



interskill
learning

INTERSKILL MAINFRAME TRAINING NEWSLETTER

June 2015

Inside This Issue:

Interskill Learning New Releases for 2015	2
What does the new z13 tell us about where the mainframe is heading?	6
Vendor Briefs	8
Tech-head Knowledge Test	10
Learning Spotlight – Rational Developer for System z	10
JCL Word Search - Answers	11
<i>Management:</i> Is Five Nines Availability Possible, or Even Needed?	12
<i>Opinion:</i> Seven Best Practices for z/OS Availability	15
<i>Technical:</i> Five z/OS Problem Prevention and Analysis Tools You Probably Aren't Using	17

Welcome to the Interskill Mainframe Training Newsletter

Interskill Mainframe Training Newsletter is an e-Zine published by Interskill Learning, which provides world-class elearning Mainframe training programs for the Information Communications and Technology industry.

As a subscriber to the Interskill Mainframe Training Newsletter you will receive an edition several times per year. These editions will include z/OS related articles for technical experts and management, and information on upcoming releases of Interskill's Mainframe curriculum.

Interskill Learning New Releases for 2015:

★ AVAILABLE NOW

- **IBM Rational Developer for System z (RDz) Series – including:**
 - Rational Developer for System z Basics
 - Creating and Managing Applications Using RDz

★ COMING SOON

- **PMBOK5 Project Management Curriculum**
- **IBM Wave**
- **Linux on System z**
- **z/OS Connect**
- **DevOps Fundamentals**
- **Enterprise Big Data and Hadoop**
- **C/C++ on z/OS for C Programmers**
- **Language Environment and LE Dumps**
- **Java on z/OS for Java Programmers**
- **Agile and Application Development**
- **COBOL Performance**
- **Improving Application Performance with Strobe, FreezeFrame and APA**
- **IMS TM (Transaction Manager)**

Interskill Learning Updated Courses and Curricula 2015:

★ COMING SOON

- **COBOL Curriculum**
- **DB2 Curriculum**
- **IMS Curriculum**
- **REXX Curriculum**
- **Parallel Sysplex Curriculum**

Interskill Learning 2014 Releases

- **z/OS 2.1 Series**
 - z/OS Concepts and Components 2.1
 - Initializing and Terminating the z/OS System 2.1
 - Monitoring the z/OS System 2.1
 - z/OS Architecture 2.1
 - Identifying z/OS System Problems 2.1
 - Resolving z/OS System Problems 2.1
- **JCL 2.1 Series**
 - JCL (z/OS) – Introduction to z/OS JCL 2.1
 - JCL (z/OS) – Using Special Data Sets in Batch Jobs 2.1
 - JCL (z/OS) – Identifying and Resolving Batch Problems in JCL 2.1
 - JCL (z/OS) – Coding Procedures and JES2 Control Statements 2.1
- **IBM Environment Introduction Series 2.1**
 - Introduction to the IBM Enterprise Environment 2.1
 - z/OS System Programming Fundamentals 2.1
 - IBM Development Environment Overview 2.1
- **The z/OS Management Facility 2.1**
- **Introduction to zEnterprise - EC12, BC12 & zBX**
- **IBM Explorer for z/OS**
- **TSO/ISPF 2.1 Series**
 - Using Online System Facilities - TSO/ISPF 2.1
 - Managing Data Files and Definitions with ISPF/PDF 2.1
 - Maintaining Data in Files with the ISPF Editor 2.1
- **System Display and Search Facility (SDSF) 2.1 Series**
 - SDSF Concepts and Operation 2.1
 - Using SDSF to Control Job Processing 2.1
 - Using SDSF to Display, Manipulate and Print Job Output 2.1
 - Using SDSF to Manage System Resources and Devices 2.1
- **z/OS Console Simulators 2.1 Series**
 - Console Simulations 2.1
- **z/OS Batch Utilities 2.1 Series**
 - General Data Set Utilities 2.1
 - Data Utilities 2.1
 - Introduction to VSAM 2.1

- **JES2 2.1 Series**
 - JES2 System Initialization & Shutdown 2.1
 - Monitoring Batch Jobs with JES2 2.1
 - Using JES2 in Scheduling Batch Jobs 2.1
 - Identify and Resolve JES2 Batch Problems 2.1
 - Identify and Resolve JES2 System Problems 2.1

