express devices, with nominal improvement over two devices. Figure 2 shows the elapsed time with various compression options.

The processor cost of compression by using zEDC was very low compared to the cost of generic and tailored compression. Using zEDC compression versus tailored compression reduced our processor cost by more than 80%. Adding zEDC compression to our extended format data sets increased our overall processor cost by more than 80%. Adding zEDC compression reduced our processor cost by about two requests with the zEDC Express running 80% utilized.

A winning combination!

QSAM, and BSAM data set compression with zEDC can reduce the cost of keeping sequential data online by reducing the disk space needed for your extended format sequential data sets by up to 75% or more, without the high processor cost that was often an inhibitor to using compression. Compression with zEDC is so efficient and so effective that in some cases, batch workloads run faster with zEDC compression than with no compression or with generic or tailored compression.

With a great compression ratio, low processor usage, and potentially a reduced elapsed time for batch workloads, the QSAM and BSAM zEDC support is a winning combination.

- For more on zEDC, see “z Enterprise Data Compression Express” in z/OS Hot Topics Newsletter Issue 27, August 2013, GA32-0892-00:
- For zEDC FAQs, see:
- For more on the IBM System z Network Batch Analyzer (zBNA) tool referenced in this article, see “Peeling back batch: IBM System z Batch Network Analyzer (zBNA)” on page 61.

Disclaimer:

The results presented in this article are based on controlled measurements obtained in dedicated environments. Your results can vary based on a number of variables such as workload, data, configuration, and software levels.

Figure 1. Compression ratios with generic, tailored, and zEDC compression.

Figure 2. Elapsed time and processor time with various compression options.
Internet Protocol (IP) networks are key in z/OS, allowing systems to communicate with a variety of remote hosts located on internal or external networks. And common protocols, such as HTTP and web services, rely on IP communications as the underlying network protocol. With TCP/IP, peer applications can be configured to connect to a specific IP address representing a remote peer. The difficulty with this approach is that it requires configuration changes at all connecting hosts if the IP address of the peer changes.

On the other hand, using host name resolution provides a level of abstraction allowing applications to look up the IP address associated with a remote peer dynamically. In a host name resolution model, changes in IP addressing do not impact applications connecting the host to the remote peer. Instead, the updated IP address is obtained on subsequent host name resolution queries to Domain Name System (DNS) servers.

How does host name resolution work on z/OS?
The resolver component of z/OS Communications Server handles requests for host-name-to-IP or IP-to-hostname address resolution. The resolver also provides an application programming interface (API) for use by applications and middleware to request resource resolution requests in a variety of environments (like batch, TSO, STC, z/OS UNIX System Services, WebSphere, CICS) and a variety of programming languages (including C/C++, COBOL, Assembler, PL/I, REXX and Java).

In Figure 1, an application makes a resolver request for the IP address that is associated with host name abc.raleigh.ibm.com. The resolver receives the request (1) and determines the correct TCPIP. DATA file for the requesting application. The TCPIP.DDATA file contains a list of NSINTERADDR statements, where each statement represents a possible DNS server IP address. The resolver forwards the query to the first DNS server address in the list (2). The DNS server can either:
• Provide the IP address based on local cached information.
• Send the query to another DNS that is more knowledgeable about host names in the zone (3).

In this example, the authoritative DNS server is able to provide an answer (4), which is propagated to the z/OS resolver (5), and then sent back to the application (6).

Where’s TCPIP.DATA?
In addition to the list of DNS server IP addresses, the TCPIP.DATA file holds the configuration information. Most important is the timeout value that specifies how long the resolver must wait for a response from a DNS server before it can send the request to a different DNS server (RESOLVERTIMEOUT statement). Note that the TCPIP.DATA is not a data set name; it’s a term that describes the resolver configuration file. The name and location of the TCPIP.DATA file is flexible.

Some examples:
• Specify TCPIP.DATA on a DD statement like SYSTCP.
• Use TCPIP.DATA as an environment variable such as RESOLVER_CONFIG.
• Store TCPIP.DATA in address space unique files like “userid.TCPIP.DATA”.

The TCPIP.DATA can be allocated in zSeries File System (zFS), a sequential data set, or a PDS/PDSE member.

The resolver uses a defined search order to determine the name of the TCPIP.DATA file. This search order varies depending on the environment: z/OS or z/OS UNIX.

All this flexibility has the unfortunate downside of significant administrative complexity. For example, imagine for a moment what it’s like when a network administrator needs to change the list of NSINTERADDR statements to reflect the use of a new DNS server. Locating and updating every TCPIP.DATA file in use can be a time-consuming and an error-prone task.

Best practice for resolver configuration
To help what can become a logistical nightmare, you can use the GLOBALTCPIPDATA statement in the resolver setup file. With GLOBALTCPIPDATA, you can specify the location of a TCPIP.DATA file to be used by the entire z/OS system. It overrides the TCPIP.DATA file search order.

When you specify GLOBALTCPIPDATA, all the critical configuration options for the resolver are obtained from the specified file. Individual applications and address spaces can continue to maintain a local TCPIP.DTA file to customize other non-critical resolver options, such as enabling resolver trace (TRACE RESOLVER statement). By using the GLOBALTCPIPDATA statement, TCP/IP administrators know exactly where their configuration information is and can modify the configuration when necessary.

You can have one GLOBALTCPIPDATA statement for a single stack environment (one TCP/IP stack per z/OS system or INET environment).

GLOBALTCPIPDATA data is not recommended if you are running multiple TCP/IP stacks within a single z/OS image or CINET environment, and are required to use a unique resolver configuration for the different stacks.

Resolver Autonomics – dealing with unresponsive DNS servers
The resolver sends queries to the DNS servers configured on the NSINTERADDR statements sequentially. When a DNS server is not available, the resolver waits until the RESOLVERTIMEOUT value is exceeded before sending the request to the next DNS server in the list. This process is repeated until a DNS server provides an answer to the query within the configured time interval.

Unresponsive DNS servers can add a significant delay for applications, so it’s important to establish an optimal RESOLVERTIMEOUT value for your

Figure 2. Combined server delay for unresponsive servers
environment. In Figure 2, the default time interval is 5 seconds. The combined delay from the two unresponsive servers is 10 seconds.

Beginning with z/OS V1R13, resolver autonics automatically detects any unresponsive DNS servers and avoid using them on subsequent requests. The resolver continues to monitor the state of the DNS servers, periodically polling name servers for their availability. Then, when the resolver detects that the DNS servers are responsive, it automatically resumes sending queries to the servers.

To use resolver autonics, simply:

• Specify the AUTOQUIESCE parameter on the UNRESPONSIVETHRESHOLD statement in the resolver setup configuration file.
• Specify the GLOBALTCPIPDATA statement in the resolver setup configuration file.

Resolver caching
By default, the resolver maintains a system-wide cache of all the responses received for queries sent to DNS servers. The resolver is very efficient in responding to repeated queries for the same host name or IP address resources because the cache eliminates the need to send queries to DNS servers.

How long does the resolver maintain entries in the cache? The resolver uses the Time To Live (TTL) value provided on each query response by the DNS server. The TTL value specifies how long caching name servers or resolvers can cache a response before issuing a new query for the resource. The DNS administrator responsible for the DNS zone should specify the TTL value, setting it to a value allowing DNS updates to be reflected in the network within a reasonable amount of time. If the DNS administrator specifies TTL values that are too long, you can have periods when the resolver cache contains stale DNS information. To minimize stale cache information, use the MAXTTL parameter in the resolver setup configuration file. The MAXTTL statement defines a maximum TTL setting and overrides any larger values from the DNS server. Now, the cache information is refreshed in a timely manner.

Using hard coded host names and IP address tables
While you can create local z/OS files with hard-coded IP addresses for host names that are not defined to any DNS server, it is not advisable. Doing so creates additional administration and maintenance for your system.

Instead, use the COMMONSEARCH and GLOBALIPNODES statements in the resolver setup configuration file to simplify the rules for using local z/OS resolution files.

• The COMMONSEARCH statement forces a single search order for these files regardless of whether the requesting application is using the z/OS or z/OS UNIX environment.

• The GLOBALIPNODES statement allows a single file to be used for all locally hard-coded host names and IP addresses in both IPv4 and IPv6 formats.

Used together, these statements can significantly reduce the administration and maintenance costs of your locally defined host names and IP addresses.

References
For more details about resolver autonics, see “Houston, we have a solution: Recovering and monitoring name server failures” in z/OS Hot Topics Newsletter Issue 25, August 2011, GA22-7501-21.

For all of the resolver details, see:
• z/OS Communications Server: IP Configuration Guide
• z/OS Communications Server: IP Configuration Reference
Everyone knows that technology is advancing rapidly. As technology becomes more and more advanced, the speed at which data can move through a system increases in tandem. While this constantly-improving performance allows for greater throughput, the higher volumes of work lead inevitably to an increased risk of serialization issues, including contention, hangs, and deadlocks.

Although strict rules exist that govern the order in which enqueues (ENQs) of various resources are obtained, extreme circumstances allows such rules to be broken, which inevitably leads to serialization issues. In this article, we’ll show how these issues can occur within your Catalog environment. We also describe a new Catalog feature called contention detection, which can reduce the impact of serialization issues.

Catalogs and Virtual Storage Access Method (VSAM) Volume Data Sets (VVDSs) are very important control data sets that are read quite frequently. Global Resource Serialization (GRS) uses such resources as SYSIGGV2 and SYSZVVDS to serialize access to these data sets. You can obtain these resources using ENQs—either shared or exclusive—on these resources. An exclusive ENQ locks out any other requestors, while a shared ENQ allows for multiple resource holders as long as they are all shared.

It is common for access to some data to require more than one ENQ, much as multiple tools are needed to fix a flat tire. For example, let’s pretend you loaned your jack to a friend who got a flat. As (bad) luck would have it, you get a flat tire on the way home after helping your friend. Unfortunately, your friend forgot to take the wrench when he borrowed your jack. Now you and your friend are stranded on opposite sides of town without the right tools!

How to collect documentation and eliminate contention

So what should you do when one catalog task (your friend) has the SYSIGGV2 resource (the jack) and needs SYSZVVDS (the wrench) to complete its task, while your catalog task has the SYSZVVDS (wrench) resource, but needs SYSIGGV2 (jack) to complete your task? Fortunately, mainframes don’t imitate life, so there is a way out. But first, you must capture enough documentation to diagnose the deadlock or hang later on when the dust settles. The most important piece of documentation to obtain is a dump of the catalog address space (CAS) during the hang to find out who holds which resource. You can do this by issuing the F CATALOG,TAKE_DUMP command in the console. Most deadlocks span multiple systems, so always get a dump of each system in the sysplex as well as any other systems that might be sharing DASDs or catalogs. If you don’t take dumps during the deadlock or hang, it is nearly impossible to figure out who was holding which resource. And if you only have a dump from one system, you’ll be able to tell whether the system was indeed hung or deadlocked and maybe which other systems were involved, but you won’t be able to figure out what was occurring on the other systems to cause the hang or deadlock. Also, as the first stop in serialization analysis, it is recommended that you use the D GRS,C command to obtain periodic contention reports, to
provide contention analysis over time. If there is a hang or a deadlock in resources used by components other than Catalog, there is no need to review a dump of CAS.

After getting a dump, you can take one of the following actions:

- Identify the specific task or tasks to surgically eliminate the deadlock or hang. This action is preferred because it is less invasive, but it does require a bit more work. To do this, issue the F CATALOG,LIST command to find a catalog task in contention, then ABEND this task using the F CATALOG,ABEND() command.
- Completely restart CAS, which re-drives all in-flight catalog tasks. This action is more invasive, but simpler and might be preferable when time is of the essence.

How to detect contention
To reduce downtime and improve responsiveness, we have developed a contention detection tool in CAS to detect contention earlier. This tool is available in IBM z/OS V1R13 DFSMS and later. Detecting contention earlier can lead to reduced downtime by preventing deadlocks or hangs that create long lines of tasks waiting for a resource. The syntax for this tool is:

F CATALOG,CONTENTION(resource, wait_time,action-flags)

- Resource can be ALLOCLCK, SYSIGGV2, SYSZTIOT, or SYSZVVDS.
- Wait_time is a minute value that ranges from 5 to 9999. The default value is 10.
- Action-flags can be N (for notify) or NR (for notify and redrive). The default value is N. Redrive cancels (abends) the task in contention and tries to resubmit the request.

In z/OS V1R13 DFSMS, only SYSZTIOT is supported for resource and action-flags is not available (this is the equivalent of the default N, notify).

When a task reaches the contention detection threshold specified by wait_time, the system issues message IEC393I, which provides information about the hang. If the contention persists for more than five minutes, the message is repeated, and then repeated every 15 minutes thereafter until the contention is cleared.

Contention can be scary, but if you collect proper documentation during contention, you'll be better equipped to avoid contention in the future. And using our new contention detection tool, you can identify contention sooner!

References
- INFO APAR II14744 describes the contention detection tool and documentation collecting in further detail.
- INFO APAR III4297 outlines the recommended GRS settings for multiple sysplex configurations.
- z/OS DFSMS Managing Catalogs, SC23-6853, provides information for all of your catalog-related needs.
Record-level sharing (RLS) for VSAM data sets streamlines I/O by buffering and caching data across the sysplex. There are certain cases, however, where complete caching is not needed. For these situations, directory-only caching lets the user remove those data sets from the caching process, saving coupling facility space and increasing performance.

**What is directory-only caching?**

Directory-only caching is a new option for VSAM RLS in z/OS V2R1 that limits the amount of data that is cached. VSAM RLS uses cache structures in the coupling facility to mirror the data in its buffers. During a record management request (such as a GET or PUT), in an effort to avoid an I/O to DASD, RLS first checks the buffers. If the control interval is not buffered, RLS checks the cache. Thus, cache acts as a cross-system buffering layer.

VSAM RLS can limit how much data is cached using four caching modes as defined by the DATACLAS parameter “RLS CF Cache Value”. These values are:

- **ALL** – caches all control intervals (CIs).
- **UPDATESONLY** – caches only updated control intervals.
- **NONE** – caches only the index.
- **DIRONLY** (new in V2R1) – caches nothing.

Directory-only (DIRONLY) caching does not cache the data or index control intervals, but instead, uses the cache to register interest in a control interval. VSAM RLS uses that registration to notify other systems of updates to the control interval. When another system notes the update, it rereads the control interval from DASD to get the newest copy, which keeps the entire sysplex synchronized.

**Which data sets are good candidates?**

Directory-only caching is best for those data sets where caching adds little or no benefit to the request path. The easiest example is a data set that fits entirely in buffer and is only accessed from one system at a time. Because no other system attempts to read the data set, the cross-system capabilities of caching don’t add any benefit. Examples of such data sets are the HSM control data sets, if HSM only runs on one system, or log files that are written by only one region or on only one system.

If coupling facility space is a concern, any RLS data set can be switched to directory-only. However, because the cache is removed from the request path, directory-only data sets with cross-system activity may see an increase in I/O to DASD. Avoid converting those data sets with high cross-system update activity; for those data sets, consider using the UPDATESONLY option.

**Configuration**

To use RLS Directory-Only Caching:

- Verify that RLSMAXCFFEATURELEVEL = A (defined in IGDSMxx).
- Change the DATACLAS parameter “RLS CF Cache Value” to “D” (DIRONLY).
- Activate the SMS configuration.
- Redefine the data set or close it throughout the sysplex, open and close it NON-RLS, and reopen using RLS.

Once RLS has flushed its internal knowledge of the data, the data class is re-read at next open and the new parameter definition is used.

