



z/OS 2.2 User Experiences

Ed Jaffe

Phoenix Software International

11 August 2015

Session Number 17740



#SHAREorg



SHARE is an independent volunteer-run information technology association that provides education, professional networking and industry influence.



Come Out and Play z/OS with Me



- Like many men my age, I enjoy my “toys.”
 - My electric car, my 6-way DVR, my drone, my HTC One M9
- We know that z Systems represent the most sophisticated and powerful business computers devised by man.
- Despite that, I treat them as “toys.”
- This presentation more or less documents my discoveries as I “fool around” with one of my toys aka z/OS 2.2.
- I hope you enjoy tagging along as we explore some of its cool new features...
- Final tests performed with **z/OS 2.2 ETP BLDH 2015-07-28**

Early Test Program

- Phoenix Software International (PSI) was privileged to work with early z/OS 2.2 drivers via IBM's Early Test Program.
 - <http://dtsc.dfw.ibm.com/MVSDS/%27HTTPD2.DSN01.PUBLIC.PDF%28ISVEPROG%29%27>
- This program is available only to participating ISVs and not to customers. Then why should you care about it?
- Every *serious* ISV on planet Earth should be involved in IBM Early Test programs. They provide access to early code builds for purposes of:
 - Developing/testing ISV products to ensure they support the latest operating system and middleware.
 - IMHO, all ISVs should at least tolerate new releases by the IBM Early Support Program start.
- Helping to “shake out” bugs and re-shape APIs in z/OS and related products before customers see them.

Early Test Program (*continued...*)

- Our first ETP was in 1994 with MVS/ESA 4.3. Since then, we have maintained an *As-of-GA* toleration policy.
- Before that, we had a *GA-Plus-Six-Months* toleration policy.
- Apparently, based on IBM-MAIN postings, not all ISVs avail themselves of this opportunity. Recent “surprises” include:
 - Abends after **AllowUserKeyCsa(No)** became default
 - Abends after **CaptUcb Protect=Yes** became default
 - Abends after **UseZosV1R9Rules(No)** became default
- If you have ISVs with *GA-plus-non-zero-time* toleration policies, you might ask if they are involved with IBM Early Testing programs. If not, politely suggest they should be.
- It’s an extra cost, but—for us at least—well worth it!

Table of ISV Products Supporting z/OS 2.2?



- Since the early OS/390 days, IBM has maintained a table of ISV products detailing support for recent releases of the operating system.
- THIS NO LONGER EXISTS FOR z/OS 2.2!
- I was told, “A decision was made to no longer generate similar pages. The IBM Global Solutions directory is our primary repository for this type of information, now and going forward.”

<http://www.ibm.com/systems/z/os/zos/software/isv210.html>

Company Name	Product(s)	Available	EAY
Phoenix Software International	CONDOR (E)JES FALCON Key/101 NetTester PHX-Adders PHX-KeyPlus PHX-ODE zHISR	Now for all products listed	Yes for all products listed

- Similar tables are available going back to z/OS 1.3.

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



What Difference Did Agile Make?

- Previously, IBM development used a so-called “waterfall” development approach. In recent years, they have starting using Agile development techniques.
- With the “waterfall” approach, most functions appeared in the first driver. Each subsequent driver was more stable.
- With Agile development, we now get the first driver much earlier than before! 😊 However, many functions will not appear until later drivers.
- Some planned functions won’t appear until *after* GA. We seem to be headed toward a continuous delivery model.

ETP Install is Not ServerPac

- The package is delivered in a format agreed upon by ISVs and IBM back in the early 1990s. It is a DSS logical dump of Target, DLIB, and SMP/E volumes.
 - No JCL or tooling to help build a system
 - No catalogs or operational data sets
 - Intended to be overlaid by the next driver rather than being serviced with PTFs.
- In general, this install technique has little relevance for a presentation to customers that will use ServerPac.
- You might recall z/OS 2.1 “blew the doors off” our DASD space. We encountered no such surprises transitioning from z/OS 2.1 to z/OS 2.2.

Our Environment

- zBC12 with 3 CPs, 1 ICF, 1 IFL, 1 zIIP, 1 HMC
- Various LPARs configured including:
 - Our z/OS “primary” parallel sysplex (PHXHQ)
 - Bronzeplex with multi-image JES2 and JES3 JESplexes
 - Mixed z/OS releases during the early test process
 - Eventually all are migrated to new z/OS
 - z/VM LPAR running numerous guests including:
 - New z/OS in a stand-alone system
 - New z/OS in a virtualized parallel sysplex
 - Two virtualized CFs and two z/OS images
- Storage
 - FICON-attached IBM DS8100 DASD (w/zHPF)
 - FICON-attached IBM TS1140 Tape Drives in 3584 Library

Deployment

- We first deploy new z/OS under a stand-alone z/VM guest.
- Then we deploy into a z/VM virtual parallel sysplex with two virtualized CFs and two z/OS systems. Both systems run the new z/OS release. We do some development and testing here. This is also where we apply service.
- Lastly, we deploy into one or more LPARs in our primary parallel sysplex. The other systems continue to run the current z/OS release w/necessary toleration maintenance applied.
- All new z/OS images run from exactly the same read-only SYSRES volumes—shared between z/OS and z/VM.

SHARE Requirements Satisfied in z/OS 2.2 (From 2013 “Top 50” and “Next 25” Lists)



Rank	SHARE#	Title	RFE#
2	SSMVSE99007	Externalize IEFUSI to set Storage Limits	50153
5	SSMVSE03030	Enhance handling of dynamic subsystems	49922
5b	SSMVSE11001	MVS should validate Subsystem Init Routine before adding subsystem	50057
19	SSSHARE016690	Region Size for Private and Extended Private	49889
22	SSMVSE06001	Add IPL unit address and volume serial to IPL message IEA0911	49928
26	SSMVSE12011	Private Virtual Storage Exhaustion Health Check is Needed	49952
28	SSMVSE12018	BCPii needs to have logging facilities	50311
29	SSMVSE12014	PER SLIP Traps Should Capture the BEAR	49565
36	SSMVSO07003	Need list of which ZFS is quiesced in message IOEZ00581E	47430
42	SSMVSE09008	Allow SYS1.LOGREC to be reallocated and used without an IPL	49564
N4	SSMVSE12006	z/OSMF should accept Service Request PMR format	47383
N5	SSMVSE11033	Add JCTJOBID and Sysplex to SMF types 14/15 (data set close)	51903
N10	SSMVSE064943	Create Dataset Level SMF Audit Trail for APF List	49917
N20	SSMVSE12010	Improve purge of PFA saved exception data	49589

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Externalize IEFUSI to set Storage Limits

- I searched around my early publications for this information, but could not find it.
- The information did not make the GA publications either.
- I was told it is expected to be added sometime *after* GA via new parmlib member SMFLIMxx.
- Welcome to the Agile world of continuous delivery ... 😊
- Be on the look-out for this feature when it arrives.
- **UPDATE:** During the session John Eells said it should make the GA delivery. Yay!

Enhance Handling of Dynamic Subsystems



- The problem – *I have been personally burned by this more than once* – was that if anything goes wrong when adding a subsystem, that subsystem is “broken” until the next IPL
- There were three requirements in this area:
 - Ensure initialization routine exists and runs properly before adding the subsystem
 - Allow delete of existing subsystem
 - Allow full reuse of existing subsystem
- IMHO, IBM satisfied all three 😊

Enhance Handling of Dynamic Subsystems

```
SETSSI ADD ,SUBNAME=SHAR ,INITRTN=SHARE806 ,INITPARM=DEMO4SHARE  
IEFJ023I SETSSI ADD COMMAND FOR SUBSYSTEM SHAR COMPLETED WITH ERRORS  
IEFJ027I SUBSYSTEM INITIALIZATION ROUTINE SHARE806 NOT FOUND
```

D SSI

```
IEFJ100I 11.31.21 SSI DISPLAY  
...  
SUBSYS=IOSH  
    DYNAMIC=YES      STATUS=ACTIVE      COMMANDS=REJECT  
SUBSYS=SHAR  
    DYNAMIC=YES      STATUS=INACTIVE    COMMANDS=REJECT
```



z/OS
2.1

```
SETSSI ADD ,SUBNAME=SHAR ,INITRTN=SHAREAOK ,INITPARM=DEMO4SHARE  
IEFJ023I SETSSI ADD COMMAND FOR SUBSYSTEM SHAR COMPLETED WITH ERRORS  
IEFJ026I SUBSYSTEM SHAR IS ALREADY DEFINED TO THE SSI
```

```
SETSSI ADD ,SUBNAME=SHAR ,INITRTN=SHARE806 ,INITPARM=DEMO4SHARE  
IEFJ023I SETSSI ADD COMMAND FOR SUBSYSTEM SHAR COMPLETED WITH ERRORS  
IEFJ027I SUBSYSTEM INITIALIZATION ROUTINE SHARE806 NOT FOUND FOR SUBSYSTEM SHAR
```

D SSI

```
IEFJ100I 11.34.50 SSI DISPLAY  
...  
SUBSYS=IOSH      HEX=C9D6E2C8  
    DYNAMIC=YES  STATUS=ACTIVE      COMMANDS=REJECT
```



z/OS
2.2

```
SETSSI ADD ,SUBNAME=SHAR ,INITRTN=SHARE806 ,INITPARM=DEMO4SHARE  
IEFJ023I SETSSI ADD COMMAND FOR SUBSYSTEM SHAR COMPLETED WITH ERRORS  
IEFJ027I SUBSYSTEM INITIALIZATION ROUTINE SHARE806 NOT FOUND FOR SUBSYSTEM SHAR
```

