Z/OS LITTLE ENHANCEMENTS: MANY SMALL POTATOES CAN MAKE A BIG MEAL! EDITION 2019B

MARNA WALLE, MWALLE@US.IBM.COM
MEMBER OF THE IBM ACADEMY OF TECHNOLOGY
Z/OS SYSTEM INSTALLATION
IBM Z SYSTEMS, POUGHKEEPSIE, NY, USA

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”).

No other workload processing is authorized for execution on an SE.

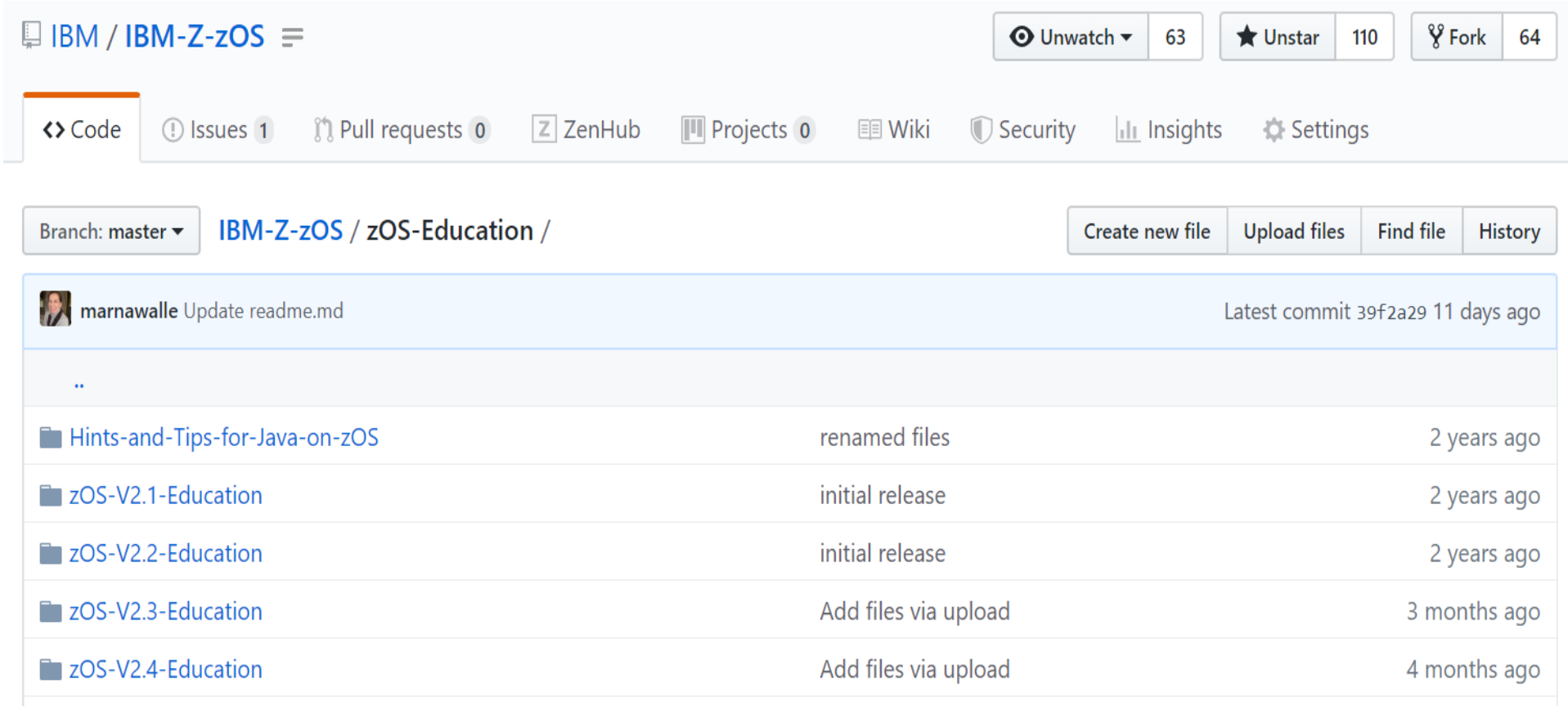
IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

z/OS Small Enhancements - Edition 2019B

- **z/OS V2.3:**
  - **BCP: GDGBIAS**
  - **z/OSMF: Looking at z/OSMF server parameters**
  - **z/OSMF: More information on the angel for z/OSMF**
  - **z/OS UNIX: SUPERUSER for consistent username**
  - **z/OS UNIX: Automatic VERSION UNMOUNT**
- **z/OS V2.2:**
  - **BCP: HZSPRMXx SYS Filter (OA49807)**
  - **BCP: HZSPRMXx Syntax Check (OA49807)**
  - **z/OSMF: Swagger support (PI96461)**
- **z/OS V2.1:**
  - **BCP: PDUU Support for HTTPS (OA55959)**
  - **BCP: Generic Tracker**
  - **BCP: WLM support for Specialty Engine Containment and Memory Capping**
- **Older than the hills:**
  - **BCP: D IPLINFO for IEASYSxx and MACHMIG**

z/OS IBM Education Modules - V2R1, V2R2, and V2R3, and V2R4!

On github! Very easy to find and download! zOS Education PDFs



V2R4 Education is here!..

Small Enhancements

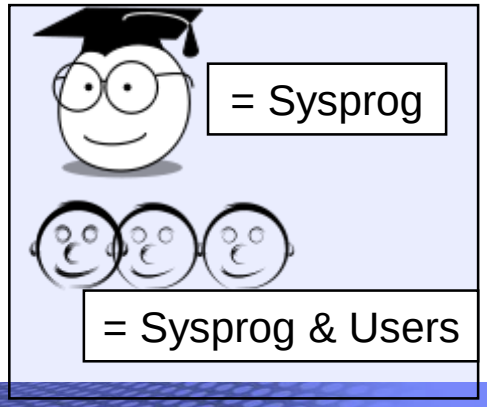
- **BCP: GDGBIAS**

- **z/OSMF: Looking at z/OSMF server parameters**

- **z/OSMF: More information on the angel for z/OSMF**

- **z/OS RACF and UNIX: user ID for UID(0) consistency**

- **z/OS UNIX: BPXPRMxx VERSION UNMOUNT**



BCP: GDGBIAS

What: When restarting jobs that use GDG(E)s, there might be complications on ensuring that the intended generation is used.

- Allocation generally establishes the absolute and relative generation association *for the life of the job*.

- If you knew that you could "reset" the establishment of the absolute and relative generation *at the step level,* then it could be easier to restart a job at a specific step.

How to use:

- New JCL keyword on JOB: `GDGBIAS=JOB|STEP` –or-
- On JES2 or JES3 JOBCLASS `GDGBIAS=JOB|STEP`
- See what is in use with message IEFA111I (system message log)

Considerations:

- Easy to view the setting in SDSF for JOBCLASS.
- There are several other considerations, such as catalog, DISP, and other jobs that might have run before the current job has been restarted.
- By default, dynamic allocation will use the same bias as the batch JCL.

BCP: GDGBIAS

Example, with GDGBIAS of JOB:

```
//JOB1 JOB GDGBIAS=JOB
//STEP1 EXEC PGM=IEFBR14
//DD01 DD DSN=GDG01(+1),DISP=(NEW,CATLG)
//STEP2 EXEC PGM=IEFBR14
//DD02 DD DSN=GDG01(+1),DISP=OLD
```

GDG01.G0001V00

GDG01.G0001V00

Associates to the same relative generation, with the same reference.

```
ICH70001I MWALLE   LAST ACCESS AT 18:57:22 ON THURSDAY, MARCH 7,
IEFA111I MWALLEG1 IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=BELOW,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
```

BCP: GDGBIAS

Example, with GDGBIAS of STEP:

```
//JOB2 JOB GDGBIAS=STEP
//STEP1 EXEC PGM=IEFBR14
//DD01 DD DSN=GDG01(+1),DISP=(NEW,CATLG)
//STEP2 EXEC PGM=IEFBR14
//DD02 DD DSN=GDG01(0),DISP=OLD
```

GDG01.G0001V00

GDG01.G0001V00

Associates to the same relative generation, but needs a different reference.

```
ICH70001I MWALLE   LAST ACCESS AT 21:07:01 ON THURSDAY, MARCH 7,
IEFA111I MWALLEGS IS USING THE FOLLOWING JOB RELATED SETTINGS:
         SWA=BELOW,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=STEP
```

BCP: GDGBIAS

Changing JES2 JOBCLASS dynamically for GDGBIAS of STEP:

```
$T JOBCLASS(Y),GDGBIAS=STEP
$HASP837 JOBCLASS(Y) 006
$HASP837 JOBCLASS(Y)          ACTIVE=YES,ACCT=NO,AUTH=(ALL),
$HASP837                      BLP=YES,COMMAND=DISPLAY,COPY=NO,
$HASP837                      DSENQSHR=ALLOW,DUPL_JOB=DELAY,
$HASP837                      GDGBIAS=STEP,GROUP=,HOLD=NO,
$HASP837                      IEFUJP=YES,IEFUSO=YES,
```

```
SDSF JOB CLASS DISPLAY ALL CLASSES                    LINE 24-38 (38)
COMMAND INPUT ===>                                    SCROLL ===> HALF
NP    CLASS      XBMProc  DupJob  JobRC   Active  SysSym    DsEnqShr PromoRt GDGBia
      V                   NO      MAXRC   YES     DISALLOW  ALLOW          0 JOB
      W                   NO      MAXRC   YES     DISALLOW  ALLOW          0 JOB
      X                   NO      MAXRC   YES     DISALLOW  ALLOW          0 JOB
      Y                   NO      MAXRC   YES     DISALLOW  ALLOW          0 STEP
      Z                   NO      MAXRC   YES     DISALLOW  ALLOW          0 JOB
      0                   NO      MAXRC   YES     DISALLOW  ALLOW          0 JOB
```

z/OS V2.3 as of PH00712 (9/2018) and PI93759 (7/2018):
z/OSMF: Looking at z/OSMF server parameters

What: You are wondering about what you are using in z/OSMF. For instance:

- What URL should you be using logon to z/OSMF?
- What plug-ins does z/OSMF think are available?
- How long has the z/OSMF server been up?
- Is z/OSMF using a default value, or one that I've specified?

