z/OS Little Enhancements: Many Small Potatoes Can Make a Big Meal

Marna WALLE
mwalle@us.ibm.com
IBM Poughkeepsie
z/OS System Installation

August 16, 2013

Permission is granted to SHARE to publish this presentation paper in the SHARE proceedings; IBM retains the right to distribute copies of this presentation to whomever it chooses.
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/390®, 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 products 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 Little Enhancements: Many Small Potatoes Can Make a Big Meal!**

- **z/OS V2.1:**
  - BCP: Add and remove MCS consoles dynamically
  - BCP: Dynamic System Symbol Support
  - BCP: DISPLAY PPT
  - BCP: BCP Parmlib Comments

- **z/OS V1.13:**
  - z/OS UNIX: Prevent Content Overlay during MOUNT
  - DFSMS: IEBPDSE Batch Program
  - SDSF: Cursor-sensitive Sort

- **z/OS V1.12:**
  - BCP: IEFSSNxx BEGINPARALLEL
  - BCP: Timed Event Data Report
  - BCP: Some PROGxx Goodies (in handout only)
  - DFSMS: IDCAMS DELETE All Members, plus more!

- **z/OS V1.11:**
  - TSO/E: LOGONHERE reconnect support
  - BCP: D ALLOC and SETALLOC commands

- Older than dirt on potatoes:
  - DFSMS: STGADMIN.DPDSRN._oldname
  - ISPF: Member Search
z/OS V2R1 Enhancements

Small Enhancements of System Programmer Interest

- **BCP**: Add and remove MCS consoles dynamically
- **BCP**: Dynamic system symbol support
- **BCP**: DISPLAY PPT
- **BCP**: BCP parmlib comments
**BCP:** Add and remove MCS consoles dynamically

- **What:** The ability to add and delete consoles without an outage using operator commands, *when in Distributed mode.***
- **How to use:**
  - `SET CON=xx` processes operational settings and creates new consoles found in that CONSOLxx parmlib member
    - New SMCS or subsystem consoles will be defined sysplex-wide (even pre-V2R1)
    - Cannot add the system console (SYSCON) dynamically
    - Only specified statements will be processed (absence will not mean default)
  - `SETCON DELETE, CN=nnnnnnnn` deletes that inactive console from the sysplex
    - Applies to MCS, HMCS, SMCS, Subsystem, as well as, EMCS consoles
    - Can delete an inactive console from a system that didn’t define it (even pre-V2R1)
- **Considerations:** May choose to replace samplib program IEARELCN (removes inactive EMCS consoles) with SETCON DELETE.
  - If you decide to use the specified CONSOLxx parmlib members for subsequent IPLs, you’ve hardened the usage of the consoles you added.
Example: remove an inactive MCS console, then add it back in.
What: The ability to add or change system symbols in a supported way on a local system. Two new methods are provided to do this: SETLOAD xx, IEASYM and IEASYMU2.

How to use:

- **SETLOAD xx, IEASYM**: This indicates to process the IEASYM statement in LOADxx found in your parmlib concatenation. (There is a SETLOAD xx, IEASYM, DSNAME=dd, VOLUME=vv. if you want to point to a data set outside the parmlib concatenation.)
  - A new complete system symbol table is built. The prior system symbol table remains allocated. Therefore, it’s better to do fewer rebuilds than more rebuilds.

- **IEASYMU2** replaces **IEASYMUP**. IEASYMU2 is similar to IEASYMUP, in that you can run a batch job to update the system symbols. The same FACILITY class profile (IEASYMUP.*) is used.
  - However, IEASYMU2 changes (or a continued use of IEASYMUP, for that matter) will **not** be reflected when you do a subsequent SETLOAD xx, IEASYM.
  - No changes in the limit in the number of system symbols: remains at least 800 symbols.

Considerations: Don’t use the old unsupported method (IEASYMUP) anymore. Understand the interactions between SETLOAD xx, IEASYM, IEASYMU2, and IEASYMUP, as your IEASYMU2 / IEASYMUP changes will be lost when a SETLOAD xx, IEASYM is done.

- If you decide to use specified LOADxx parmlib member for subsequent IPLs, you’ve hardened the usage the symbols you’ve changed with the SETLOAD xx, IEASYM.
z/OS V2R1 Enhancements

**BCP:** A BONUS system symbol - &SYSOSLVL !!

- **What:** A new system-defined symbol to indicate the z/OS release.
  - The format for z/OS is: Z1vvrrmm
  - z/OS V2 R1 is: z1020100

- **How to use:**
  - Use it as you wish and where allowed to: in data set names, etc.