- **Operations Assessment Series 2.1**
 - CICS Concepts and Operation Assessment 2.1
 - JES2 Concepts and Operation Assessment 2.1
 - JES2 Problem Resolution Assessment 2.1
 - SNA/VTAM Concepts and Operation Assessment 2.1
 - z/OS Concepts Assessment 2.1
 - z/OS Problem Diagnosis and Resolution Assessment 2.1
 - z/OS System Operation Assessment 2.1
 - JCL Concepts Assessment 2.1
 - JCL Coding Assessment 2.1
 - JCL Problem Resolution Assessment 2.1
 - TSO/ISPF Concepts Assessment 2.1
 - TSO/ISPF Operation Assessment 2.1

- **Cloud Computing for Data Centers Series 2.1**
 - Cloud Computing for Data Centers 2.1

- **IBM Mainframe Communications Series 2.1**
 - IBM Mainframe Communications Concepts 2.1
 - VTAM Commands 2.1
 - Mainframe TCP/IP Commands 2.1

- **Data Center Storage Management Series**
 - Storage Fundamentals for Data Centers 2.1
 - Using DFSMS to Manage the z/OS Storage Environment 2.1
 - Storage Networks, Administration, and DASD Management Using ICKDSF 2.1

- **CA OPS/MVS® Event Management and Automation Series**
 - CA OPS/MVS® Event Management and Automation - Overview, Components, and Features
 - CA OPS/MVS® Event Management and Automation - Rules and OPS/REXX
 - CA OPS/MVS® Event Management and Automation - Automating Events Using the Relational Data Framework
 - CA OPS/MVS® Event Management and Automation - Automating Events Using the SSM
 - CA OPS/MVS® Event Management and Automation - Schedule and Group Managers for Event Management

- **CA ACF2™ Series**
 - CA ACF2™ - Introduction
 - CA ACF2™ - Protecting Data Integrity
 - CA ACF2™ - Protecting System Access
 - CA ACF2™ - Defining Environment Controls
 - CA ACF2™ - Protecting General Resources
 - CA ACF2™ - Maintaining ACF2™
 - CA ACF2™ - For Auditors
- **CICS Transaction Server Series**
 - CICS TS - Introduction 5.2
 - CICS Explorer Fundamentals 5.2
 - CICS TS - Controlling CICS Operations 5.2
- **z/VM 6.3 Series**
 - z/VM Concepts, System Initialization and Shutdown
 - Monitoring and Controlling z/VM Operations
 - Managing Guest Operating Systems
 - Identifying and Resolving z/VM Problems
- **IBM System z and z/OS Fundamentals Mastery Test Learning Plan and Practice Assessments updated to z/OS 2.1**
- **VSAM Series**
 - Introduction to VSAM 2.1
- **z/OS Batch Utilities Series**
 - General Data Set Utilities 2.1
 - Data Utilities 2.1
- **IBM i Programming Fundamentals Series**
 - CL Programming
 - CL Programming Functions and Messaging
 - RPG/400 Introduction
 - RPG/400 Coding
 - RPG/400 Programming
 - RPG/400 Workstation Programming Introduction
 - RPG/400 Advanced Workstation Programming
- **IBM i Application Programming Introduction Series**
 - DB2 for IBM i - Introduction for Programmers
- **IBM i Query Series**
 - Database Basics and the Need for Query
 - Creating a Simple Query
 - Advanced Query Features and Management



What does the new z13 tell us about where the mainframe is heading?

By Greg Hamlyn

The perennial question on most mainframe manager's lips is: Will the mainframe be around in the future to service our business needs? The recently released z13, along with several other IBM initiatives, holds the key to where IBM is heading with the mainframe.

Most mainframers will expect with any new hardware release that there will be increased processing power, capacity and throughput, energy efficiency, and security improvements. The z13 delivers on all these fronts, but it is some of the other features that indicate how IBM sees the future of enterprise data.

Before looking at these features in more detail, we need to backtrack and look at events leading up to the z13's release so that we can put into perspective what the z13 is delivering, both now and for the future. IBM in recent years has offloaded business that it considers not as profitable, which includes its System x servers to Lenovo, and its semiconductor-manufacturing business

to Globalfoundries Inc. The fact that it kept the mainframe development business speaks volumes for the value it places on it. To spend five years and \$1 billion dollars building it from the ground up indicates they have planned a future for the mainframe.

IBM's vision

It isn't hard to see what IBM's vision is of the data processing world in the years to come. For starters, they are pumping \$3 billion dollars into Internet of Things. The Internet of Things (IoT) is a future where common everyday objects are connected to the internet and provide data that can be analyzed for many different purposes. For example, you are driving to work for an important meeting. Your Google calendar notified your car on the destination before you even hopped in and has determined the most efficient path to get there. As you get closer the car identifies where there are free parking spaces and based on your average walking speed determines that you will be ten minutes late for the meeting.