How to use:

- `D IZU[,SERVER=`*`servername`*`]` –or-
- `F` *`servername`*`,DISPLAY IZU`
- Look at the nice long response.

Considerations:

- Note the syntax on the modify (`F`) command, it is not `D IZU`, it is `DISPLAY IZU`.

z/OS V2.3 as of PH00712 (9/2018) and PI93759 (7/2018):
z/OSMF: Looking at z/OSMF server parameters

```
F IZUSVR1,DISPLAY IZU
+CWWKB0004I: z/OSMF PARMLIBs DISPLAY
  IZUG013I The home page of z/OSMF server                    (S1)
  in AUTOSTART_GROUP(IZUDFLT) can be accesse
  https://MVS1.CENTERS.IHOST.COM-  :443/zosmf
  IZUG014I The server started at  03/05/2019 15:47:
   and has been running for  05(hh):39(mm):07(ss)

  Current z/OSMF settings                                   Source

  HOSTNAME(MVS1.CENTERS.IHOST.COM- )                        IZUPRM01
+CWWKB0061I CONTINUATION 1 FOR MESSAGE IDENTIFIER 771778365
  HTTP_SSL_PORT(443)                                        IZUPRM01
  LOGGING('*=warning:com.ibm.zosmf.*=info:com.ibm.zosmf.
  environment.ui=fi')                                       IZUPRM01
  UNAUTH_USER(IZUGUEST)                                     IZUPRM01
  SEC_GROUPS
      ADMIN(IZUADMIN)                                       IZUPRM01
```

Mind the wrap!

Up time

z/OS V2.3 as of PH00712 (9/2018) and PI93759 (7/2018):
z/OSMF: Looking at z/OSMF server parameters

```
 Status of z/OSMF plugins

 Configuration Assistant(STARTED)                          IZUPRM01
CWWKB0061I CONTINUATION 5 FOR MESSAGE IDENTIFIER 771778365
 Capacity Provisioning(STARTED)                            IZUPRM01
 Workload Management(STARTED)                              IZUPRM01
 Resource Monitoring(STARTED)                              IZUPRM01
 Incident Log(STARTED)                                     IZUPRM01
 Software Management(S                                     IZUPRM01
 WebISPF(STARTED)                                          IZUPRM01
 Sysplex Management(UNSPECIFIED)                           DEFAULT
```

z/OSMF: More information on the angel for z/OSMF

What: You are unable to tell from the angel's joblog what level of WebSphere Liberty profile is being used.

- This makes it very hard to debug, as you quite possibly would have multiple angels on the same system.
- More information is added during angel startup, concerning the service level of Liberty.

How to use:

- Look at angel startup for message CWWK0079I in joblog.
- Issue command to get the level: `F angelname,VERSION`

Considerations:

- The joblog message and the MODIFY response show very different things!

z/OSMF: More information on the angel for z/OSMF

From joblog:

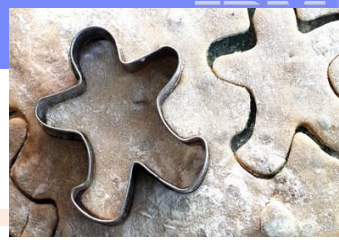
```
STC19240 ---- MONDAY,    04 MAR 2019 ----
STC19240  $HASP373 IZUANG1  STARTED
STC19240  CWWKB0079I THE ANGEL BUILD LEVEL IS 18.0.0.3 20180906-0409 /  743
          2018.9.0.0 20180906-0409
STC19240  CWWKB0069I INITIALIZATION IS COMPLETE FOR THE IZUANG1 ANGEL PROCESS.
```

As a command:

```
MODIFY IZUANG1,VERSION
CWWKB0053I ANGEL VERSION 9
```

- My assumption on why these are so different:
  - If you are an exploiter or Liberty and are using certain functions, you care about what functional level Liberty is at: "VERSION".
  - The “BUILD LEVEL” contains the information on the fixes incorporated, which is probably more interesting to a z/OS sysprog who doesn't write Liberty programs.

z/OS RACF and UNIX: user ID for UID(0) consistency



What: You quite possibly have many user IDs assigned a UNIX UID of 0. You quite possibly want to see what user ID is associated with a file that is owned by UID 0 using common utilities (such as ls –l). Which user ID would that necessarily be?

- This could actually vary from time to time, making it confusing to some.

- Now, RACF will use the BPXPRMxx `SUPERUSER(username)` for reporting consistency.
- *Bonus!* A matching z/OS UNIX health check to make sure that your `SUPERUSER(username)` is identified to your security product and has a UID of 0.

How to use:

- Make sure you have the desired user ID in `SUPERUSER(username)`. `BPXROOT` is the default.
  - The `username` must be identified to RACF and have a UID of 0.

z/OS V2.3:

z/OS RACF and UNIX: user ID for UID(0) consistency

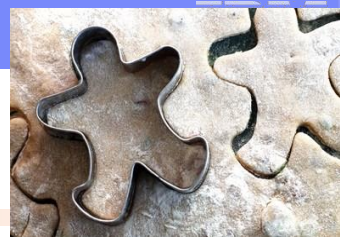
From ls –l in shell:

```
MVS1:MARNA:/shareuser/marna: > ls -l /bin/alias
-rwxr-xr-x   17 BPXROOT  STCGROUP    77824 Apr 27  2017 /bin/alias
```

From RACF data base:

```
 BROWSE - RACF COMMAND OUTPUT---------------------- LINE 000
 COMMAND ===>
********************************* Top of Data ****************
USER=BPXROOT  NAME=UNKNOWN  OWNER=STEVEM    CREATED=18.017
OMVS INFORMATION
----------------
UID= 0000000000
HOME= /
PROGRAM= /bin/sh
```

z/OS RACF and UNIX: user ID for UID(0) consistency



From Health Checker:

```
********************************* TOP OF DATA **************************
CHECK(IBMUSS,USS_SUPERUSER)
SYSPLEX:    SHARPLEX  SYSTEM: S1
START TIME: 03/04/2019 10:18:21.237842
CHECK DATE: 20160331  CHECK SEVERITY: HIGH

BPXH003I z/OS UNIX System Services was initialized using OMVS=(SH),
where each 2-character item is a BPXPRMxx suffix.
...

BPXH086I No problems were found with userID BPXROOT.

END TIME: 03/04/2019 10:18:21.318046  STATUS: SUCCESSFUL
******************************** BOTTOM OF DATA ************************
```

This check's interval is ONETIME per IPL and will re-run if SUPERUSER value changes.

z/OS UNIX: Automatic VERSION UNMOUNT

What: BPXPRMxx `VERSION` statement is used in a shared file system environment to allow multiple release and service levels of a software to coexist in the hierarchy.

- Once you've "passed through" that level, you should remember to unmount that unused VERSION file system from the environment.
  - And any file systems under it too.
- z/OS can now automatically detect that that VERSION file system is no longer in use.
  - Meaning, no system is using it or using any file system mounted under it, and unmount them.

How to use:

- In BPXPRMxx: `VERSION=('string',UNMOUNT|NOUNMOUNT)`
- Can change dynamically with SET OMVS or SETOMVS
- Validates hardening in health check USS_PARMLIB

Considerations:

- You'll see messages in the hardcopy log when unmounts occur.

z/OS UNIX: Automatic VERSION UNMOUNT

Example: Current options:

```
-D OMVS,OPTIONS
 BPXO043I 23.13.43 DISPLAY OMVS 405
 OMVS     0012 ACTIVE                OMVS=(CB)
 CURRENT UNIX CONFIGURATION SETTINGS:
 MAXPROCSYS       =      10000     MAXPROCUSER      =       2000
MAXQUEUEDSIGS     =     100000     SHRLIBRGNSIZE    =   67108864
SHRLIBMAXPAGES    =       4096     VERSION          = CMRS41   ,N
SYSCALL COUNTS    = NO             TTYGROUP         = TTY
```

Changing to VERSION UNMOUNT
(notice the handy * to designate current VERSION string):

```
-SETOMVS VERSION=(*,UNMOUNT)
 BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.
```

z/OS V2.3:

z/OS UNIX: Automatic VERSION UNMOUNT

Current options have changed:

```
-D OMVS,OPTIONS
 BPXO043I 23.13.43 DISPLAY OMVS 405
 OMVS     0012 ACTIVE                OMVS=(CB)
 CURRENT UNIX CONFIGURATION SETTINGS:
 MAXPROCSYS       =       10000    MAXPROCUSER      =        2000
MAXQUEUEDSIGS    =      100000    SHRLIBRGNSIZE    =    67108864
SHRLIBMAXPAGES   =        4096    VERSION          = CMRS41   ,U
SYSCALL COUNTS   = NO             TTYGROUP         = TTY
```

z/OS UNIX: Automatic VERSION UNMOUNT

Health Check kindly reminds of the dynamic and hardened mismatch:

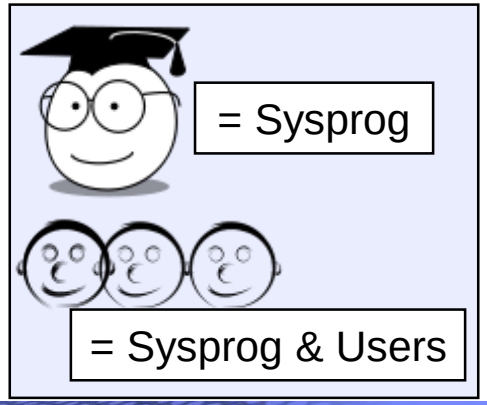
```
CHECK(IBMUSS,USS_PARMLIB)
SYSPLEX:    UTCPLXCB  SYSTEM: CB89
START TIME: 03/07/2019 23:23:52.103861
CHECK DATE: 20060112  CHECK SEVERITY: LOW

BPXH003I z/OS UNIX System Services was initialized using OMVS=(CB),
where each 2-character item is a BPXPRMxx suffix.

BPXH041I The following differences were found between the system
settings and the BPXPRMxx parmlib members:


Option                 BPXPRMxx Value          System Value
-------------------------------------------------------------
VERSION                  CMRS41   ,N             CMRS41   ,U
```

Small Enhancements

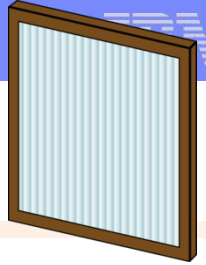
- **BCP: HZSPRMXx SYS Filter**

- **BCP: HZSPRMXx Syntax Check**

- **z/OSMF: Swagger support**



BCP: HZSPRMXx SYS Filter



What: Typically users make changes to health checks, and harden those into an HZSPRMxx. However, the differences from system to system, or sysplex to sysplex might be small. This caused full separation of HZSPRMxx parmlib members for those different environments.

- As of z/OS V2.3, you can now use filters to consolidate those differences into a single HZSPRMxx parmlib member

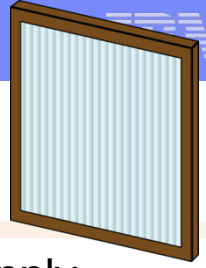
How to use:

- **`WHEN`** **(***condition)* **`DO`** (*some_change)* **`END`** to scope around the differences for an environment.
- *Condition* can use SYSTEMNAME, SYSPLEXNAME, HWNAME, LPARNAME, VMUSERID, or a *textstring (system symbol!),* and can be compounded
- Comparison operators: =, <>, >, >=, <, <=, IN, NOTIN
- Can wildcard with * or ?, with some reasonable restrictions

Considerations:

- Good programming practice would be to include DO and ENDs for clarity, although there are rules if that is not done.
- You can put any HZSPRMxx statement within the WHENs: HZSPDATA, …
- Now, you can collapse perhaps several HZPRMxx parmlib members which differ slightly into a single parmlib member to maintain.

BCP: HZSPRMXx SYS Filter



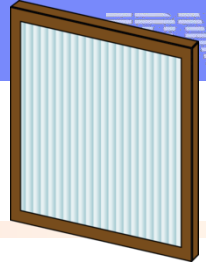
Example: I want to maintain a single HZSPRMxx, and in one sysplex I want to apply some health check policy differences:

- On all zOS V2.4 production systems I want to have CA_RECLAIM check changed …

```
WHEN (&SYSPLEX. = UTCPLXCB)
DO /* FOR production UTCPLXCB */

   WHEN (&MWPARM. = 'PROD' &SYSOSLVL. >= Z1020400 )
   DO /* PRODuction */
      /* On z/OS V2.4 and up */
      UPDATE CHECK(IBMVSAM,VSAM_CA_RECLAIM)
             DATE(20190717)
             SEVERITY(HIGH)
             INTERVAL(00:10)
             REASON('CA Reclaim medium severity, every 10m')
   END /* PRODuction */
```

- All other systems will remain as is.
- Original settings: INTERVAL: **ONETIME** SEVERITY: **MEDIUM**

BCP: HZSPRMXx SYS Filter



Example: …and

- On Test systems, at any z/OS level, I want MAXVIRTUAL check changed

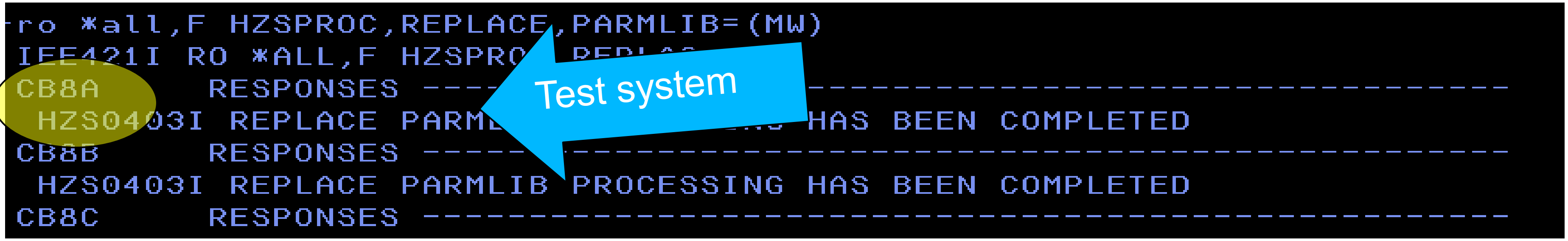
```
   WHEN (&MWPARM. = 'TEST')
   DO /* All TEST systems */
      UPDATE CHECK(IBMVLF,VLF_MAXVIRT)
             DATE(20190717)
             SEVERITY(MEDIUM)
             INTERVAL(0:10)
             REASON('VLF medium severity, every 10m')
   END /* All TEST systems */

END /* FOR production UTCPLXCB */
```

- All other systems, will remain as is.
- Original settings: INTERVAL: **1:00** SEVERITY: **LOW**
- **Note: &MWPARM. is my own system symbol to identify which systems are Production vs. Test.**

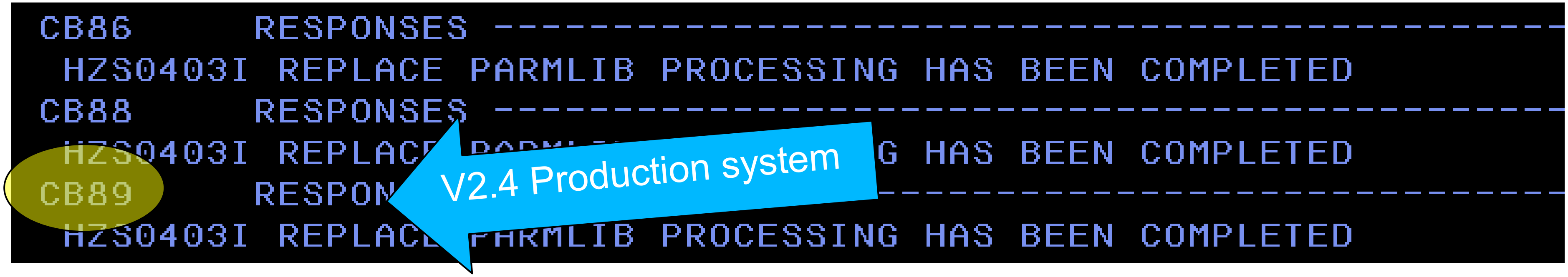
z/OS V2.2 with APAR OA49807 (RSU1612)

BCP: HZSPRMXx SYS Filter

Example: …then made the change across the sysplex.



- …



BCP: HZSPRMXx SYS Filter

Example: …then verified the changes:
zOS V2.4 production systems for CA_RECLAIM

- Production V2.4 systems (changed to HIGH and :10):

```
CB89      RESPONSES ---------------------------------------
HZS0201I 18.19.10 CHECK DETAIL      499
CHECK(IBMVSAM,VSAM_CA_RECLAIM)
 STATE: ACTIVE(ENABLED)                STATUS: SUCCESSFUL
 EXITRTN: IDAHCADD
 LAST RAN: 07/19/2019 18:16     NEXT SCHEDULED: 07/19/2019 1
 INTERVAL: 0:10
 EXCEPTION INTERVAL: SYSTEM
 SYNCVAL: SYSTEM
 SEVERITY: HIGH
```

- Non-production systems (remains MEDIUM and ONETIME):

```
CB8A      RESPONSES ---------------------------------------
HZS0201I 18.19.10 CHECK DETAIL      900
CHECK(IBMVSAM,VSAM_CA_RECLAIM)
 STATE: ACTIVE(ENABLED)                STATUS: SUCCESSFUL
 EXITRTN: IDAHCADD
 LAST RAN: 07/19/2019 17:20     NEXT SCHEDULED: (NOT SCHEDULED)
 INTERVAL: ONETIME
 EXCEPTION INTERVAL: SYSTEM
 SYNCVAL: SYSTEM
 SEVERITY: MEDIUM
```

BCP: HZSPRMXx SYS Filter

Example: …then verified the changes:
Test systems for MAXVIRT

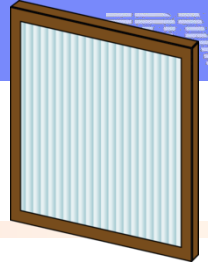
- Production V2.4 systems (remains at LOW and 1:00):

```
CB89      RESPONSES ---------------------------------------
HZS0201I 18.29.56 CHECK DETAIL      648
CHECK(IBMVLF,VLF_MAXVIRT)
 STATE: ACTIVE(ENABLED)                STATUS: SUCCESSFUL
 EXITRTN: COFMHCVA
 LAST RAN: 07/19/2019 18:20     NEXT SCHEDULED: 07/19/2019
 INTERVAL: 1:00
 EXCEPTION INTERVAL: SYSTEM
 SYNCVAL: SYSTEM
 SEVERITY: LOW
```