**How to verify**

To confirm the change, issue “D XCF,STR,STRNM=cachename” and examine the directory entries-to-data elements ratio. The number of ELEMENTS should be close to zero, and the number of ENTRIES should rise as usage increases.

```plaintext
IXC360I 12.03.06 DISPLAY XCF
STRNAME: CACHE01
STATUS: ALLOCATED
[..trimmed for brevity..]

SPACE USAGE IN-USE TOTAL % CHANGED %
ENTRIES: 63 4742 1 1 0
ELEMENTS: 2 4737 0 2 0
```

DIRONLY enablement can also be verified using the SMF42 subtype 16 fields SMF42GAJ, SMF2AGAJ, SMF42A09, and SMF2AA09. To use the subtype 16 records, monitoring must be specifically enabled using the VARY MONDS (data.set.name),ON command.
Welcome to the block party
z/OS Fixed Block Architecture services

BY STEPHEN M. KOCIK AND SCOTT COMPTON

Is using traditional networking to transfer data between z/OS and other operating systems slowing down your application data transfer? Are the transfer rates with traditional Transmission Control Protocol / Internet Protocol (TCP/IP), channel-to-channel (CTC), Systems Network Architecture (SNA), and user-defined type (UDT) your best choice for the most efficient data exchange method? Have you considered z/OS® Fixed Block Architecture (FBA) services? FBA was introduced for z/OS Version 2 Release 1 (V2R1), and is also available for z/OS V1R13 with a PTF.

With z/OS FBA, you can manage and perform I/O to and from z/OS systems and distributed (non-z/OS) systems by using the z/OS Distributed Data Backup (zDDB) multiplatform access feature on IBM System Storage DS8700 and DS8800 devices. The new application-programming interface (API) provides functions to build and run channel programs to read from or write to fixed block devices. Those same fixed block devices can be accessed from distributed systems using Small Computer System Interface (SCSI) I/O commands. In effect, the disk device with the distributed data feature becomes a data transfer device between z/OS systems and distributed systems.

The z/OS I/O subsystem communicates with the disk controller using channel programs that contain channel command words (CCWs). These disk devices are set up for extended count key data (ECKD™) architecture. ECKD includes commands that can be used to improve performance.

Not a programmer?
If you are not a programmer, FBA is still helpful. IBM Sterling Connect:Direct® is using z/OS FBA Services for large file transfers between z/OS and UNIX.

By using FBA devices as the transport for the bulk of the data, Connect:Direct can help reduce:

- CPU consumption
- Time Spent On Data Transfer.

z/OS Distributed Data Backup
z/OS Distributed Data Backup (zDDB) is a no-charge licensed feature. When you install zDDB on DS8000 devices, distributed host systems that are attached through Fibre Channel connection (FICON®) interfaces can access logical unit number (LUN) devices containing fixed block data. Typically, FBA LUN devices are connected to Linux®, Windows®, and UNIX® operating systems.

With zDDB, two views of the disk devices are presented, one for z/OS and one for the distributed system, as shown in Figure 2.

Defining FBA devices in z/OS
FBA devices are implemented as a new device and control unit type in the unit record (UR) class of devices. Hardware Configuration Definition (HCD) is used to define these new devices connected to a new control unit of type 2107-FBA with a device type of FBA and a device class of Unit Record. The 2107-FBA control unit uses FICON channels to connect to the devices.
FBA devices can be designated as “online at IPL” (the default) or can be varied online explicitly using the VARY device command. Devices must be online before being used by z/OS FBA services.

**z/OS FBA services**

z/OS FBA services are supported by the IOSFBA API. The IOSFBA API is an authorized, supervisor state service that provides a way for callers to manage (allocate and release) and perform I/O (read and write) to allocated FBA devices. To communicate with the IOSFBA API, new mapping macros, IOSDFBA and IOSDIOST, are provided. IOSDFBA defines the input and output areas for IOSFBA. IOSDIOST provides an area to obtain the status of a given read or write I/O operation.

IOSFBA must be invoked in task mode and enabled for I/O interrupts; it can be called in 31-bit or 64-bit addressing mode.

The IOSFBA API provides the following functions:
- QUERY
- ALLOCATE
- READ
- WRITE
- ERASE
- UNALLOCATE.

The general flow for reading or writing to an FBA device follows:

1. Query the devices to obtain self-describing and physical data.
2. Allocate available FBA devices using the self-describing and physical data.
3. For reading from FBA devices, identify the data location on the FBA device and the size of the data that is being read.
4. For writing to FBA devices, identify the data, the size of the data, and the location where the data can be written to on the FBA device.
5. Repeat steps 3 and 4 as needed.
6. If needed, erase any data from the FBA devices.
7. UALLOCATE the devices using the information returned from the allocate step.

**Querying FBA devices**

Before allocating an FBA device or set of devices, the physical and self-describing information of the FBA device or devices must be obtained. To obtain this information, the QUERY function is used. In addition to the physical and self-describing information, the QUERY function gives an indication if the device is allocated.

The caller must provide a list of devices to query. This list can be built using the UCBSSCAN API, to find potential FBA devices that can be used, through an initialization file, which contains a list of FBA devices to use, or through startup parameters, which specify a list of FBA devices to use.

**Allocating FBA devices**

With the physical and self-describing information obtained, the FBA device or devices can be allocated. An FBA device must be online and allocated before it is used to read, write, or erase data from the device. How you use the device indicates how to allocate it.

More on that in a moment, but first, some important notes: Serialization of the data to be read or written is left to the calling application. You can use:
- Candidate masks to isolate the FBA devices to specific LPARs
- Security controls to control access on z/OS systems
- LUN masking to isolate access on distributed systems.

**z/OS Distributed Data Backup**

The same DS8700 and DS8800 logical devices have two views:

- **Target/LUN address for FCP access** / z/OS UCB for FICON/FBA access

*Figure 2. z/OS Distributed Data Backup*
The ALLOCATE function provides four access types for allocating an FBA device:

- Specifying SINGLE limits usage of the device to a single z/OS system within the sysplex.
- Specifying READ or WRITE allows two z/OS systems within the sysplex to use the device (one as READ and the other as WRITE).
- Allocating an FBA device as ANY gives the caller control over how the device is serialized on the z/OS systems within the sysplex. Regardless of the access type, distributed systems can still share the device—that is, read from or write to the device.

As with the QUERY function, ALLOCATE requires a list of devices that are to be allocated. ALLOCATE can allocate the device dynamically (through SVC99) and provide the same information (plus a bit more) that the QUERY function provides. This information is placed in a device descriptor list, which is used as input to other IOSFBA functions.

**Reading, writing, but no ‘rithmetic**

With the FBA device or devices allocated, reading from and writing to the FBA device can begin. To read from or write to an FBA device or set of devices, a device I/O list entry for each device is created that contains, among other things:

- Device descriptor (returned from the ALLOCATE function)
- I/O status block (mapped by IOSDIOST)
- Storage buffers
- Blocks on the device (to be read from or written to).

That's right! The caller simply passes in storage areas and targets on the FBA devices; z/OS FBA Services builds the channel programs and starts the I/Os. The API allows the read and write functions to be synchronous or asynchronous.

For asynchronous reading and writing, an entry control block (ECB) is passed to IOSFBA and control is returned to the calling program after the I/O is started. The calling program must wait, at some point, for the ECB to be posted. The ECB is posted after all I/Os are completed (successfully or unsuccessfully). The status of the I/O for the device is stored in the corresponding status block (IOSDIOST).

For synchronous reading and writing, an ECB is not passed and control to the calling program is returned after all of the data has been read or written successfully or unsuccessfully.

What's cool is that the caller of the IOSFBA service can stripe the data to many FBA devices for parallelism to speed up the data transfer. For example, a program wanting to transfer 16 MB to a peer could write 4 MB to four FBA devices easily using four device descriptors and four storage buffers.

**Cleanup time**

After all of the data is read or written, it's time to clean up.

Data written to the FBA device persists when IOSFBA operations have completed. Fortunately, IOSFBA provides an ERASE function to erase a contiguous set of blocks on the disk that were referenced during read and write operations. The device descriptor list (returned by the ALLOCATE function) is provided as input. The IOSFBA API writes ‘00’x to the blocks.

With the data swept aside, the UNALLOCATE function is used to release the FBA devices.

The device contained in the device descriptor list (returned by the ALLOCATE function) is input to the UNALLOCATE function. UNALLOCATE returns the device to the state for use in the next IOSFBA invocation and frees the devices dynamically (through SVC99).

We mentioned using z/OS FBA Services for transferring data to a peer. Many more use cases can be explored if you dream up the need. Sharing an FBA device with distributed systems can open a world of possibilities.

**Find out more**

- Check out the following article in IBM Knowledge Center for more information about how FBA works:
  ibm.com/support/knowledgecenter/SSFGBN_5.2.0/com.ibm.help.cdzos.admin.doc/ZOS_How_zFBA_works.html
- Additional information is available in the IBM Redpaper IBM System Storage DS8000: z/OS Distributed Data Backup
  www.redbooks.ibm.com/abstracts/redp4701.html
We know that the z/OS Migration publication has become popular. In fact, it’s one of the most referenced publications in our z/OS library! Always working for improvement, we’ve moved the migration information into another dimension—to z/OS Management Facility (z/OSMF).

z/OSMF V2R1 has a new function called Workflow. The z/OS Migration documentation is now available as an XML workflow in z/OSMF. The migration workflow provides interactive, step-by-step instructions for migrating to z/OS V2R1, based on the z/OS V2R1 Migration publication.

The XML workflow is identical to the publication, and continues to be refreshed on the same schedule. You can, if you want, replace the publication with the workflow. In addition, because the information is identical, you needn’t feel guilty about replacing your longtime migration-information companion. The same information is still there, helping you along the way. Instead of only documenting what to do, the z/OSMF workflow can now take your input and help you interactively organize and assign your migration activities!

Using the z/OS Migration workflow in z/OSMF offers many benefits, including the ability to do these tasks:

- Assign migration tasks to other users on the system (and send them corresponding notifications).
- Skip certain migration actions as “not applicable” to your system so that those actions are not be tracked as part of your migration.
- Show how far through the overall migration you have progressed, including the total number of necessary steps and the number of completed steps.

You can also find other usability and process improvements as you become familiar to anyone who has used the previous versions of the documentation.

To access the XML migration workflow, go to the IBM z/OS home page: ibm.com/systems/z/os/zos/. Go to Tools > Downloads. A link to “z/OSMF z/OS V2R1 Migration Workflow” is at the top of the page. Migration workflows are available for migration from z/OS V1R12 and z/OS V1R13 to z/OS V2R1. Choose the one that is appropriate for you.

After you download the XML, you must upload it as a binary file to your z/OSMF V2R1 system.

Even though you cannot use the workflow to assist you with this migration (because the system must already be at z/OS V2R1), early adoption still has many benefits:
• After z/OSMF V2R1 is running, you can assign and track the z/OS V2R1 post-installation migration actions. Many of these post-installation migration actions might be appropriate for a migration to a future z/OS release, meaning that you are better positioned for the future.

• You can determine if you missed any z/OS V2.1 migration actions by using the workflow to ensure that you didn’t overlook any applicable actions.

• You can learn how to maneuver through the z/OSMF workflow now so that for later migrations, the workflow is familiar to you.

• You can provide early feedback to IBM on using the z/OSMF workflow for the z/OS V2R1 Migration. Feedback can help IBM improve deliverables for z/OS Migration in later releases.

• You can identify any wish list items that would be helpful to improve z/OS Migration within the z/OSMF workflow itself.

The migration workflow continues to include improvements made to the underlying z/OS Migration documentation, as well as enhancements made directly to the workflow itself. Our vision is that z/OSMF workflows can take advantage of programmatic capabilities to do more than provide an alternate way to provide documentation to you. For instance, wouldn’t it be great if a specific migration step could run a health check for you or examine your customization, and tell you what to change?

As you begin exploring the new migration capabilities that are available through the z/OSMF workflow, be aware that any customizations that you make to the workflow can’t be carried forward when you upload a new XML workflow. Customizations include status of tasks, skipping non-applicable steps, assigning task owners, and so on. As the migration workflows function today, you must customize each new workflow according to your requirements and preferences.

IBM continues to explore improvements to make the transition to a new workflow as seamless as possible.

To watch a brief tutorial about creating and using a migration workflow, go to YouTube to find the IBM z/OSMF V2.1 Migration Workflow Demo: youtu.be/7QBhC2yMEwM. You can send your comments and suggestions to: zosmig@us.ibm.com.

We hope you find the migration workflow to be a valuable new tool to assist with your z/OS migrations. We are looking forward to seeing you at z/OS V2R1 and beyond!

Happy migrating!
As a result of participation in the z/OS Version 2 Release 1 (V2R1) Early Support Program (ESP) and regression testing, some important tips surfaced. These findings can help ease preparation for your migration to z/OS V2R1 from V1R12.

**Start the migration to z/OS V2R1!**

1. In V2R1, by default, when SDSF attempts to activate an extended console and the default console name (the TSO/E logon user ID) is already in use, SDSF activates a new console with a different name. The new name is derived by appending a single-character suffix to the default name. SDSF tries up to 32 different characters to create a unique console name. This new behavior also applies to the extended console names that are assigned by the SET CONSOLE commands. However, extended console names that have the maximum length of 8 characters are not modified.

   **Tip:** You can restore the previous behavior, even in z/OS V2R1. Set the ISFPARMS custom property Console.EMCS.NoConMod to TRUE.

   The extended console name is logged in the hardcopy log when you issue a system command. However, that name is not likely to be the TSO/E logon user ID by default in z/OS V2R1. Specifying that name in other system command, such as CANCEL U=userid, results in an error message like “IEE324I userid NOT LOGGED ON”.

2. In z/OS V2R1, when the TYPRUN=SCAN job ended normally, SDSF reports a Max-RC of “blank” on the Held Output (H), Output Queue (O), and Status (ST) panels. In prior releases, such as z/OS V1R12 and V1R13, “CC 0000” was displayed. This change is helpful to distinguish jobs that did not run from jobs that ended with “CC 0000”.

   **Tip:** When the TYPRUN=SCAN job ends normally in z/OS V2R1, the response message $HASP890 from $DJ,LONG and $DJ,CC command continues to show “CC=(COMPLETED)”. And the job end notify message, $HASP165, again does not have the completion code information.

3. Starting with z/OS V2R1, the JES2 input phase still detects errors. Jobs are always queued to the conversion phase. The input errors are added to errors found by conversion and reported in the same way, which means a JCL ERROR.

   **Tip:** The JESYMSG data set in the job log contains both HASP error and warning messages in addition to the existing IEFC converter error messages. The HASP messages from the JES2 input phase that write to the JESYMSG can be mixed-case. In the job log, you can change the messages to uppercase by specifying the LOGMSG=FOLD option in JOBDEF initialization statement. Also, JES2 DOC APAR QA43513 mentions the mixed-case HASP messages that were introduced in z/OS V2R1.

---

**Valuable hints and tips for migrating to z/OS V2R1**

*Advice from Professor Kimura*

**BY SHIGEKI KIMURA**
4. Beginning in z/OS V2R1, IBM Health Checker for z/OS starts automatically during the system initialization and runs under SUB=JES2 when the primary subsystem is JES2.

**Tip:** To shut down JES2 successfully, issue the P HZSPROC command before the $PJES2 command to stop the Health Checker address space.

5. In z/OS V2R1 Program Directory, GI11-9848, Appendix C “Figure 90 Storage required for distribution libraries for z/OS V2.1.0” mentions that the AFOMHFS library needs 53,785 3390 tracks, which are larger than the capacity of some models.

**Tip:** This number is not correct. As is described in the PSP bucket (Upgrade ZOSV2R1, Subset ZOSGEN), the amount of storage required for the AFOMHFS library is approximately 24,000 3390 tracks.

6. In z/OS V2R1 Program Directory, GI11-9848, “8.1.7 Allocate target and distribution libraries for Wave 1 elements, Note 14” mentions that the sizes of the SCUNTBL and ACUNTBL libraries required for installing the Unicode element have increased since the previous z/OS release.

**Tip:** The overall space requirements for z/OS V2R1 have grown a bit. Much of the growth is coming from the new font element and the new z/OS Unicode Dynamic Locale support, and so on. For example, the support of the new item Dynamic Locale Service that ships in SYS1.SCUNTBL. Some of the Asian locales are large. The z/OS Unicode Services dynamic locale service dynamically builds and loads locale data into the z/OS Unicode Services environment.

The space allocation of SYS1.SCUNTBL data set in z/OS V2R1, for example, increased to roughly 41,500 tracks while in z/OS V1R13 the allocation was roughly 1,100 tracks. However, z/OS V2R1 target libraries still fit comfortably on most volumes. As always, the z/OS Program Directory and PSP bucket contain details about individual data set space requirements.