- **Considerations:** If you defined a system symbol to do this yourself, consider using the system-defined one instead. z/OS defines six for you.

```
00- SY1 d symbols
SY1 IEA007I STATIC SYSTEM SYMBOL VALUES 840
   &SYSALVL. = "2"
   &SYSCLONE. = "Y1"
   &SYSNAME. = "SY1"
   &SYSOSLVL. = "Z1020100"
   &SYSPLEX. = "LOCAL"
   &SYSR1. = "ZDR21"
```

What: A new system command to display the currently effective program properties table (PPT). Including options to show:
- **ALL:** all the entries in the PPT
- **PARMLIB:** entries specified from SCHEDxx
- **DEFAULT:** entries that are IBM-supplied defaults and have not been re-specified by SCHEDxx
- **NAME=pattern:** entries that match a pattern (wildcards accepted)

How to use: `D PPT` or `D PPT,options`

Example:

```
D PPT                                         IEF386I 20.58.57 DISPLAY PPT 837              Parmlib Values                ...        NS   Non-swappable                         NOSWAP          PR   Privileged                            PRIV
```

z/OS V2R1 Enhancements

**BCP:** `DISPLAY PPT`

**Reference**

<table>
<thead>
<tr>
<th>Synonym</th>
<th>Meaning</th>
<th>SCHEDxx keyw</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC</td>
<td>Non-cancelable</td>
<td>NOCANCEL</td>
</tr>
<tr>
<td>NS</td>
<td>Non-swappable</td>
<td>NOSWAP</td>
</tr>
<tr>
<td>PR</td>
<td>Privileged</td>
<td>PRIV</td>
</tr>
</tbody>
</table>
z/OS V2R1 Enhancements

BCP: Parmlib comments

- **What:** The ability to add comments to certain parmlib members, throughout the member. This allows for better documentation of the member (and can be less error-prone).
  - Previously, some parmlib members only allowed comments at the end of the line, or at the end of the member, or possibly no comments at all.
  - For Parmlib members COMMNDxx, GTFPARM, IEAABD00, IEACMD00, IEADMP00, IEADMRO00, IEAPAKxx, IEASYSxx, LPALSTxx, VATLSTxx.

- **How to use:** Put an asterisk in column 1. That line is ignored.

- **Considerations:** With OA38328 back to R12, for parmlib sharing.

- **Example:**
  ```
  SYS1.PARMLIB.POK(COMMNDMW) - 01.00
  Command ==>>
  ***** ***************************************** Top of Data **********
  COLS> ------1--------2--------3--------4--------5--
  00001 **********************************************
  00002 * USE MPFLSTAI for MPF Table     Marna 7/31/2013 *
  00003 **********************************************
  00004 COM='SET MPF=AI'
  ***** ***************************************** Bottom of Data **********
  ```
z/OS R13 Enhancements
Small Enhancements of System Programmer Interest

- **z/OS UNIX**: Prevent Content Overlay during MOUNT
- **DFSMS**: IEBPDSE Batch Program
- **SDSF**: Cursor-sensitive Sort
What: Have you ever “lost” updates to files when they’ve been over-mounted? Do you want the system to let you know when you’re mounting over something? Now you can have the system warn or even deny a mount on a mountpoint that is not empty.

How to use: Specify WARN or DENY (default is NOWARN, same as today’s behavior) on BPXPRMxx or on SETOMVS NONEMPTYMOUNTPT statement. `D OMVS,OPTIONS shows current setting.

Considerations: WARN produces a syslog message (not back to user). DENY goes back to the user.
  – Advanced consideration: Nonprivileged User Mount (in R13) must occur on an empty mountpoint (will act like a “DENY”).
### z/OS UNIX: Prevent Content Overlay during MOUNT

- **WARN Usage Example:**
  ```
  SETOMVS NONEMPTYMOUNTPT=WARN
  BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.
  BPXO062I NONEMPTYMOUNTPT WAS CHANGED FROM NOWARN TO WARN.
  ```

  Then do a mount:

  ```
  READY
  MOUNT FILESYSTEM('IBMUSER.PRODUCT.ZFS')
    MOUNTPOINT('/samples/') TYPE(ZFS) MODE(RDWR)
  READY
  ```

  User saw no warning here!

  However, in the syslog:

  ```
  0290  BPXF263I FILE SYSTEM 901
  0290  IBMUSER.PRODUCT.ZFS
  0290  HAS BEEN MOUNTED ON A NONEMPTY DIRECTORY
  0290  IEF106I IEF237I 0290 0800 ALLOCATED TO SYS00050
  ```
DENY Usage Example:

SETOMVS NONEMPTYMOUNTPT=DENY
BPXO015I THE SETOMVS COMMAND WAS SUCCESSFUL.
BPXO062I NONEMPTYMOUNTPT WAS CHANGED FROM WARN TO DENY.

Then do a mount:

A bpxmtext 063C gives:
Notice: unknown modid, reason text may be incorrect
JrNonEmptyMntPtDir: The mount point directory is not empty.
Action: Retry the mount on an empty mount point directory.
***
- DENY Usage Example:

Or from ISHELL:

```
Mounted with DENY:
Errno=88x The directory is not empty; Reason=055B063C. Press Enter to continue.
```
What: Want to check the structural integrity of a PDSE? It’s a good idea to verify a PDSE’s integrity before it’s deployed, and a problem would be propagated. A new utility can help you do that.

How to use: Invoke PGM=IEBPDPSE, with SYSLIB indicating your PDSEs to verify.

Considerations: If the DUMP parameter option is specified, the PDSE validation utility issues an ABEND in the PDSE address space, which results on an SVC dump. Utility output is meant to tell you if the PDSE was valid or not – it does not correct the problem, nor is intended for you to correct the PDSE yourself. A dump would help IBM Service to diagnose the problem.