- Non-production systems (changed to MEDIUM and :10):

```
CB8A      RESPONSES ---------------------------------------
HZS0201I 18.29.56 CHECK DETAIL      043
CHECK(IBMVLF,VLF_MAXVIRT)
 STATE: ACTIVE(ENABLED)                STATUS: SUCCESSFUL
 EXITRTN: COFMHCVA
 LAST RAN: 07/19/2019 18:29     NEXT SCHEDULED: 07/19/2019
 INTERVAL: 0:10
 EXCEPTION INTERVAL: SYSTEM
 SYNCVAL: SYSTEM
 SEVERITY: MEDIUM
```

BCP: HZSPRMXx SYS Filter



Example: …Just a little nice thing to incidentally notice in SDSF:
CB89 system (Production V2.4, CA_RECLAIM changes, MAXVIRT does not):

```
SDSF HEALTH CHECKER DISPLAY  CB89                          LINE 172-189 (235)
COMMAND INPUT ===>                                           SCROLL ===>
NP    NAME                                vCode WTOType    ModifiedBy
      VLF_MAXVIRT                             4 INFO
      VSAM_CA_RECLAIM                        12 CRITICAL   PARMLIB(HZSPRMMW)
      VSAM_INDEX_TRAP                         8 EVENTUAL
```

CB8A system (Test system, CA_RECLAIM does not change, MAXVIRT does):

```
SDSF HEALTH CHECKER DISPLAY  CB8A                          LINE 166-183 (232)
COMMAND INPUT ===>                                           SCROLL ===>
NP    NAME                                vCode WTOType    ModifiedBy
      VLF_MAXVIRT                             8 EVENTUAL   PARMLIB(HZSPRMMW)
      VSAM_CA_RECLAIM                         8 EVENTUAL
      VSAM_INDEX_TRAP                         8 EVENTUAL
```

BCP: HZSPRMXx Syntax Check

What: Also introduced is the capability to perform a syntax check only on complete HZSPRMxx parmlib member(s)

- Does not apply the contained statements to any health checks or to any Health Checker global settings.

How to use:

- `MODIFY hzsproc,ADD,PARMLIB=(aa,..,CHECK|C)`

Considerations:

- `ASA021I SYNTAX CHECKING IS COMPLETE FOR PARMLIB MEMBER=HZSPRMMW. NO ERRORS WERE FOUND`
- `ASA020I SYNTAX CHECKING IS COMPLETE FOR PARMLIB MEMBER=HZSPRMMW. ERROR(S) WERE FOUND`
- `HZS0403I ADD PARMLIB PROCESSING HAS BEEN COMPLETED` ← *this message is also produced*
- New option is on the ADD (not the REPLACE)

BCP: HZSPRMXx Syntax Check

Example:

```
-F HZSPROC,ADD,PARMLIB=(MW,CHECK)
 ASA009I SYNTAX ERROR IN PARMLIB MEMBER=HZSPRMMW ON LINE 22,
 POSITION 1: WHEN END IS SPECIFIED,
 THE FOLLOWING MUST ALSO BE SPECIFIED:
 (DO).
 DETECTING MODULE IS HZSIPMU1. INPUT LINE:
 END /* NOT production UTCPLXCB */
 ASA003I SYNTAX ERROR IN PARMLIB MEMBER=HZSPRMMW ON LINE 27,
 POSITION 26: QUOTED-STRING WAS SEEN, WHERE ONE OF
 (= GREATER_THAN IN LESS_THAN
 NOTIN)
 WOULD BE CORRECT.
 DETECTING MODULE IS HZSIPMX. INPUT LINE:
    WHEN (&MWPARM. = 'PROD' 'Z1020400 = 'Z1020400')
 ASA009I SYNTAX ERROR IN PARMLIB MEMBER=HZSPRMMW ON LINE 34,
 POSITION 4: WHEN END IS SPECIFIED,
 THE FOLLOWING MUST ALSO BE SPECIFIED:
 (DO).
```

```
ASA020I SYNTAX CHECKING IS COMPLETE FOR PARMLIB MEMBER=HZSPRMMW. ERROR(S) WERE
HZS0403I ADD PARMLIB PROCESSING HAS BEEN COMPLETED
```

Then when all problems were fixed:

```
ASA021I SYNTAX CHECKING IS COMPLETE FOR PARMLIB MEMBER=HZSPRMMW. NO ERRORS WE
HZS0403I ADD PARMLIB PROCESSING HAS BEEN COMPLETED
```

z/OSMF Swagger support

•**What: "Swagger"** is an open-source software framework backed by a large ecosystem of tools that helps developers design, build, document, and consume RESTful Web services. (definition from Wiki).

•z/OSMF Swagger support allows users to:

- browse z/OSMF REST APIs by connecting to any z/OSMF instance:

  https:// <hostname>:<port>/zosmf/api/explorer/

- Try z/OSMF REST API without having to do any coding

•Only subset of current z/OSMF REST APIs support Swagger today: jobs services, data set and file services, cloud provisioning for z/OS, …

How to use:

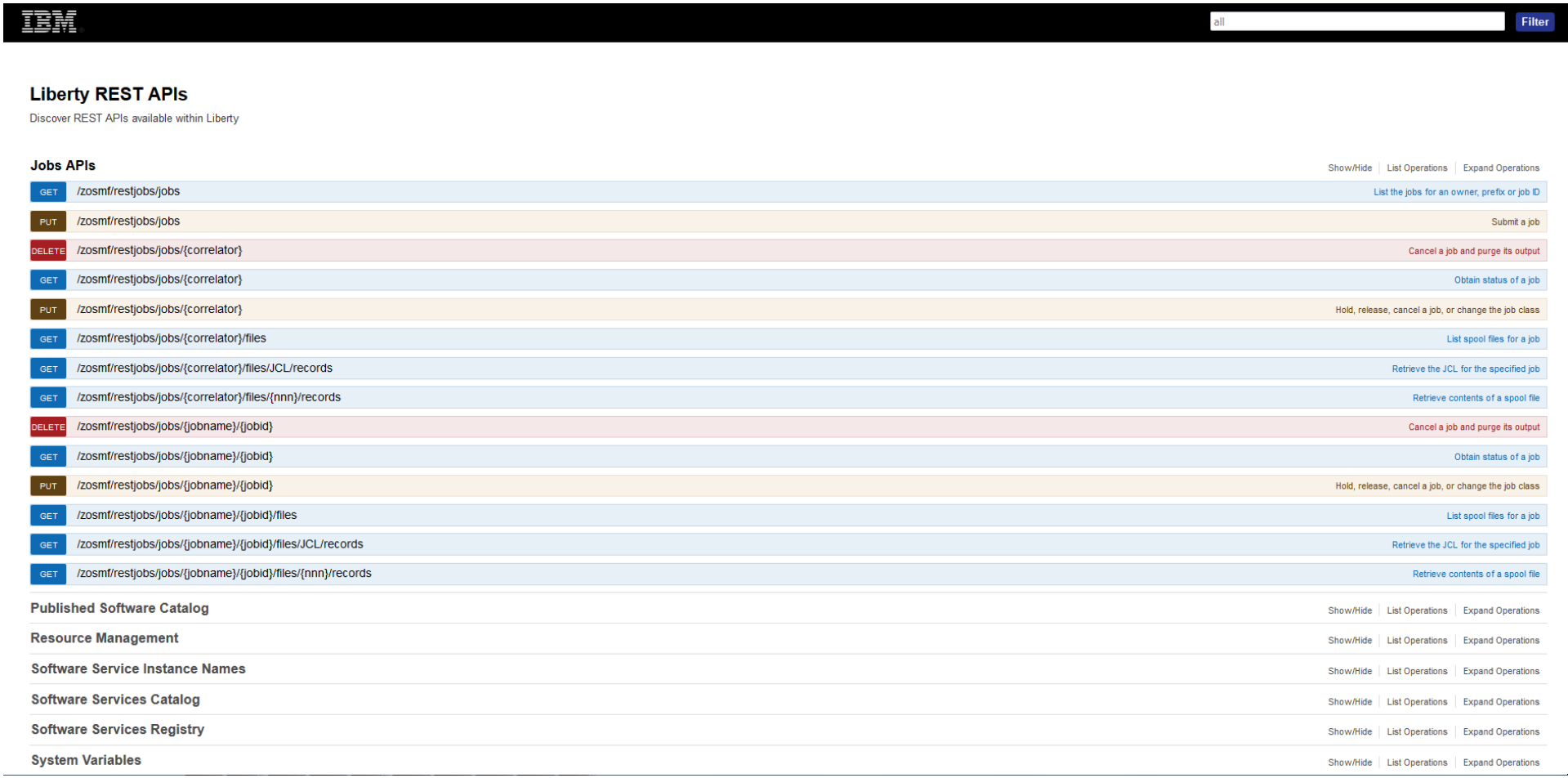
•Additional security resource set up is necessary.

- `IZUDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers` resource in the `EJBROLE` class
- Go to web location, and investigate and try.

z/OS V2.2 with APAR OA49807 (January 2019)

z/OSMF Swagger support

Example : Go to URL and see what is there. Jobs APIs looks interesting…



z/OS V2.2 with APAR OA49807 (January 2019)

z/OSMF Swagger support

Example : Want to see all my MWALLE jobs.