Enhance Handling of Dynamic Subsystems

- Full reuse requires DELETE, which is now supported, even for subsystems defined at IPL!
- It requires use of the FORCE keyword – a big clue that this should not be attempted by the faint of heart.
- SETSSI DELETE also allows SUBNAME to be specified as a hex value (e.g., 'E2C8C1D9'X)

```
SETSSI DELETE, SUBNAME=SHAR
ASA100I SYNTAX ERROR: ,
EXPECTED BEFORE <END_OF_LINE>.
DETECTING MODULE IS IEFJSMMSG
```

```
SETSSI DELETE, SUBNAME=SHAR, FORCE
IEFJ023I SETSSI DELETE COMMAND FOR SUBSYSTEM SHAR COMPLETED WITH ERRORS
IEFJ024I SUBSYSTEM SHAR NOT DEFINED
```

```
SETSSI ADD, SUBNAME=SHAR
IEFJ022I SETSSI ADD COMMAND FOR SUBSYSTEM SHAR COMPLETED
SUCCESSFULLY
```

```
SETSSI DELETE, SUBNAME=SHAR, FORCE
IEFJ022I SETSSI DELETE COMMAND FOR SUBSYSTEM SHAR COMPLETED
SUCCESSFULLY
```

IPL saved!



Region Size for Private and Extended Private



- This has been needed since way, Way, WAY back when MVS/XA was first written. A single REGION= value required numerous silly rules and inferences to be established.

JCL REGION= Parm	Results Below 16MB	Results Above 16MB
0	The job step is allocated all storage available below 16M, resulting in region size of 16M minus the amount of virtual storage allocated to MVS.	The job step is allocated all the storage available above 16M, resulting in an extended region size of 2G minus the amount of virtual storage allocated to MVS, minus 16M.
0 > REGION >= 16M	Establishes the size of the private area below 16M. If the region size specified is not available, an ABEND S822 occurs.	The extended region size is exactly 32M
16M > REGION >= 32M	The job step is allocated all storage available below 16M, resulting in region size of 16M minus the amount of virtual storage allocated to MVS.	The extended region size is exactly 32M
32M > REGION >= 2G	The job step is allocated all storage available below 16M, resulting in region size of 16M minus the amount of virtual storage allocated to MVS.	The extended region size is the specified value. If the region size specified is not available, the job step receives the maximum amount available.

Region Size for Private and Extended Private

- New REGIONX= keyword allows both values to be explicitly specified.
- I was given incomplete documentation on this new keyword, so I was able to do basic syntax checking only.

```
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(4M,256M),REGION=0M  
IEFC009I KEYWORD REGION IS MUTUALLY EXCLUSIVE WITH KEYWORD REGIONX ON THE JOB STATEMENT
```

```
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=1K  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=15M  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(4M,)  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(4M,256M)  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=16380K  
IEF142I IEFBR14 - STEP WAS EXECUTED - COND CODE 0000
```

```
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGION=16380K  
IEF085I REGION UNAVAILABLE, ERROR CODE=20  
IEF450I IEFBR14 - ABEND=S822 U0000 REASON=00000014
```

```
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=16M  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=256M  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(,16M)  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(,256M)  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=256M  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(,15M)  
//IEFBR14 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,REGIONX=(,4M)  
IEF638I SPECIFIED NUMERIC EXCEEDS MAXIMUM ALLOWED IN THE REGIONX FIELD
```

← This test proves the function isn't actually enabled yet on my system.

Region Size for Private and Extended Private



- Like SMFLIMxx, this feature is expected to be added sometime *after* GA.
- Welcome to the Agile world of continuous delivery ... 😊
- Be on the look-out for this feature when it arrives.
- **UPDATE:** During the session John Eells said it should make the GA delivery. Yay!

IPL Unit and Volume Serial on IPL Message

- Who *hasn't* been burned by this in the past 40 years?
- *You IPLed from the wrong volume!*
- Now, this information is captured in the log.
- Instead of updating IEA091I, the developers satisfied the requirement with message IOS128I.

```
IEA371I SYS4.IPLPARM ON DEVICE 8100 SELECTED FOR IPL PARAMETERS.
IEA246I LOAD ID 22 SELECTED
IEA246I NUCLST ID 00 SELECTED
IEA519I IODF DSN = SYS4.IODFA1
IEA520I CONFIGURATION ID = ZOS . IODF DEVICE NUMBER = 8100
IEA091I NUCLEUS 1 SELECTED
IOS128I IPL DEVICE: 08110 VOLUME: T2RES1
IEA370I MASTER CATALOG SELECTED IS CATALOG.MCATZ.PHXHQ
IEA009I SYMBOLIC DEFINITIONS WILL BE READ FROM: 025
      IEASYM00
      IEASYM28
IEE252I MEMBER IEASYM00 FOUND IN SYS2.PARMLIB
IEE252I MEMBER IEASYM28 FOUND IN SYS2.PARMLIB
IEA008I SYSTEM PARMS FOLLOW FOR z/OS 02.02.00 HBB77A0 028
      IEASYS28
      IEASYS02
IEE252I MEMBER IEASYS00 FOUND IN SYS2.PARMLIB
IEA325I IEASYS00 PARAMETER LIST
ALLOC=00, Allocation policies
CEE=(00,L), Language Environment Options
CLPA, Create Link Pack Area
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

Private Virtual Storage Exhaustion Health Check



- This health check is delivered as part of the Predictive Failure Analysis component. The name of the new check is PFA_PRIVATE_STORAGE_EXHAUSTION.
- It analyzes samples looking for private area spike, leak and creep in a manner similar to the existing PFA check for common storage exhaustion.
- We don't run PFA anymore, so I was unable to try this. ☹️
- The most fun thing for me is the new LDAX control block, provided by MVS, which contains this information.
- Anyone: IBM, customers, ISV products, etc. can access the information within this control block. Even unauthorized code can get to it because it exists in non-fetch-protected common storage.
- For monitor products, this means no more need for an SRB into every address space to examine its LDA control block.

Private Virtual Storage Exhaustion Health Check

- The new LDAX control block exists in the Nucleus for ASID=1, in SP 245 for address spaces with RCEInitialized off, and in 64bit common for all other address spaces
 - Does REXX STORAGE function support 64-bit pointers?

LDAX_LDAASCB	DS	A	Address of ASCB FOR THIS ADDRESS SPACE
LDAX_LDASTRTA	DS	A	Low addr of Region
LDAX_LDASIZA	DS	F	Size of Region
LDAX_LDAESTRA	DS	A	Low addr of extended Region
LDAX_LDAESIZA	DS	F	Size of extended Region
LDAX_LDACRGTP	DS	A	Current high address of PRIVATE AREA Region
LDAX_LDAERGTP	DS	A	Current high address of EXTENDED PRIVATE AREA Region
LDAX_LDALIMIT	DS	A	< 16M V=V Region limit value
LDAX_LDAVVRG	DS	A	< 16M V=V Region high value
LDAX_LDAELIM	DS	A	> 16M V=V Region limit value
LDAX_LDAEVVRG	DS	A	> 16M V=V Region high value
LDAX_LDALOAL	DS	F	< 16M USER Region alloc value
LDAX_LDAHIAL	DS	F	< 16M AUTH Region alloc value
LDAX_LDAELOAL	DS	F	> 16M USER Region alloc value
LDAX_LDAEHIAL	DS	F	> 16M AUTH Region alloc value
LDAX_TCTVALUS	DS	CL16	TCT Water marks
	ORG		LDAX_TCTVALUS
LDAX_TCTHWM	DS	F	< 16M max virt stor in SWA and LSQA
LDAX_TCTLWM	DS	F	< 16M max virt stor in user subpools
LDAX_TCTEHWM	DS	F	> 16M max virt stor in SWA and LSQA
LDAX_TCTELWM	DS	F	> 16M max virt stor in user subpools
LDAX_CurHighBot	DS	A	< 16M Cur bot Auth area
LDAX_CurEHighBot	DS	A	> 16M Cur bot EAuth area
LDAX_LDASMD	DS	A	Storage Management Area
LDAX_LDASMSZ	DS	A	Storage Management Size

BCPii Logging Facilities

- BCPii can cause unwanted disruptions if misused, either by accident or on purpose.
- It now generates SMF 106.1 records to record HWISET activity and SMF 106.2 records to record HWICMD activity.

SMF6ACTP DS	H	Connect type from HWICONN	} Common
SMF6ACPC DS	CL17	CPC Name	
SMF6ANLL DS	CL1	Word boundary use	
SMF6ARTN DS	CL16	Request parameter	
SMF6AASD DS	H	2 Byte Address space ID ASID	
SMF6AJOB DS	CL8	Job name	
SMF6AUSR DS	CL8	User name	
SMF6ATYP DS	F	Set Type	} HWISET only
SMF6ATVL DS	F	Set Type Value Len	
SMFSETDA DS	CL257	Set Parameter Data	
SMF6ACTY DS	F	Command type	} HWICMD only
SMF6AAOS DS	F	CMD Parameter Data Offset	
SMFCMDPM DS	CL328	CMD Passed in Parameter.	

BCPii Logging Facilities (*continued...*)

- Unlike the vast majority of z/OS components, the BCPii folks deliver JCL to format their SMF records in a friendly, human-consumable way.
 - I highly commend this above-and-beyond approach!
- Unfortunately, the report is formatted using ICETOOL
 - An extra cost option, but most everyone pays for SORT
 - Hopefully, alternative SORT products support this as well
 - IBM's ICETOOL is limited to a maximum of twenty HEADER statements.
- Presumably, MXG and similar SMF reporting products will also provide support for the new BCPii records.

BCPii Logging Facilities (continued...)

- The JCL can be found in member HWI6AFMT in SYS1.SAMPLIB
- Here are the statements that format the HWISET summary report from the SMF 106.1 records.
- Note how some available fields are commented out due to ICETOOL restrictions.