7. z/OS V2R1 removes some DFSMSrmm samples and modules from the SYS1.SAMPLIB and SYS1.LINKLIB data sets including the documentation in EDGCM01. You can find related information, such as JCL samples to generate the removed modules and documentation, on the “as-is z/OS downloads” web page. Look for the heading, z/OS DFSMSrmm Conversion Support on ibm.com/systems/z/os/zos/tools/downloads/index.html

8. In z/OS V2R1, when the IEBGENER step is completed with CC12 (for example, IEB311I CONFLICTING DCB PARAMETERS), the “PROCESSING ENDED AT EOD” message is no longer written to the SYSPRINT data set.

**Tip:** The “PROCESSING ENDED AT EOD” implies that the data was copied during the IEBGENER step, and that the end of data on the input data set was reached. However, all the data is not going to be copied during the IEBGENER step because there are DCB parameters that differ between the input and output data set. For this reason, the “PROCESSING ENDED AT EOD” message is not issued if IEBGENER did not copy all of the data.

The same change is in z/OS V1R13 with APAR OA40972 applied, which is in z/OS V2R1.

9. Beginning in z/OS V1R13, when defining the storage management subsystem (SMS) ACDS, COMMDS, or both with NOREUSE option, the attribute is automatically altered to REUSE during the SMS initialization to avoid space problems.

**Tip:** Before starting z/OS V2R1, you can change the attribute to REUSE for control data set (CDS) by using IDCAMS ALTER command in batch. This action is needed to eliminate the security messages that are issued when the CDS is protected by RACF® and if the SMS address space does not have the TRUSTED attribute. See DOC APAR OA43577 for details.

The CDS defined with NOREUSE attribute is still usable, but the recommendation is that all SMS control data sets are defined with REUSE. In z/OS V1R13, SMS detects that they are not and issues the ALTER command to change it.
10. In z/OS V2R1, when you enable the VSAM EDIT, BROWSE, and VIEW commands, the default File Manager command to be internally invoked is changed.

**Tip:** If you explicitly specify the old default commands in the ISPF Configuration table, you can remove them to make the new default commands active. The new default command is FMNINV DSE / for EDIT, FMNINV DSB / for BROWSE, and FMNINV DSV / for VIEW. See DOC APAR OA43596 for more details.

11. Beginning in z/OS V2R1, the dialog variable ZSCR is changed after invoking the BROWSE service. For example, when running the ISPF application on z/OS V2R1, the variable ZSCR resets itself to a value of “OFF” after the CLIST runs the ISPEXEC BROWSE, and then returns.

**Tip:** In z/OS V2R1, the ISPF team added the ZSCR variable to support new function to expand member count field display in member lists, which takes values “ON” or “OFF”. If any applications have a variable named ZSCR before z/OS V2R1, the application needs to be reviewed and you might have to change your dialog (exec and panel) to use another variable name. See following information on this change.

[ibm.com/support/docview.wss?uid=swg21673038](ibm.com/support/docview.wss?uid=swg21673038)

12. Differences in ISPF panel specification: Panel ISRUMVC (OPT3.4 COPY) contains “Pack Option” 1. Default and 2. Pack. Panel ISRUMC2B (OPT3.3 COPY) provides 1. Yes, 2. No, and 3. Default. The meaning of “Default” is different between the two panels. On the ISRUMVC panel, Default is equivalent to NOPACK. The problem comes when selecting a member from OPT3.4 DSLIST, and trying to copy a packed member from panel ISRUMVC with “Pack Option” set to 1. Default, the member is unpacked when copied to the other data set.

**Tip:** To avoid confusion, apply PTF OA43201 to ISPF to change the values. The ISRUMVC “Pack Option” selection changes to 1. No and 2. Yes. The panel change clarifies the options. The meaning and behavior of selection 1 and 2 is not changed.
Get started with SMC-R on z/OS V2R1 for fast and efficient network communication

BY MIKE FOX AND GUS KASSIMIS

Shared Memory Communications over remote direct memory access (RDMA) (SMC-R) is a new network connectivity technology providing fast and efficient communications across z/OS systems. SMC-R was introduced with z/OS V2R1 and is available on IBM zEnterprise EC12 (zEC12) and IBM zEnterprise BC12 (zBC12) processors that are equipped with the new 10 gigabit Ethernet (10 GbE) RDMA over Converged Ethernet (RoCE) Express feature. SMC-R uses RDMA to quickly and efficiently move data into the shared memory allocated by each host that participates in SMC-R communications. The RDMA operations are performed over the same Ethernet network fabric as the z/OS OSA Express adapters, using the RoCE protocol and the 10 GbE RoCE Express feature, which is an RDMA capable Network Interface Card (RNIC).

SMC-R leverages existing TCP/IP network connectivity by allowing TCP connections between peers to be established over OSA Express adapters as. After a connection is established, SMC-R uses the connection to negotiate and set up the out-of-band RDMA connection to transfer the data. You don’t have to change the IP addresses or host names for these connections, or your TCP/IP applications to use SMC-R. The TCP connection remains active but idle and controls the out-of-band RDMA connection.

You can enable SMC-R on z/OS V2R1 with minimal setup work, and then check out the additional configuration options to optimize your implementation. z/OS V2R1 Communications Server also provides lots of commands, displays and network management interfaces to help you control and monitor SMC-R in your network.

Setting up and operating a z/OS host for SMC-R

The following points address setting up SMC-R in your network:

• Because RDMA traffic is not routable, you must attach SMC-R peer hosts to a common IP subnet to allow a TCP connection to go directly over the OSA Express attached subnet without requiring use of a router. You must also attach the RNICs to the same layer 2 Ethernet switch fabric as the common IP subnet.

• Using virtual local area network (VLAN) is fully supported. If you have a VLAN defined for OSA Express, they are automatically inherited by the RNICs that share physical network fabrics. You can only enable SMC-R over OSA Express adapters configured in QDIO mode with a CHPID type OSD using the INTERFACE statement.

First, your hardware configuration

After you install the 10 GbE RoCE Express, you must configure it using the Hardware Configuration Definition (HCD). Because the RNIC is a Peripheral Component Interconnect Express (PCIe) feature, you assign it a PCIe function ID, or PFID. The PFID value is used to reference the RNIC in the TCP/IP profile. You must also assign a Physical Network ID (PNET ID) to both RNICs and to the OSA Express adapters that support SMC-R to tell the system which adapter ports are plugged into a common physical network fabric.
To make PFID assignments for RNICs, do the following:

- Obtain the processor list from the main HCD panel
- Select a processor, and then select the appropriate menu item to work with either the RoCE Express or OSA Express feature.
- Add the RoCE Express or OSA Express feature and define its function ID and PNET ID.
- Similarly, to add a PNET ID for an OSA Express OSD PCHID, use the HCD “Add/Modify Physical Network IDs” from the defining channel paths panels, as shown in Figure 1.

In Figure 1, notice that the PNET ID is specified on a port basis. For example, RoCE Express, entry 1 correlates to Port 1 and entry 2 correlates to Port 2. For OSA Express, entry 1 correlates to Port 0, entry 2 correlates to Port 1, and so on.

**TCP/IP basic setup**

TCP/IP set up for SMC-R is quick and easy. You must enable SMC-R and control its use on individual hosts, but there is no requirement for configuration of remote host information.

To enable SMC-R in TCP/IP, specify the new SMCR parameter on the GLOBALCONFIG configuration statement, and list the PFIDs (defined in HCD) that you want to use. Each 10GbE RoCE Express feature has two ports but only one port can be used at a time. You can optionally specify a specific port to use—the default is port 1.

For example, the following configuration statement enables SMC-R using port 1 of the RNIC with PCIe Function ID 001A and port 2 of the RNIC with PCIe Function ID 001B:

```
GLOBALCONFIG SMCR
PFID 001A
PFID 001B PORT 2
```

In this configuration, SMC-R is automatically enabled for TCP connections that use the OSA Express features where the PNET ID matches the RNIC PNET ID.

Some things that you don’t have to do:

- No need to define interfaces or TRLEs for 10 GbE RoCE Express features; TCP/IP and VTAM do that for you dynamically.
- You don’t have to activate the 10 GbE RoCE Express features; they’re activated automatically when the first SMC-R capable OSA Express feature is activated, and they remain active unless stopped by an operator.
- SMC-R does not require any changes to your TCP/IP security settings. Connection level security (such as SSL, TLS, and AT-TLS), IP filters, traffic regulation, SAF-based network resource controls (such as NETACCESS, STACKACCESS), and auditing based on IP addresses and ports all work seamlessly with SMC-R.

Note also that IPSec and multilevel security (MLS) with packet tagging are not compatible with SMC-R. Communications Server automatically opts out of SMC-R for any TCP connections that require IPSec and MLS.

This minimal configuration is all you need to be up and running with SMC-R!

**TCP/IP advanced configuration**

Although SMC-R is fully operational with a minimal amount of TCP/IP configuration, you can control the function in more detail if necessary.

By default, Communications Server uses SMC-R for TCP connections over any OSA Express interface with the same PNET ID as a 10 GbE RoCE Express feature. But you can selectively disable SMC-R on a specific OSA Express interface using the NOSMCR keyword on the INTERFACE statement—the system cannot use SMC-R for any TCP connection on an OSA Express interface defined with NOSMCR. To re-enable SMC-R, specify or default to the corresponding SMCR keyword.

To disable SMC-R for specific TCP Server applications that might not benefit from it, specify the NOSMCR keyword on the PORT or PORTRANGE statement for the server. For example, applications with very short lived connections and very small payloads might not see performance benefits from SMC-R. The overhead of setting up an RDMA relationship for a short duration can outweigh the performance gains of SMC-R.
To control the amount of fixed memory that SMC-R uses, specify the FIXEDMEMORY keyword on the GLOBALCONFIG SMCR statement. This might be useful for systems constrained for real memory. SMC-R primarily exploits 64-bit TCP/IP private fixed memory for allocating the buffers used in RDMA operations. The default FIXEDMEMORY value is 256 MB, which is a good starting point for most users. When the SMC-R function reaches the FIXEDMEMORY maximum, it does not attempt SMC-R on any additional TCP connections, until its fixed memory usage drops below this value. Existing connections are not affected and continue using SMC-R.

**Monitoring SMC-R on a z/OS host**

z/OS Communications Server provides displays and network management functions to help you monitor the usage of SMC-R on your host. Examples include:

- SMC-R information is integrated into the relevant Netstat displays, providing SMC-R information with TCP/IP information. For example:
  - Use Netstat ALL/-A for SMC-R information about TCP connections, as shown in Figure 2:
  - Use Netstat DEVLINKS/-d for information about RNIC features and their associations with OSA Express features
  - Use Netstat STATS/-S for SMC-R connection statistics and TCP statistics
- DISPLAY PCIE command shows the status of 10 GbE RoCE Express Features
- DISPLAY TCPIP,STOR shows the 64 bit fixed memory in use by SMC-R and the high water mark
- SMC-R and RNIC information is included in VTAM display commands for TRLs and tuning statistics (TNSTAT).

Communications Server also provides SMC-R information on its network management interfaces. The System Management Facilities (SMF) and network management interface (NMI) interfaces provide information about RNICs as well as SMC-R connection records when appropriate. Examples of the data provided from SMF 119 records include:

- The TCP Termination (subtype 2) record reports the SMC-R capability of the connection.
- The TCP/IP Profile (subtype 4) records report SMC-R configuration settings.
- The TCP Statistics (subtype 5) record now reports SMC-R statistics, including storage usage.
- The Interface Statistics (subtype 6) record reports SMC-R capability for OSA Express interfaces
- New RNIC Interface (subtype 44) record reports information and statistics for RNIC interfaces.

For more information about SMC-R, including FAQs, best practices, common problems, and reference documents, see:

ibm.com/software/network/commserver/SMCR/

For more guidance on using FIXEDMEMORY for tuning, see z/OS Communications Server IP Configuration Guide in IBM Knowledge Center:

ibm.com/support/knowledgecenter/SSLTBW_2.1.0

---

Figure 2. Output from DISPLAY TCPIP command with NETSTAT,ALL
Last year IBM announced the zEnterprise Business Class (zBC12) server and an update of the zEnterprise Enterprise Class (zEC12) server, including the support for two new PCI Express (PCIe) adapters: a RoCE adapter and the zEDC-Express adapter.

There are several differences between these new PCIe adapters and ordinary CHPIDs. Sure, the PCIe adapters are adapters that have a physical channel ID (PCHID) and you assign them to an LPAR. But you don’t need to assign them to a channel subsystem, and you don’t need control units or devices. These adapters follow a new paradigm for your I/O definition.

PCIe adapters provide PCIe functions for an LPAR. These functions are identified by a function ID. For zEC12 and zBC12, a function ID (FID) is a number in the range 00 - FF. The function ID must be unique for your server.

These two PCIe adapter types have different attributes and uses, as follows:

- The RoCE adapter is great for high throughput, low latency communication. It has an FID, PCHID and physical network IDs (PNET IDs).
- The zEDC-Express adapter is better for offloading data compression from the processor. It has no PNET IDs, can be virtualized, and holds up to 15 different Virtual Function (VF) IDs in the range 1–15.

Similar to reconfigurable CHPIDs, a PCIe function can only be operated by one LPAR at a time, so you must define a PCIe function to an LPAR. The PCIe function can have only one LPAR in its access list, but up to 15 LPARs in its candidate list. Because a PCIe adapter is not accessed through a channel subsystem, you can choose any LPAR of any channel subsystem.

As with all objects in HCD, a PCIe function has a description field.

Unlike CHPIDs, the PCHID value is a required input field when you add a PCIe adapter. (For a CHPID, you have the option to not specify the PCHID value when you add the adapter. You can assign that value later by using the CHPID mapping tool.)
To help you monitor PCHIDs that are used in the IODF, HCD provides a new report. This PCHID report provides information about which PCHIDs are already used by your processor, which type of adapter uses the PCHID values, and, if available, which PNET IDs are used by that adapter. For PCIe adapters, the report also contains information about the function IDs and, if applicable, the Virtual Function IDs used by that adapter.

**Defining PCIe function with HCD**

To define the PCIe functions you can use one of the following:

- The HCD ISPF dialogs
- HCM as graphical user interface
- The IOCP deck migration.

Note that the IOCP program performs less checking than the other methods, and you cannot explicitly state the PCIe type in an IOCP deck. If you want to migrate a zEDC-Express adapter, you must use extended migration and update the IOCP statement with HCD specific information.

In a stand-alone IOCP deck, PCIe-related statements for a zEDC-EXPRESS adapter look similar to the following statement:

```plaintext
FUNCTION FID=005,VF=1,PART=((LP14),(LP01)),PCHID=105
```

HCD requires type information to perform better checking.

Because there is no type information in the IOCP deck, HCD always assumes a RoCE function, which means that theVF parameter is not allowed. To migrate a zEDC function, you must define the profile option MIGRATE_EXTENDED = YES, and you must add the $HCDC$ comments to the IOCP deck to provide the PCIe type information. For example:

```plaintext
FUNCTION FID=005, VF=1, PART= ((LP14),(LP01)), PCHID=105
$HCDC$ UNIT=ZEDC-EXPRESS
$HCDC$ DESC='myDescription'
```

Specify the PCIe type in IOCP indirectly by using a different combination of parameters. For example, RoCE is defined in IOCP by using a statement similar to the following example:

```plaintext
FUNCTION FID=005, VF=1, PART= ((LP14),(LP01)), PCHID=105
```

There is no type information in the preceding statement (no VF= parameter). That means that the virtual function (VF) is not verified by IOCP regardless of whether the FID is of type RoCE for zEDC-Express.

PCIe adapters can be defined and activated with z/OS Version 2 HCD. Earlier HCD versions cannot modify IODFs that hold PCIe adapters. In addition, dynamic hardware activation is restricted (see APAR OA39234).

For more details about defining a PCIe function, see *HCD User’s Guide*, SC34-2669.