Jobs APIs Show/Hide | List Op

GET /zosmf/restjobs/jobs List the jo

Implementation Notes

You can use this operation to list the jobs for an owner, prefix, or job ID.

Response Class (Status 200)

On completion, the z/OS jobs REST interface returns an HTTP response with an array of matching jobs, each as a JSON job document.

Model | Example Value


```
{
  "application/json": [
    {
      "jobid": "JOB00023",
      "jobname": "TESTJOB2",
      "subsystem": null,
      "owner": "IBMUSER",
      "status": "OUTPUT",
      "type": "JOB",
      "class": "A",
```


Response Content Type application/json

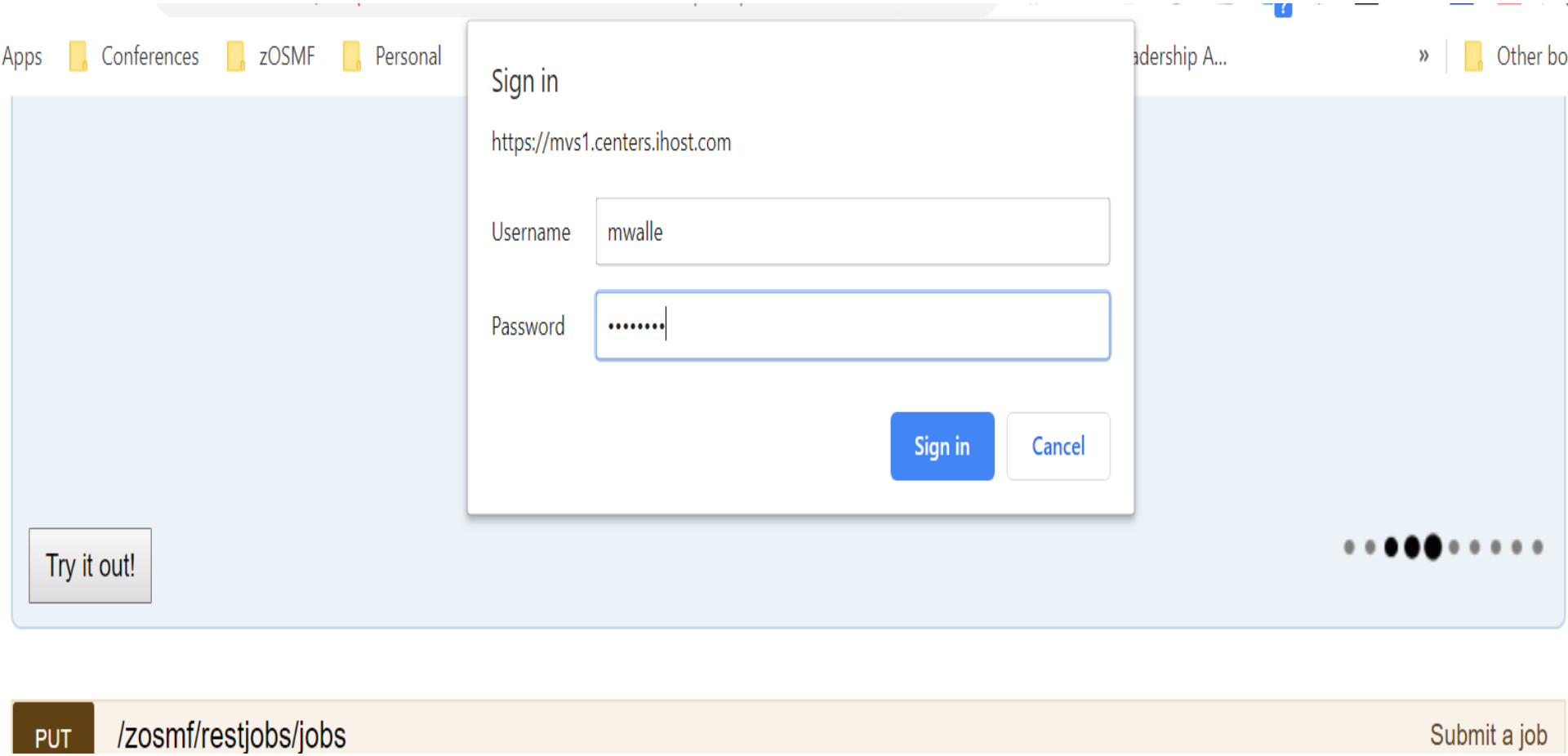
Parameters

Parameter	Value	Description	Parameter Type	Data Type
owner	mwalle	User ID of the job owner whose jobs are being queried; the default is the z/OS user ID. Folded to uppercase; cannot exceed eight characters. Wildcard characters are permitted in the owner and prefix query parameter values. Use an asterisk (*) for multiple characters, and a question mark (?) for a single character.	query	string

z/OS V2.2 with APAR OA49807 (January 2019)

z/OSMF Swagger support

Example : "Try it out!" needs me to identify myself to the server, if I'm going request a service.



z/OS V2.2 with APAR OA49807 (January 2019)

z/OSMF Swagger support

Example : Results are shown: correctly coded Request URL and the Response body returned. Already tested for my program!

Try it out! Hide Response

Curl

```
curl -X GET --header 'Accept: application/json' 'https://mvs1.centers.ihost.com:443/zosmf/restjobs/jobs?owner=mwalle&max-jobs=1000'
```

Request URL

```
https://mvs1.centers.ihost.com:443/zosmf/restjobs/jobs?owner=mwalle&max-jobs=1000
```

Response Body


```
[
  {
    "owner": "MWALLE",
    "phase": 20,
    "subsystem": "JES2",
    "phase-name": "Job is on the hard copy queue",
    "job-correlator": "T0023791N1......D606AB1E.......:",
    "type": "TSU",
    "url": "https://mvs1.centers.ihost.com:443/zosmf/restjobs/jobs/T0023791N1......D606AB1E.......%3A",
    "jobid": "TSU23791",
    "class": "TSU",
    "files-url": "https://mvs1.centers.ihost.com:443/zosmf/restjobs/jobs/T0023791N1......D606AB1E.......%3A/files",
    "jobname": "MWALLE",
    "status": "OUTPUT",
    "retcode": "ABEND S622"
  },
  {
    "owner": "MWALLE",
    "phase": 20,
    "subsystem": "JES2",
```


Response Code

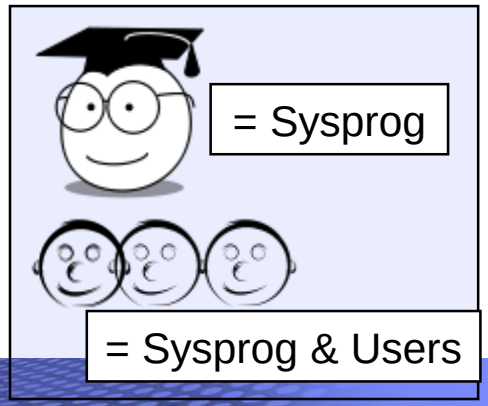
```
200
```

Small Enhancements

- **BCP: PDUU support for HTTPS (OA55959)**

- **BCP: Generic Tracker**

- **BCP: WLM support for Special Engine Containment and Memory Capping**



z/OS V2.1 with OA55959 (June 2019) and higher:
BCP: PDUU Support for HTTPS

What: AMAPDUPL: Problem Documentation Upload Utility.

- Used to sent a dump to IBM, can be compressed, encrypted, and sectioned into smaller data sets.
- FTP (existing capability, default) was not a popular choice (firewall issues, …)
- Now with OA55959, HTTPS can be used!

How to use:

- Uses private virtual storage for buffering, so specify `WORK_SIZE` adequately.
- Use `USE_HTTPS=Y` on the SYSIN.
- Needs necessary certificates to access the IBM sites, via `HTTPS_KEYRING` or `HTTPS_KEYFILE`. Read certificate info here.

Considerations:

- Need to select a feasible `WORK_SIZE`: these are allocated in 31-bit private storage, which limits it to less than 2GB (known restriction). Failures will be rc 12 w/ AMA761E (Unable to obtain necessary storage).
- Might result in longer processing times, due to smaller work sizes, however, still may be appropriate if FTP is not an option.
- Alas, z/OSMF Incident Log today uses PDUU FTPS.

z/OS V2.1 with OA55959 (June 2019) and higher:
BCP: PDUU Support for HTTPS example

```
//SEND2IBM EXEC PGM=AMAPDUPL
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//DEBUG DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=D10SWL1.DUMP30
//HTTPDEBG DD DISP=MOD,DSN=MWALLE.HTTPDEBG
//SYSIN DD DISP=SHR,DSN=MWALLE.FTP.PDUU.OPTIONS
// DD *
TARGET_SYS=testcase.boulder.ibm.com
TARGET_DSN=TEST.DUMP00
CC_HTTPS=03
WORK_SIZE=50
DIRECTORY=/toibm/mvs/
PMR=00000.000.000
USE_HTTPS=Y
HTTPS_KEYRING=*AUTH*/*
```

I used:

- 3 parallel transfer sessions
- Default of work size of 50 MB
- HTTPS mode

```
** AMA732I  START 001 HTTPS SESSION
** AMA733I  SEND FILE A001
** AMA732I  START 002 HTTPS SESSION
** AMA733I  SEND FILE A002
** AMA732I  START 003 HTTPS SESSION
** AMA733I  SEND FILE A003
** AMA733I  SEND FILE B001
** AMA764I  WAITING FOR COMPLETION OF FTP THREAD: 001 REMOTE FILE: ...F00002
** AMA733I  SEND FILE B002
** AMA764I  WAITING FOR COMPLETION OF FTP THREAD: 002 REMOTE FILE: ...F00003
** AMA733I  SEND FILE B003
** AMA764I  WAITING FOR COMPLETION OF FTP THREAD: 003 REMOTE FILE: ...F00004
…..
** AMA725I  PROCESSING COMPLETED
** AMA728I  ELAPSED TIME: 162.38 SECONDS
** AMA729I  NUMBER OF RECORDS READ FROM SYSUT1: 380,310 TOTAL BYTES: 1,582,089,600
** AMA730I  TOTAL BYTES COMPRESSED DATA TRANSMITTED = 565,547,008
** AMA731I  EFFECTIVE THROUGHPUT = 9,743,100 BYTES/SECOND
```

BCP: Generic Tracker

What: Generic Tracker is a facility that an help assess usage of specific requestable incidents that happen on a system. (z/OS MVS Diagnostics: Tools and Service Aids.)