```
*****
* RPTSUM01:
* PRODUCE A SUMMARY REPORT OF SUBTYPE 1 RECORDS
* - ICETOOL HAS A MAXIMUM OF 20 FIELDS TO DISPLAY
*****
DISPLAY FROM(SMF01SUM) LIST(RPTSUM01) -
  TITLE(' SUBTYPE 1 RECORDS SUMMARY REPORT') DATE TIME PAGE -
  BLANK -
  BTITLE('CPC Name') BREAK(SMF6ACPC) -
  HEADER('LEN') ON(SMF6ALEN) -
  HEADER('SEG') ON(SMF6ASEG) -
  HEADER('FLG') ON(SMF6AFLG,HEX) -
  HEADER('RTY') ON(SMF6ARTY,U03) -
  HEADER('TME') ON(SMF6ATME,TM1,E'99:99:99',NOST) -
  HEADER('DTE') ON(SMF6ADTE,DT1,E'9999/99/99',NOST) -
  HEADER('SID') ON(SMF6ASID) -
  HEADER('WID') ON(SMF6AWID) -
  HEADER('STP') ON(SMF6ASTP) -
* HEADER('SDL') ON(SMF6ASDL) -
* HEADER('POF') ON(SMF6APOF) -
* HEADER('PLN') ON(SMF6APLN) -
* HEADER('PON') ON(SMF6APON) -
* HEADER('EOF') ON(SMF6AEOF) -
* HEADER('ELN') ON(SMF6AELN) -
* HEADER('EON') ON(SMF6AEON) -
* HEADER('DOF') ON(SMF6ADOF) -
* HEADER('DLN') ON(SMF6ADLN) -
* HEADER('DON') ON(SMF6ADON) -
  HEADER('VERSION') ON(SMF6ARVN) -
  HEADER('PROD NAME') ON(SMF6APNM) -
  HEADER('MVS PROD') ON(SMF6AOSL) -
  HEADER('SYSTEM NAME') ON(SMF6ASYN) -
  HEADER('CONN TYPE') ON(SMF6ACTP) -
  HEADER('CPC') ON(SMF6ACPC) -
  HEADER('RTN') ON(SMF6ARTN) -
  HEADER('ASD') ON(SMF6AASD) -
  HEADER('JOBNAME') ON(SMF6AJOB) -
  HEADER('USER') ON(SMF6AUSR) -
  HEADER('SET TYPE') ON(SMF6ATYP) -
  HEADER('SET PARM LEN') ON(SMF6ATVL) -
  HEADER('TDA') ON(SMF6TDA,HEX)
```

PER SLIP Traps Should Capture the BEAR

- When I first saw this requirement, I was confused.
- How is it possible the BEAR is *not* being captured?
- It turns out the BEAR is indeed always being captured in LCCABEA2 as IPCS CBF address STR(LCCA) shows:

```
+0448 BEA1..... 00000000 00000000          BEA2..... 00000000 00FE7ECC
+0458 BEA3..... 00000000 00000000
```

- The requirement was satisfied simply by formatting the BEAR on the message produced by SLIP with A=WAIT
- **NOTE:** is that the BEA fields in the LCCA were relocated between z/OS 2.1 and z/OS 2.2
 - Moved as part of SIMD support
- Be sure to use the right IPCS level when inspecting them.

PER SLIP Traps Should Capture the BEAR



```
00: HCPGIR450W CP entered; disabled wait PSW 00020000 80000000 00000000 0000001B
00: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from
CPU 02.
01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop from
CPU 02.
01: HCPGIR450W CP entered; disabled wait PSW 00020000 80000000 00000000 0000001B
02: HCPPCM6533A Following is a priority message received by the service proces
sor - use the VINPUT command to respond:
02: *IEE844W SLIP TRAP 0003 MATCHED. ACTION=WAIT TYPE=PER
02: PER INFO: 40F0 00000000_0CB65116 BEAR: 00000000_7F32C212
02: PSW: 47142001 80000000 00000000 0CB6511C
02: CR 3-4: 40000022 00000022
02: AR/GR 0: 00000000/FFFFFFFF_FFFFF000 1: 00000000/00000000_7F44A800
02: 2: FFFFFFFFF/FFFFFFFF_FFFFF000 3: FFFFFFFFF/FFFFFFFF_FFFFF000
02: 4: FFFFFFFFF/FFFFFFFF_FFFFF000 5: FFFFFFFFF/FFFFFFFF_FFFFF000
02: 6: FFFFFFFFF/FFFFFFFF_FFFFF000 7: FFFFFFFFF/FFFFFFFF_FFFFF000
02: 8: FFFFFFFFF/FFFFFFFF_FFFFF000 9: FFFFFFFFF/FFFFFFFF_FFFFF000
02: A: FFFFFFFFF/FFFFFFFF_FFFFF000 B: FFFFFFFFF/00000000_7F32C100
02: C: FFFFFFFFF/00000000_7F32D100 D: FFFFFFFFF/FFFFFFFF_FFFFF000
02: E: 00000000/00000000_00000000 F: 00000000/00000000_0CB66F00
02: RESTART THE SYSTEM TO CONTINUE
02: *
```

HOLDING VM80

Supercool SLIP “Thingy” Discovered

- While researching the SLIP/BEAR issue, I found that SLIP in z/OS 2.2 allows MVS commands to be issued when a SLIP trap matches.
- Any command?! WOW!! I can think of *a million* cool uses!

```
SLIP SET,IF, ID=SHAR, J=EDJAFFE, PVTMOD=(EJESML,116), MODE=HOME, A=CMD, CMD='D T', END
```

```
IEE727I SLIP TRAP ID=SHAR SET
```

```
IEA992I SLIP TRAP ID=SHAR MATCHED. JOBNAME=EDJAFFE , ASID=0034.
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.55 DATE=2015.212 UTC: TIME=23.16.55 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.56 DATE=2015.212 UTC: TIME=23.16.56 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.56 DATE=2015.212 UTC: TIME=23.16.56 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.57 DATE=2015.212 UTC: TIME=23.16.57 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.58 DATE=2015.212 UTC: TIME=23.16.58 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.59 DATE=2015.212 UTC: TIME=23.16.59 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.16.59 DATE=2015.212 UTC: TIME=23.16.59 DATE=2015.212
```

```
D T
```

```
IEE136I LOCAL: TIME=16.17.00 DATE=2015.212 UTC: TIME=23.17.00 DATE=2015.212
```

```
SLIP DEL, ID=SHAR
```

```
IEE727I SLIP TRAP ID=SHAR DELETED
```

Which zFS is Quiesced in IOEZ00581E?

- According to the response in the RFE:

```
APAR OA44214 has the code which will issue the message 830 with the name of one quiesced aggregate. The explanation text for the message explains that a complete list of quiesced aggregates can be obtained by issuing the modify command f zfs,query,file,quiesced.
```
- The z/OS 2.1 PTF for OA44214 is UA72508, which is superseded by UA76562. Both have been installed on my systems for some time, yet I never saw the name of any quiesced aggregate when IOEZ00581E is produced.
 - **UPDATE!** According to IBM, the ‘830’ message is produced only when there is 30 seconds or more of *actual* contention.
- The F ZFS,QUERY,FILE QUIESCED command *does* produce a list of quiesced aggregates.
- I’m considering using MPF (or automation) to issue this command when IOEZ00581E is issued.

Supercool zFS “Thingy” Discovered

- While researching this issue, I found a new MODIFY ZFS command in z/OS 2.2!

F ZFS,FSINFO

IOEZ00836I zFS kernel: MODIFY command - FSINFO accepted.

IOEZ00670I Starting FSINFO command.

IOEZ00850I File System Status: 011

AEW11A.SAEWZFS	MVS60	RW,NS
CICSTS51.ZFS	MVS70	RW,NS,L
CICSTS52.ZFS	MVS60	RW,NS

...

... (lots of other file systems listed)

...

HQOMVS.PHXHQ.LOCAL.ZFS	MVS60	RW,NS
HQOMVS.PHXHQ.ROOT.ZFS	MVS60	RW,NS
JVA601.ZFS	MVS60	RO
JVA700.MVSA0.ZFS	MVSA0	RO
JVA700.ZFS	MVS60	RO
JVB601.ZFS	MVS60	RO
JVB700.MVSA0.ZFS	MVSA0	RO
JVB700.ZFS	MVS60	RO
OMVS.VERSYSB.VERSION.ZFS	MVSA0	RO
SYS2.EJES540.ZFS	MVS70	RW,NS
SYS3.HKC.HKCCFGFZ	MVS60	RW,NS,L
SYS3.HKC.HKCDATFZ	MVS70	RW,NS
SYS3.HKC.HKCLOGFZ	MVSA0	RW,NS
ZFS.Z21Z.VERSION	MVS70	RW,NS

Legend: RW=Read-write,NS=Mounted NORWSHARE,L=Low on space

RO=Read-only

IOEZ00849I FSINFO command done.

MODIFY ZFS,FSINFO Command

- An MVS command alias of the zfsadm fsinfo command
- I did not have the documentation for this command, but I found OA48385 (using Google) which suggested issuing:
 - F ZFS,FSINFO,ALL,FULL,SELECT=Q
- The complete description was behind a firewall, but I quickly inferred from the syntax that ALL,FULL provided “verbose” output that SELECT=Q would consider quiesced file systems only.
- I issued F ZFS,FSINFO,ALL,FULL,SELECT=RW to get this information for my file systems mounted read/write and it worked!!!
- **UPDATE!** z/OS 2.2 doc now publicly available from:
 - <http://www-03.ibm.com/systems/z/os/zos/library/bkserv/v2r2pdf/>

MODIFY ZFS,FSINFO Command

F ZFS,FSINFO,ALL,FULL,SELECT=RW

IOEZ00836I zFS kernel: MODIFY command - FSINFO,ALL,FULL,SELECT=RW accepted.

IOEZ00670I Starting FSINFO command.

IOEZ00850I File System Status:

File System Name: HQOMVS.MVSA0.Z22Z.WBEM.ZFS.

*** owner information ***