---

**IBM System z security portal**

As a best practice, IBM strongly recommends that you get access to the IBM System z security portal. Subscribe to the automatic notification process to get access to the latest service information about security and system integrity APARs for z/OS and z/VM. IBM treats vulnerability information in connection with System z as IBM Confidential. By accessing the security portal, you agree to treat such information as confidential in accordance with the terms set forth. Visit:

[ibm.com/systems/z/advantages/security/integrity_sub.html](http://ibm.com/systems/z/advantages/security/integrity_sub.html)
Accelerating Java applications with new JZOS features

BY GINA YUAN, MICHAEL WANG, AND CLARK GOODRICH

The JZOS Batch Toolkit (JZOS) component of the IBM z/OS Java Software Development Kit (SDK) product is the z/OS Java strategic direction for accessing z/OS legacy data sets and services.

JZOS, the z/OS Java strategic direction

With continuous enrichments and enhancements, JZOS offers a great vehicle for z/OS customers to modernize their applications and to facilitate application development through easy-to-use Java APIs. This article presents two enhancements: query contention support and field conversion with data access accelerator, which were added to the JZOS component in IBM z/OS Java Technology Edition, Version 7 Release 1 (z/OS Java V7R1). It also demonstrates how easily you can use these enhancements to improve productivity and performance.

Query contention to reduce resource contentions and deadlocks

The JZOS query contention feature introduced with z/OS Java V7R1 provides a new Java wrapper for the z/OS SYSEVENT QRYCONT macro that was delivered with IBM z/OS V2R1. Through this new JZOS Java wrapper, an application can query the system to determine whether there are resource contentions, as well as the types of contentions, the subsystems in contention, and the time and frequency of the contentions that have occurred. Given the information obtained from the query, the application can choose to commit resources that are causing contentions, adjust checkpoints, or take other appropriate mitigation actions to optimize system performance and resource usage.

The following example will amaze you with how easy it is to use this new z/OS capability through JZOS APIs. Just imagine how much time you can save by not having to write your own native code to access z/OS SYSEVENT QRYCONT directly!

```java
// Query resource contentions in the system
QueryContentionResult result = QueryContention.query();

// Retrieve contention information
ResourceContention[] resourceContentions = result.getResourceContentions();
int numContentions = result.getNumberOfContentions();

// Traverse the contentions list for each contention
for (int i = 0; i < resourceContentions.length; i++) {
  ResourceContention contention = resourceContentions[i];
  String subsystemType = contention.getSubsystemType();
  String subsystemName = contention.getSubsystemName();
  byte[] startTime = contention.getStartTime();
  int contentionID = contention.getContentionId();
  int count = contention.getCount();
}
```
z/OS field conversion enhanced with Data Access Accelerator (DAA)
The JZOS field conversion package, com.ibm.jzos.fields, consists of convenient methods for z/OS Java applications to map z/OS unique data type fields into Java data types. This JZOS package was further enhanced through the new Data Access Accelerator (DAA) capabilities in the com.ibm.dataaccess package added to z/OS Java V7R1. DAA is a high-performance data marshalling, conversion, and arithmetic library that handles a range of non-primitive and primitive Java types. The library operates directly on raw byte arrays, removes the requirements of intermediate processing, and accelerates data manipulations.

The JZOS CobolDatatypeFactory and AssemblerDatatypeFactory classes have been enhanced to return instances of DAA-accelerated field converter types:
- Binary fields converted to or from Java int and long types
- Packed Decimal fields converted to or from Java int, long, and BigDecimal types
- Zoned Decimal fields converted to or from Java int, long, and BigDecimal types.

With DAA exploitation in z/OS Java V7R1, field data conversions through the com.ibm.jzos.fields JZOS package were accelerated greatly, as shown in the performance measurements in Figure 1. Figure 1 shows the processor time for a sample JZOS Medicare record parsing benchmark. This enhancement is particularly beneficial in improving the interoperability between COBOL and Java applications.

See the MedicareRecord benchmark class description for more information:

Smooth ride
With JZOS, developing Java applications on z/OS is the smoothest it has ever been. The new JZOS query contention API provides a way for Java users to learn about resource contention, so they can take the appropriate mitigation actions of their choice to optimize resource usage. And by exploiting DAA capabilities in the Java virtual machine (JVM) for high-performance data marshalling and conversion, the JZOS fields package offers a significant performance boost that will further increase productivity and improve interoperability between COBOL and Java applications.

Acknowledgments
We would like to acknowledge Qi Yong Xin and Weng Shu Han from System z Performance and Brian Beegle from z/OS Java Development for their contributions to this article.

Figure 1. JZOS Medicare Record Parsing Benchmark with and without DAA

31-bit Java V7R1 with DAA versus Java V7 SR4 CPU Time improved by 2.4x
64-bit Java V7R1 with DAA versus Java V7 SR4 CPU Time improved by 1.9x
A growing number of middleware products are getting on board with z Enterprise Data Compression (zEDC). The IBM Java V7.0.0 SR7 and Java V7R1 runtime environments, IBM Encryption Facility, IBM Sterling Connect:Direct for z/OS V5.2, and IBM WebSphere MQ for z/OS V8 all provide the ability to compress data with zEDC.

### Data compression and decompression using Java

The Java runtime environment provides a set of classes under the java.util.zip package to perform data compression and decompression. These classes allow users to read, create, and update compressed/uncompressed data using the ZIP, GZIP, or deflate data compression file formats. The table below lists some of the frequently used compression API classes in the java.util.zip package:

<table>
<thead>
<tr>
<th>Class</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deflater</td>
<td>Supports general compression using the zlib compression library.</td>
</tr>
<tr>
<td>Inflater</td>
<td>Supports general decompression using the zlib compression library.</td>
</tr>
<tr>
<td>InflaterInputStream</td>
<td>Reads a stream that is compressed in the “deflate” compression format and decompresses it.</td>
</tr>
<tr>
<td>DeflaterOutputStream</td>
<td>Writes compressed data in “deflate” compression format.</td>
</tr>
<tr>
<td>GZIPInputStream</td>
<td>Reads a stream that is compressed in the GZIP format and decompresses it.</td>
</tr>
<tr>
<td>GZIPOutputStream</td>
<td>Writes compressed data in GZIP format.</td>
</tr>
<tr>
<td>ZipInputStream</td>
<td>Reads a stream that is compressed in the ZIP format and decompresses it.</td>
</tr>
<tr>
<td>ZipOutputStream</td>
<td>Writes compressed data in ZIP format.</td>
</tr>
</tbody>
</table>

Table 1. Frequently used compression API classes in the java.util.zip package

The classes listed in this table all use the zlib compression library using the Deflater and Inflater classes to compress and decompress data. The version of zlib used by IBM Java 7.0.0 SR7 and Java V7R1 on z/OS has been updated to include the changes required to use zEDC.

Java applications can potentially use zEDC for compression and decompression, provided the following criteria are met:

- zEDC Express devices are available on the system. This can be confirmed with the D PCIE z/OS console command.
- The application has READ access to the SAF resource FPZ.ACCELERATOR.COMPRESSION.
- The _HZC_COMPRESSION_METHOD environmental variable is either unset or set to hardware. Setting this environmental variable to software causes zEDC not to be used.
- The input buffer size meets the threshold specified in IQPPRMxx.

For further information, see z/OS Programming: Callable Services for High-Level Languages, SA23-1377.

### Java code example

The following sample Java program illustrates the use of zEDC. Using GZIPOutputStream, it reads data from one file and then writes compressed data to another file. Note that imports and try/catch logic have been removed for brevity.

```java
final static int BUFFERSIZE = 16 * 1024;
byte buffer[] = new byte[BUFFERSIZE];
byte outputFile[];
input = new FileInputStream(argv[0]);
output = new ByteArrayOutputStream();
gzStream = new GZIPOutputStream(output, 4096);
for(;;) {
    readBytes = input.read(buffer);
    if(readBytes < 0) {
        break;
    } else {
        gzStream.write(buffer, 0, readBytes);
    }
}
```

In this program, the input buffer for each deflate call is the 16 KB area represented by buffer, with a 4 KB output area. If the output buffer is not large enough to contain the entire output of a compressed 16 KB input buffer, the gzStream.write blocks processing until all output is processed.
Does the output buffer size really matter? Yes and no. The output buffer size does not impact the decision to use zEDC. It does, however, impact some of the software-based efficiency. Internally, the GZIPOutputStream class keeps calling the Deflate API in a loop to collect all of the output in 4 KB increments. Each call does incur some overhead, as well as the additional memory usage to buffer the additional output that exceeds the initial 4 KB output buffer.

Which buffer sizes matter...and when

The java.util.zip classes, in general, provide external means of changing the internal buffer sizes used for both input and output of compression or decompression operations. It can be non-obvious, however, which specific changes need to be made in the Java application to get the benefit of zEDC.

The input buffer size represents the size of the input buffer, which contains the data that needs to be compressed or uncompressed. This size determines if zEDC can be used for compression or decompression. If this value is greater than or equal to the threshold, zEDC is used. Otherwise, zlib software-based compression/decompression algorithms are to be used.

The output buffer size represents the size of the output buffer where output data is stored – that is, compressed data for compression or uncompressed data for decompression. The output buffer size provided to the inflate or deflate method does not impact the decision to use zEDC. If this value is smaller than the amount of data in the input buffer, the zEDC code allocates a buffer to hold the overflow output, and subsequent calls to the inflate or deflate methods consumes the output from this buffer and does not issue additional zEDC requests.

Now, let's look at the Java classes in java.util.zip package and see which buffer values are important...

When using the Inflater and Deflater classes directly, the input buffer size is nothing but the size parameter passed using the setInput method.

The GZIPInputStream, DeflaterInputStream and InflaterInputStream classes provide a constructor that allows the input buffer size for the deflate or inflate operation to be specified. The buffer passed to the read method determines the size of the output buffer.

The GZIPOutputStream and DeflaterOutputStream classes provide a constructor that allows the output buffer size for deflate and inflate operations to be specified. For these classes, the size of the buffer passed to the write method sets the input buffer size.

The ZipInputStream and ZipOutputStream classes do not provide a constructor that allows the buffer size to be manipulated. For the ZipOutputStream class, the input buffer for the write method can be large enough to qualify for zEDC to be used for compression of the data. The output buffer size is the default, which is 512 bytes. The ZipInputStream always uses 512 byte input buffers and does not qualify for zEDC.

The JarInputStream and JarOutputStream classes inherit from the ZipInputStream and ZipOutputStream classes respectively, and have the same behavior as described above.

Java Performance Results

Using 256 KB buffers, zEDC can reduce java.util.zip.Deflater elapsed time by up to 55 times, and processor time by up to 240 times, when compared to zlib software compression. The compression ratio achieved using zEDC, however, may not match the compression ratio achieved using zlib compression in software.

DISCLAIMER: Results are based on internal controlled measurements using java.util.zip.Deflater on data already in memory. Results may vary based on the application's use of java.util.zip classes and other work done by the application.

Figures 1 and 2 demonstrate GZIPOutputStream compression of three files containing different types of data. Books in text format written by authors Mark Twain, Herman Melville, and Jonathan Swift were used in the public domain books file. SVC dumps and SMF data contain varying amounts of binary and textual data. This GZIPOutputStream program used 256 KB input and output buffers. A significant reduction in the processor and elapsed time was observed in compressing all the three types of files when zEDC hardware compression was used versus zlib software compression, with:

- Up to 90% reduction in processor time
- Up to 74% reduction in elapsed time

IBM Encryption Facility

The IBM Encryption Facility (EF) is a Java middleware application that uses java.util.zip classes to compress and decompress data. The EF for z/OS V1.2 (FMID HCF7740) code has been updated to use proper buffer sizes for java.util.zip classes, to exploit zEDC when processing and generating compression OpenPGP (RFC 4880) compliant data. EF minimum Java requirements to exploit zEDC are either z/OS, Java Technology Edition, Version 7 Release 1 (Java V7.1) or z/OS, Java Technology Edition, Version 7 SR7.

1. IBM Encryption Facility having environments where compression is already in use, zEDC can provide significant reductions of:
   a. Up to 70% in processor time, and
   b. Up to 60% in elapsed time.

2. For IBM Encryption Facility users not already using compression, compression with zEDC can provide reductions of:
   a. Up to 46% in processor time, and
   b. Up to 44% in elapsed time.
DISCLAIMER: Results based on internal controlled measurements using IBM Encryption Facility for files containing public domain books. Results may vary by customer based on individual workload, data, configuration and software levels.

Figure 3 shows the total processor time of EF performing no compression, zlib compression, and zEDC compression plus software encryption for public domain books in text format. The encrypted output file size is also shown. zEDC makes it possible to compress the file, consuming very little processor time prior to encryption. After the file is compressed, the processor time to encrypt the compressed file is further improved, since there are fewer bytes to encrypt. zEDC hardware compression consumed the lowest processor time to produce an encrypted file.

WebSphere MQ for z/OS V8
WebSphere MQ for z/OS V8 exploits zEDC for channel message compression. Currently there are several options for channel message compression, which are specified using the COMPMSG attribute, and two of these, ZLIBHIGH and ZLIBFAST, provide DEFLATE compliant compression. The COMPMSG(ZLIBFAST) now uses zEDC for compression and decompression, when available.

For more information about WebSphere MQ, see [ibm.com](http://www.ibm.com/software/products/en/wmq)

IBM Sterling Connect:Direct for z/OS V5.2
The managed file transfer product IBM Sterling Connect:Direct for z/OS now automatically leverages zEDC for file compression and decompression as files are transferred when the extended compression option is specified. The support is fully compatible with zlib compression used in IBM Sterling Connect:Direct today, so there are no changes required at end points. The only software requirement for Connect:Direct for z/OS is version V5R2M00, which provides the ability to exploit zEDC. Using zEDC for compression over software reduces the elapsed time for file transfers with a dramatic reduction in processor usage.

For more information about IBM Sterling Connect: Direct for z/OS, see [ibm.com](http://www.ibm.com/software/products/en/connect-direct)

But wait, there’s more!
Middleware adoption of zEDC provides significant additional value on top of the z/OS operating system exploitation of zEDC. The standard zlib compression algorithm provided with zEDC is especially useful because it makes it possible to decompress data on any platform using standard services. Plus, the savings in elapsed time and processor time with zEDC open up new opportunities to perform data compression in applications where previously compression was not realistic.

- For more on zEDC, see the “z Enterprise Data Compression Express” article in Hot Topics Issue 27: [ibm.com](http://www.ibm.com/systems/z/os/zos/library/hot-topics/hot-topics.html)
- For zEDC FAQs, see [ibm.com](http://www.ibm.com/support/docview.ws?uid=ts1fq131484&aid=1)
- For more information about Java on z/OS, see [ibm.com](http://www.ibm.com/systems/z/os/zos/tools/java/)

We would like to acknowledge Steven Hart (z/OS Cryptography) and Qi Yong Xin (System z Performance) for their contributions to this article.
When mainframes have a large number of daily transactions, analyzing data in real time can be a difficult task, especially if you need to extract the system log during a long-running period of time and analyze it. The problem is that even though the system log is updated very quickly in z/OS, the entire system log cannot be generated due to time constraints. To avoid this obstacle, use REXX scripts to update and extract system log in real time. The extracted information is then written into an assigned data set, at which point the data analysis can begin.

The problem and the solution
There are two logs, the operations log (OPERLOG) and the system log (SYSLOG). The operations log is a log stream that uses the system logger to record write-to-operator system messages from each system in a sysplex. The system log is a single-system message log that resides on a JES spool.

To determine whether the hardcopy medium is a system log or an operations log, issue “DISPLAY CONSOLES,HARDCOPY.” If the returned messages contain "LOG=(SYSLOG)", we know that the operations log is not available. To get a sysplex-wide log of system messages, you can use ADDRESS SDSF “ISFLOG READ TYPE(OPERLOG)” in REXX, but you must set up the operations log first. However, ADDRESS SDSF “ISFLOG READ TYPE(SYSLOG)” only gets a single-system message log in REXX.

With the SDSF feature of z/OS, you can choose one of those two REXX functions to read and save the system log into the STEM variable “isfline.”. But ADDRESS SDSF “ISFLOG READ TYPE(SYSLOG)” only extracts part of the most recent system log before it is run. It does not obtain earlier or later system logs.