Usage Example:

```plaintext
//STEPCHK EXEC PGM=IEBPDPSE,PARM='DUMP'
//SYSPRINT DD SYSOUT=*                    
//SYSLIB DD DSN=SYS1.SIEALNKE,DISP=SHR   
// DD DSN=MWALLE.UTIL.JOBS,DISP=SHR
```

Successful output:

```
************************************************  TOP OF DATA ***
IGW700I PDSE Directory Validation Successful
DSN:SYS1.SIEALNKE
ADPages:36 IXRecords:1054
NDPages:6 IXRecords:315
AD ND Tree Nodes:315
ingw700i PDSE Directory Validation Successful
DSN:MWALLE.UTIL.JOBS
ADPages:65 IXRecords:3985
NDPages:29 IXRecords:1590
AD ND Tree Nodes:1590
************************************************  BOTTOM OF DATA *
```
z/OS R13 Enhancements

**DFSMS: IEBPDSE Batch Program**

- **Usage Example:**
  
  Unsuccessful output:

  ********************************** TOP OF DATA ***

  IGW702I PDSE Directory Validation Unsuccessful
  DESC:<ND> Structure is corrupted
  LTK:
  00000000000000000000000000000000000000000000000000000000000000000*
  ERROR NUM:3
  DSN:IBMUSER.PDSEVAL.PDSE1
  VOLSER:338001
  RC:8 RS:01198018 R14:040130A8
  RPN:0
  VPTVFN:N/A
  IGW699I PDSE Directory Validation Unsuccessful
  DESC:PDSE structure is corrupted
  ERROR NUM:101
  DSN:IBMUSER.PDSEVAL.PDSE1
  VOLSER:338001
  ADPages:1 IXRecords:14
  NDPages:0 IXRecords:0
  RC:8 RS:01198018
  IGW702I PDSE Directory Validation Unsuccessful
  DESC:<ND> Structure is corrupted
  LTK:
  00000000000000000000000000000000000000000000000000000000000000000*
  ERROR NUM:23
  DSN:IBMUSER.PDSEVAL.PDSE1
  VOLSER:338001
  RC:8 RS:01198018 R14:04012F8E
  RPN:0
  VPTVFN:N/A

  ********************************** BOTTOM OF DATA *
What: Under ISPF, you now have “point and shoot” support. Meaning, you can sort on a column by tabbing the cursor to the column title and pressing Enter.

How to use:
- Tab to column title, hit Enter. Easy!
- Enter toggles through ascending, descending, then original order.

Considerations:
- Can control it with SDSF command `SET CSORT OFF` or `ON`.
- Make sure you have “Tab to point-and-shoot fields” enabled under ISPF Settings in Option 0.
### z/OS R13 Enhancements

**SDSF: Cursor Sensitive Sort**

- **Usage Example:**
  
  **Before:**

  ![SDSF Output Example](image)

<table>
<thead>
<tr>
<th>NP</th>
<th>JOBNAME</th>
<th>JobID</th>
<th>Owner</th>
<th>Prty</th>
<th>C</th>
<th>ODisp</th>
<th>Dest</th>
<th>Tot-Rec</th>
<th>Tot-Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MWALLEAT JOB15623 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,727</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB20279 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,772</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB18568 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,793</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB00102 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,768</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB09312 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,750</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB21325 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,751</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAT JOB21530 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>157,807</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEP JOB29260 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEP JOB29281 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LISTCSI JOB31937 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>264</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLERM JOB31939 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>119</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLERM JOB31942 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MWALLEAP JOB31958 MWALLE 7 H HOLD LOCAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4,615,948</td>
<td></td>
</tr>
</tbody>
</table>
Usage Example:
First enter (ascending):

```
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 5,720,943 LINE 1-13 (13)
COMMAND INPUT ===> SCROLL =-9 HALF
PREFIX=* DEST=(ALL) OWNER=MWALLE SORT=Tot-Rec/A SYSNAME=
NP JOBNAME JobID Owner Prty C ODisp Dest Tot-Rec Tot-
MWALLEP JOB29260 MWALLE 7 H HOLD LOCAL 62
MWALLEM JOB31942 MWALLE 7 H HOLD LOCAL 87
MWALLEP JOB29281 MWALLE 7 H HOLD LOCAL 95
MWALLEM JOB31939 MWALLE 7 H HOLD LOCAL 119
LISTCST JOB31937 MWALLE 7 H HOLD LOCAL 264
MWALLEAT JOB15623 MWALLE 7 H HOLD LOCAL 157,727
MWALLEAT JOB09312 MWALLE 7 H HOLD LOCAL 157,750
MWALLEAT JOB21325 MWALLE 7 H HOLD LOCAL 157,751
MWALLEAT JOB00112 MWALLE 7 H HOLD LOCAL 157,768
MWALLEAT JOB20279 MWALLE 7 H HOLD LOCAL 157,772
MWALLEAT JOB18568 MWALLE 7 H HOLD LOCAL 157,793
MWALLEAT JOB21530 MWALLE 7 H HOLD LOCAL 157,807
MWALLEAP JOB31958 MWALLE 7 H HOLD LOCAL 4,615,948
```
**z/OS R13 Enhancements**