An automatic text message is sent to the meeting participant's phones warning them of the delay. Did I mention that your alarm clock woke you earlier because of perceived traffic delays?

This simplistic example shows how objects will be communicating with each other through the internet in the future. Some of things are already happening! The end result is a lot of data that needs to be captured, processed, quickly analyzed, and converted to a meaningful user experience, whilst also ensuring stringent security. In part, that is where the z13 comes in.

z13 technology

The z13 addresses this ever increasing data processing need by providing increased memory (up to 10TB), I/O capabilities (through simultaneous multithreading in ZiiP and IFL cores), and processing (single instruction multiple data (SIMD) execution). These along with a myriad of other improvements allow the z13 to be able to process an amazing 2.5 billion transactions a day.

On the security front, the z13 cryptographic hardware provides the highest standard of file, communication, and system security available using the Central Processor Assist for Cryptographic Functions (CPACF) and Crypto Express5S through in-kernel cryptography APIs and, for Linux on z Systems, the libica cryptographic functions library. IBM knows that in a world where data is becoming increasingly important that business' cannot afford any security hiccups.

Real-time analytics (as opposed to moving data offsite to a data warehouse for analysis) is also high on the z13's achievement list. For example, fraud detection can be detected while a transaction is being performed rather than being analyzed minutes, hours,

or days later and then trying to catch the perpetrator. The z13's increased memory and I/O capabilities assist with the amount of data that now needs to be managed and processed, while SIMD facilitates data parallel processing allowing improvements of complex analytical mathematical calculations.

The IBM mainframe has been involved with cloud initiatives for some time now and with the z13 this continues. The z13 is the first generation that supports an enterprise-grade Linux, OpenStack, and KVM, and could be configured as a private cloud allowing the provision of up to 8,000 virtual servers.

z13 costs

There are no real advertised costs of the z13 but IBM will undoubtedly feel pressure to compete with the x86 market citing superior speeds and security for a greater upfront cost. With this in mind it is likely that current mainframe customers are likely to be the today's z13 buyers...big business that wants security and the ability to uncover any advantage or opportunity. In recent years, mainframe revenue from related software and services has overtaken hardware as the most profitable area, and this seems like a trend that will continue.

Name change

Can we read anything into the subtle mainframe name change? In recent years, the name has been split to accommodate business class (BC12) and enterprise class (EC12) models. While the number represents the 13th generation of CMOS mainframe, the dropping of EC/BC could indicate a number of things, including integration to a single class of mainframe? This may just be my paranoia kicking in as it may have been introduced just to freshen the image (note that even the "System z" name has changed and is now known as "z Systems").

Summary

The key trends of increasing use of mobile devices for all workloads, the proliferation of sensors, and the growth of the IoT all combine to point to a future where systems will need to manage and analyze extremely large volumes of data and very high numbers of transactions securely and quickly. IBM seems to have embraced these and point the z13 in this direction...dare I say it, reinventing itself again.

I believe that with the foundation that the z13 has built that the mainframe will be around for many years to come, especially when technology demands are becoming greater by the day.



Greg Hamlyn, a senior technical course designer and developer with Interskill, has over 25 years experience working in the mainframe and z/OS data center environments, having

performed in numerous operational roles ranging from computer operator to national mainframe technical training manager for a large Government agency. He has presented at AFCOM and has written a number of articles relating to training methods, trends, and everything that is z/OS.

You can contact him at g.hamlyn@interskilllearning.com

Vendor Briefs

Our Vendor Briefs section usually introduces products released by vendors over the last few months, and while this is covered at the end of this article, initially we look at IBM and its acquisitions and partnerships as it

presents an interesting picture of tomorrow's data frontier.

IBM

One of our main articles in this newsletter focused on the January release of the z13 mainframe, and how IBM has positioned it to cater for today's hottest trends - data analytics, cloud, security, and the processing of mobile data. Since the z13 release, IBM has announced a number of other initiatives that are aiming to take it into the future. Here's a taste:

Healthcare - IBM has unveiled its Watson Health Unit, which along with acquisitions and collaboration is looking to move the healthcare industry to the next level by providing computing and data analytic capabilities. For example, they are collaborating with Johnson & Johnson to create a healthcare delivery model that will provide patients with personalized health and rehabilitation solutions, and with Medtronic to produce personalized diabetes management solutions. IBM has also recently acquired healthcare cloud companies Explorys and Phytel.