- It is composed of:
  - A callable tracking service (GTZTRACK).
  - A callable query service to extract previously stored records (GTZQUERY).
  - Operator commands to display and maintain information and configuration.
  - Batch utility program (GTZPRINT)
  - Parmlib members for desired customization (GTZPRMxx).
- *Why it is helpful?* This facility can help with knowing when something is used (when it might be deleted in a future release), or to position for a new function (when changes might be required to use it).
- GTZ will be automatically started at IPL, however by default, it is not enabled for capturing data.

How to use:

1. Enable GTZ. SETGTZ TRACKING=ON
2. Check on collected information: DISPLAY GTZ or with GTZPRINT

BCP: Generic Tracker

Considerations:

- Currently, this is a list of the Generic Track exploiters
  - DFSMS tracking – GDGLIMIT and EAV
  - JES3 control statement tracking – JES3 JECL
  - JES2 control statement tracking – JES2 and JES3 JECL
  - MVS Allocation tracking – IEF348I message control
  - SDSF tracking – NOPARM FALLBACK and MENU TABLE DISABLED
  - TSO/E tracking – MVSSERV executed to invoke Enhanced Connectivity Facility
  - VSM tracking – V=R request

- TRACKDATA can be persisted in SMF type 125, for historical review.

- Your DDDEF'd SYS1.PARMLIB contains a shipped GTZPRM00 which contains currently know exceptions that are acceptable and are "not interesting anymore".
  - These exceptions will be excluded from tracking as to not clutter up new data being collected. (Currently we have about 33 of them.)

- Change MEMLIMIT on the GTZ proc to control how much storage is used → how many events you can store. 2MB is miniumum, default if 200MB.

BCP: Generic Tracker

Very simple example: I want to know when JES3 JECL is used on my JES2 system. (I've got that V2.2/V2.3 function enabled already.)

1. Check is Generic Tracker is enabled for tracking:

```
-D GTZ,STATUS
 GTZ1001I  13.24.11 GTZ STATUS 797
  TRACKING:  DISABLED              2019-07-20 13:23:57
  TRACKED:   UNIQUE=19             TOTAL=2443
  EXCLUDE:   DEFINED=34            APPLIED=126
  DEBUG:     DEFINED=0             APPLIED=0
  GTZPRMXX:  00
  MEMORY:    99% AVAILABLE         PERSIST:   OFF
  DIRLOAD:   YES
```

2. If not, enable it:

```
-setgtz tracking=on
 GTZ1105I SETGTZ TRACKING PROCESSING IS COMPLETE
```

BCP: Generic Tracker

Very simple example: I want to know when JES3 JECL is used on my JES2 system.

3. Now, we wait…then, check if any instance has been captured.

```
-d gtz,trackdata=(owner=ibmjes2)
 GTZ1002I  14.21.42 GTZ TRACKDATA 780
 FOUND 87 MATCHING TRACKED INSTANCE(S)
 ----------------------------------------------------------------------
  INSTANCE:       1                        COUNT:          1
  EVENTDESC:      '|00000000 0|10000000 0|  INTRDR      PFACMDS  PFASUB '+
                  '   JES2          '
  OWNER:          IBMJES2                  SOURCE:         HASCINJR
  EVENTDATA:      x0000000000000000        x0000000000000000
```

…

```
INSTANCE:       87                       COUNT:          6
EVENTDESC:      '|00100000 0|00000000 0|  INTRDR      MWALLCB  MWALLE '+
                '   JES2          '
OWNER:          IBMJES2                  SOURCE:         HASCINJR
EVENTDATA:      x0000000000000000        x0000000000000000
PROGRAM:        *OMITTED                 PROGRAMOFFSET:  x0000000000000000
HOMEJOB:        MWALLE                   HOMEASID:       x0048
EVENTJOB:       MWALLE                   EVENTASID:      x0048
AUTHORIZED:     YES                      FIRST TIME:     2019-07-20 14:10:57
******************************** BOTTOM OF DATA ********************************
```

BCP: Generic Tracker

Very simple example: I want to know when JES3 JECL is used on my JES2 system.

4. Also nice to view in SDSF:

```
SDSF GENERIC TRACKER  CB8A     CB8A                          LINE 98-106 (106)
COMMAND INPUT ===>                                             SCROLL ===> HALF
NP    OWNER          EJobName  HJobName  EASIDX HASIDX Auth Count First-Date-Tim
      IBMJES2  0000 NOSAPURG  NOSAPURG  010C   010C   YES       1 07/19/2019 07:
      IBMJES2  0000 NOSAPURG  NOSAPURG  010C   010C   YES       1 07/19/2019 07:
      IBMJES2  0000 NOSAPURG  NOSAPURG  010C   010C   YES       1 07/19/2019 07:
      IBMJES2  0000 IZUSVR2   IZUSVR2   00FE   00FE   YES      20 07/19/2019 14:
      IBMJES2  0000 JRL4      JRL4      006D   006D   YES       1 07/20/2019 04:
      IBMJES2  0000 JRL5      JRL5      0100   0100   YES       1 07/20/2019 04:
      IBMJES2  0000 JRL6      JRL6      006D   006D   YES       1 07/20/2019 04:
      IBMJES2  0000 JRL3      JRL3      010C   010C   YES       1 07/20/2019 04:
      IBMJES2  0000 MWALLE    MWALLE    0048   0048   YES       6 07/20/2019 14:
```

EVENTDESC:

4th position, 1 = //*MAIN statement

BCP: Generic Tracker

What: Generic Tracker is a facility that an help assess usage of specific requestable incidents that happen on a system. (z/OS MVS Diagnostics: Tools and Service Aids.)

- It is composed of:
  - A callable tracking service (GTZTRACK).
  - A callable query service to extract previously stored records (GTZQUERY).
  - Operator commands to display and maintain information and configuration.
  - Batch utility program (GTZPRINT)
  - Parmlib members for desired customization (GTZPRMxx).
- *Why it is helpful?* This facility can help with knowing when something is used (when it might be deleted in a future release), or to position for a new function (when changes might be required to use it).
- GTZ will be automatically started at IPL, however by default, it is not enabled for capturing data.

How to use:

1. Enable GTZ. SETGTZ TRACKING=ON
2. Check on collected information: DISPLAY GTZ or with GTZPRINT

z/OS V2.1 with OA50845 (March 2017) and higher:

BCP: WLM support for Special Engine Containment and Memory Capping

What:

- New workloads (especially those that come from other platforms and are zIIP-eligible) can demand significant processor capacity which might impact traditional workload running on regular CPs.
  - Previously this containment was only controllable system-wide via IEAOPTxx's IIPHONORPRIORITY.
  - WLM's "Honor Priority by Service Class" – prevents overflow from zIIPs to CPs at an individual service class.

Thanks to Mike Shorkend again for this suggestion!

- Likewise, this new workload might want to use a significant amount of memory which might impact traditional workload.
  - WLM's "Memory Limit by Resource Group"

How to use:

- New field "Honor Priority" on the Create and Modify Service class panels.
  - `NO` means do not spill over, `DEFAULT` means to use IIPHONORPRIORITY.
- New field "Memory Limit" on the Create and Modify a Resource Group panel. Can specify an upper memory limit in GB for resource groups and associated address spaces with those resource groups through classification on the local system.

z/OS V2.1 with OA50845 (March 2017) and higher:

BCP: WLM support for Special Engine Containment and Memory Capping

Via z/OSMF WLM Plug-in for Honor Priority (1 of 2):

Welcome | Workload Manage...