**SDSF: Cursor Sensitive Sort**

- **Usage Example:**

  Second enter (descending):

  ```
  SDSF HELEI OUTPUT DISPLAY ALL CLASSES LINES 5,720,943 LINE 1-13 (13)
  COMMAND INPUT ===>
  PREFIX=* DEST=(ALL) OWNER=MWALLE SORT=Tot-Rec/D SYSNAME=
  NP  JOBNAME  JobID  Owner   Prty  ODisp  Dest        Tot-Rec  Tot-
  MWA00398  J0B31958  MWA00398  7  H  HOLD  LOCAL  4,615,948
  MWA00398  J0B21530  MWA00398  7  H  HOLD  LOCAL  157,930
  MWA00398  J0B18568  MWA00398  7  H  HOLD  LOCAL  157,793
  MWA00398  J0B20279  MWA00398  7  H  HOLD  LOCAL  157,727
  MWA00398  J0B00102  MWA00398  7  H  HOLD  LOCAL  157,688
  MWA00398  J0B21325  MWA00398  7  H  HOLD  LOCAL  157,751
  MWA00398  J0B09312  MWA00398  7  H  HOLD  LOCAL  157,750
  MWA00398  J0B15623  MWA00398  7  H  HOLD  LOCAL  157,727
  MWA00398  J0B09318  MWA00398  7  H  HOLD  LOCAL  264
  MWA00398  J0B31939  MWA00398  7  H  HOLD  LOCAL  119
  MWA00398  J0B29281  MWA00398  7  H  HOLD  LOCAL  95
  MWA00398  J0B31942  MWA00398  7  H  HOLD  LOCAL  87
  MWA00398  J0B29260  MWA00398  7  H  HOLD  LOCAL  62
  ```
**Usage Example:**

Third enter (original):

```
PREFIX=* DEST=(ALL) OWNER=MWALLE SYSNAME=
NP JOBNAME JobID Owner Prty C ODisp Dest Tot-Rec Tot-
MWALLEAT JOB15623 MWALLE  7 H HOLD LOCAL  157,727
MWALLEAT JOB20279 MWALLE  7 H HOLD LOCAL  157,772
MWALLEAT JOB18568 MWALLE  7 H HOLD LOCAL  157,793
MWALLEAT JOB00102 MWALLE  7 H HOLD LOCAL  157,768
MWALLEAT JOB09312 MWALLE  7 H HOLD LOCAL  157,750
MWALLEAT JOB21325 MWALLE  7 H HOLD LOCAL  157,751
MWALLEAT JOB21530 MWALLE  7 H HOLD LOCAL  157,807
MWALLEP JOB29260 MWALLE  7 H HOLD LOCAL   62
MWALLEP JOB29281 MWALLE  7 H HOLD LOCAL   95
LISTCSI JOB31937 MWALLE  7 H HOLD LOCAL  264
MWALLERM JOB31939 MWALLE  7 H HOLD LOCAL  119
MWALLEAM JOB31942 MWALLE  7 H HOLD LOCAL   87
MWALLEAP JOB31958 MWALLE  7 H HOLD LOCAL 4,615,948
```

Arrow pointing to ALF.
z/OS R12 Enhancements
Small Enhancements of System Programmer Interest

- **BCP**: IEFSSNxx BEGINPARALLEL
- **BCP**: Timed Event Data Report
- **BCP**: Some PROGxx Goodies (in handout only)
- **DFSMS**: IDCAMS DELETE Members
What: In order to help with Mean Time To Recovery (MTTR), we want to reduce initialization paths where possible. With this in mind, you can specify that Subsystem Initialization Routines be run in parallel.

How to use: Code `BEGINPARALLEL` in IEFSSNxx at the point in which you would like the SSI routines to be executed in parallel. Everything before `BEGINPARALLEL` will be executed serially, as before.

Considerations:
- The order that the parallelized initialization routines are run is now unpredictable. Therefore, the routines must have not any execution order dependencies.
- SMS should be started before the `BEGINPARALLEL` statement. Also, Communications Server’s VMCF and TNF subsystems should be specified before `BEGINPARALLEL`. Check with the subsystem product documentation to see if it should be placed before or after `BEGINPARALLEL`.
- How much benefit you get will depend on how many initialization routines you have, the complexity of the routines, serialization requirements of routines, and available CPs. If a subsystem doesn’t specify an initialization routine, then there is no effect on that subsystem.
- Duplicate specifications of `BEGINPARALLEL` (within multiple concatenated IEFSSNxx members, for instance) result in subsequent specifications being rejected.
z/OS R12 Enhancements

BCP: BEGINPARALLEL in IEFSSNxx

- Usage Example*:

  `BROWSE SYS1.PARMLIB(IEFSSNS2) - 01.25 Command ===>`

  `SUBSYS SUBNAME(JES2) PRIMARY(YES) START(NO)`
  `SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)`
  `  INITPARM('ID=99,PROMPT=DISPLAY')`
  `SUBSYS SUBNAME(BFL) INITRTN(BFLSSI)`
  `SUBSYS SUBNAME(JESA)`
  `SUBSYS SUBNAME(JESB)`
  `SUBSYS SUBNAME(TNF) INITRTN(MVPTSSI)`
  `SUBSYS SUBNAME(VMCF) INITRTN(MVPXSSI) INITPARM(&SYSNAME.EIP)`
  `BEGINPARALLEL`