IBM's relationship with Apple will also be leveraged to deliver healthcare-related iPhone and iPad applications.

Weather - IBM and the Weather Company have created an alliance, which will result in the Weather Company shifting its massive weather data services platform to the IBM Cloud and integrating its data with IBM analytics and cloud services. The weather has an immense impact on many businesses, so being able to analyze data that has come from a myriad of devices and sensors is going to provide huge benefits to many businesses.

Social Media - IBM has teamed with Twitter to provide cloud data services that allows Twitter data to be analyzed and integrated with public and business data (i.e. sales data, weather) to identify business opportunities. Organizations have for some time been attempting to harness the huge amount of social data to gain an edge...this could be the start of something big!

Master the Mainframe Competition

IBM's Master the Mainframe competition has been run and won for another year. Interestingly, the contest attracted the biggest turnout since 2005, with over 8,100 high school and university students around the world involved. Having this many people participating is encouraging for IBM and the mainframe world.

IBM Research and Tape Storage

In collaboration with FUJIFILM, IBM has been able to demonstrate that they are capable of creating a 220 TB tape that fits in the palm of your hand. While traditional tape use has been largely seen as a medium being phased out in many organizations, the amount of data now being produced and regulations regarding how long it is to be stored has seen a resurgence of this medium in some business sectors. Considering that as little as nine years ago, the maximum amount of data that could be stored on tape was 8 TB, this breakthrough will ensure that tape remains around for many more years.

Name Change

It would be amiss of me not to mention the subtle name change of "System z" to "z Systems". This rebranding occurred at the same time as the z13 release to signal a shift in direction for its mainframe line and will now encompass every IBM Mainframe from S/360 through the new z13, and all future systems.

Personally I would have liked to have seen a more exciting name, as flipping the words around does little to excite me.

CA Technologies

CA Technologies has recently released a new version of its CA ERwin data modeling solution that provides business and technical users with a simple, visual interface to their organization's complex data. The CA ERwin 9.6 release includes a redesigned interface, simplified reporting, performance improvements, new platform support, and ease-of-use enhancements.

GT Software

While we are on big data management, GT Software has recently unveiled their enVue data virtualization solution. This product enables users to easily access and interact with enterprise-wide data (cloud, relational, NoSQL and mainframe data) via one simple, single-source access point, and use it for analysis or inclusion in BI tools.

Compuware

Compuware's new Topaz platform is designed so that today's new generation of developers can effectively work with and understand mainframe code and data. It is based on the Eclipse framework, which many IBM products in recent times have also adopted (i.e. RDz, z/OSMF), and contains tools that show relationships between data objects, and allows mainframe data to be edited and copied across repositories. The most recent upgrade also includes the ability to analyze COBOL and PL/I mainframe programs and present the results of that analysis visually.

Tech-head Knowledge Test

With every release of this newsletter a mini online test will be provided of a product that you are likely to be using in the workplace. You may want to challenge your work colleagues or just want to confirm your own knowledge!

The test for this newsletter focuses on the use of z/OSMF, and consists of 13 questions. Click the link below to start.

[z/OSMF 2.1](#)

Learning Spotlight – Rational Developer for System z

Rational Developer for System z (RDz) is a development environment for COBOL, CICS, IMS, PL/I and mainframe applications and is fast becoming the defacto standard tool for developers.

The module that we have provided for you here is [Getting Started with Rational Developer for System z](#), which describes the general steps involved when implementing RDz, the RDz product interface components, and explains how workspaces and projects are utilized by RDz personnel.

We hope you enjoy it.

JCL Word Search - Answers

Below is the word search from our last newsletter, where we asked you to find the hidden JCL parameters. At the bottom are the ones you should have found.

R	T	I	X	O	E	Z	I	S	K	L	B	S	C	S
J	E	N	O	T	I	F	Y	U	C	C	M	E	E	E
P	I	F	E	T	S	E	D	E	O	S	L	I	X	T
B	S	Q	D	M	U	E	R	N	G	A	E	P	E	S
L	O	I	T	D	M	L	D	C	S	A	B	O	I	P
D	C	B	D	L	P	O	L	F	Y	F	A	C	V	A
N	K	U	J	D	R	A	C	E	S	K	L	T	E	C
T	T	D	L	O	S	T	M	X	O	W	L	J	T	E
R	X	O	N	S	B	A	S	C	U	Q	U	Y	Y	T
T	H	A	V	R	N	N	E	I	T	J	P	O	I	K
P	D	W	P	S	E	B	K	Z	Q	R	B	J	T	X
X	I	P	D	Q	C	G	F	A	U	Y	M	M	U	D
Z	B	H	X	D	W	W	I	N	T	I	C	R	Z	U
L	U	M	R	E	C	F	M	O	B	N	Z	R	K	A
D	N	E	Y	B	D	K	D	U	N	X	G	J	E	Z