```
Owner:           MVSA0           Converttov5:     OFF,n/a
Size:            108000K         Free 8K Blocks:  4595
Free 1K Frags:   7             Log File Size:   1080K
Bitmap Size:     16K            Anode Table Size: 1064K
File System Objs: 4212          Version:         1.4
Overflow Pages:  0             Overflow HiWater: 0
Thrashing Objs:  0             Thrashing Resol: 0
Token Revc:      0             Revc WaitTime:   0.000
Devno:           6155           Space Monitoring: 0,0
Quiescing Sys:   n/a           Quiescing Job:   n/a
Quiescor ASID:   n/a           File System Grow: ON,0
Status:          RW,NS
Audit Fid:       E5D7C9E9 E4C60006 0000
```

```
File System Creation Time: Jul 29 21:26:28 2013
Time of Ownership:        Jul 25 22:43:14 2015
Statistics Reset Time:    Jul 25 22:43:14 2015
Quiesce Time:             n/a
Last Grow Time:           n/a
Connected Clients:        n/a
```

*** local data from system MVSA0 (owner: MVSA0) ***

```
Vnodes:          40             LFS Held Vnodes:  3
Open Objects:    2             Tokens:            0
User Cache 4KPGs: 122          Meta Cache 8KPGs: 110
App. Rds:        4305          Avg. Rd RespTime: 0.058
App. Wrs:        21            Avg. Wr RespTime: 0.460
Rd XCF Calls:    0             Avg. Rd XCF RespTime: 0.000
Wr XCF Calls:    0             Avg. Wr XCF RespTime: 0.000
ENOSPC Errors:   0             Disk IO Errors:   0
XCF Comm. Errors: 0           Canceled Ops:     0
```

```
DDNAME:          SYS00015
Mount Time:      Jul 25 22:43:14 2015
```

VOLSER	Reads	KBytes	Writes	KBytes	Waits	Average
MVSUFS	135	1384	10947	43804	146	1.067
TOTALS	135	1384	10947	43804	146	1.067

- SELECT=RW produced the info you see for all of the read/write zFS on my system.
- This info includes location, size, owner, time stamps, response time statistics, options in effect, etc.
- Spec-freaking-tacular! 😊

SYS1.LOGREC Reallocated without IPL

- I have wanted this since *Old Man Noah* cornered the market on gopher wood!
- The SETLOGRC command has been enhanced to allow specification of a new data set or log stream name.

D LOGREC

```
IFB090I 09.55.26 LOGREC DISPLAY
CURRENT MEDIUM = DATASET
MEDIUM NAME = SYS2.MVSA0.LOGREC
```

```
//IFCDIP00 JOB 1, 'INITIALIZE LOGREC', CLASS=S, NOTIFY=&SYSUID
/*JOBPARM S=SA0
//*MAIN SYSTEM=MVSA0
//NEWDIP EXEC PGM=IFCDIP00, COND=(8,LT)
//SERERDS DD DSN=SYS2.MVSA0.LOGREC1, DISP=(,CATLG),
//          VOLUME=(,RETAIN, SER=MVSA0B), UNIT=3390,
//          SPACE=(CYL, (30))
```

SETLOGRC DATASET=SYS2.MVSA0.LOGREC1

```
IEF196I IEF237I 8207 ALLOCATED TO SYS00009
IFB097I LOGREC RECORDING MEDIUM CHANGED FROM DATASET TO DATASET
```

D LOGREC

```
IFB090I 10.00.31 LOGREC DISPLAY
CURRENT MEDIUM = DATASET
MEDIUM NAME = SYS2.MVSA0.LOGREC1
```

JCTJOBID and Sysplex in SMF Types 14/15

- We have NOTYPE(14:16) in our SMFPRMxx member.
- Obviously, someone cares about this. Maybe you?
- See below for the relevant updates to IFGSMF14
- Note the questionable choice of field names. ☹️

```
.DSECT9 ANOP @LDA
* @L2A
* THIS DESCRIBES THE STEP INFORMATION SECTION OF THE @L2A
* EXTENDED INFORMATION SEGMENT. THE JOB INFORMATION @L2A
* HAS A TYPE CODE OF 3. IT CONTAINS THE STEP NAME, @L2A
* JOB ID AND SYSPLEX NAME OF THE JOB, PLUS THE @P9A
* ACTIVE PROGRAM NAME OF THE JOB. @L2A
* @L2A
AIF ('&DSECT' EQ 'NO').NODSECT10 @LDA
&FRSTSYM.SPI DSECT @LDA
.NODSECT10 ANOP @LDA
SMF14SPI EQU * STEP INFORMATION SECTION @L2A
SMF14SPN DS CL8' ' STEP NAME @L2A
SMF14PGN DS CL8' ' ACTIVE PROGRAM NAME @L2A
SMFJOBID DS CL8' ' JOBID @P9A
SMFPLXID DS CL8' ' SYSPLEXID @P9A
SMF14SPE EQU * END OF STEP INFORMATION SECTION @L2A
```


Dynamic APF List SMF Record

- New member CSVAPSMF
- This extremely robust mapping macro has Peter Relson's fingerprints all over it! 😊

```

SMF90T37          DSECT          SMF record (type 90 subtype 37)      *
                  data                                             *
SMF90T37Function  DS      X      See equates SMF90T37Add,          *
                  SMF90T37Delete, SMF90T37DynFormat,          *
                  SMF90T37StatFormat                          *
SMF90T37Flags     DS      B      Flags                               *
* Bit definitions:
SMF90T37_SETPROG  EQU      X'80'  Update via SETPROG command      *
SMF90T37_SET_PROG EQU      X'40'  Update via SET PROG command      *
SMF90T37_CSVAPF   EQU      X'20'  Update via CSVAPF macro          *
SMF90T37ParmMemSuffix DS      CL2  When SMF90T37_SET_PROG, the PROGxx *
                  parmlib member name's "xx".                    *
SMF90T37Dsname    DS      CL44   The data set name when function is *
                  not "DynFormat" or "StatFormat"                *
                  (otherwise undefined)                            *
SMF90T37Volume    DS      CL6    The volume when function is not   *
                  "DynFormat" or "StatFormat"                    *
                  (otherwise undefined). This will be             *
                  zeroes for an add or delete                     *
                  indicating SMS-managed (such as by               *
                  SMS on SETPROG APF ADD)                          *
SMF90T37TimeStamp DS      CL2    Reserved                          *
                  DS      CL8    Time value (via STCK) of the      *
                  update                                           *
...
... (continued on the right)
...
SMF90T37Jobname   DS      CL8    The jobname of issuer of the APF *
                  request. For a SETPROG command or              *
                  SET PROG processing, this will be              *
                  the job name of the ASID processing            *
                  the command (ASID 1 - *MASTER*)                *
SMF90T37CHKKey    DS      CL8    The CHKKEY field from the CSCB (for *
                  a started task, this is the                     *
                  stepname). Zeroes if there is no                *
                  CSCB.                                           *
SMF90T37Consid    DS      F      Console ID of issuer of the APF *
                  request. The value is -1 if the                 *
                  request was via the CSVAPF macro                *
                  Reserved                                        *
SMF90T37UToken    DS      CL80   Security product user token of   *
                  issuer of the APF request                       *
SMF90T37Add       EQU      1     This record is a result of ADD to *
                  APF                                             *
SMF90T37Delete    EQU      2     This record is a result of DELETE *
                  from APF                                        *
SMF90T37DynFormat EQU      4     This record is a result of changing *
                  to dynamic format                               *
SMF90T37StatFormat EQU      6     This record is a result of changing *
                  to static format                               *
SMF90T37_Len      EQU      *-SMF90T37
    