  `SUBSYS SUBNAME(CSQB) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,!MQSB0,S')`
  `SUBSYS SUBNAME(CSQD) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,!MQSD0,S')`
  `SUBSYS SUBNAME(CSQF) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,!MQSF0,S')`
  `SUBSYS SUBNAME(CSQH) INITRTN(CSQ3INI) INITPARM('CSQ3EPX,!MQSH0,S')`

* on nine of our test systems in Poughkeepsie, we start 108 subsystem initialization routines in parallel each week!
What: A new authorized service, Timed Event Data Service (IEATEDS), allows a program to record events to a Timed Event Data Table: `REQUEST=REGISTER` and `REQUEST=RECORD`. (Hint: use and document a good description on your event!)

IBM-supplied REXX exec, IEAVFTED, can be used to produce a Timed Event Data Report from the Timed Event Data Table in either a TSO or IPCS environment.

As of z/OS R12, some components (XCF, GRS, JES2, and others) use the IEATEDS service. This should help you gather some Mean Time To Recovery statistics during IPL (such as the benefit of using the BEGINPARALLEL statement in IEFSSNxx).

– Other methods still help with gathering IPL statistics.

How to use: IEAVFTED is compiled REXX, and requires the full REXX compiler runtime libraries (from the REXX Library product).

– Note that the REXX Alternate Runtime Library incorporated into the z/OS product is not sufficient!

Several options on IEAVFTED, but most common may be the simple invocation from TSO/E of `=== > IEAVFTED DA('output_data_set')`

– Where ‘output_data_set’ must be the name of a pre-allocated data set with an LRECL of 512 and a RECFM of V or VB

Reference information: z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDT-IXG)
BCP: Timed Event Data Report

- **Considerations:** The IEAVFTED-produced report has two sections with headers: formatted entries, and spreadsheet data.
  - You can use the spreadsheet portion to import in a spreadsheet program. That makes it easy to sort on data – such as quickly finding out which events took the longest time.
  - Once you have the report downloaded:
    a) Delete everything in the top half (the formatted entries)
    b) Import to the spreadsheet program, indicating that the delimiter was a semicolon (or other delimiter that you indicated you wanted used when you created the report).

- **Usage Example:**
  1. Before using IEFSSNxx, from TSO:  

     ```
     ===> IEAVFTED DA('SAOUSER.TED.REPORT.SL0.BEFORE')
     IEAVFTED Processing Started
     IEAVFTED Processing Complete
     ***
     ```
  2. Incorporate BEGINPARALLEL in IEFSSNxx, and IPL at the next scheduled window
  3. After using IEFSSNxx, from TSO:  

     ```
     ===> IEAVFTED DA('SAOUSER.TED.REPORT.SL0.AFTER')
     IEAVFTED Processing Started
     IEAVFTED Processing Complete
     ***
     ```
IBM z/OS Timed Event Data Report
Level: HBB7770-V1.04 Report Date/Time: 8 Jan 2013 13:20:52 Component
Sysplex: UTCPLXSB System: SLO FMID: HBB7780 z/OS V01R13M00
Machine: 2817-0035FDF5 Online Standard CPs: 10 zAAPs: 2 zIIPs: 2
IPL Start Date/Time: 8 Jan 2013 11:18:31.078886

Total Timed Event Data Table Storage: 0000FE10

*
Sort by “Event Time” column to see the events in chronological order.

- Look for the event “Start” and “End” to see how long an event took.
- For BEGINPARALLEL exploitation: look for “SSN=subsystem” Start and End and see how long it took. Compare times on before and after spreadsheets to see any benefits!
Benefit = 6.151519 seconds for BEGINPARALLEL
**What:** Prior to z/OS R12, when you did an LPA ADD you had to also remember to include all the aliases for the module. As of z/OS R12, you can use ADDLIAS (not the default) to automatically include the aliases.

**Usage Example:**
- SETPROG LPA,ADD,MODNAME=(CNMCNETV),DSNAME=MWALLE.MY.LPAMODS,ADDALIAS
- Output:
  CSV551I 10.42.50 LPA ADD 925
  SUCCESSFUL: 2  UNSUCCESSFUL: 0  NOT PROCESSED: 0
  MODULE RESULT
  CNMCNETV SUCCESSFUL
  CNMNETV SUCCESSFUL

**What:** Prior to z/OS R12, when you wanted to replace a Dynamic Exit routine, you had to do a DELETE and then an ADD. This meant for some period of time that exit routine was not available on the system. As of z/OS R12, you can use REPLACE so that the exit routine is not unavailable for that period of time.

**Considerations:** You can’t REPLACE an exit routine that hadn’t been ADD’d. In other words, make sure the exit routine has been ADD’d before trying to do a REPLACE.