BLKSIZE	COMMENT	COND	COPIES
DCB	DISP	DSNAME	DUMMY
EXEC	EXPDT	HOLD	JOB
LABEL	LRECL	MSGCLASS	NOTIFY
RECFM	REFDD	REGION	SET
SPACE	SYSOUT	TYPRUN	

Management: Is Five Nines Availability Possible, or Even Needed?

By David Stephens, Lead Systems Programmer at Longpela Expertise

For the past six years, Information Technology Intelligence Consulting (ITIC) has performed a survey of server and operating system reliability. In 2014, this survey found z/OS as the system with the highest availability. This was no different from previous years, confirming what most people think as fact: z/OS on IBM mainframes is king when it comes to stability and reliability.

Another interesting fact from this survey is that 79% of respondents required five-nines availability for their mission critical systems. In fact, five-nines, or 99.999% availability is quickly becoming the standard target for key systems. But can you really get five-nines availability, and do you really need it?

Scheduled vs Unscheduled

'Five-nines' availability comes from the telecommunications industry, where this 'carrier grade' availability is expected for hardware and systems. To achieve five-nines, a system can be down for at most 5.26 minutes per year (or 5.27 for a leap-year). There is no standard, committee or ruling body formalising five-nines and what it means. So it is an often-used term that can mean different things, depending on where you're coming from.

Let's start with IPLs. Regular z/OS IPLs are essential for upgrades and system maintenance. Assuming each IPL takes an

hour from start to finish, we can IPL at most once every seven years for five-nines. This isn't going to happen. What we can do is put together a Parallel Sysplex with workload sharing or something similar. So while one system is IPLed, others handle the workload. This kind of duplication will eat up a lot of money and resources to setup and maintain. Even with this configuration, you may still need to take down the entire Sysplex from time to time.

Many providers claiming five-nines availability will add small print to get around this problem. By excluding scheduled downtime, five-nines become a lot easier.

Components

A complete computing application or system is made up of many pieces. Think hardware, software, networks, and applications. A failure in any affects availability. The foundation of it all is the data centre. The Uptime Institute has created a standard classifying data centre performance and availability into four Tiers. In a previous whitepaper, they state that the tier with the highest availability (Tier IV) will aim for a maximum availability of 99.995%. Even more interesting, they state that most Tier IV data centres won't get this 99.995%. So if 99.995% is the best we can do, anything running in only one data centre cannot make five-nines. We need to divide our workload between two data centres in different cities or countries: an exponential increase in costs.

Let's take another component: the network. Australia's largest telecommunications provider, Telstra, offer availability options for leased lines (at a cost) of up to 99.99% (four-nines). So again, we need to do something extra to bring this up to five-nines And

remember that this is only the components provided by the vendor. Your own network infrastructure adds more components that could fail.

Similar arguments can be found for all components needed by any application. For example, an online z/OS CICS/DB2 application receiving traffic from Websphere MQ relies on six major components: data centre, z/OS, CICS, DB2, Websphere MQ and the network. And this is just the mainframe side. To achieve five-nines availability here, average downtime for each component can be at most one minute per year: six-nines availability. The required average reliability increases with the number of components.

More fine print can help if you want five-nines bragging rights: limiting the components covered. So a hardware manufacturer may talk about availability for their hardware device alone, or a z/OS team their z/OS systems excluding hardware and data centre outages.

What Does Unavailable Mean?

Anyone working with availability or service level agreements will be familiar with the problem of defining availability (or unavailability). For example Automated Telling Machine (ATM) transaction authorisations must be approved within 20 seconds. So if response times exceed this, the system is effectively down, even though everything may be working fine, but slowly.

Even if systems are performing, end-users may consider the system unavailable. Consider a computing system where one function is unavailable, but the rest are working fine. Or an end-user that can't use a system until a security change is made. Training and application complexity are

other areas to muddy the waters. If an end-user cannot do something because of lack of training or complicated user interfaces, that function is effectively unavailable.