Loops are an obvious answer to this problem, but will introduce another problem. Part of the system log that is generated in the previous loop will inevitably occur at the beginning of the next loop. But if the system log is not updated, almost the identical system log is repeatedly read, extracted, and saved from the system. Saving the duplicated information wastes disk space, and the duplicated information interferes with the subsequent analysis of the system log.

To avoid the repeated extraction, use the time stamp in each line of the system log.

How it works
First, allocate a source PS data set to store the keywords and the related parameters. The related parameters specify the number of lines above and below the line that contains specific keywords. The keywords and parameters must be separated by a separator that is not used in the system log. In the following procedure, <!-----> is used as a separator.
1) Read the source PS data set
Read the keywords from the source PS data set and save them into the STEM variable “keyword.” for later use.

```
KEYINPUT = YOURID.REXX.DATASET /*the source PS data set*/
KEYINPUT = STRIP(KEYINPUT) /*remove the blanks around*/
/*------------------------------------------------------*/
/* Read the source PS data set to get the keywords and */
/* related parameters.   */
/*------------------------------------------------------*/
CALL BPXWDYN "ALLOC DA('KEYINPUT') FI(INPUT) SHR REUSE"
ADDRESS MVS "EXECIO * DISKR INPUT (STEM keyword. FINIS"
CALL BPXWDYN "FREE FI(INPUT)"
```

2) Initialize the Public Time Stamp
Initialize the Public Time Stamp for later use.

```
zDatebp = date(s) /*get the date when the script begins*/
/*---------------------------------------------------------*/
/* Initialize the Public Time Stamp. */
/*---------------------------------------------------------*/
zTimebp1=0
zTimebp2=0
zTimebp3=0
zTimebp4=0
```

3) Verify the existence of the object data set
Determine whether the object PS data set exists. If there is one, set it to the modification mode for subsequent addition. Otherwise, allocate a new one and set it to the modification mode.

```
CALL BPXWDYN "Rtext = SYSDSN('YOURID.REXX.OUTPUT')"
If Rtext <> 'OK' then /* if it doesn't exist. */
do
/*---------------------------------------------------------------*/
/* Allocate a new object PS data set and set it to the */
/* MOD(modification) mode.          */
/*---------------------------------------------------------------*/
CALL BPXWDYN "ALLOC FI(OUTDD) DA('YOURID.REXX.OUTPUT')"
NEW CATALOG RECFM(V,B) SPACE(5000,5000) BLKSIZE(27998)
REUSE" /* allocate a new object PS data set */
CALL BPXWDYN "FREE FI(OUTDD)"
CALL BPXWDYN "ALLOC FI(OUTDD) DA('YOURID.REXX.OUTPUT')"
MOD REUSE" /* set it to MOD mode */
end
else /* if it exists*/
CALL BPXWDYN "ALLOC FI(OUTDD) DA('YOURID.REXX.OUTPUT')"
MOD REUSE" /* Set it to MOD mode */
```

4) Loop to do the separation work
Use an endless loop to fetch each line of the source PS data set from keyword 0. Extract the keywords and the related parameters for later use.

```
do forever

do k = 1 To keyword.0 /*Loop to process each 'keyword.'*/
  ck_jn = strip(keyword.k) /*get the line with keywords and parameters*/
  /* Find the position of the parameters. */
  position = POS('<!---!',ck_jn) + 7
  if position = "7" then /*if there is no '<!---->'*/
    x = y = 0 /* No additional lines to be extracted */
  else /*if there is '<!---->!'*/
    /* Extract the two numbers in the source PS data set. */
    do
      POStar = POS('<!---!',ck_jn) - 1
      target = strip(substr(ck_jn,1,POStar))
      key = strip(substr(ck_jn,position))
      x = subword(key,1,1)
      y = subword(key,2,1)
    end
  end
end
```

5) Get the time stamp of the line containing specific keywords
Use a loop to look through the system log for the time stamp of the line containing specific keywords.

```
do k = 1 To keyword.0
  do
    rc = isfcalls('ON') /* open the SDSF host environment */
    ADDRESS SDSF "ISFLOG READ TYPE(SYSLOG)" /* read the SYSLOG */
    do ix = 1 to isfline.0 /* loop through the SYSLOG */
      strPos = pos(target,isfline.ix)/* locate the keywords */
      if strPos <> 0 then /* find the keywords */
        do
          cnt = 0 /* initialize the variable */
          sum = 0 /* initialize the variable */
          out.0 = 0 /* initialize the variable */
          zString = strip(isfline.ix) /* get the line containing specific keywords */
          /* Find the position of the time separators, here these */
          /* are "!" and "." */
        end
    end
end
```

6) Update the latest required lines

Compare the time stamp of the current line containing specific keywords with the Public Time Stamp to decide whether the current line and its around lines are to be written into the object PS data set.

Downloading the z/OS Collection Kit

To download the z/OS Collection Kit, follow these steps:

1. Go to the IBM Publications Center:
   ibm.com/shop/publications/order
2. Select your country, then click “Search for publications.”
3. Type SK4T-4949 in the “Publication number” field and press Enter. SK4T-4949 is the order number for the electronic-only z/OS Collection Kit and matches what was previously available in physical media for base and features.
4. Use a modern zip app such as 7-zip to unzip the file. See the appropriate download pages at:
   ibm.com/systems/z/os/zos/library/bkserv/index.html

Every “shelf” has its own order number, so you can pick and choose to download only the products you need from the IBM Publications Center. For example, z/OS HCM, SK2T-1377, or z/OS Messages and Codes, SK5T-9251.

Otherwise, to download the entire collection kit:

z/OS V2R1    SK4T-4949
z/OS V1R13   SK3T-4271
else if zTime1 = zTimebp1 &
  zTime2 = zTimebp2 &
  zTime3 = zTimebp3 &
  zTime4 < zTimebp4 &
  zDate = zDatebp then
  nop
else
  do
  /*---------------------------------------------*/
  /* Judge whether 'x' exceeds the scope of the SYSLOG. */
  /*---------------------------------------------*/
  if ix <= x then
    do
      sum = sum + (x - ix) + 1
      iz = 1 /*read from the first line*/
    end
    else
      do
        iz = ix - x /*read from the requisite line*/
      end
    /*---------------------------------------------*/
    /* Get the requisite lines around the line containing */
    /* keywords. */
    /*---------------------------------------------*/
    out.0 = x + y + 1 - sum
    do iy = iz to (iz + x + y - sum)
      cnt = cnt + 1
      out.cnt = isfline.iy
    /*---------------------------------------------*/
    /* Judge whether 'iy' exceeds the scope of the SYSLOG. */
    /*---------------------------------------------*/
    if iy >= isfline.0 then
      do
        out.0 = cnt
      /*---------------------------------------------*/
      /* If 'iy' exceeds the scope, exit the current loop. */
      /*---------------------------------------------*/
      LEAVE iy
    end
  end
  /*---------------------------------------------*/
  /*Write the requisite lines into the object PS data set.*/
  /*---------------------------------------------*/
  ADDRESS MVS 'EXECIO' out.0 'DISKW OUTDD (STEM out.
  FINIS'/*---------------------------------------------*/
  /*Update the Public Time Stamp. */
  /*---------------------------------------------*/
  zTimebp3 = zTime3
  zTimebp4 = zTime4
  end
  end
  end
  /*---------------------------------------------*/
  /* At the end of each loop, update the Public Time Stamp*/
  /* on zTimebp4 to make sure what to be extracted is the */
  /* latest with no repetition. */
  /*---------------------------------------------*/
  zTimebp4 = zTimebp4 + 1
end
CALL BPXWDYN "FREE FI(OUTDD)"
exit 0

Summary
Using REXX to update and extract the system log in z/OS in real
time can benefit industries that depend on timely data analysis,
such as banks and stock trading centers.
A n increasing number of z/OS shops find themselves in the cloud business, offering a private cloud or acting as the service provider for a broader audience. Among the benefits of the cloud model is the ability to scale capacity by pooling processing resources. Sometimes, workload fluctuations level out, which results in a more consistent overall capacity demand. But other times, cloud service providers experience multiple workloads that increase concurrently, resulting in higher and more dramatic capacity spikes. And transactions originating from mobile devices can aggravate this unpredictable spikiness.

As a service provider, you must react quickly to these capacity bottlenecks, while keeping costs under control. You've got to meet your service level requirements, such as response time goals and transaction rates. Providing timely capacity increases can help keep work from backing up and can be the most cost-effective way to handle periods of increased workload demand. In addition, the ability to remove capacity that is no longer needed can help control overall costs. Capacity Provisioning Manager (CPM) can help with both of these!

Conventional capacity planning and its limits
On/Off Capacity on Demand (OOCoD) record: As part of the capacity planning process, you can determine the capacity required for both the individual LPARs and the entire processor using the processor model. You can then use OOCoD record to increase hardware processor capacity when needed.

Software pricing: When setting and controlling LPAR capacity, it helps to consider the specific way you price your software products. Frequently, z/OS shops use billing models like the Variable Workload License Charge (VWLC) or Sub-Capacity Advanced Workload License Charges (AWLC), which are usage-based pricing models. z/OS measures the utilization and builds a four-hour rolling average (4HRA) as the base for determining the software charge.

To control the 4HRA, you can define two types of limits and use them concurrently:

- Defined capacity limit for an LPAR running one z/OS system.
- Group capacity limit for a group of LPARs, called group capacity limit.

z/OS then checks the current 4HRA against the defined limits, and if the current 4HRA value exceeds the limit, it caps the LPAR so that the system consumes less capacity.

However, if your service levels are at risk, you might want to increase your capacity limits. For particularly critical workloads, for example, you can increase the defined capacity limit even before capping begins. Monitoring products, such as z/OS Resource Measurement Facility (RMF), provide a time-to-cap metric, showing the projected time until capping will begin. As this duration diminishes, you can consider increasing the limit.
For conventional workloads, this kind of dynamic by-hand analysis often works perfectly well. But with cloud workloads, you might need a much faster reaction time and a predefined policy that lets the system automatically balance capacity—as needed. Luckily, you can rely on z/OS to provide this function, using z/OS CPM.

**CPM provides the answer to your flexibility needs**

CPM is a base element of z/OS that you can use to manage capacity for general-purpose processors as well as zSeries Application Assist Processor (zAAP)

zAAP and zSeries Integrated Information Processor (zIIP) processors. As of z/OS V2R1, CPM can also manage defined capacity and group capacity. Using a CPM policy, you can define the actions that you want CPM to take when workload performance suffers because of capacity bottlenecks.

You can run CPM in one of four modes, each providing different degrees of automation:

- In manual mode, you can control server capacities using CPM commands. The system does not use the CPM policy.
- In analysis mode, CPM processes the policy and informs the operator when action is needed using the criteria specified in the policy. Then it’s up to the operator to take action to increase capacity.
- In confirmation mode, CPM processes the policy and proposes possible actions to increase capacity to the operator. If the operator accepts a suggested capacity change, CPM implements it.
- In autonomic mode, CPM processes the policy and implements capacity changes immediately, as needed.

**Defining a CPM policy**

Defining a policy is easy using z/OSMF from your Web browser. Prior to z/OS V2R1, you could define which workloads might trigger capacity changes and when such changes are allowed. But only physical OOCd capacity can be activated. In the policy, you can also specify the following in your CPM policy:

- The Workload Management (WLM) service class periods that identify work critical enough to warrant additional capacity when capacity bottlenecks pop up.
- The time periods when additional capacity might be needed.
- How much additional capacity to add.

With z/OS V2R1, you can also include the following additional facets for maximum defined capacity scope in your CPM policy:

- The z/OS systems for which CPM should manage defined capacity. CPM automatically detects the CPC and LPAR where the specified system runs.
- The maximum number of MSUs by which CPM can increase the originally defined capacity limit.
- The increments for step-wise capacity limit adjustments. You can specify values for initial increments and for subsequent increments.

Figure 1 shows an example of a Defined Capacity scope in z/OSMF.

Note that both physical (OOCd) and defined capacity can be managed at the same time. You can also define capacity scope for LPAR group capacity.
Using the CPM policy, CPM monitors the specified systems and workloads. If, and only if, insufficient workload performance is caused by a capacity bottleneck, CPM takes an appropriate action in the workload. For example, a workload performance can suffer because of capacity capping. In this situation, CPM evaluates whether to relieve the defined capacity limit.

However, depending on your policy definition, CPM can also anticipate a capacity bottleneck due to capping and raise the limit before workload performance suffers. When the time-to-cap metric falls below a threshold of your choice (called lead time), CPM can proactively increase the defined capacity limit in accordance with your settings as shown in Figure 2.

CPM performs only as many capacity increments as needed to allow all workloads to perform as defined in the policy and always stay within the specified maximum limits. As soon as the workload permits, CPM decrements the limit again to the original value.

A timely decrease of defined capacity limit when no longer needed helps prevent increased costs. You can also simply define a time limit that you want additional defined capacity to last.

**Manual management options**

While managing defined capacity according to the policy, CPM detects capacity changes initiated outside of CPM, including manual CPM commands. CPM won’t interfere with capacity added manually, so that you can manually override your policy when, for example, you have to add capacity more aggressively than defined in your policy.

CPM commands can also help you with Linux for z or cloud workloads. With z/OS V2R1, you can temporarily increase your Linux capacity using commands for OCoD Integrated Facility for Linux (IFL) capacity from the z/OS console.

With z/OS V2R1, CPM added support so that you can manage defined capacity and group capacity. With this enhancement, you can manage capacity in varied environments.

For more information about capacity provisioning and its management capabilities for defined capacity and LPAR group capacity, see:

ibm.com/systems/z/os/zos/features/cpm/
Q: So tell us, just what is Suite B?

A: Suite B is a set of cryptographic algorithms defined by the National Security Agency (NSA) that can be used as the interoperable cryptographic base for unclassified information and for most classified information.

Starting with IBM z/OS V2R1, System SSL supports Suite B algorithms within TLS 1.2 – as defined in RFC 5430, Suite B Profile for Transport Layer Security (TLS).

System SSL allows TLS client and server applications to specify a profile that is compliant with Suite B cryptography. This profile restricts the cryptographic algorithms used for the session to the set of algorithms that Suite B cryptography supports. Communication is possible between TLS clients that require Suite B cryptography and TLS servers that do not explicitly support Suite B cryptography, and vice versa, as long as the non-Suite-B entity supports the Suite-B-compliant cryptographic algorithms.

Suite B cryptography does not define cryptographic algorithms. Instead, it specifies which cryptographic algorithms can be used in a Suite-B-compliant TLS 1.2 session.

Q: Which cipher suites does Suite B support?

A: Table 1 describes the cipher suites that are allowed for the 128-bit and 192-bit Suite B profiles.

Here’s a different depiction of how these cipher suites are defined:

- C023 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- C024 TLS_ECDHE_ECDSA_WITH_AES_192_CBC_SHA384
- C02B TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- C02C TLS_ECDHE_ECDSA_WITH_AES_192_GCM_SHA384

<table>
<thead>
<tr>
<th>Cipher suite</th>
<th>128-bit security level</th>
<th>192-bit security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>C023</td>
<td>Encryption: 128-bit AES Advanced Encryption Standard (AES) Message authentication: Secure Hash Algorithm 256 (SHA-256) Ephemeral Elliptic Curve Diffie Hellman (ECDH) key exchange signed with an Elliptic Curve Digital Signature Algorithm (ECDSA) certificate</td>
<td>N/A</td>
</tr>
<tr>
<td>C02B</td>
<td>Encryption: 128-bit AES in Galois/Counter Mode (GCM) Message authentication: SHA-256 Ephemeral ECDH key exchange signed with an ECDSA certificate</td>
<td>N/A</td>
</tr>
<tr>
<td>C024</td>
<td>N/A</td>
<td>Encryption: 256-bit AES in GCM Message authentication: SHA-384 Ephemeral ECDH key exchange signed with an ECDSA certificate</td>
</tr>
<tr>
<td>C02C</td>
<td>N/A</td>
<td>Encryption: 256-bit AES in GCM Message authentication: SHA-384 Ephemeral ECDH key exchange signed with an ECDSA certificate</td>
</tr>
</tbody>
</table>

Table 1. Suite B cipher suites (Exhibit A)
Whenever a Suite-B-compliant client and a Suite-B-compliant server establish a TLS 1.2 session, only Suite B algorithms are employed. For this session to be fully Suite-B-compliant, the TLS 1.2 protocol must be used to establish a connection. Any other protocols that are configured for a connection are ignored and only TLS 1.2 Suite B cryptographic algorithms are used to establish the connection.