```

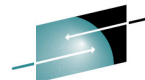
Additional Exploration

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

More JES3 Function Rolled Back to JES2



- In z/OS 2.1, JES2 added 8-character job class names, job class groups, and pre-execution C/I processing.
- In z/OS 2.2, JES2 adds Dependent Job Control and Deadline Scheduling functions, and raises the number of jobs on SPOOL from 400K to one million like JES3.
- We never approach a million jobs and have never used JES3's Deadline Scheduling feature, but we do use DJC *extensively*.
- All of our product builds are DJC networks. To learn about JES2's new function in this area, we re-engineered one of our DJC networks to work under z/OS 2.2 JES2.



Existing JES3 DJC Network



- This simple, yet effective, DJC Network was first developed in 1988
- Two jobs run sequentially
 - They run *very* quickly
- Five jobs run in parallel
 - These are HLASM steps that run for approximately 1/2 hour on our system.
- Three jobs run sequentially
 - They run for just a couple/few minutes each

Existing JES3 DJC JCL

```
//EJES00 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, RL=EJES10  
//*MAIN CLASS=GENERATE, LINES=(100, W)
```

```
//EJES10 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=(EJES20, EJES21, EJES22, EJES23, EJES24)  
//*MAIN CLASS=GENERATE, LINES=(100, W)
```

```
//EJES20 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES30  
//*MAIN CLASS=GENERATE, LINES=(300, W)
```

```
//EJES21 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES30  
//*MAIN CLASS=GENERATE, LINES=(300, W)
```

```
//EJES22 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES30  
//*MAIN CLASS=GENERATE, LINES=(300, W)
```

```
//EJES23 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES30  
//*MAIN CLASS=GENERATE, LINES=(600, W)
```

```
//EJES24 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES30  
//*MAIN CLASS=GENERATE, LINES=(600, W)
```

```
//EJES30 JOB 1, 'EJES GENERATION', REGION=512M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=5, RL=EJES40  
//*MAIN CLASS=GENERATE, LINES=(100, W)
```

```
//EJES40 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1, RL=EJES50  
//*MAIN CLASS=GENERATE, LINES=(150, W)
```

```
//EJES50 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES  
//*NET ID=EJESGENR, NHOLD=1  
//*MAIN CLASS=GENERATE, SYSTEM=MVS70
```

- EJES00 releases EJES10
- EJES10 releases EJES20, 21, 22, 23 and 24
- EJES20, 21, 22, 23 and 24 release EJES30
- EJES30 releases EJES40
- EJES40 releases EJES50

Equivalent JES2 JCL

```
//EJESGENR JOBGROUP OWNER=EJES
//EJES00 GJOB
//
//EJES10 GJOB
//
//EJES10 BEFORE NAME=EJES10
//
//EJES2SET JOBSET
//
//EJES20 SJOB
//EJES21 SJOB
//EJES22 SJOB
//EJES23 SJOB
//EJES24 SJOB
//EJES2SET ENDSET
//EJES30 GJOB
//
//EJES40 GJOB
//
//EJES50 GJOB
//EJES50 BEFORE NAME=EJES50
//
//EJESGENR ENDGROUP
//EJES00 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(100, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES10 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(100, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES20 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(300, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES21 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(300, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES22 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(300, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES23 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(600, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES24 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(600, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES30 JOB 1, 'EJES GENERATION', REGION=512M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(100, WARNING), TIME=NOLIMIT
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES40 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A, LINES=(150, WARNING)
//
// SCHEDULE JOBGROUP=EJESGENR
//EJES50 JOB 1, 'EJES GENERATION', REGION=256M, MSGCLASS=E, USER=EJES,
//
// CLASS=A
//
// SCHEDULE JOBGROUP=EJESGENR
```

Job Set

Job Group

Jobs

- More JCL statements required than for JES3
- Must define job group, job sets, etc. to describe the processing order.
- *Then* supply the jobs themselves and associate each with job group via SCHEDULE JCL

DJC Experiments

EXPERIMENT #1

- Removed all product build steps. Inserted a single IEFBR14 for every job except EJES23.
- EJES23 runs a program called WAIT4REP which issues a WTOR and waits.
- This is how I hope to prove EJES30 is really waiting for the “job set” to complete.

EXPERIMENT #2

- Removed all product build steps. Inserted a single IEFBR14 for every job except EJES23.
- EJES23 runs a program called ABEND806 which issues does not exist.
- This is how I hope to prove I can restart the network after a failure.

DJC Experiment #1 – JES3 Results



```
M7      2015213 1306352 T2SY2 R= EJES23   IAT2000 JOB EJES23   (JOB01171) SELECTED T2SY2   SRVCLASS=
M7      2015213 1306352 T2SY2 R= EJES23   NET-ID=EJESGENR
      2015213 1306353 T2SY2 R= EJES23   IEF403I EJES23 - STARTED - TIME=13.06.35
16      2015213 1306353 *T2SY2 R= EJES23   0251 PHX0001 Job paused, reply to continue
      EDJAFFE 2015213 1307170 -R 251,HELLO
16      EDJAFFE 2015213 1307170 T2SY2 R=          IEE600I REPLY TO 0251 IS;HELLO
      2015213 1307171 T2SY2 R= EJES23   IEF404I EJES23 - ENDED - TIME=13.07.17
MLG     2015213 1307171 T2SY2 R= EJES23   -EJES23 ENDED. NAME-EJES GENERATION   TOTAL TCB CPU TIME=
MLG     2015213 1307171 IAT7100 (MAIN    ) *S DJCUPDAT,EJES23 ,EJESGENR,1
INTERNAL2015213 1307171 +S DJCUPDAT,EJES23 ,EJESGENR,1
M7      2015213 1307171 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1307172 T2SY2 R= EJES30   IAT2000 JOB EJES30   (JOB01173) SELECTED T2SY2   SRVCLASS=
M7      2015213 1307172 T2SY2 R= EJES30   NET-ID=EJESGENR
      2015213 1307172 T2SY2 R= EJES30   IEF403I EJES30 - STARTED - TIME=13.07.17
      2015213 1307172 T2SY2 R= EJES30   IEF404I EJES30 - ENDED - TIME=13.07.17
MLG     2015213 1307173 T2SY2 R= EJES30   -EJES30 ENDED. NAME-EJES GENERATION   TOTAL TCB CPU TIME=
MLG     2015213 1307173 IAT7100 (MAIN    ) *S DJCUPDAT,EJES30 ,EJESGENR,1
INTERNAL2015213 1307173 +S DJCUPDAT,EJES30 ,EJESGENR,1
M7      2015213 1307173 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1307174 T2SY2 R= EJES40   IAT2000 JOB EJES40   (JOB01175) SELECTED T2SY2   SRVCLASS=
M7      2015213 1307174 T2SY2 R= EJES40   NET-ID=EJESGENR
      2015213 1307174 T2SY2 R= EJES40   IEF403I EJES40 - STARTED - TIME=13.07.17
      2015213 1307174 T2SY2 R= EJES40   IEF404I EJES40 - ENDED - TIME=13.07.17
MLG     2015213 1307174 T2SY2 R= EJES40   -EJES40 ENDED. NAME-EJES GENERATION   TOTAL TCB CPU TIME=
MLG     2015213 1307175 IAT7100 (MAIN    ) *S DJCUPDAT,EJES40 ,EJESGENR,1
INTERNAL2015213 1307175 +S DJCUPDAT,EJES40 ,EJESGENR,1
M7      2015213 1307175 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1307175 T2SY2 R= EJES50   IAT2000 JOB EJES50   (JOB01176) SELECTED T2SY2   SRVCLASS=
M7      2015213 1307175 T2SY2 R= EJES50   NET-ID=EJESGENR
      2015213 1307176 T2SY2 R= EJES50   IEF403I EJES50 - STARTED - TIME=13.07.17
      2015213 1307176 T2SY2 R= EJES50   IEF404I EJES50 - ENDED - TIME=13.07.17
MLG     2015213 1307176 T2SY2 R= EJES50   -EJES50 ENDED. NAME-EJES GENERATION   TOTAL TCB CPU TIME=
MLG     2015213 1307176 IAT7100 (MAIN    ) *S DJCUPDAT,EJES50 ,EJESGENR,1
INTERNAL2015213 1307177 +S DJCUPDAT,EJES50 ,EJESGENR,1
M7      2015213 1307177 IAT2001 OS MVS WAITING FOR WORK - T2SY2
LOG     2015213 1307177 IAT7310 NET EJESGENR HAS COMPLETED
LOG     2015213 1307177 IAT7315 DJC PROCESSING HAS COMPLETED
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OS 2.2 User Experiences