**Usage Example:**
- SETPROG EXIT,REPLACE,EXITNAME=IRREVX01,MODNAME=IRREVX1A,DSNAME=COMMON.LOOKFEEL.LINKLIB
- Output:
  CSV420I MODULE IRREVX1A HAS BEEN REPLACED FOR EXIT IRREVX01
What: Prior to z/OS R12, it was easy to make common mistakes on Dynamic LPA and LNKLST in PROGxx and SETPROG, and the DISPLAY PROG,EXIT command. As of z/OS R12, you can use the DEFAULTS statement in PROGxx to make it less error-prone.

Considerations: `D PROG,DEFAULTS` shows what you currently have.

Usage Example:

- To your PROGxx member add your preferences:
  ```
  DEFAULTS LPA ADDALIAS
  DEFAULTS LNKLST REQCOPYFROM COPYFROMCUR
  DEFAULTS EXIT EXITTYPE(INSTALLATION) (or ALL, NOTPROGRAM)
  ```

- SETPROG LPA,ADD,MODNAME=(CNMCNETV),DSNAME=MWALLE.MY.LPAMODS
  - Output:
    ```
    CSV551I 10.42.50 LPA ADD 925
    SUCCESSFUL: 2  UNSUCCESSFUL: 0  NOT PROCESSED: 0
    MODULE RESULT
    CNMCNETV SUCCESSFUL
    CNMNETV SUCCESSFUL
    ```

- SETPROG LNKLST DEFINE NAME(MARNALL)
  - Output: the MARNALL LNKLST set was copied from the current LNKLST.

- `D PROG,EXIT`
  - Output: same as the D PROG,EXIT,INSTALLATION output.
**z/OS R12 Enhancements (OK, really z/OS R11!)**

**DFSMS: IDCAMS DELETE MASK**

- **What:** As of z/OS R11, IDCAMS DELETE MASK allows you to delete more than one data set at a time by specifying multiple qualifiers (and within a qualifier). Previously you could only use a wildcard delete on a single qualifier (for instance, MWALLE.*.JOBS).
- **Considerations:** Only one data set mask can be provided at a time. **All** data sets matching the mask will be deleted – be careful what you ask for!
- **Usage Example:**
  ```
  //DELMEM   EXEC PGM=IDCAMS
  //SYSPRTN DD SYSOUT=*
  //SYSIN DD *
  DELETE MWALLE.TEST%%%.PDS* MASK
  /*
  ```
  - **Output:**
    ```
    DELETE MWALLE.TEST%%%.PDS* MASK
    IDC0550I ENTRY (A) MWALLE.TESTDEL.PDS1 DELETED
    IDC0550I ENTRY (A) MWALLE.TESTDEL.PDS10 DELETED
    IDC0550I ENTRY (A) MWALLE.TESTDEL.PDS100 DELETED
    /*(for all the data sets, which were more than 100!)
    ```
    - **Warning!** Don’t put MASK before the entry name...  
      DELETE MASK MWALLE.TEST%%%.PDS*
      IDC3211I KEYWORD 'MWALLE.TEST%%%.PDS*' IS IMPROPER
      IDC3202I ABOVE TEXT BYPASSED UNTIL NEXT COMMAND. CONDITION CODE IS 12
What: Prior to z/OS R12, IDCAMS DELETE could only delete only one member at a time. You’d have to invoke the DELETE command for each member you wanted to delete. Now, you can delete all members at once!

Usage Example:
- //DELMEM EXEC PGM=IDCAMS
- //SYSPRINT DD SYSOUT=* 
- //SYSIN DD *
- DELETE MWALLE.TESTDEL.MEMS(*)
- /*

Output:
- DELETE MWALLE.TESTDEL.MEMS(*)
- IDC0553I ALL MEMBERS IN DATA SET MWALLE.TESTDEL.MEMS DELETED
- IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
Before deleting all members:

```
Command ==> ...

Data Set Name : MWALLE.TESTDEL.PDS1

General Data
Volume serial : ZD18
Device type   : 3390
Organization  : P0
Record format : FB
Record length : 90
Block size    : 32760
1st extent blocks : 1
Secondary blocks : 1

Current Allocation
Allocated blocks : 16
Allocated extents : 16
Maximum dir. blocks : 1

Current Utilization
Used blocks : 16
Used extents : 16
Used dir. blocks : 1
Number of members : 5

Dates
Creation date : 2013/04/05
Referenced date : 2013/04/05
Expiration date : ***None***
```
DFSMS: IDCAMS DELETE All Members

After deleting all members:

```
Command ==> 
```

Data Set Information

Data Set Name . . . : MWALLE.TESTDEL.PDS1

General Data
Volume serial . . : ZDI18
Device type . . . : 3390
Organization . . : P0
Record format . . : FB
Record length . . : 90
Block size . . . : 32760
1st extent blocks : 1
Secondary blocks : 1

Current Allocation
Allocated blocks . : 16
Allocated extents . : 16
Maximum dir. blocks : 1

Current Utilization
Used blocks . . . : 1
Used extents . . . : 2
Used dir. blocks : 1
Number of members : 0