Workload Management

Overview | Service Definitions | Modify SHARPLEX

Service Definitions

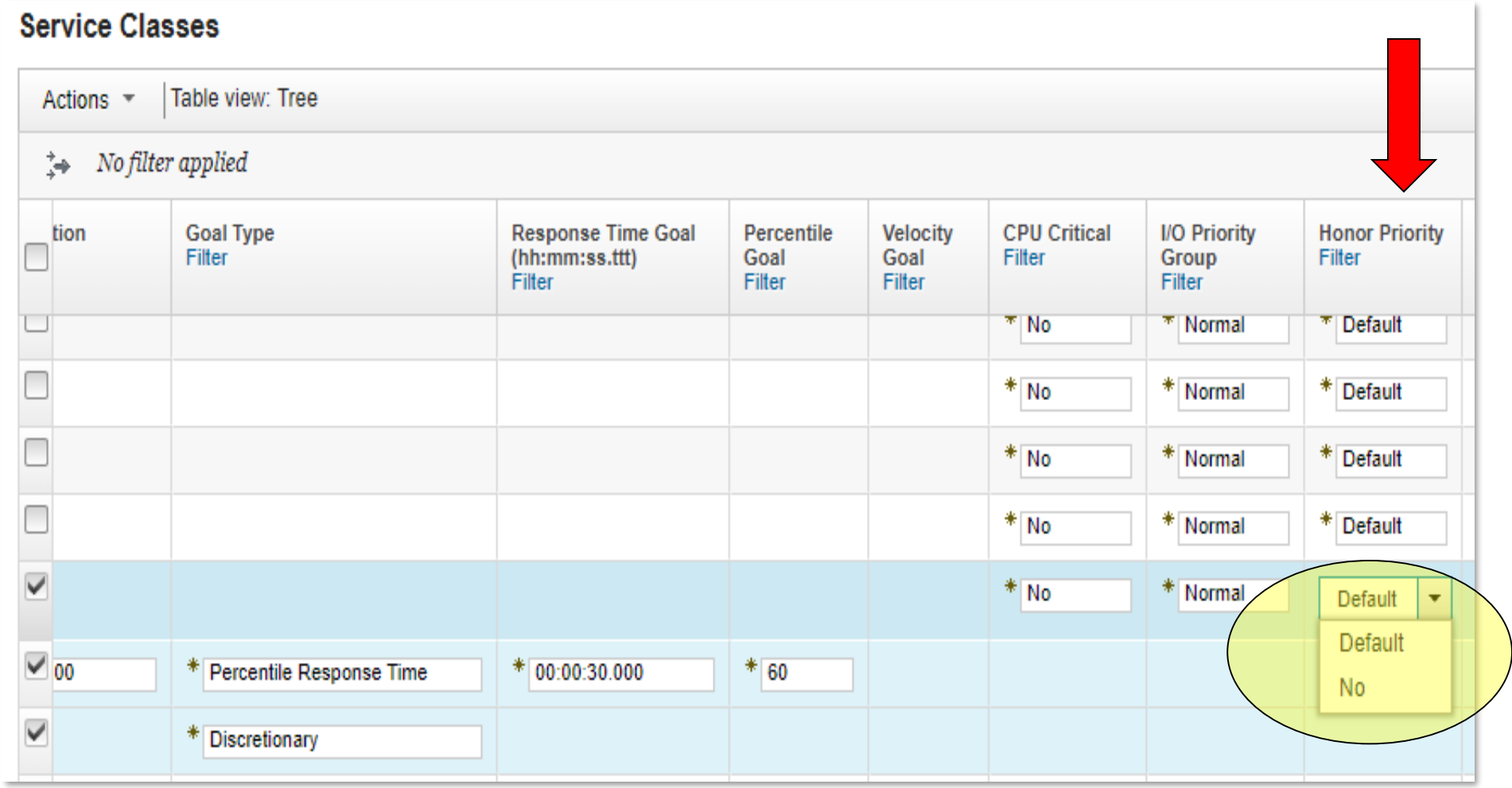
Actions

- Modify Service Definition ▸
  - Service Definition Details
  - Service Policies
  - Workloads
  - Service Classes
  - Resource Groups
  - Report Classes
  - Classification Groups
  - Classifications
  - Application Environments
  - Resources
  - Scheduling Environments
  - Tenant Resource Groups
  - Tenant Report Classes
- View Service Definition ▸
- View Messages
- View History
- Print Preview
- Install and Activate...
- Copy...
- Delete...
- Export ▸
- View WLM Status
- New...
- Import ▸
- Select All
- Deselect All
- Configure Columns...
- Hide Filter Row
- Clear Sorts
- Clear Search

		Activity Filter	Sysplex Filter
		Being modified	SHARPLEX
	WLMPROD to use for z/OSMF lab		
	WLMTEST to use for z/OSMF lab		
	Test Lab		
WLMTST04			

BCP: WLM support for Special Engine Containment and Memory Capping

Via z/OSMF WLM Plug-in for Honor Priority (2 of 2):



z/OS V2.1 with OA50845 (March 2017) and higher:

BCP: WLM support for Special Engine Containment and Memory Capping

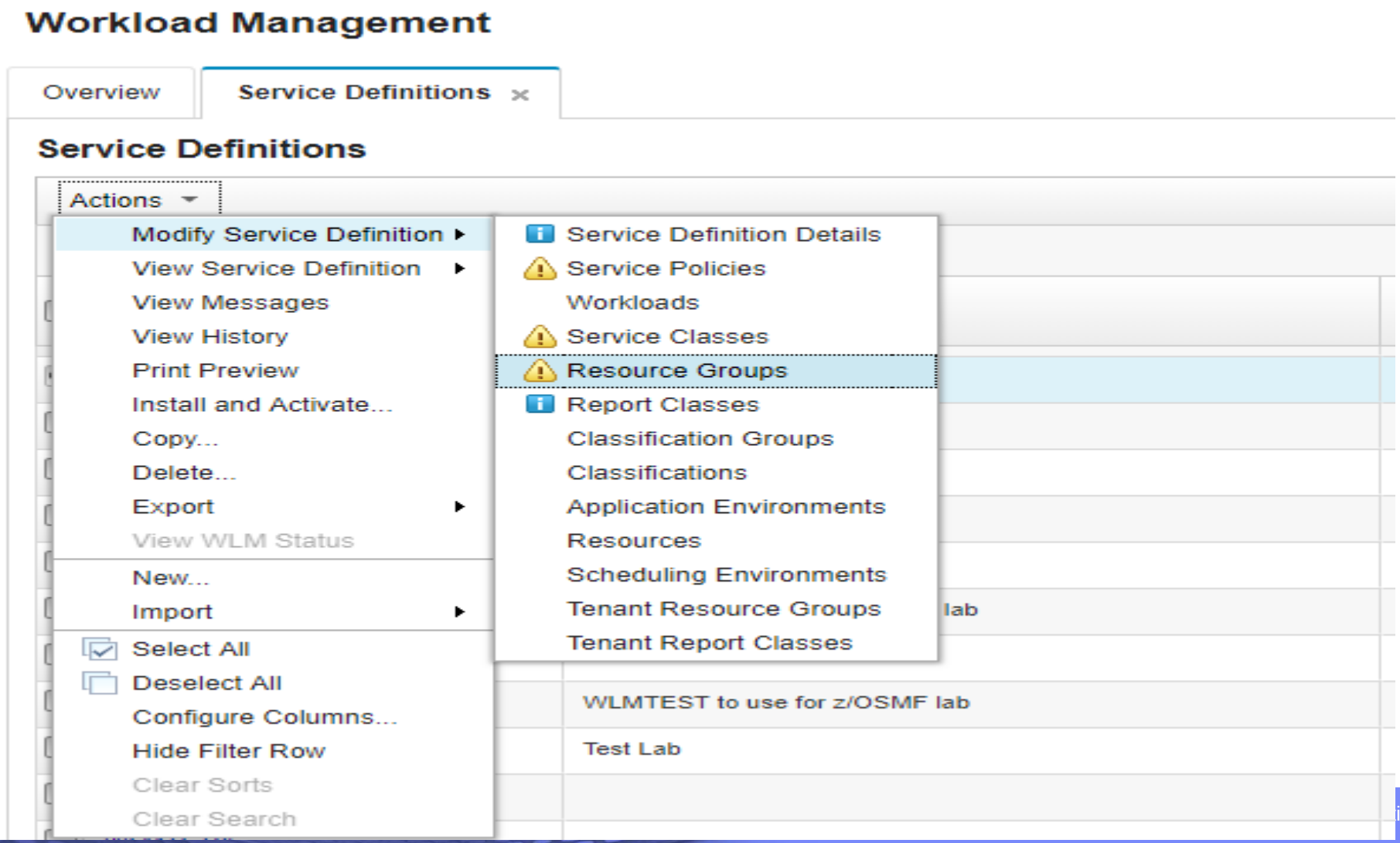
Via WLM ISPF Panels for Honor Priority :

```
  Service-Class  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                          Modify a Service Class                        Row
Command ===> ____________________________________________________________

Service Class Name . . . . . : WEBSLOW
Description  . . . . . . . . . WebServer Slow_____________________
Workload Name  . . . . . . . . IMWEB___   (name or ?)
Base Resource Gro  . . . . . . _________   (name or ?)
Cpu Critical . . . . . . . . . NO_        (YES or NO)
I/O Priority Gr p . . . . . . NORMAL      (NORMAL or HIGH)
Honor Priority . . . . . . . . DEFAULT    (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

         -- Period --  ------------------- Goal --------------------
Action   #  Duration    Imp.  Description
  __     _  _________   _     ____________________________________________
  __     1  3600        3     60% complete within 00:00:30.000
  __     2  _________   _     Discretionary
****************************** Bottom of data ********************************
```

z/OS V2.1 with OA50845 (March 2017) and higher:

BCP: WLM support for Special Engine Containment and Memory Capping

Via z/OSMF WLM Plug-in for Memory Limit (1 of 2):



z/OS V2.1 with OA50845 (March 2017) and higher:

BCP: WLM support for Special Engine Containment and Memory Capping

Via z/OSMF WLM Plug-in for Memory Limit (2 of 2):

Workload Management

Overview | Service Definitions × | Modify WLMPR03 ×

Add comments to histor

Resource Groups

Actions

No filter applied

Name Filter	Type Filter	Capacity Minimum Filter	Capacity Maximum Filter	Include Specialty Processor Consumption Filter	Memory Limit (GB) Filter	Descr Filter
ECPE	* NumberCPsTimes100	100	100	No		AVT
ECPMAX	* NumberCPsTimes100		80	No		AVT
ECPMIN	* NumberCPsTimes100	50		No		AVT
ELPAR	* PercentageLPARShare	50	60	No		AVT
ELPMAX	* PercentageLPARShare		60	No		AVT
ELPMIN	* PercentageLPARShare	70	80	No	400	AVT
EWLM	* CPUServiceUnits	5000	15000	No		Resc