Humans and Black Swans

The 2014 ITIC survey included another interesting point: 44% of network errors were from human error. Other surveys differ in this figure. But regardless of the details, human error is a large cause of errors and downtime. It can be minimised by ongoing training, change control, monitoring and problem post-mortems. However it cannot be predicted or eliminated. Nor can any other Black Swan event such as viruses, hacker attacks, terrorist activity, or rogue weather.

Aiming for a maximum of five minutes downtime per year effectively limits downtime to very small events that are automatically resolved by monitoring systems. If a human is required to resolve the problem, they are not going to detect, identify and fix it within five minutes. The moment a human is needed, your five-nines are gone for the year.

SLAs

Five-nines, or any percentage availability figure looks good in Service Level Agreements (SLAs). It is easy to understand, and 99.999% (or even 99.99%) are attractive figures. However SLAs can be very creative when dancing around five-nines. For example, Amazon's EC2 Cloud service offers 99.95% availability, with credits if this is not met. However it includes much of the expected fine-print:

- Events outside of Amazon's control are excluded.

- Scheduled outages are excluded.
- Downtime is measured from when the Cloud service is in a Region Unavailable state: i.e. when Amazon systems detect that the service is down.
- Downtime is measured monthly: last month's failure does not affect this month's statistics.

Another SLA from a different company claimed 99.999% availability with credits and rebates when this was not met. However the fine print was again interesting. Credits and rebates were only paid after total outages exceeded one hour per month. Or in other words, the company really offered 99.85%, or two-nines availability.

Are Five-Nines Even Necessary?

A study by Compuware and Forrester Research in 2011 found an average business cost of US\$14,000 per minute for mainframe outages. Using this figure, outages for systems with five-nines cost about US\$73,500 per year. For systems with four-nines (99.99%, or 52.56 minutes downtime) this increases to US\$735,000, and three-nines (99.9%, or 525.6 minutes) US\$7.3 million. So a business case to improve availability from four to five-nines needs the extra hardware, software and resources to cost less than about US\$660,000 per year. When we're talking about mainframe hardware and software, \$660,000 doesn't buy much. A jump from three-nines to four-nines may save millions, and is a far easier business case.

Another question is: does the business really need such high availability? Here mainframes may be victims of their own reliability. If a system has no outages for two years, the business will expect and demand that this continue. But the fact remains

that outages still happen. Just ask Air New Zealand in 2009, HSBC in 2011, or the Royal Bank of Scotland in 2013. Most businesses with mature mainframe-based applications have developed procedures in the case of an outage. For example, ATMs may provide limited service while the back-end host is unavailable. Similarly retail stores continue to have a manual way of accepting credit card payments.

Summary

The fact is that five-nines availability for an entire computing service is impossible to guarantee. There is too little room for error and Black Swan or unexpected events are impossible to eliminate. Many service providers advertising such five-nines availability use fine print to make it an easier target.

For many of those original 79% of respondents requiring five-nines availability for their critical systems, this may not be possible to cost-justify. Four-nines, or even three-nines may give them what they need for a far smaller price tag.

Source:

LongEx Mainframe Quarterly - February 2015, Retrieved from <http://www.longpelaexpertise.com.au/ezone/FiveNinesAvailability.php>

Opinion: Seven Best Practices for z/OS Availability

By David Stephens, Lead Systems Programmer at Longpela Expertise

As z/OS systems programmers, we're lucky enough to work on the most stable computing platform and operating system available. We have more control, more information, and more ways to minimise downtime and maximise availability. This is all great, but errors and problems can still occur. The good news is that there are ways to minimise these errors and problems, and the downtime they cause.

So here are my seven top ways to maximise z/OS availability, and minimise downtime. Most of these are best-practices that most systems programmers and shops are already doing.

1. Health Checker

Implementing the IBM Health Checker that comes free with z/OS is a no-brainer. It's easy to setup, easy to use, and consumes little CPU or other resources. The first time you switch it on, it always finds some areas of your z/OS system you hadn't thought of. If you haven't enabled it yet, you will with z/OS 2.1 – it's started by default.

An add-on to the Health Checker, Predictive Failure Analysis (PFA) is another freebie with z/OS that helps detect soft errors – errors that by themselves may not be a problem, but may indicate or become one. Think of your JES spool filling up, or running out of CSA. Again, easy to setup, easy to use, little CPU or resource overhead.

2. Six Months Behind Up-to-Date

Keeping current with software maintenance is key to maximising uptime. However I don't like to keep that up-to-date. My rule of thumb is to apply Hipers as soon as possible and other PTFs six months after they're released. This lets others find any bugs from PTFs. If I'm installing new software, I like software to be 6-12 months old before installing it – the bleeding edge is not where I want to be.