Dates
Creation date . . : 2013/04/05
Referenced date . : 2013/04/05
Expiration date . : ***None***
```
**What:** Note that on z/OS V2.1, we now have “partial” wildcard delete support!

**Usage Example:**
```
//DELJCL EXEC PGM=IDCAMS
-SYSPRINT DD SYSOUT=*  
-SYSIN DD *
   DELETE MWALLE.TESTDEL.MEMS(*JCL*)
-/*
```

**Results in:**
```
-DELETE MWALLE.TESTDEL.MEMS(*JCL*)
-IDC0549I MEMBER JCLE DELETED
-IDC0549I MEMBER JCLEF DELETED
-IDC0549I MEMBER JCLMLW DELETED
-IDC0549I MEMBER JCL1 DELETED
-IDC0549I MEMBER JCL12 DELETED
-IDC0549I MEMBER JCL2 DELETED
-IDC0549I MEMBER MYJCL DELETED
-IDC0549I MEMBER YOURJCL DELETED
-IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
-IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```
TSO/E: LOGONHERE reconnect support

BCP: D ALLOC and SETALLOC commands
What: Support for VTAM unconditional reconnect.
   - Allows you to reconnect to your session even if no disconnection has been detected. You “resume” right were you were before.
   - So easy to switch from one computer to another now! Hopefully should reduce the number of times that operators have to cancel TSO/E user IDs.

How to use:
   - Support is turned on by default, but you can turn it off:
     - IKTSOxx LOGON statement LOGONHERE (OFF) then SET IKJTSO=xx, or
     - TSO/E PARMLIB UPDATE(xx) command
   - D IKJTSO,LOGON can tell you what you currently have

Considerations: Verify your TSOKEYxx RECONLIM= setting to make sure it is non-zero. (Zero means a zero wait reconnection time, or reconnection is not possible.) RECONLIM default is 3 minutes.
**Usage Example:**

- Verify that TSOKEYxx RECONLIM isn’t zero, then logon selecting “-Reconnect”.

![Logon Window](image-url)
CSI CROSS-ZONE QUERY - ENTRY SELECTION

Entry Type: SYSMOD
Entry Name: PTF00002

To return to the previous panel, enter END.
To select an entry from a zone, enter $ next to the zone.

* - Entry not found in zone.
** - Zone could not be allocated or is not initialized.

<table>
<thead>
<tr>
<th>ZONE</th>
<th>TYPE</th>
<th>FMID</th>
<th>STATUS</th>
<th>DATE</th>
<th>TIME</th>
<th>REWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLB111</td>
<td>*</td>
<td></td>
<td></td>
<td>12.309</td>
<td>09:29:40</td>
<td>2012003</td>
</tr>
<tr>
<td>DLB112</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLB113</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLIB</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLOBAL</td>
<td>PTF</td>
<td></td>
<td>REC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGT111</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TGT112</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**What:** You can now see your current ALLOCxx settings with a command, and change most of them dynamically. This is great for availability, since you don’t need an IPL to change the settings.

  – Exception: policy for 2 digits years (2DGT_EXPDT)

**How to use:** `SETALLOC options and D ALLOC,OPTIONS`

  – The options statement on the command is different than the parmlib syntax. For example command syntax:
    ```
    SYSTEM, IEFBR14_DELMIGDS=NORECALL vs. parmlib syntax: SYSTEM IEFBR14_DELMIGDS(NORECALL)
    ```

**Considerations:** Note this is for changing specific settings, not the whole ALLOCxx parmlib member. It is not `SET ALLOCxx`!

  – The more dynamics we use, the more we need to ensure that we harden the newly desired values!
BCP: D ALLOC and SETALLOCCOMMANDS

- **Usage Example:**

  - SETALLOCCOMMAND SYSTEM, IEFBR14_DELMIGDS=NORECALL, TAPELIB_PREF=BYDEVICES
    
      IEFA010I SETALLOCCOMMAND SUCCESSFUL 829
      IEFBR14_DELMIGDS SET TO NORECALL.
      TAPELIB_PREF SET TO BYDEVICES.
    
  - D ALLOC, OPTIONS
    
      IEFA003I 16.16.38 ALLOC OPTIONS 832
    
      ...  
      SYSTEM IEFBR14_DELMIGDS: NORECALL
      TAPELIB_PREF: BYDEVICES
      REMIND_INTV: 90
      VERIFY_UNCAT: FAIL
Older than dirt on potatoes…

Small Enhancements of System Programmer Interest

- **DFSMS:** STGADMIN.DPDSRN.\textit{oldname}
- **ISPF:** Member search
Older than dirt on potatoes:

**DFSMS: STGADMIN.DPDSRN.** _oldname_

- **What:** Provided by the operating system, a way to rename a non-SMS data set whose name is in use by another address space. (That is, to rename a duplicately named data set.) *This support carries inherent risks that must be understood and used wisely.*
  - Data sets that are currently in use on the driving system, while you are maintaining a target system, are good examples of those data sets that are duplicately named that you may want to rename.
  - System-Specific Aliases (SSAs) is the way that ServerPac gets around this problem during system replace installs.
  - There are probably many “home grown” tools to do this already, as the need has been around forever.