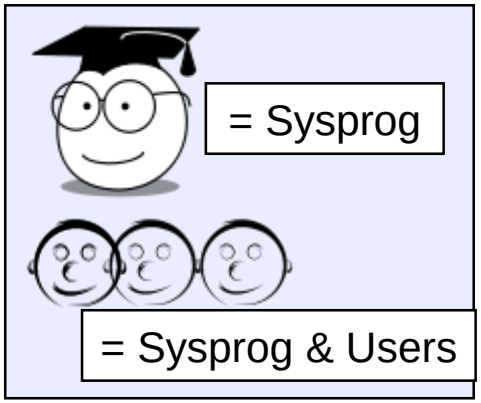
BCP: WLM support for Special Engine Containment and Memory Capping

Via WLM ISPF Panels for Memory Limit :

```
  Resource-Group  Xref  Notes  Options  Help
------------------------------------------------------------------------------
                          Modify a Resource Group
Command ===>

Enter or change the following information:

Resource Group Name  . . . . . : MWPIG
Description  . . . . . . . . . . Test Resource Group for a pig

Define Capacity:
2    1.  In Service Units (Sysplex Scope)
     2.  As Percentage of the LPAR share (System Scope)
     3.  As a Number of CPs times 100 (System Scope)
     4.  In accounted workload MSU (Sysplex Scope)
Minimum Capacity  . . . . . . . . . . . . . . . 50
Maximum Capacity  . . . . . . . . . . . . . . . 80
Include Specialty Processor Consumption NO          (YES or NO)

Memory Limit (System Scope) . . . . . . . 400_____  GB
```

Small Enhancements

❖BCP: D IPLINFO for IEASYSxx (and MACHMIG)



BCP: D IPLINFO for IEASYSxx (and MACHMIG)

What: New IEASYSxx statements have been added recently.

- Did you know that any IEASYSxx statement can be used on the D IPLINFO command to see if you are using them?

How to use:

- `D IPLINFO,`*`sysparm`*
- Also nice for `D IPLINFO,LOADXX,MACHMIG`
  - Note the value is `LOADXX` , regardless of your actual LOAD suffix.
- Look at response of IEE255I or IEE580I.

Considerations: This gives you a quick starting place to identifying what is in use, however, you still need to issue component-specific commands for more details (e.g. `D IZU`).

- It is great for “one value” investigation.

Older than the hills:

BCP: D IPLINFO for IEASYSxx (and MACHMIG)

```
-D IPLINFO,OSPROTECT
 IEE255I SYSTEM PARAMETER 'OSPROTECT': SYSTEM
```

```
-D IPLINFO,IZU
 IEE255I SYSTEM PARAMETER 'IZU': 01
```

```
-D IPLINFO,IEFOPZ
 IEE255I SYSTEM PARAMETER 'IEFOPZ': NOT_SPECIFIED
```

```
-D IPLINFO,LOADXX,MACHMIG
 IEE580I  20.49.10 LOADXX MACHMIG 346
 NO MACHMIG STATEMENTS WERE PROVIDED
```

Summary of What We Might Want to Share:

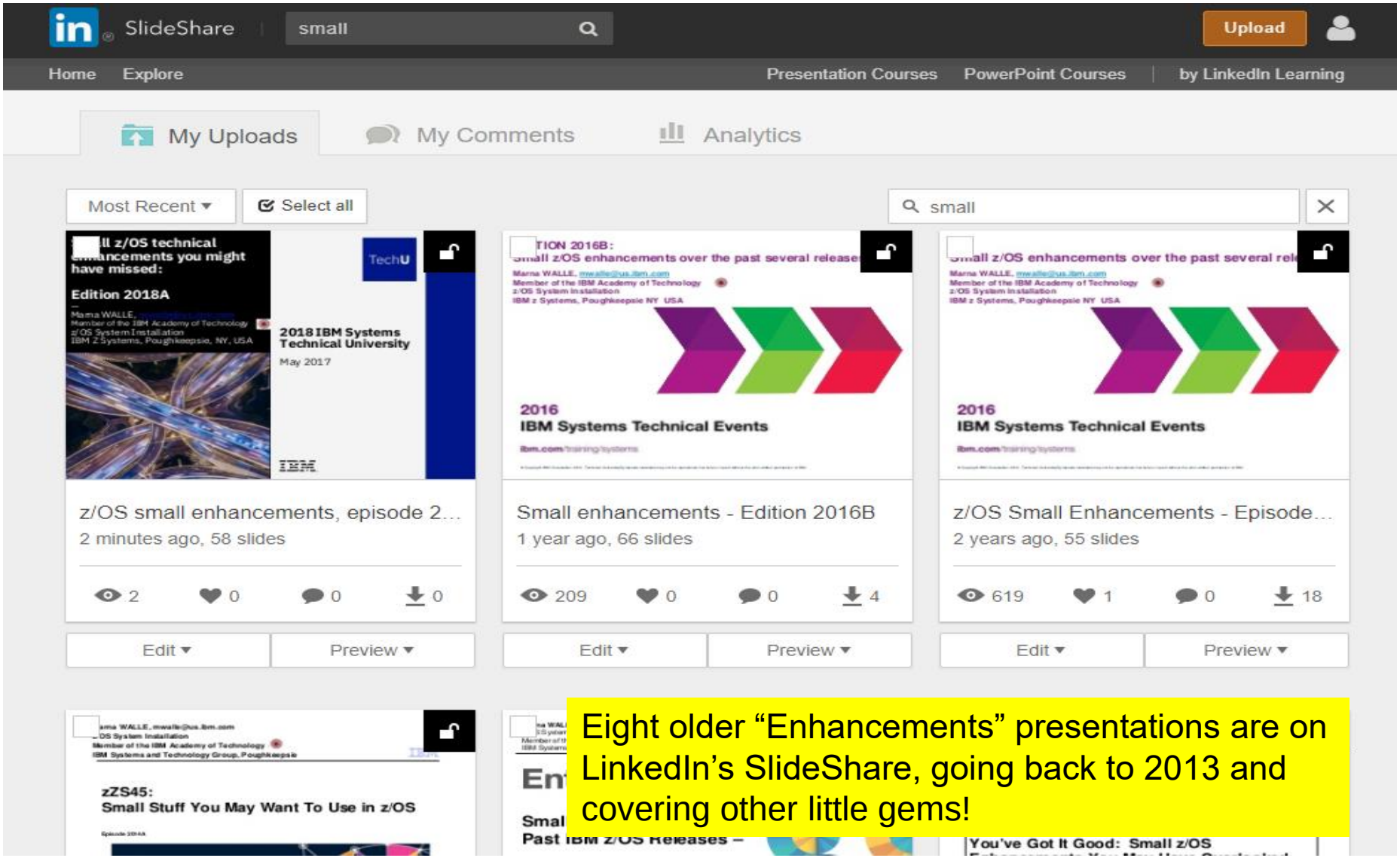
- ***System Programmer & User Items:***
  - **BCP (V2.3): GDGBIAS**
  - **z/OS RACF and UNIX (V2.3): user ID for UID(0) consistency**
  - **z/OSMF (V2.2): Swagger support**

- ***System Programmers' Items:***
  - **z/OSMF (V2.3): Looking at z/OSMF server parameters**
  - **z/OSMF (V2.3): More information on the angel for z/OSMF**
  - **z/OS UNIX (V2.3): BPXPRMxx VERSION UNMOUNT**
  - **BCP (V2.2): HZSPRMxx filter**
  - **BCP (V2.2): HZSPRMxx syntax check**
  - **BCP (V2.1): PDUU support for HTTPS**
  - **BCP (V2.1): Generic Tracker**
  - **BCP (V2.1): WLM support for Special Engine Containment and Memory Capping**
  - **BCP: D IPLINFO for IEASYSxx (and MACHMIG)**

z/OS Summary Enhancements – Edition 2019B

- **z/OS V2.3:**
  - **BCP: GDGBIAS** Can be very useful for restarting jobs
  - **z/OSMF: Looking at z/OSMF server parameters** Config evaluation, in a traditional way
  - **z/OSMF: More information on the angel for z/OSMF** For Liberty levels used
  - **z/OS UNIX: SUPERUSER for consistent username** Always the same user ID
  - **z/OS UNIX: Automatic VERSION UNMOUNT** Autonomic manamagent of fs
- **z/OS V2.2:**
  - **BCP: HZSPRMxx filter** Consolidate your parmlib members across your enterprise
  - **BCP: HZSPRMxx syntax check** Validate your syntax before using
  - **z/OSMF Swagger support** Incredibly useful and helpful for REST API programs
  - **z/OS UNIX: TRNG for /dev/(u)randow** helps with simplificiation
  - **z/OS UNIX: updates to zlsof, and adding jsonprint** see more and use with programs
  - **SDSF: snapshot** on tabular panels, find information fast
- **z/OS V2.1:**
  - **BCP: PDUU HTTPS support** For those that prefer HTTPS over FTP
  - **BCP: Generic Tracker** Helpful for upgrading and exploiting new functions
  - **BCP: WLM support for Specialty Engine Containment and Memory Capping** Limit those new guys in the pool.
- **Older than the hills: BCP: D IPLINFO for IEASYSxx (and MACHMIG)** Quick lookup.

z/OS Little Enhancements - A history



Eight older “Enhancements” presentations are on LinkedIn’s SlideShare, going back to 2013 and covering other little gems!