3. PROG

IBM over the years has introduced the PROGxx parmlib members to create linklists, add libraries to APF and add exits. However many sites still use the old LNKLSTxx, IEAAPFxx and EXITxx parmlib members. Using PROGxx has many advantages. One is that if there is a problem with the member or statement, it doesn't stop the IPL. The PROG statement fails, and everything moves on. Put as much as you can into PROGxx to reduce chances of a Parmlib or dataset change stopping and IPL in its tracks.

4. Always have a z/OS System

I hate not having a z/OS system up and running. If I IPL a system, I want another z/OS system up in case of problems. This is possible with those in a Parallel Sysplex – most of the time. However hardware upgrades, DASD moves and other maintenance may require the entire Sysplex to be brought down. This is where my [One Pack System](#) gives me piece of mind. If all else fails, my regularly tested One Pack System comes to the rescue.

5. No Super Logons for Everyday Use

Systems programmers often need to do things that require a high-powered logon. As do DASD administrators, security administrators and others. However these high-powered logons should only be used when needed. For day-to-day use, another, normal logon should be used. This prevents accidents with high-powered logons.

6. Always Have a Plan B

What do you do if someone modifies all your TSO logon procs, stopping anyone from logging on. Or a pack with one dataset used in all TSO logon procs fails? How about if a user catalog needed for any TSO logon fails? Or if you've shutdown all your z/OS systems, and none will come up again – parmlib failure, master JCL failure, VTAM failure?

If you don't know what you would do, then you need to do some serious disaster recovery planning. When many think about disaster recovery, they think about backups, hot sites, GDPS and similar. However there are many other potential problems that could occur. If they do, you need a Plan B. Let's look at some of my favourite Plan Bs.

- A TSO logon proc with the absolute minimum dataset allocations is very handy to get ISPF up and running if other datasets become unavailable.
- Duplex JES2 checkpoints, and a spare SPOOL volume I can add if SPOOL suddenly fills up.
- Spare page datasets I can add if the normal ones fill up.
- A minimal VTAM and TCPIP configuration that can be used in emergencies.
- An emergency parmlib LOADxx and friends for to startup a minimal z/OS system.

- A RACF (or CA- ACF2 or CA Top Secret) backout plan in case someone removes RACF access to sys1.vtam1st (I've seen this happen) or similar.

And there are so many more. But you get the idea. I like to try and think of problems, and see if I can figure out how to fix it.

Plan B also applies to any changes. What if the change doesn't work. How to undo it. What if the change breaks something else – how to fix it?

7. Review and Change Control

Look back through your problem management systems, and you'll see that most of the problems are from human error. And we've all done it – entered a command on system PROD rather than system DEV, or deleted dataset A rather than dataset B. Perhaps the best way to minimise human error is to have someone else check your changes. Set the changes up, include instructions, and then have someone else check it.

When you think about it, add this to a Plan B for changes, and you have a good chunk of a change control system. And change control is the best way to minimise damage and outages from changes. I've worked at sites with and without change control, and although I can get frustrated with change control, it's better.

Oh yes, and that means no 'on the fly' changes.

Source:
LongEx Mainframe Quarterly - February 2015, Retrieved from
<http://www.longpelaexpertise.com.au/ezone/FavouritesAvailability.php>

Technical: Five z/OS Problem Prevention and Analysis Tools You Probably Aren't Using

By David Stephens, Lead Systems Programmer at Longpela Expertise

z/OS continues to be the availability leader when compared with any other operating systems. Over the past years z/OS has included software and features to manage or prevent problems and further extend this leadership – many of which you're probably not using now.

Runtime Diagnostics

Runtime Diagnostics takes the hard work out of z/OS problem determination. Set it off and it will search the z/OS OPERLOG for messages or message combinations that indicate a problem – all within a minute. It will also check things like ENQ and latch contention, address spaces using too much CPU or locks, looping tasks, and more. Rather than run continuously, it is designed to be started when a problem has been detected, and you want a place to start looking. In many ways it does initial problem diagnosis a Systems Programmer would do, but faster.

- What It Is: A z/OS based feature to quickly identify potential problems on a z/OS system.
- Why it's cool: Simple to setup, simple to use, and free with z/OS. Simple reports quickly indicate if a problem was found.

- Why it's not: It burns CPU. Only used when there is a problem to speed up diagnosis.