- **How to use:**
  1. Create **FACILITY** class profile `STGADMIN.DPDSRNN._oldname_` for the data set you want to rename. Obtain READ access to this profile.
  2. Rename the data using ISPF PDF, after understanding consequences.
     - You could write your own program as well, using the correct CAMLIST macro expansion and authorization check.
     - SMF type 18 record written for successful duplicately-named renames: "So let it be written, so let it be done"

- **Considerations:** `STGADMIN.DPDSRNN.*` wildcarding of the profile name is strongly not recommended, as it could cover more data sets than intended. IDCAMS and IEHPROGM do not exploit `STGADMIN.DPDSRNN`. 
Older than dirt on potatoes:
**DFSMS:** STGADMIN.DPDSRN.*oldname*

- **User Example:**

  Before:
Older than dirt on potatoes:

**DFSMC**: STGADMIN.DPDSRN.*oldname*

- **User Example:**

  After:

```
Command ==> 

Data Set Name . : SYS1.LINLIB
Volume . . . . : PAGE08

The system detected that a data set with the above name is in use
(possibly on another system) but it cannot determine whether it is the
data set you wish to rename. If it is the same data set and any program
has it open, renaming it could cause serious system and data integrity
problems.

You have the extra security authority to rename the data set even though
its name is in use. Refer to the DFSMS documentation on the RENAME macro
for further information.

Instructions:
Press ENTER to override data set name protection and rename the data
set.  Enter CANCEL or EXIT to cancel the rename request.
```

[Image of a terminal session showing a command to rename a data set.]
What: Look for data sets containing a member name from ISPF 3.4 quickly. You can exclude data sets to search and say if you want to search migrated data sets.

- Wildcards are supported for member names.
- First data set containing the member name (or matching name) is placed at the top of the list.

How to use: On the DSLIST command line:

\( \text{MEMBER } \) name (or MEM name or M name)

- Options you can specify are:
  - X or EX: search only excluded
  - NX: search only not excluded data sets
  - RECALL1: Also, search data sets that are migrated to DASD
  - RECALL2: Also, search data sets that are migrated to tape

Considerations: Easy as pie!
Before: Which data sets have members that start with ALT in them?

- One of them is migrated to DASD, so I'll use RECALL1.
- I've excluded 10 data sets I don't care about, so I'll use NX

```dsh
DSLIST - Data Sets Matching MVSBUILD.OS*.SCPPBENU
Command ===> M ALT* NX RECALL1

10 data set(s) not displayed
```
After: 7 data sets have member names that start with ALT

- The one data set migrated to DASD was recalled.
- First data set with ALT* is at the top
Summary of What We Might Want to SHARE with Our User Community:

**System Programmer & User Items:**
- **SDSF:** Cursor-sensitive Sort
- **DFSMS:** IDCAMS DELETE Members
- **TSO/E:** LOGONHERE reconnect support
- **ISPF:** Member search

**System Programmers Items:**
- **BCP:** Add and remove MCS consoles dynamically
- **BCP:** Dynamic System Symbol Support
- **BCP:** DISPLAY PPT
- **BCP:** BCP Parmlib Comments
- **z/OS UNIX:** Prevent Content Overlay during MOUNT
- **DFSMS:** IEBPDSE Batch Program
- **BCP:** IEFSSNxx BEGINPARALLEL
- **BCP:** Timed Event Data Report
- **BCP:** Some PROGxx Goodies
- **BCP:** D ALLOC and SETALLOC commands
- **DFSMS:** STGADMIN.DPDSRN._oldname
z/OS Little Enhancements: Many Small Potatoes Can Make a Big Meal!

Summary

- **z/OS V2.1:**
  - **BCP:** Add and remove MCS consoles dynamically
  - **A**
  - **BCP:** Dynamic system symbol support
    - A supported way to change system symbols. Understand considerations.
  - **BCP:** DISPLAY PPT
    - **N**
  - **BCP:** Parmlib comments *(in handout only)*
    - Help document parmlib members with comments.

- **z/OS V1.13:**
  - **z/OS UNIX:** Prevent Content Overlay during MOUNT
    - Good to protect from overmounts.
  - **DFSMS:** IEBPDSE Batch Program
    - Helpful to see if your PDSEs are structurally sound.
  - **SDSF:** Cursor-sensitive Sort
    - Easy to use, and can be helpful when looking for something.
z/OS Little Enhancements: Many Small Potatoes Can Make a Big Meal!

Summary

- **z/OS V1.12:**
  - **BCP:** IEFSSNxx BEGINPARALLEL
    - A time saver that is easy to implement.
  - **BCP:** Timed Event Data Report
    - Data proof of where time is spent.
  - **BCP:** Some PROGxx Goodies
    - Nice defaults to set up.
  - **DFSMS:** IDCAMS DELETE Members
    - Something long desired!

- **z/OS V1.11:**
  - **TSO/E:** LOGONHERE reconnect support
    - It really works.
  - **BCP:** D ALLOC and SETALLOC commands
    - An availability aid.

- **Older than dirt on potatoes:**
  - **DFSMS:** STGADMIN.DPDSRN.
    - Nice to have this option, but beware of using it.
  - **ISPF:** Member search
    - Find members across many data sets very quickly.
z/OS Little Enhancements:
Many Small Potatoes Can Make a Big Meal!

August 16, 2013