DJC Experiment #1 – JES2 Results



```
N 40C0000 T2SY1      2015213 13:23:07.26 J0033352 00000090 $HASP1301 EJES23 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 13:23:07.39 J0033352 00000281 $HASP373 EJES23   STARTED - INIT 5   - CLASS A       - SYS SY1
N 0000000 T2SY1      2015213 13:23:07.42 J0033352 00000281 IEF403I EJES23 - STARTED - TIME=13.23.07
W 0001000 T2SY1      2015213 13:23:07.58 J0033352 00000090 @0252 PHX0001 Job paused, reply to continue
NC0000000 T2SY1      2015213 13:28:08.29 EDJAFFE 00000290 252,HELLO
NR0001000 T2SY1      2015213 13:28:08.32 J0033352 00000090 IEE600I REPLY TO 0252 IS;HELLO
N 0000000 T2SY1      2015213 13:28:08.40 J0033352 00000000 IEF404I EJES23 - ENDED - TIME=13.28.08
N 0004000 T2SY1      2015213 13:28:08.41 J0033352 00000290 -EJES23   ENDED.   NAME-EJES GENERATION       TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 13:28:08.46 J0033352 00000281 $HASP395 EJES23   ENDED - RC=0000
N C000000 T2SY1      2015213 13:28:08.49                00000281 $HASP309 INIT 5   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 13:28:08.50 J0033354 00000090 $HASP1301 EJES30 in job group EJESGENR queued for execution
N 0020000 T2SY1      2015213 13:28:08.54 J0033354 00000281 ICH70001I EJES   LAST ACCESS AT 13:23:07 ON SATURDAY, AUGUST 1, 2015
N 4000000 T2SY1      2015213 13:28:08.55 J0033354 00000281 $HASP373 EJES30   STARTED - INIT 1   - CLASS A       - SYS SY1
N 0000000 T2SY1      2015213 13:28:08.57 J0033354 00000281 IEF403I EJES30 - STARTED - TIME=13.28.08
N 0000000 T2SY1      2015213 13:28:08.61 J0033354 00000000 IEF404I EJES30 - ENDED - TIME=13.28.08
N 0004000 T2SY1      2015213 13:28:08.61 J0033354 00000290 -EJES30   ENDED.   NAME-EJES GENERATION       TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 13:28:08.66 J0033354 00000281 $HASP395 EJES30   ENDED - RC=0000
N C000000 T2SY1      2015213 13:28:08.69                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 13:28:08.69 J0033355 00000090 $HASP1301 EJES40 in job group EJESGENR queued for execution
N 0020000 T2SY1      2015213 13:28:08.73 J0033355 00000281 ICH70001I EJES   LAST ACCESS AT 13:28:08 ON SATURDAY, AUGUST 1, 2015
N 4000000 T2SY1      2015213 13:28:08.74 J0033355 00000281 $HASP373 EJES40   STARTED - INIT 1   - CLASS A       - SYS SY1
N 0000000 T2SY1      2015213 13:28:08.75 J0033355 00000281 IEF403I EJES40 - STARTED - TIME=13.28.08
N 0000000 T2SY1      2015213 13:28:08.78 J0033355 00000000 IEF404I EJES40 - ENDED - TIME=13.28.08
N 0004000 T2SY1      2015213 13:28:08.78 J0033355 00000290 -EJES40   ENDED.   NAME-EJES GENERATION       TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 13:28:08.84 J0033355 00000281 $HASP395 EJES40   ENDED - RC=0000
N C000000 T2SY1      2015213 13:28:08.86                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 13:28:08.86 J0033356 00000090 $HASP1301 EJES50 in job group EJESGENR queued for execution
N 0020000 T2SY1      2015213 13:28:08.92 J0033356 00000281 ICH70001I EJES   LAST ACCESS AT 13:28:08 ON SATURDAY, AUGUST 1, 2015
N 4000000 T2SY1      2015213 13:28:08.93 J0033356 00000281 $HASP373 EJES50   STARTED - INIT 1   - CLASS A       - SYS SY1
N 0000000 T2SY1      2015213 13:28:08.93 J0033356 00000281 IEF403I EJES50 - STARTED - TIME=13.28.08
N 0000000 T2SY1      2015213 13:28:08.96 J0033356 00000000 IEF404I EJES50 - ENDED - TIME=13.28.08
N 0004000 T2SY1      2015213 13:28:08.96 J0033356 00000290 -EJES50   ENDED.   NAME-EJES GENERATION       TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 13:28:09.06 J0033356 00000281 $HASP395 EJES50   ENDED - RC=0000
N C000000 T2SY1      2015213 13:28:09.49                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 13:28:09.51 G0033346 00000090 $HASP1304 job group EJESGENR is complete
```

DJC Experiment #2 – JES3 Results



```
M7      2015213 1417063 T2SY2 R= EJES23 IAT2000 JOB EJES23 (JOB01184) SELECTED T2SY2 SRVCLASS=
M7      2015213 1417063 T2SY2 R= EJES23 NET-ID=EJESGENR
      2015213 1417064 T2SY2 R= EJES23 IEF403I EJES23 - STARTED - TIME=14.17.06
SEC     2015213 1417065 T2SY2 R= EJES23 CSV028I ABEND806-04 JOBNAME=EJES23 STEPNAME=
MLG     2015213 1417065 T2SY2 R= IEA989I SLIP TRAP ID=X806 MATCHED. JOBNAME=EJES23 , ASID=002C.
      2015213 1417066 T2SY2 R= EJES23 IEF450I EJES23 - ABEND=S806 U0000 REASON=00000004
      2015213 1417066 T2SY2 R= EJES23 TIME=14.17.06
      2015213 1417066 T2SY2 R= EJES23 IEF404I EJES23 - ENDED - TIME=14.17.06
M7      2015213 1417067 IAT2001 OS MVS WAITING FOR WORK - T2SY2
LOG     2015213 1418011 IAT6100 (JOB01182) JOB EJES23 (JOB01192), PRY=03, ID=EJES NET-ID=EJESGENR
LOG     2015213 1418011 IAT6100 SUB=JOB01108
LOG     2015213 1418011 IAT7450 JOB INTRDR (JOB01182) PURGED
M7      2015213 1418012 T2SY2 R= EJES23 IAT2000 JOB EJES23 (JOB01192) SELECTED T2SY2 SRVCLASS=
M7      2015213 1418012 T2SY2 R= EJES23 NET-ID=EJESGENR
      2015213 1418013 T2SY2 R= EJES23 IEF403I EJES23 - STARTED - TIME=14.18.01
      2015213 1418013 T2SY2 R= EJES23 IEF404I EJES23 - ENDED - TIME=14.18.01
MLG     2015213 1418013 T2SY2 R= EJES23 -EJES23 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
MLG     2015213 1418014 IAT7100 (MAIN ) *S DJCUPDAT,EJES23 ,EJESGENR,1
INTERNAL2015213 1418014 +S DJCUPDAT,EJES23 ,EJESGENR,1
M7      2015213 1418014 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1418014 T2SY2 R= EJES30 IAT2000 JOB EJES30 (JOB01187) SELECTED T2SY2 SRVCLASS=
M7      2015213 1418014 T2SY2 R= EJES30 NET-ID=EJESGENR
      2015213 1418015 T2SY2 R= EJES30 IEF403I EJES30 - STARTED - TIME=14.18.01
      2015213 1418015 T2SY2 R= EJES30 IEF404I EJES30 - ENDED - TIME=14.18.01
MLG     2015213 1418015 T2SY2 R= EJES30 -EJES30 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
MLG     2015213 1418015 T2SY2 R= EJES30 .00 TOTAL ELAPSED TIME= .0
MLG     2015213 1418015 IAT7100 (MAIN ) *S DJCUPDAT,EJES30 ,EJESGENR,1
INTERNAL2015213 1418015 +S DJCUPDAT,EJES30 ,EJESGENR,1
M7      2015213 1418016 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1418016 T2SY2 R= EJES40 IAT2000 JOB EJES40 (JOB01189) SELECTED T2SY2 SRVCLASS=
M7      2015213 1418016 T2SY2 R= EJES40 NET-ID=EJESGENR
      2015213 1418016 T2SY2 R= EJES40 IEF403I EJES40 - STARTED - TIME=14.18.01
      2015213 1418016 T2SY2 R= EJES40 IEF404I EJES40 - ENDED - TIME=14.18.01
MLG     2015213 1418017 T2SY2 R= EJES40 -EJES40 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
MLG     2015213 1418017 IAT7100 (MAIN ) *S DJCUPDAT,EJES40 ,EJESGENR,1
INTERNAL2015213 1418017 +S DJCUPDAT,EJES40 ,EJESGENR,1
M7      2015213 1418017 IAT2001 OS MVS WAITING FOR WORK - T2SY2
M7      2015213 1418018 T2SY2 R= EJES50 IAT2000 JOB EJES50 (JOB01191) SELECTED T2SY2 SRVCLASS=
M7      2015213 1418018 T2SY2 R= EJES50 NET-ID=EJESGENR
      2015213 1418018 T2SY2 R= EJES50 IEF403I EJES50 - STARTED - TIME=14.18.01
      2015213 1418018 T2SY2 R= EJES50 IEF404I EJES50 - ENDED - TIME=14.18.01
MLG     2015213 1418018 T2SY2 R= EJES50 -EJES50 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
MLG     2015213 1418019 IAT7100 (MAIN ) *S DJCUPDAT,EJES50 ,EJESGENR,1
INTERNAL2015213 1418019 +S DJCUPDAT,EJES50 ,EJESGENR,1
M7      2015213 1418019 IAT2001 OS MVS WAITING FOR WORK - T2SY2
LOG     2015213 1418019 IAT7310 NET EJESGENR HAS COMPLETED
LOG     2015213 1418019 IAT7315 DJC PROCESSING HAS COMPLETED
```