Health Checker

"Ahah!", you say. "We've already implemented Health Checker." And you're probably right. A 2014 SHARE survey showed 77% of sites had implemented Health Checker. And from z/OS 2.1, Health Checker will be started by default, which will take care of the rest. However by 'implement', most sites will have it switched on with the default settings. Which is great – IBM provide a suite of checks that are brilliant, and are automatically implemented with little or no extra configuration effort. So far, so good.

The problem is that most sites consider Health Checker a handy tool for Systems Programmers. So there's no automation to detect Health Checker alerts, and notify people / raise problem tickets when they occur. Nor are there procedures or requirements to regularly check and resolve Health Checker alerts. This can result in issues being overlooked for some time.

- What It Is: Regularly and automatically check for problems or configurations that do not adhere to best practices.
- Why it's cool: Easy to implement, easy to use, almost zero CPU overhead, free with z/OS.
- Why it's not: No reasons. From z/OS 2.1, it will be started by default.

Predictive Failure Analysis

Predictive Failure Analysis (PFA) piggy-backs on the z/OS Health Checker infrastructure to add an extra layer of problem detection: soft failures. The classic example is a z/OS system

running out of CSA memory. Many tasks use CSA, however you get little warning before it runs out. When this happens, your solution is often an IPL, and then go back to find out the cause.

PFA can detect this problem before it becomes critical. It analyses historical and current data (thanks to Health Checker facilities), saving historical data in a z/OS UNIX file system. From this it will raise Health Checker alerts when it thinks there is a problem. Current areas PFA checks includes CSA, memory usage, LOGREC record rates, SMF record rates, JES SPOOL usage and console message rates. IBM has indicated that more will be added in the future.

This all sounds great, but how much CPU does it burn? Most of PFA is Java, so offloading to zIIP is on the cards.

- What It Is: Enhance Health Checker to detect soft failures.
- Why it's cool: Easy to implement, easy to use, free with z/OS.
- Why it's not: Some small CPU overhead, though much can be offloaded to zIIP.

zAware

First released in 2013 with the EC12, zAware is an extra partition on your System z mainframe. It analyses console message traffic patterns, learning what is normal over a 90 day period. Using some nifty analysis techniques, it identifies patterns that may indicate a problem. With GUI screens for problem analysis, it's a nice tool to detect problems as they occur, and simplify problem determination.

The flipside is that it's an extra piece of hardware on your mainframe. It uses ½

to 2 IFL or GP processors (2 partial CPs recommended), so it will reduce the CPU available. It will also slightly increase the CPU usage of every z/OS system monitored – say doubling the System Logger CPU. The zAware partition also needs around 500GBytes of DASD, 4GBytes+ of memory, and adds load to your OSA or Hipersockets.

Bottom line – a solution for larger shops where availability is critical.

- What It Is: Advanced Workload Analysis Reporter (AWARE) - Additional LPAR installed on a System z mainframe to monitor OPERLOG messages. Uses analytics to identify console message traffic patterns that may indicate a problem. Can work with software such as NetView, z/OSMF, BMC MAINVIEW AutoOperator and Omegamon XE to automate monitoring of anomalies.
- Why it's cool: Finds problems other monitors may miss, and possibly before they impact service. Nice GUI for problem determination.
- Why it's not: It's an extra processing on your mainframe. It is firmware supplied with hardware, with quite a few configuration tasks required. It adds a bit of z/OS CPU, uses more CPU for the partition, DASD and memory. And one more thing: it isn't free.

SCA-LA

In 2013 IBM released the Linux based SmartCloud Analytics Log Analysis (SCA-LA) software to quickly analyse log data from many different sources. It included GUI screens to simplify message searching and analysis. The z/OS Insight Packs expand this to z/OS: z/OS syslog and Websphere Application Server logs.

- What It Is: GUI to analyse, index and search through messages from multiple systems from a nice GUI screen.
- Why it's cool: Quickly search and process messages from multiple systems: z/OS and other.
- Why it's not: It isn't free. It is an extra Linux box, with message collectors on each z/OS system - so some z/OS CPU overhead.

Source:

LongEx Mainframe Quarterly - February 2015, Retrieved from

<http://www.longpelaexpertise.com.au/ezine/ProbDetermineTools.php>

Copyright Disclaimer: Interskill Learning product names and images referenced herein are registered trademarks, or trademarks of Interskill Learning, or one of its subsidiaries. All other trademarks, trade names, service marks, content, images, and logos referenced herein belong to their respective companies.