Q: Which types of certificates are required to use Suite B?

A: Suite B requires the key establishment and authentication algorithms used in TLS 1.2 sessions to be based on Elliptic Curve Cryptography (ECC) and the encryption algorithm to be Advanced Encryption Standard (AES).

Two security levels are defined in the Suite B profile:
128-bit security level, which corresponds to an elliptic curve size of 256 bits and AES-128
192-bit security level, which corresponds to an elliptic curve size of 384 bits and AES-256.

The Suite B standard specifies the elliptic curves that are allowed in a TLS connection. Table 2 shows the curves that are allowed for the 128-bit and 192-bit Suite B profiles.

<table>
<thead>
<tr>
<th>Named curve</th>
<th>128-bit security level</th>
<th>192-bit security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>secp256r1 – {1.2.840.10045.3.1.7}</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>secp384r1 – {1.3.132.0.34}</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2. Suite B elliptic curves (Exhibit B)

Server and client certificates that are used to establish a Suite-B-compliant connection must be signed with an ECDSA certificate.

- For certificates that are used at the 128-bit security level, the subject public key must use the secp256r1 curve and be signed with the secp256r1 curve or the secp384r1 curve.
- For certificates that are used at the 192-bit security level, the subject public key must use the secp384r1 curve and be signed with the secp384r1 curve.

Q: How is Suite B deployed?

A: Enabling client and server connections to be Suite-B-compliant is as easy as setting the System SSL environment variable GSK_SUITE_B_PROFILE. Table 3 describes the valid settings for this variable.

<table>
<thead>
<tr>
<th>GSK_SUITE_B_PROFILE setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSK_SUITE_B_PROFILE_OFF</td>
<td>Specifies that Suite-B-compliant profiles are not in use for TLS sessions. This is the default.</td>
</tr>
<tr>
<td>GSK_SUITE_B_PROFILE_128</td>
<td>Specifies that only ciphers defined within a 128-bit Suite-B-compliant profile can be used for a TLS session.</td>
</tr>
<tr>
<td>GSK_SUITE_B_PROFILE_192</td>
<td>Specifies that only ciphers defined within a 192-bit Suite-B-compliant profile can be used for a TLS session.</td>
</tr>
<tr>
<td>GSK_SUITE_B_PROFILE_ALL</td>
<td>Specifies that ciphers defined within the 128-bit and 192-bit Suite-B-compliant profiles can be used for a TLS session.</td>
</tr>
</tbody>
</table>

Table 3. GSK_SUITE_B_PROFILE settings (Exhibit C)

Q: What if the client application has set GSK_SUITE_B_PROFILE to a supported value that is not GSK_SUITE_B_PROFILE_OFF, but the server application was allowed to use the default GSK_SUITE_B_PROFILE_OFF?

A: Even though the server can allow a connection with a choice from many different cryptographic algorithms, the client will only allow a connection using a Suite-B-compliant algorithm up to the highest security bit setting defined. Hence, the connection is a Suite-B-compliant session. That is to say, if the client has set GSK_SUITE_B_PROFILE to GSK_SUITE_B_PROFILE_128, only cipher C023 or C02B can be used for the connection.
Q: And if the setup is the other way around?
A: In that case, the server application will allow a connection using a cipher that is defined within its GSK_SUITE_B_PROFILE. If the server has set GSK_SUITE_B_PROFILE to GSK_SUITE_B_PROFILE_192, only cipher C024 or C02C can be used for the connection.

Q: What if the client application has set GSK_SUITE_B_PROFILE to GSK_SUITE_B_PROFILE_128 and the server application has set GSK_SUITE_B_PROFILE to GSK_SUITE_BPROFILE_128?
A: Those settings are mutually exclusive and no connection can be made. If this scenario were changed so that the client or server application’s GSK_SUITE_B_PROFILE is set to GSK_SUITE_B_PROFILE_ALL, the connection will use a Suite-B-compliant cipher within the set that is allowed by the other application.

Looking at Table 4, one can see how client and server profile settings affect which Suite B cryptographic algorithms are allowed for that connection.

Q: What if the client’s GSK_PROTTOCOL_TLSV1_2 variable is set to GSK_PROTTOCOL_TLSV1_2_OFF and the server’s GSK_SUITE_B_PROFILE variable is set to GSK_SUITE_B_PROFILE_ALL?
A: To be a Suite-B-compliant connection, the TLS 1.2 protocol must be enabled. This scenario cannot create a connection between the client and server applications. TLS 1.2 is required to be Suite-B-compliant by the server. However, TLS 1.2 is not configured to be used by the client application.

Q: Can the Suite B cryptographic algorithms be defined for a connection by using the GSK_V3_CIPHER_SPECS_EXPANDED environment variable?
A: Yes, this would work. It is simply easier and clearer to use the System SSL GSK_SUITE_B pre-defined environment variables.

Q: Finally, what is needed to take advantage of the Suite-B-compliant cryptographic algorithms?
A: Suite-B-compliant cryptographic functions require the support of ECC and GCM block cipher mode cipher suites from either of the following:

- Integrated Cryptographic Service Facility (ICSF) PKCS#11 callable services
- CP Assist (CPACF).

System SSL uses ICSF PKCS#11 Web Deliverable #9 callable services to perform required encryption and key generation functions. ICSF determines whether to use the hardware functions provided by CPACF or its own internal software implementations.

Q: Oh, and where can we find more information?
A: Check out Cryptographic Services System Secure Sockets Layer Programming, SC24-5901. The truth is out there.
The beast release has arrived! Cryptographic Support for z/OS V1R13 and V2R1 (also known as HCR77A1, or web deliverable #13) has turned out to be one of the most impressive releases to date, packed with new enhancements to satisfy recent client requirements. Along with better performance, security improvements, and serviceability enhancements, the release introduces:

- New key types for the financial sector
- Key reference tracking
- AES PIN system
- RSA-MK set from TKE
- EMV™ support
- PKCS #11 currency updates

Let’s take a closer look at some of the beast-release enhancements.

**Expanded EMV support**

The Diversified Key Generate callable service generates a key based on a key-generating key, processing method, and supplied parameters. The control vector of the key-generating key determines the type of target key that is generated. ICSF now supports the TDES-CBC diversification process to align with the EMV standards, which were created by the member companies of EMVco to facilitate secure smart card transactions worldwide.

**Initial Pin Encryption Key (IPEK)**

The Unique Key Derive callable service performs the key derivation process as defined in ANSI X9.24 Part 1. The process derives keys from two values: the base derivation key and the derivation data. The Initial Pin Encryption Key (IPEK) is derived from the base derivation key and the initial derivation data. Specify the K3IPEK rule array keyword to return the IPEK.

**DESUSECV support**

DESUSECV is a type of variable-length symmetric key token where the DES key and its control vector are wrapped using the AES key wrapping method. This method ensures that key management and key usage fields are maintained from export to import. The following callable services now support the new AESKWCV formatting method with the resultant key token of type DESUSECV:

- Symmetric Key Export
- Symmetric Key Import2
- Key Test2

**Fixed-length payloads versus variable-length payloads**

Fixed-length payloads for AES keys obscure the length of the encrypted key in the payload section. All new AES key types have the fixed-length payload format, while existing AES key types (CIPHER, IMPORTER and EXPORTER) default to use the variable-length payloads, unless keywords indicate the use of the fixed-length payloads. This change ensures compatibility with older releases of ICSF and hardware where fixed-length payloads are not supported. Fixed-length payload support requires an IBM zEnterprise EC12 with a Crypto Express3 Feature (CEX3C) or Crypto Express4 Feature (CEX4C) and the September 2013 or later Licensed Internal Code (LIC).

**UDX simplification and reduction**

Customers can create extensions to the base services of ICSF and IBM Common Cryptographic Architecture (CCA). An extension is called a UDX (user-defined extension). The following extensions are merged into the CCA base and are now available to all customers through ICSF callable services:

- Authentication Parameter Generate: generates an authentication parameter (AP) and optionally returns it encrypted under a supplied encrypting key.
- Recover PIN from Offset: calculates the encrypted customer-entered PIN from a PIN-generating key, account information, and an IBM-PIN0 Offset, returning the PIN properly formatted and encrypted under a PIN encryption key.
• Symmetric Key export with Data: Exports a symmetric key, along with some application-supplied data, encrypted using an RSA key.

**PKT UDX support**
Beginning with V1R12 (HCR7790), additional extensions are merged into the CSNDPKT callable service by adding support for three new output formats: EMV CRT, EMV DDA, and EMV DDAE.

**Remote Key Export (RKX) enhanced key wrapping**
The Remote Key export callable service uses the trusted block to generate or export Data Encryption Standard (DES) keys for local use and for distribution to an ATM or other remote device. RKX uses a special structure to hold encrypted symmetric keys in a way that binds them to the trusted block and allows sequences of RKX calls to be bound together as if they were an atomic operation. Rule array keywords can be specified to indicate whether to wrap an output DES CCA key token using the default wrapping mode, enhanced wrapping mode (WRAP-ENH), or original ECB wrapping mode (WRAP-ECB). The DEFAULTWRAP parameter specifies the default key wrapping for symmetric keys. Application programs can override this default method using the WRAP-ENH (enhanced key wrapping method, which is compliant with the ANSI X9.24 standard) or WRAP-ECB (original key wrapping method, which uses ECB wrapping for DES key tokens and CBC wrapping for AES key tokens).

**Enterprise PKCS #11 Phase 2**
Enterprise PKCS #11 (EP11) Phase 2 delivers stronger Cryptographic Support, improved performance, and more algorithms that are designed to meet FIPS 140-2 Level 1 criteria. This update includes support for secure Diffie-Hellman, elliptic curve Diffie-Hellman, secure RSA-PSS, clear key RSA-PSS, and secure key Brainpool EC. Other enhancements include the ability to change compliance mode for keys, use of clear key Brainpool EC curves in FIPS mode, and support to wrap/unwrap symmetric keys with symmetric keys.

**RSA Master Key set from TKE**
ICSF now allows the RSA Master Key to be set from the Trusted Key Entry (TKE) workstation to aid with disaster recovery.

**AES MAC enhancement**
To align more closely with Request for Comments (RFC) 3566 and 4434, the Symmetric Mac Generate and Symmetric Mac Verify services now allow zero-length text for XCBC-MAC on the LAST call of a multi-part operation. Additionally, both services are updated to support a new key rule, KEY-DRV, which specifies that the key parameter contains up to 256 bits of key material to derive a 128-bit AES key for the XCBC-MAC operation.

**SAF ACEE selection**
SAF ACEE Selection is a new callable service for authorized callers (either system key or supervisor state) that enables ICSF to use the caller’s authority for SAF checking of resources in the CSFKEYS, CSFSERV, XCSFKEY, and CRYPTOZ classes. This new callable service takes an ENVR object data structure as input for authorization checks for the requested service.

**One Way Hash (OWH) and Random Number Generation (RNG) access**
CSFSERV SAF checks for the One Way Hash (OWH) and Random Number Generation (RNG) ICSF services, which contribute to significant CPU consumption.

The following two new resources were added to the XFACILIT SAF resource class for disabling OWH and RNG SAF checking:

• CSF.CSFSERV.AUTH.CSFOWH.DISABLE
• CSF.CSFSERV.AUTH.CSFRNG.DISABLE

**Dynamic Special Secure Mode (SSM)**
A new resource, CSF.SSM.ENABLE, has been added to the XFACILIT SAF resource class for dynamically enabling SSM. Previously, the only way to configure SSM was in the ICSF options data set with an ICSF restart. This change eliminates the required ICSF restart to change the setting.

**AP configuration simplification**
Cryptographic Coprocessors must be configured online by the Support Element (SE), and then activated on the ICSF Cryptographic Coprocessor Management Panel. To reconfigure a Cryptographic Coprocessor you must deactivate it on the ICSF Cryptographic Coprocessor Management Pane, and then configure it offline using the SE.

In prior releases of ICSF, you might experience communication problems with Cryptographic Coprocessors when the process is not followed. Now, ICSF has redesigned its adjunct processor (AP) support to better handle Cryptographic Coprocessors being added, removed, or reconfigured.

**Improved ICSF CTRACE support**
ICSF Component Trace has been beefed up to include more diagnostic data and granular trace filtering. Trace options can be configured in parmlib or dynamically changed using the TRACE CT command. Additionally, you can now attach an external writer for offloading large volumes of ICSF trace data to DASD or disk.
KDS key utilization statistics
HCR77A1 introduces a common record format for all key data stores (CKDS, PKDS, and TKDS) along with the ability to track a reference date for key objects (“last used”). To facilitate migration to this new format, ICSF created a coordinated function that can convert a KDS to the new common record format. There is also a new keyword in the installation options data set to control the granularity of reference data tracking.

IQF access
Beginning in V2R12, a new lightweight callable service, CSFIQF2, is added. CSFIQF2 provides cryptographic algorithm information from internal ICSF control blocks without calling cryptographic coprocessors, and without requiring SAF protection.

DK AES PIN support
DK refers to the German Banking Industry Committee, which defines standards for electronic banking to be used by all member banks. Beginning with (HCR77A0) z/OS V1R12, ICSF implemented DK standards that use AES keys for Personal Identification Number (PIN) transactions. This support introduces new key types, new callable services, and the ability to define a set of “weak PINs” values that can be prevented from use with callable service. The table of weak PINs is managed by TKE and can be queried by the ICSF query facility.

CCF removal
Beginning with V1R12, ICSF removed support for Cryptographic Coprocessor Features (CCF) that was available with z800/z900 servers.

ICSF migration checks
In V1R13 (HCR77A1), ICSF delivers the following migration checks for:
- ICSFMIG77A1_UNSUPPORTED_HW
- ICSFMIG77A1_TKDS_OBJECT
- ICSFMIG77A1_CCA_COPROCESSOR_ACTIVE

Unleash the beast
The ICSF team is proud of this jam-packed release and appreciates the client feedback that made it possible to unleash this Cryptographic beast. ICSF Cryptographic Support for z/OS V1R13 and V2R1 really is a beast of a release that contains some great enhancements to further secure your enterprise.

For more information, see the z/OS V2R1 Cryptographic Services documentation in IBM Knowledge Center:
ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.csf/csft.htm

zFavorites
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 downloads, and much much more!

ibm.com/systems/z/os/zos/library/zfavorites/
New Encryption Facility goodies

BY STEVEN R. HART, PAUL TAUKATCH, AND RITA BEISEL

New enhancements to the Encryption Facility for z/OS product have recently been delivered in the service stream via SPEs. New features include RFC 4880 Compatibility for the OpenPGP Message Format, Speculative Key ID Support, batch mode settings for key generation, and public key export. To maintain compatibility with other OpenPGP products, Encryption Facility now includes support for the preferred OpenPGP symmetrically encrypted integrity protected data packet as well as notation data sub-packets, which contain raw binary data. You can also use Encryption Facility with the latest level of IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1. And, last but certainly not least, Encryption Facility has added support for zEnterprise Data Compression (zEDC), which includes a new complementary compression-only option.

RFC 4880 support
Encryption Facilities OpenPGP support was originally implemented based on the RFC 2440 Internet Standard for the OpenPGP Message Format. Since then, the RFC 4880 Internet Standard was published, which replaces RFC 2440. To remain compatible with other OpenPGP products, Encryption Facility has been upgraded to the RFC 4880 specification. This enhancement includes support for Speculative Key IDs. Speculative Key ID support zeros Key ID fields within OpenPGP public-key encrypted messages. By default, OpenPGP public-key encrypted messages contain the key IDs of the public keys that were used to wrap the symmetric session key that encrypted the data. When the key IDs are included in the encrypted message, an OpenPGP product can easily find the associated private key in the keystore. The private key is then used to unwrap the session key, and then the session key is used to decrypt the encrypted data.