DJC Experiment #2 – JES2 Results



- Not at all what I expected! The job group kept running! ☹️

```
N 0000000 T2SY1      2015213 14:47:39.62 J0033366 00000281 IEF403I EJES23 - STARTED - TIME=14.47.39
N 0020000 T2SY1      2015213 14:47:39.83 J0033366 00000281 CSV003I REQUESTED MODULE ABEND806 NOT FOUND
N 00A0000 T2SY1      2015213 14:47:39.84 J0033366 00000090 CSV028I ABEND806-04 JOBNAME=EJES23 STEPNAME=
M 0000000 T2SY1      2015213 14:47:39.93 J0033366 00000000 IEF450I EJES23 - ABEND=S806 U0000 REASON=00000004 058
E                                058 00000000                                TIME=14.47.39

N 0000000 T2SY1      2015213 14:47:40.08 J0033366 00000000 IEF404I EJES23 - ENDED - TIME=14.47.40
N 0004000 T2SY1      2015213 14:47:40.08 J0033366 00000290 -EJES23 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 14:47:40.17 J0033366 00000281 $HASP395 EJES23 ENDED - ABEND=S806
N C000000 T2SY1      2015213 14:47:40.22                                00000281 $HASP309 INIT 5 INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 14:47:40.24 J0033368 00000090 $HASP1301 EJES30 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 14:47:40.31 J0033368 00000281 $HASP373 EJES30 STARTED - INIT 2 - CLASS A - SYS SY1
N 0000000 T2SY1      2015213 14:47:40.32 J0033368 00000281 IEF403I EJES30 - STARTED - TIME=14.47.40
N 0000000 T2SY1      2015213 14:47:40.35 J0033368 00000000 IEF404I EJES30 - ENDED - TIME=14.47.40
N 0004000 T2SY1      2015213 14:47:40.37 J0033368 00000290 -EJES30 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 14:47:40.42 J0033368 00000281 $HASP395 EJES30 ENDED - RC=0000
N C000000 T2SY1      2015213 14:47:40.44                                00000281 $HASP309 INIT 2 INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 14:47:40.45 J0033369 00000090 $HASP1301 EJES40 in job group EJESGENR queued for execution
N 0020000 T2SY1      2015213 14:47:40.50 J0033369 00000281 ICH70001I EJES LAST ACCESS AT 14:47:40 ON SATURDAY, AUGUST 1, 2015
N 4000000 T2SY1      2015213 14:47:40.51 J0033369 00000281 $HASP373 EJES40 STARTED - INIT 1 - CLASS A - SYS SY1
N 0000000 T2SY1      2015213 14:47:40.52 J0033369 00000281 IEF403I EJES40 - STARTED - TIME=14.47.40
N 0000000 T2SY1      2015213 14:47:40.55 J0033369 00000000 IEF404I EJES40 - ENDED - TIME=14.47.40
N 0004000 T2SY1      2015213 14:47:40.55 J0033369 00000290 -EJES40 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 14:47:40.60 J0033369 00000281 $HASP395 EJES40 ENDED - RC=0000
N C000000 T2SY1      2015213 14:47:41.01                                00000281 $HASP309 INIT 1 INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 14:47:41.04 J0033370 00000090 $HASP1301 EJES50 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 14:47:41.09 J0033370 00000281 $HASP373 EJES50 STARTED - INIT 1 - CLASS A - SYS SY1
N 0000000 T2SY1      2015213 14:47:41.10 J0033370 00000281 IEF403I EJES50 - STARTED - TIME=14.47.41
N 0000000 T2SY1      2015213 14:47:41.13 J0033370 00000000 IEF404I EJES50 - ENDED - TIME=14.47.41
N 0004000 T2SY1      2015213 14:47:41.13 J0033370 00000290 -EJES50 ENDED. NAME-EJES GENERATION TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 14:47:41.19 J0033370 00000281 $HASP395 EJES50 ENDED - RC=0000
N C000000 T2SY1      2015213 14:47:41.21                                00000281 $HASP309 INIT 1 INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 14:47:41.21 G0033360 00000090 $HASP1304 job group EJESGENR is complete
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Making Job Group Suspend after Failure

- The ERROR= keyword controls error handling
- Multiple conditions, such as ERROR=(RC>4 | ABEND)
- I updated my JOBGROUP statement and tried again

```
//EJESGENR JOBGROUP OWNER=EJES,ERROR=ABEND
```

- This time, the job group *suspended* after the abend! 😊
EJES30, 40 and 50 remained in the SETUP queue!

Jobs	Resources	Devices	Tools	Filter	View	Options	Help			
T2SY1 STATUS 468S 31X 434W 3H 0T 137,342 Records							Row 1 of 468			
Command ==>							Scroll ==> CSR			
FILTER=MaskChar=*%,PSelect=(Type=ALL),XSelect=(),Filter=(CTRUE Queue=SETUP)										
Cmd	JobName	JobID	Status	Queue	AMbr	JP	Pos	WPos	MaxComp	Records
RMULLIN	T0033399	ACTIVE	EXEC	SY1	15					0
EJES30	J0033396	QUEUED	SETUP		9	15				0
EJES40	J0033397	QUEUED	SETUP		9	16				0
EJES50	J0033398	QUEUED	SETUP		9	17				0
EJES24	J0033395	QUEUED	PRINT		1	410		CC 0000		34
EJES20	J0033390	QUEUED	PRINT		1	411		CC 0000		34
EJES21	J0033392	QUEUED	PRINT		1	412		CC 0000		34
EJES22	J0033393	QUEUED	PRINT		1	413		CC 0000		34
EJES23	J0033394	QUEUED	PRINT		1	414		AB S806		86
EJESGENR	G0033387	QUEUED	SETUP		9	14				0
EJES00	J0033388	QUEUED	PRINT		1	408		CC 0000		34
EJES10	J0033389	QUEUED	PRINT		1	409		CC 0000		34

New DJC Experiment #2 – JES2 Results



- EJES23 was resubmitted and all worked as expected. 😊

```
N 0200000 T2SY1      2015213 15:14:45.50 J0033385 00000281 $HASP100 EJES23   ON INTRDR      EJES GENERATION      FROM T0033332
S
N 40C0000 T2SY1      2015213 15:14:45.57 J0033385 00000090 $HASP1300 EJES23 registered to job group EJESGENR
N 40C0000 T2SY1      2015213 15:14:45.58 J0033385 00000090 $HASP1301 EJES23 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 15:14:45.65 J0033385 00000281 $HASP373 EJES23   STARTED - INIT 1   - CLASS A           - SYS SY1
N 0000000 T2SY1      2015213 15:14:45.66 J0033385 00000281 IEF403I EJES23 - STARTED - TIME=15.14.45
N 0000000 T2SY1      2015213 15:14:45.73 J0033385 00000000 IEF404I EJES23 - ENDED - TIME=15.14.45
N 0004000 T2SY1      2015213 15:14:45.74 J0033385 00000290 -EJES23   ENDED.      NAME-EJES GENERATION      TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 15:14:45.79 J0033385 00000281 $HASP395 EJES23   ENDED - RC=0000
N C000000 T2SY1      2015213 15:14:45.81                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 15:14:45.82 J0033382 00000090 $HASP1301 EJES30 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 15:14:45.88 J0033382 00000281 $HASP373 EJES30   STARTED - INIT 1   - CLASS A           - SYS SY1
N 0000000 T2SY1      2015213 15:14:45.89 J0033382 00000281 IEF403I EJES30 - STARTED - TIME=15.14.45
N 0000000 T2SY1      2015213 15:14:45.91 J0033382 00000000 IEF404I EJES30 - ENDED - TIME=15.14.45
N 0004000 T2SY1      2015213 15:14:45.92 J0033382 00000290 -EJES30   ENDED.      NAME-EJES GENERATION      TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 15:14:45.97 J0033382 00000281 $HASP395 EJES30   ENDED - RC=0000
N C000000 T2SY1      2015213 15:14:46.51                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 15:14:46.52 J0033383 00000090 $HASP1301 EJES40 in job group EJESGENR queued for execution
N 0020000 T2SY1      2015213 15:14:46.55 J0033383 00000281 ICH70001I EJES   LAST ACCESS AT 15:14:45 ON SATURDAY, AUGUST 1, 2015
N 4000000 T2SY1      2015213 15:14:46.57 J0033383 00000281 $HASP373 EJES40   STARTED - INIT 1   - CLASS A           - SYS SY1
N 0000000 T2SY1      2015213 15:14:46.58 J0033383 00000281 IEF403I EJES40 - STARTED - TIME=15.14.46
N 0000000 T2SY1      2015213 15:14:46.60 J0033383 00000000 IEF404I EJES40 - ENDED - TIME=15.14.46
N 0004000 T2SY1      2015213 15:14:46.61 J0033383 00000290 -EJES40   ENDED.      NAME-EJES GENERATION      TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 15:14:46.68 J0033383 00000281 $HASP395 EJES40   ENDED - RC=0000
N C000000 T2SY1      2015213 15:14:46.70                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 15:14:46.72 J0033384 00000090 $HASP1301 EJES50 in job group EJESGENR queued for execution
N 4000000 T2SY1      2015213 15:14:46.79 J0033384 00000281 $HASP373 EJES50   STARTED - INIT 1   - CLASS A           - SYS SY1
N 0000000 T2SY1      2015213 15:14:46.80 J0033384 00000281 IEF403I EJES50 - STARTED - TIME=15.14.46
N 0000000 T2SY1      2015213 15:14:46.82 J0033384 00000000 IEF404I EJES50 - ENDED - TIME=15.14.46
N 0004000 T2SY1      2015213 15:14:46.82 J0033384 00000290 -EJES50   ENDED.      NAME-EJES GENERATION      TOTAL TCB CPU TIME=
N 4000000 T2SY1      2015213 15:14:46.88 J0033384 00000281 $HASP395 EJES50   ENDED - RC=0000
N C000000 T2SY1      2015213 15:14:46.90                00000281 $HASP309 INIT 1   INACTIVE ***** C=A
N 40C0000 T2SY1      2015213 15:14:46.90 G0033374 00000090 $HASP1304 job group EJESGENR is complete
```

Complete your session evaluations online at www.SHARE.org/Orlando-Eval



Up to 999 GDG Generations

- 255 generations is *highly* restrictive!
- If you create a new generation every day (e.g., of your system log), you can't even keep a year's worth of data!
- Specify GDGEXTENDED(YES) in IGGCATxx to raise the GDG limit from 255 to 999.
 - NOTE: Back-level systems cannot tolerate GDGEs. Period!
- Unfortunately, it seemed as if this function wasn't enabled on my z/OS 2.2 system.
 - Even after IPL with GDGEXTENDED(YES) in IGGCATxx, I received the following:

```
DEFINE GDG (NAME(EDJXADM.SHARE.TEST) -  
           LIMIT(999) OWNER(EDJXADM) SCRATCH)  
0IDC3221I CONSTANT '999' NOT WITHIN VALUE RANGE  
0IDC3202I ABOVE TEXT BYPASSED UNTIL NEXT COMMAND. CONDITION CODE IS 12
```

TSO/E Password Pre-prompt Function

- A list of all valid TSO/E userids can be obtained by brute-force, programmatic attack on the TSO/E logon screen
 - Some people are not happy about this!
- New PASSWORDPREPROMPT(ON|OFF) keyword has been added to LOGON statement in IKJTSoxx
- I “cloned” IKJTSo00 to IKJTSoB1 and added the new keyword

```
LOGON
  PASSPHRASE (ON)                /* ENABLE PASSWORD PHRASES    */ +
  PASSWORDPREPROMPT (ON)        /* MORE SECURE LOGON PROTOCOL */
```

TSO/E Password Pre-prompt Function

```
READY
parmlib update(00)
IKJ712I DEFAULT VALUES WERE USED FOR CONSOLE
IKJ712I DEFAULT VALUES WERE USED FOR TEST
IKJ712I DEFAULT VALUES WERE USED FOR PLATCMD
IKJ712I DEFAULT VALUES WERE USED FOR PLATPGM
IKJ55100I THE PARMLIB UPDATE REQUEST WAS SUCCESSFUL. +
READY
logon edjxadm
IKJ56470I EDJXADM LOGGED OFF TSO AT 16:20:41 ON AUGUST 2, 2015
*** _
```

----- TSO/E LOGON -----

Enter LOGON parameters below:

RACF LOGON parameters:

Userid ==> EDJXADM

Password ==> _

Procedure ==> \$IKJTEST

Group Ident ==>

Acct Nmbr ==> 1

Size ==> 2096128

TSO/E Password Pre-prompt Function

- With the new protocol, you get only one shot at a good password, so type *carefully*.
- A wrong password ends the TCAS session *with prejudice*.

```
READY
parmlib update(b1)
IKJ712I DEFAULT VALUES WERE USED FOR CONSOLE
IKJ712I DEFAULT VALUES WERE USED FOR TEST
IKJ712I DEFAULT VALUES WERE USED FOR PLATCMD
IKJ712I DEFAULT VALUES WERE USED FOR PLATPGM
IKJ55100I THE PARMLIB UPDATE REQUEST WAS SUCCESSFUL. +
READY
logon edjxadm
IKJ56470I EDJXADM LOGGED OFF TSO AT 16:27:23 ON AUGUST 2, 2015
IKJ56476I ENTER PASSWORD

IKJ56474I USERID OR PASSWORD IS INCORRECT OR NOT AUTHORIZED
*****
```

TSO/E Password Pre-prompt Function

```
READY
logon edjxadm
IKJ56470I EDJXADM LOGGED OFF TSO AT 16:31:14 ON AUGUST 2, 2015
IKJ56476I ENTER PASSWORD
```

```
***
_
```

----- TSO/E LOGON -----

Enter LOGON parameters below:

RACF LOGON parameters:

Userid ==> EDJXADM

Procedure ==> \$IKJTEST

Group Ident ==>

Acct Nmbr ==> 1

Size ==> 2096128

```
ICH70001I EDJXADM LAST ACCESS AT 16:30:42 ON SUNDAY, AUGUST 2, 2015
IKJ56455I EDJXADM LOGON IN PROGRESS AT 16:35:13 ON AUGUST 2, 2015
IKJ56951I NO BROADCAST MESSAGES
PROCEDURE NAME IS $IKJTEST
*** ALLOCATING SYSPROC ***
*** ALLOCATING SYSEXEC ***
*** ALLOCATING ISPPROF ***
*** ALLOCATING ISPTABL ***
*** ALLOCATING ISPLLIB ***
*** ALLOCATING ISPPLIB ***
*** ALLOCATING ISPMLIB ***
*** ALLOCATING ISPTLIB ***
*** ALLOCATING ISPSLIB ***
```

MCS Console Inactivity Time-out

- Supports automatic MCS console logoff to be performed by the system after a specified time interval elapses
- New TIMEOUT() parameter specifies minutes

```
CONSOLE DEVNUM (SMCS)
        NAME (SMCS&SYSCLONE.00)
        ROUTCODE (1-2,7-10,16-96,99-112,115-128)
        PFKTAB (PFKTAB1)
        AUTH (MASTER)
        MONITOR (JOBNAMES-T)
        CON (N) SEG (16) DEL (RD) RNUM (5) RTIME (1/4) MFORM (J,S) AREA (14)
        RBUF (15)
        LOGON (REQUIRED)
        TIMEOUT (5)
```

```
N 0040000 T2SY1      2015214 13:43:57.35      00000090  IEE055I CONSOLE SMCSY100 (LU:A60TCP24) IS ACTIVE
NR0000000 T2SY1      2015214 13:44:05.04 SMCSY100 00000090  IEE185I LOGON EDJXADM COMPLETE FOR LU=A60TCP24 CN=SMCSY100
NC0000000 T2SY1      2015214 13:44:20.61 SMCSY100 00000290  D T
NR0000000 T2SY1      2015214 13:44:20.62 SMCSY100 00000090  IEE136I LOCAL: TIME=13.44.20 DATE=2015.214 UTC: TIME=20.44.20
SR
...
... Other activity
...
NC0000000 T2SY1      2015214 13:50:11.18 SMCSY100 00000290  LOGOFF ISSUED FOR USER EDJXADM DUE TO INACTIVITY
NR0000000 T2SY1      2015214 13:50:11.18 SMCSY100 00000090  IEE185I LOGOFF EDJXADM COMPLETE FOR LU=A60TCP24 CN=SMCSY100
N 00C0000 T2SY1      2015214 13:50:11.19      00000090  CNZ2205I USER EDJXADM ON CONSOLE SMCSY100 HAS BEEN LOGGED OFF DUE TO
S
AT LEAST 5 MINUTES OF INACTIVITY
N 0040000 T2SY1      2015214 13:50:11.23      00000090  IEE055I CONSOLE SMCSY100 (LU:A60TCP24) IS INACTIVE
```


New Health Checks

- CATALOG_ATTRIBUTE_CHECK
- CTRACE_DEFAULT_OR_MIN
- DMO_REFUCB
- ICSF_KEY_EXPIRATION
- IOS_DYNAMIC_ROUTING
- JES3_DATASET_INTEGRITY
- JES3_DOT_POOL_USAGE
- JES3_JET_POOL_USAGE
- JES3_OST_POOL_USAGE
- JES3_SEE_POOL_USAGE
- RACF_ENCRYPTION_ALGORITHM
- RACF_PASSWORD_CONTROLS
- RACF_RRSF_RESOURCES
- PFA_PRIVATE_STORAGE_EXHAUSTION
- TSOE_OPERSEWAIT_SETTING
- USS_KERNEL_RESOURCES_THRESHOLD
- ZFS_CACHE_REMOVALS

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

New Macros in MACLIB and MODGEN

Macro	Description
CSFZIXIB	ICSF Exit Information Block
CSVAPSMF	Dynamic APF SMF record
CSVFTCHX	CSVFETCH Exit info
ERBR7410	Monitor III Storage Class Memory Data
GTZZSMxF	GTZ tracking service GTZTRACK via SMF write services and mapping macros
HWISMF6A	BCPii mapping for SMF Record Type 106
IARRAX64	RSM Address Space Block 64-bit Extension
IARRCE64	RSM Control and Enumeration 64-bit Extension
IAZASINF	JES Active Job Step information
IAZJPROC	PROCLIB concatenation parm list for SSI 82
IAZLGxxx	EVENTLOG data service IAZLGDT and mapping macros
IEANTRTR	Name/Token Retrieve Register Interface
IHAESSA	Extended Status Save Area
IHALDAX	VSM Local Data Area Extension
IOSDSCxx	IOSSCM Output Mapping macros
IOSSCM	IOS Storage Class Memory Service
IVTBFL64	CSM Buffer Descriptor Mapping
IWMWOPTI	WLM Parmlib Option Information Area
IWMWQHAA	IWM4QHLT answer area (QHAA)
IWM4OPTQ	Query Parmlib Information
IWM4QHLT	Query Server Health Indicator

Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OS 2.2 User Experiences

There were many other z/OS 2.2 features I knew about, but did not explore due to lack of time, lack of pre-reqs, or just good ol' fashioned lack of interest...

I hope you had fun exploring with me...



Complete your session evaluations online at www.SHARE.org/Orlando-Eval

z/OS 2.2 User Experiences