When Speculative Key ID support is used to zero the Key IDs in the encrypted messages, the OpenPGP product must search through its keystore for a private key that is able to unwrap the symmetric session key. This search is more work for the OpenPGP product, but the advantage is that if the encrypted message is intercepted by a malicious user, this user would not know which key IDs were used to wrap the session key, making it more difficult to crack the encrypted message.

Enhancements have been made to support multiple recipients when you use the preferred Symmetrically Encrypted Integrity Protected Data Packet. By using this feature, you can encrypt data by using one symmetric session key, wrap that key with multiple recipients’ public keys, and send all of your recipients the same encrypted file. Each of your recipients will have a private key that can un-wrap the symmetric session key and decrypt the data.

With Encryption Facility, you can create digital signatures on plain text data and also on encrypted data. Digital signatures can now be made by using both OpenPGP certificate primary keys and subkeys.

You can use Notation Data Sub-packets within OpenPGP certificates to append a note. Originally, Encryption Facility only allowed these notes to be encoded in UTF-8 human-readable text. Other OpenPGP products append notes in binary format. To maintain compatibility with other OpenPGP products, Encryption Facility now supports binary-format notes.

Batch mode enhancements
In the original implementation of Encryption Facilities OpenPGP support, the generate and export functions were commands that required user responses to command line prompts. Clients, though, wanted these commands to be enhanced to support batch processing.

In response, a new set of command options are now added so that you can pre-answer all the prompts that were issued by the generate and export commands. You can configure these options in the Encryption Facility configuration file, ibmef.config, or within your batch jobs.

Support for zEnterprise Data Compression (zEDC)
In today’s z/OS environment, many installations compress certain types of data to occupy less space when not in use.
use, and then restore the data as needed. Using zEnterprise Data Compression (zEDC) to compress data can help to reduce processor cost and elapsed time of data compression compared to traditional software-based compression services, such as CSRCESRV and CSRCMPSC. zEDC can also lower the cost of applications that use host-based compression that are running on z/OS. With zEDC, you can use hardware-based data compression by using the zlib compression library.

Requirements for zEDC include:

- z/OS V2R1 operating system
- IBM zEnterprise EC12 (with GA2 level microcode) or IBM zEnterprise zBC12
- zEDC Express feature. This System z compression accelerator can improve the speed of data compression and is sharable across up to 15 partitions and up to 8 cards per CPC
- The z/OS zEDC Express software feature, which must be enabled in an IFAPRDxx parmlib member

Encryption Facility for OpenPGP uses the zEDC feature for compression of OpenPGP messages if the zEDC Express feature is available on the system and the required level of Java (IBM 31-bit SDK for z/OS, Java Technology Edition, Version 7 Release 1 or later) is installed.

There are a few things to keep in mind to get the most out of the new zEDC support. First, you must ensure that your Encryption Facility user ID has the proper READ access to the RACF FPZ.ACCELERATOR.COMPRESSION profile that protects the zEDC. If you use the zEDC hardware compression, the input data must be larger than the minimum buffer sizes for compression and decompression. If the input data is smaller than the minimum buffer size, the data is processed by using traditional software-based compression and decompression. The default thresholds for compression and decompression are 4 KB and 16 KB respectively, although you can override these values.

The original Encryption Facility CSDFILEN and CSDFILDE services provide hardware compression support through the use of the CMPSC instruction available on System z processors. Encryption Facilities OpenPGP supports use of zEDC hardware compression, providing several advantages over the existing CMPSC instruction in terms of compatibility and performance. While the CMPSC instruction uses an IBM-proprietary, dictionary-based algorithm to compress its data, zEDC uses the industry-standard ZLIB library, which provides more flexibility when moving the data off-platform. The zEDC hardware compression is also optimized to provide better compression performance across various file sizes, while the CMPSC compression is more tailored for short records, such as database rows.

**Compression-only command**

Encryption Facility is also enhanced with a new command, compress, which can be used to compress user data into a “compression only” OpenPGP message. Previously, Encryption Facility would only perform compression prior to signing or encrypting data, providing no option to do only the compression. This new enhancement extends the current capabilities of Encryption Facility and allows it to be used as a stand-alone compression tool, utilizing the standard ZIP and ZLIB libraries, and leveraging the new zEDC hardware compression support when applicable. Generated, compressed output is based on the OpenPGP message format, meaning that the data can only be decompressed by using Encryption Facility or another OpenPGP-compliant product.

**Conclusion**

Try the new Encryption Facility for z/OS enhancements today. Between the RFC 4880 support, batch mode processing enhancements, and the zEDC support, Encryption Facility for z/OS is easier and faster to use than ever.

For more information about the RFC 4880 specification for the OpenPGP Message Format, see:

[ietf.org/rfc/rfc4880.txt](https://ietf.org/rfc/rfc4880.txt)

For more information about batch mode processing as well as zEDC, see *Encryption Facility for z/OS: Using Encryption Facility for OpenPGP*, SA23-2230.

For more information about zEDC, see *MVS Programming: Callable Services for High-Level Languages*, SA23-1377.
Quick service for z/OS Management Facility

BY MIKE YOUNG

The z/OS Management Facility (z/OSMF) was released in 2009 as a web-based interface for several z/OS functions. The “new face of z/OS” has grown over the last five years to encompass a host of new features, including cutting-edge cloud application programming interfaces. Along the way, z/OSMF moved from the WebSphere Application Server OEM Edition to a much smaller and customized WebSphere Liberty Profile server.

Over the years, the goal has been to supply an interface for z/OS that requires no specialized WebSphere knowledge while simultaneously making z/OS administration easier for newcomers to the platform. Now that it is easier and simpler to manage your z/OS system, a new question arises: “How do I manage z/OSMF?”

Problems in the original configuration

The izusetup.sh script generates log and report files in your configuration log directory (by default /var/zosmf/configuration/logs/). Use the log and report files to verify whether any configuration problems exist on your system. These files are also helpful if you need to contact IBM Support.

Verifying your web browser setup

If the z/OSMF server seems to be operating correctly, but the interface has display problems, you might have an issue with your web browser. z/OSMF ships with an “Environment Checker” for browser-based issues. Direct your browser to: https://hostname:port/zosmf/izuUICommon/environment.jsp and review the data presented there.

IBM makes every effort to ensure that z/OSMF operates across a wide range of modern browsers, but occasionally there can be a particular configuration or version that causes a problem. Check the z/OS Management Facility “Browser and Operating System” web page for specific recommendations regarding your browser and operating system configuration:

ibm.com/systems/z/os/zos/features/zosmf/browser_notes.html

Problems during runtime

Every system programmer likes to be prepared for the worst. Fortunately, z/OSMF strives to exceed the System z reliability, availability, and serviceability (also known as RAS) guidelines. Even though z/OSMF does not operate quite like a traditional z/OS system component, z/OSMF provides several powerful diagnostic and error-correcting functions to help you with management and service.

z/OSMF logging

The z/OSMF server has a logging subsystem that records important details in case a problem occurs during operation. Like an aircraft’s flight recorder, z/OSMF has an in-memory buffer of trace data that records the incoming requests and operations taking place on the server. Both client-side data, from the browser, as well as server-side data are stored in the buffer. If a serious warning or error occurs, the buffer is written to a disk so that system operators and IBM Support can piece together the actions leading up to the problem. If a critical problem such as a crash occurs, IBM Service can extract the buffer data from an SVCDUMP.

Sending data to IBM Support

If you ever need to contact IBM Support, the first two things they ask for are the configuration details and logs covered in this article. It is simple and quick for you to package up this data and send it to IBM.

Issue the following z/OS UNIX commands to generate two archive package files:

pax -wz -x pax -vf config.pax.Z /etc/zosmf
pax -wz -x pax -vf logs.pax.Z /var/zosmf

The resulting pax.Z files are already compressed for direct binary FTP transfer to IBM. (There’s no need to TERSE.)

Now you understand why this article has an image of a pit crew. Servicing z/OSMF is as quick and easy as making a pit stop! 

izusetup.sh -file izuconfig1.cfg -verify all
How do Chief Information Officers (CIOs) keep pace with the requirements of the lines of business while still providing the services that keep the company viable — such as security and availability — within a shrinking budget? This is what most CIOs are facing today. This is a seemingly impossible task and is especially important because information technology (IT) is competing with external cloud providers for the money that lines of business have to spend on applications.

While there are many different strategies a CIO needs to employ to become a viable service provider, one of the key tools in his or her toolbox is platform as a service (PaaS). With PaaS, a programmer writes code around a set of components that are building blocks for an application. The programmer is not concerned with the underlying infrastructures and codes merely to a container. The rest is handled by the platform. This allows a programmer to focus on the business logic while the implementation is handled elsewhere.

**Advantages**

This provides a CIO with two advantages:

- The development process happens quickly.
- The programmer has no control over the implementation.

These advantages allow the CIO to implement according to a policy that serves the business as a whole rather than focusing solely on a single line of business. This also serves the line of business by enabling new applications to come out in a fraction of the time than they would have come out before.

What IT needs is a powerful operating system for creating PaaS solutions. Luckily, IBM has z/OS. IBM z/OS has the ability to support multiple organizations securely, providing the capabilities that do not exist on other systems. Imagine the possibility of a PaaS environment that take advantage of such z/OS subsystems as the System Authorization Facility (SAF) or the coupling facility.

**No experience required**

IBM z/OS Cloud Services allow businesses to leverage all of the power of the mainframe for people who are not familiar with the platform. These services use HTTP protocols to allow a programmer to make use of z/OS capabilities in any language. Imagine suddenly being able to take advantage of our hardware cryptographic capability through the use of a simple HTTPS call. Not only does make it easier for the programmer to use, it makes management simpler.

**Security fortress**

Here is an example of using a z/OS service that uses IBM System z security services. Programmers can issue a straightforward HTTPS call using the Representational State Transfer (REST) architecture with the text he or she wants to encrypt and it returns the encrypted token of that text. Data encrypted by one program is unavailable to another program without the appropriate keys. Because z/OS becomes the one-stop security bastion for the enterprise, it becomes easier to manage as well. All of the keys are in one place. Certificates can be issued and revoked in one place and the systems that are involved are all collocated. Security policies can be managed at an enterprise level, allowing data to be secured consistently regardless of which applications are performing the encryption.

It becomes much easier for the programmer to create solutions that use encryption. They can take advantage of the mainframe’s cryptographic superiority without needing to comprehend the infrastructure that supports it. It is a simple REST call to the service. A programmer can request the service from a catalog, have it provisioned, and use it within minutes.

It is also easier for the data center to track usage and determine cost. Unlike distributed systems, where costs are often hidden in a number of different ways, it is relatively simple to understand how many resources someone has used and the different resources that are required to fulfill a service.

So, as you can see clearly now, z/OS Cloud Services can provide benefits to each person in the IT chain. These services make it easier to deploy new applications quickly and easily with an understanding of which resources are consumed. The management of the underlying infrastructure is consolidated and straightforward. As the number of cloud services grows, these services can move up the stack and provide the transactional and secure business functions that continue to make IBM System z the premier platform for business.
Performance management and capacity planning are constant considerations for System z customers. Luckily, there are numerous tools available to make these tasks more efficient. In 2013, the System z Capacity Planning Tools team released one such tool: the IBM System z Batch Network Analyzer (zBNA). zBNA is designed to help customers optimize their batch windows and view the impact of new technologies.

Who, what, where, and why
The System z Batch Network Analyzer tool is available at no charge to all System z customers with the minimum PC requirements.

zBNA’s functional user interface allows customers to gain insights into their batch windows quickly by:

- Identifying job time sequences from the graphical view
- Filtering jobs by key attributes
- Reviewing the resource consumption of batch jobs and their individual steps
- Identifying candidate jobs to run on different processors
- Identifying direct access storage device (DASD) data sets used by key jobs and the Top 10 DASD data sets overall
- Studying the “life of a data set”
- Finding job steps with speed-of-engine issues.

In addition to analyzing the current state of their batch windows, zBNA users can perform “what if” analyses to see how the batch windows would perform in a different environment with an alternative processor model (“fewer, faster” or “more, slower” processors, for example).

The zBNA process
Figure 1 represents the flow of data throughout the zBNA process from System Management Facility (SMF) records to zBNA-generated reports. zBNA accepts various SMF record types, including 70, 72, 42 (subtype 6), 30 (subtype 4 and 5), 14, and 15 records. With all of these records combined, zBNA inputs a massive amount of data to allow users to study their batch windows with unprecedented granularity. Due to the large amount of data that is analyzed, a single batch shift is a good time frame in which to begin the analysis.

Figure 1. The flow of data throughout the zBNA process
Customers choose which records they want, which batch shift to use, and other variables, using a predefined Job Control Language (JCL) statement that is provided in the CP3KEXTR extractor program. CP3KEXTR's member EXTRZBNA gathers the data that is specific to zBNA and creates the two input files that are needed to start a zBNA study. After the Enterprise Data File (.edf) and .dat files have been transferred to a PC with zBNA installed, a customer can input the files and begin analyzing the batch window.

**zEDC exploitation**

zBNA's newest feature is its ability to estimate the basic sequential access method (BSAM) and queued sequential access method (QSAM) data sets that might be eligible for IBM's zEnterprise Data Compression (zEDC) Express Peripheral Component Interconnect Express (PCIe) cards. The zEDC reports provide customers with a starting point for the amount of zEDC Express capacity they might need to support their workloads.

zBNA uses the SMF 14 and 15 records to locate BSAM and QSAM data sets that meet the zEDC compression criteria. These criteria stipulate that the data sets:

- Are BSAM or QSAM data sets and are not virtual storage access method (VSAM) data sets
- Are extended format (EF) or are not EF
- Are EXCP=NO
- Cannot be open for update
- Cannot be open with electronic data interchange (EDI) processing

- Have a data set size (initial allocation) that is greater than 5 MB (or greater than 8 MB if there is no secondary allocation)
- Are not compressed previously or are convertible from generic / tailored to zEDC.

Figure 2 shows a sample zBNA zEDC report that estimates the number of zEDC cards required by hour.

As you can see, the top zEDC candidates are separated into three categories: compressed, EF not compressed, and physical sequential (PS) not EF. These categories represent the data sets that are currently using hardware compression, but can use zEDC compression, extended format data sets that can be compressed, and data sets that might be zEDC candidates, but are not in extended format. Again, this study is meant as a starting point for customers. More research needs to be done to verify the benefits of the zEDC Express cards.

**Summary**

The batch window continues to be important in meeting service level agreements. zBNA can help customers identify batch resource consumption and evaluate technology options to reduce the batch window. With the addition of zEDC BSAM/QSAM compression analysis, zBNA becomes an even more valuable tool for all System z customers to help analyze their batch windows.

**More information**

To learn more about the minimum PC requirements and to download zBNA, visit the IBM Technical Support Library website at:


For more information or to provide feedback, send an email to the Capacity Planning Support team at zpcr@us.ibm.com.

---

**Est. zEDC QSAM/BSAM Demand (All DSNs)**

![Image](image-url)

*Figure 2. A sample zBNA zEDC report*
System z Platform Evaluation Test
The final verification blog

IBM System z Platform Evaluation Test is the team for all releases of z/OS as well as System z hardware and System Storage! This team consists of system programmers and system testers that run a Parallel Sysplex where they perform the final verification of a z/OS release. The team tests z/OS along with the current System z hardware and System Storage products before the release becomes generally available to you. They report their findings and results directly and honestly to you to help you know what to expect.

Perhaps you already know the team from its infamous System z Platform Test Report for z/OS and Linux Virtual Servers. Today, to provide you with the same type of information in a more timely manner, they are blogging their results. There’s no longer any need for you to wait for documented information on a test report boundary!

To find the team’s latest test experiences, check out the Large Configuration System Test / enterprise (LCST/e) blog at: ibm.com/developerworks/community/groups/community/zpet

To find prior test reports, check out the IBM Platform Test—System z overview website at: ibm.com/systems/services/platformtest/servers/systemz.html
Recently there was a commotion in the halls of Building 707 at the Poughkeepsie, New York offices of IBM. “It was hard to determine the cause, so I went to my inbox for a clue and saw that IBM Knowledge Center was live,” recalls Jodi. That news was great, so what could possibly be causing the hubbub?

It turns out that all of the information centers were unexpectedly redirecting readers to IBM Knowledge Center. The move happened before we had an opportunity to tell you more about IBM Knowledge Center. The team quickly moved us back to the information centers. After explaining the information that follows to our colleagues, the halls went silent.

We know that you have questions
You want to know why z/OS is now hosting documentation in a different interface. How did IBM arrive at this new delivery method for its information? Most of all, you want to know how to use IBM Knowledge Center, and if it works as well as our information center.

Information centers are history
Until now, all IBM documentation was hosted in individual information centers — generally one per product. When they were introduced nearly 10 years ago, a significant advantage offered by information centers was that they were easily crawled by search engines like Google. People could easily search for documentation. The downside was that all content was pillared into separate information center instances. You had to bookmark multiple URLs for frequently-used product documentation. If you wanted or needed a local copy of IBM documentation, you had to install multiple information centers.

Over the years, you made it clear that you did not want to manage information centers. You told us that the information seemed fragmented. You want one location that provides all of the information for the products used in your shop. The ideal solution must contain all of the z/OS messages, as well as other products in your stack.

IBM Knowledge Center is the future
In December 2013, the total information experience (TIE) team introduced IBM Knowledge Center in a beta program. The positive response spurred the team forward to hosting all IBM publications in IBM Knowledge Center as quickly as possible.

Those outside z/OS typically find it difficult to realize just how large our library is and why every page is important. Lucky for us, the TIE team understands your needs and wants to create solutions that work for you. They are working to find a solution to let you scope your search to a specific element or task topic. By the time you read this article, there might possibly be a solution for searching messages across products.

United
IBM Knowledge Center was developed to address several things that information center delivery could not. First, IBM Knowledge Center unites all IBM documentation, creating a “one-stop shopping experience” for all users of IBM documentation. Bringing this content together makes it easier to integrate documentation for new and emerging
Having a single repository makes it easier to reuse and build upon preexisting information. This is the first time much IBM documentation has been together in one location.

Collections and commenting
You can easily create personal custom collections for disparate information, and when you log in to IBM Knowledge Center, you can save these collections for future use. You can “bookmark” and share searches. A comment facility lets you provide feedback and corrections in real time. This commenting feature can also be used to share tips with other users, which is especially important in a world where social media is becoming an important way to share your insights.

Whether you need only z/OS messages or messages from many products across the library, they can all be stored in your personal collection. Any time you want to check a certain message, you can go to your personal collection. Migration information is another example of what you can store in your personal collection while you need it, which might be only for a couple of months. Then you can delete the links to the migration information. Some of you might constantly need to look up System Management Facilities (SMF) records, or refer to JES2 or JES3, Workload Manager (WLM), Global Resource Serialization, or diagnosis material. Links to topics can easily be stored in your personal collection. Whatever your specific information needs, you can save them to your collection.

Mobile
Although information centers could be used with portable devices, the user experience was not optimal for small-screened smartphones. The IBM Knowledge Center team is looking at how IBM Knowledge Center can be optimized for devices with smaller screen real estate.

Email
Another convenience is that you can email topics to people directly from the page that you are looking at in IBM Knowledge Center. For example, during a morning status call, someone mentioned a problem that Predictive Failure Analysis (PFA) spotted. You can find the documented solution in IBM Knowledge Center and share it with the person experiencing the problem.

Your comments
Over the years, you probably discovered how much your comments mean to us. Now you can comment directly on the topic that you are questioning in IBM Knowledge Center. You no longer have to take the extra step of sending an email with your comments. Make a direct comment in IBM Knowledge Center, and you are automatically notified when the question or commented is answered.

We look forward to working with you in IBM Knowledge Center. Check out:

ibm.com/support/knowledgecenter

Look for the follow-on to this article, “Making the best use of the IBM Knowledge Center” on the z/OS Hot Topics Facebook page:

facebook.com/zosHotTopics
Our contributors

Jürgen Baumann is a software engineer for z/OS Resource Measurement Facility (RMF) at the Boeblingen Laboratory. His expertise includes UI design and development in various z/OS system management products. Jürgen has been with IBM for more than 20 years.

David Betten has over 30 years of experience in I/T management consulting and services for a broad range of industries worldwide. Considered an expert in his field, David is known for his excellent performance management and problem determination skills.

Rita Beisel is a technical writer at IBM. She is currently writing for Integrated Cryptographic Services Facility (ICSF), Trusted Key Entry (TKE), Encryption Facility for z/OS, and Cryptographic Services System Secure Sockets Layer Programming (SSL). She has written for RACF, z/VM, DB2, DFSMS, and MVS in the past. She also likes to do interface design, usability testing, and surveys.

Harald Bender has been with IBM at the Boeblingen Laboratory for 25 years. His expertise includes z/OS Performance, WLM and Parallel Sysplex. Today he is responsible for the strategy and architecture of z/OS Resource Measurement Facility (RMF).

Neal Bohling is an Advisory Software Engineer and lead of the VSAM RLS sub-component of the DFSMS Defect Support team. He holds a bachelor’s degree in computer science from Northern Arizona University and an MBA from the University of Arizona.

Kirsten Brunner McDonald is a senior software engineer in the System z Development User Experience and Design Area. She is the user experience architect for System z working on mainframe simplification. Past projects have included z/OS workload management, WebSphere Application Server, e-business on z/OS, System z Capacity Provisioning, z/OS Management Facility, and the z Hardware Management Console.

Scott B. Compton is a Senior Technical Staff Member for I/O Supervisor at IBM Poughkeepsie, NY. With IBM for more than 25 years, he has expertise in many of the core MVS components and holds a B.S. degree for the Ohio State University.

Anuja Deedwaniya is a Senior Technical Staff Member in the z/OS Development lab at IBM, and system z architect for System Management and IT analytics. Anuja has over 30 years experience at IBM in various areas of mainframe design and development.

Frank De Gilio is a Distinguished Engineer in the World Wide Client Centers. He is the Chief architect for Cloud in STG. Frank helps clients use IBM’s latest technology to modernize their IT environments. His latest focus has been on making z/OS the premier Cloud Platform for high level Cloud Services. Frank is rarely wrong.

Amar Devegowda is a senior developer in the Java Technology Centre (UTC) team, IBM India Software Labs. He has been with IBM for about 11 years and is the technical lead for the Java API Library development team. He has worked on some of the key Java SDK components like io, networking, security and time zone. He also has extensive experience in debugging and diagnosing problems in Java SDK on various platforms.

Adrian Alvarez Diez is a software engineer at IBM Germany. Since joining IBM he has worked for several projects, such as the z/OS Workload Manager. Adrian is currently a developer for z/OS Capacity Provisioning.

Jodi Everdon is a software engineer at IBM Poughkeepsie, New York. She is an executive editor for z/OS Hot Topics, and the lead planner for the z/OS information development and user design team.

Robert Fairlie is an Advisory Software Engineer on the z/OS System SSL team. When not developing new and improved support for System SSL, Robert brews his own beer, gardens, enjoys long walks on the beach, and flower arrangement. He also designs and builds trebuchets.

Peter Fatzinger is a Senior Software Engineer with over 25 years at IBM. While most of his time has been spent in the consoles component he has occasionally been spotted in logger, IOS and TSO, among other places.

Mike Fox is a senior software engineer for IBM’s Enterprise Network Solutions Design and Strategy group in Research Triangle Park, working on routing and connectivity software technologies for z/OS. Mike holds a B.S. degree in Electrical Engineering from Virginia Tech and has worked on mainframe networking with IBM for over 25 years.

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 over 20 years, most recently focusing on enhancements to the z/OS Communications Server system resolver.

Garth Godfrey is a Senior Software Engineer in the IBM zAware development team at IBM Poughkeepsie. With 34 years of IBM development experience, Garth’s expertise includes IBM zAware, z/OS UNIX and z/OS Ported Tools. Garth holds a bachelor’s degree in System Software Science from RIT.

Clark Goodrich is a Senior Software Engineer for System z Performance Organization at IBM Systems & Technology Group in Poughkeepsie, New York. Clark’s focus is performance analysis of Java on System z.

Michael Groetzner is an advisory IT specialist for z/OS Programming at IBM Germany. His expertise includes z/OS Capacity Provisioning and Hardware Configuration Definition. Michael holds a degree in Computer Science from the University of Cooperative Education in Stuttgart, Germany.

Eric Harris supports and is an expert of DFSMS Catalog. He has supported the DFSMS Catalog component for 5 years. Before becoming an IBMer, Eric earned both Information Technology and Computer Science degrees from Northern Arizona University.

Steven R. Hart, CISSP®, works on design, development and strategy for z/OS Cryptographic Services at IBM Poughkeepsie, New York. In his eleven years with IBM, he has made significant contributions to Integrated Cryptographic Services Facility (ICSF), Trusted Key Entry (TKE), Encryption Facility for z/OS, Tivoli Key Lifecycle Manager (TKLM) for z/OS, and Cryptographic Coprocessor firmware.

Samantha Hegarty has just started her career at IBM working on the zGrowth team (formerly ATS/WSC). At the moment, she is beginning her transition to the field as a System z Client Technical Specialist for the insurance industry based out of Hartford, CT.

Zhan Peng Huo joined IBM on 2013/4/15 through campus hire, worked on System Verification Test (SVT) for z/OS UNIX for ten months, and is now focused on SVT for Language Environment. Zhan obtained a master’s degree in Xi’an Jiaotong University in 2013.
**Gus Kassimis** is a Senior Technical Staff Member for IBM’s Enterprise Network Solutions group. He is a designer for the z/OS Communications Server, focusing on TCP/IP technologies, performance, and High Availability functions. He has over 25 years experience working on z/OS and is a frequent speaker at user conferences.

**Shigeki Kimura** is the technical lead for z/OS migration at IBM Japan. He expertise includes deep knowledge of z/OS release-to-release migration, and has participated in key reviews of z/OS Migration book since 2006. Also, he is a faithful contributor to z/OS Hot Topics since 2010. Shigeki has been with IBM for 28 years.

**Stephen M. Kocik** is an Advisory Software Engineer for I/O Supervisor at IBM Poughkeepsie, NY. Stephen is the lead developer for z/OS FBA Services and FICON DCM. He has been with IBM for over ten years.

**Bob Petti** is the Test Team Leader in the z/OS Security area. He has worked on numerous security and middleware components, including Integrated Cryptographic Services Facility (ICSF), Lightweight Directory Access Protocol (LDAP), Secure Socket Layer (SSL), and zFS.

**Kathy Pfeiffer** is a Senior Software Engineer at IBM in Poughkeepsie, New York. She is the System z Hardware and Software Information Solutions Manager.

**Bertold Reddemann** is an advisory IT specialist for z/OS Programming at IBM Germany, working for Hardware Configuration Definition. Bertold holds a degree in Psychology from the University of Marburg, Germany.

**Horst Sinram** is the architect for z/OS Workload Management and z/OS Capacity Provisioning.

**Geoffrey Smith** is a software engineer in z/OS Information Development in Poughkeepsie, New York. He has been with IBM 31 years and is the z/OS information strategist. He is a regular presenter at SHARE.

**Paul Smith** ("Smitty") is an IBM Service Management Architect located in Research Triangle Park, North Carolina. Smitty has 30+ years of experience working on System z service management solutions. His most recent focus area is IT Analytics for System z.

**Anthony Sofia** is a Software Engineer at IBM Poughkeepsie. His expertise includes the new z Enterprise Data Compression offering, the z/OS I/O Subsystem and the z/OS System Management Facilities. He has been with IBM for 10 years and enjoys spending his free time with his wife and son.

**Bob St. John** is a Senior Technical Staff Member from the IBM Poughkeepsie lab. He has more than 30 years of experience with MVS and z/OS and is one of the technical leaders in the IBM System z Performance Organization.

**Paul Tawkatch** is a Software Engineer for z/OS Cryptographic Services at IBM Poughkeepsie, New York. He is currently developing new features for Integrated Cryptographic Services Facility (ICSF) and Encryption Facility for z/OS.

**Mike Todd** served as the Master the Mainframe Contest team leader in the US and Canada for nine years while helping global IBM teams to launch their own contests. He recently joined the z/OS User Technologies team.

**Marna Walle** is an IBM Senior Technical Staff Member in z/OS System Install at IBM Poughkeepsie. Her current responsibilities include z/OS release migration and new installation technologies of the z/OS platform. Marna is a frequent speaker on z/OS installation and exploitation topics at System z Technical University, SHARE, and other customer conferences.

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

**Steve Wehr** is the mobile offering manager for IBM System z. Steve has held various technical positions with IBM for 35 years, with the last year focused on building the System z strategy for mobile.

**Dave Wilson** is Worldwide Market Manager in Branding for Power and z Systems, working with the IBM Academic Initiative team to bring skilled talent to market.

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

**Mike Young** is the team lead of z/OS Management Facility Client Support, also known as “Level 2”. He’s been involved with z/OSMF since its first release and is on a mission to improve its serviceability with every PTF. Mike joined IBM in 2005 as a member of the z/OS Java support team, where he continues to dabble.

- **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/library/hot-topics/hot-topics.html
Looking for opportunities to build your enterprise system skills and knowledge at no cost? Look no further. The only things you'll need to get started are an internet connection and a desire to learn!

As a part of the 50th anniversary of the mainframe, IBM announced, through partnerships with Syracuse University, Marist College and The Linux Foundation, the availability of three MOOCs (Massive Open Online Courses) providing students and professionals access to technology education for enterprise computing:

• An Introduction to Enterprise Computing — Marist College
• Introduction to Linux — The Linux Foundation
• Enterprise Computing Strategies — Syracuse University

These self-paced, web-based learning courses are open to anyone, anywhere, anytime, and have become increasingly popular. The best part is, these three offerings are all 100% free of charge!

MOOC descriptions
Each MOOC contains foundational and introductory content; no prerequisites are required. You are invited to take any of the MOOCs that interest you — register for one, two, or all three.

All of these courses teach skills and concepts that can help you to build your skills portfolio and assist with your career development goals. You can learn further details about the MOOCs by visiting the individual MOOC home page locations provided below.

Marist College: An Introduction to Enterprise Computing
This course, which traditionally runs in the spring and early summer (check the home page for the current schedule), introduces the concepts of enterprise computing and explains the role of IBM System z hardware and z/OS in the enterprise computing environment. It is intended for participants who wish to begin their studies of System z or expand their understanding of large systems with an emphasis on commercial computing. This course provides an overview of different types of workloads, such as transaction processing, business analytics, end user interactive computing, web serving, data serving, batch processing, and many others.

MOOC home page: https://mooc.marist.edu/web/ecc
Estimated time to complete: 35-45 hours

The Linux Foundation: Introduction to Linux
This course, presented by the Linux Foundation (a nonprofit consortium focused on open source), is available beginning August 1, 2014, with ongoing rolling enrollment. The MOOC is designed for those who have limited or no previous exposure to Linux. Linux appears in many different architectures, from mainframes to server to desktop to mobile, and on a staggeringly wide variety of hardware. Participants will explore tools and techniques commonly used by Linux programmers, system administrators, and end users to achieve day-to-day work in a Linux environment.

MOOC home page: https://www.edx.org/course/linuxfoundations/linuxfoundationx-lfs101x-introduction-1621
Estimated time to complete: 40-60 hours

Syracuse University: Enterprise Computing Strategies
This course, which runs in the fall semester (registration open now), is designed to help participants develop new appreciation for the issues and challenges faced by the modern technology manager. Students will learn to develop the skills and tools needed to make more informed computing infrastructure decisions in the enterprise computing environment. The course also focuses on strategies in computing resource provisioning, including ways to help organizations:

• Save money on equipment, acquisition, and labor
• Reduce energy consumption
• Simplify and secure IT infrastructure

MOOC home page: http://ischool.syr.edu/ecsmooc
Estimated time to complete: 40-60 hours