z/OS 2.1 User Experience

Ed Jaffe
Phoenix Software International

August 14, 2013
14250
Early Test Program

• Phoenix Software International (PSI) was privileged to work with early z/OS 2.1 drivers via IBM’s Early Test Program.
  • [http://dtsc.dfw.ibm.com/MVSDS/%27HTTPD2.DSN01.PUBLIC.PDF%28ISVEPROG%29%27](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. 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 a coincident-with-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 2.1


<table>
<thead>
<tr>
<th>Company Name</th>
<th>Product(s)</th>
<th>Available</th>
<th>EAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS Solutions</td>
<td>Thruput Manager SE (PTF TMT6219)</td>
<td>At z/OS GA for all products listed</td>
<td>Yes for all products listed</td>
</tr>
<tr>
<td></td>
<td>Thruput Manager AE (PTF TMT6219)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thruput Manager AE+ (PTF TMT6219)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Era</td>
<td>ImageFOCUS</td>
<td>At z/OS GA for all products listed</td>
<td>Yes, Yes, Yes, Yes Limited</td>
</tr>
<tr>
<td></td>
<td>The Control Editor</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPL Check</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Stand Alone Environment (SAE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fast DASD Erase for z/OS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Step One</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>z/OS New Release Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phoenix Software International</td>
<td>CONDOR (E)</td>
<td>Now for all products listed</td>
<td>Yes for all products listed</td>
</tr>
<tr>
<td></td>
<td>JES</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FALCON</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Key/101</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetTester</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHX-Adders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHX-KeyPlus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHX-ODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>zHISR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Part of IBM’s Global Solutions Directory
- Consult this table as you plan your upgrade to z/OS 2.1
- If an ISV product you use is not listed, then ask “Why not?”
- Similar tables are available going all the way back to z/OS 1.3
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.
  • Agile is similar to what most development organizations have done for years, but more formalized and with rich project management tooling that is integrated with the development process
• 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.
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.
- However, we did encounter one big surprise at install time that should be quite relevant…
Biggest Surprise: DASD Space “Explosion”

- Space requirement nearly *doubled*!
- Upon inspection, we found that much of the difference was attributable to fonts, which are now part of the base.
  - It’s nice that software can now depend on the fonts always being there. Too bad they take up so much space… 😞
- We chose not to restore the fonts. Everything we needed worked just fine without them. But, this is not a suggestion for you.
- There is no ServerPac option for leaving out the fonts like there is for the other JES. And, unilaterally choosing not to install a base component is not supported.
May 17, 2013

Get your DASD installation requirements in early!

By Marna Walls

It's been a while since I've been able to blog... z/OS V2.1 is strongly underway here, and that has been taking me away from my "beat" of blogging. For this entry, I wanted to pass along some information about planning for z/OS V2.1 that you may find helpful. (Remember, I've already mentioned to use the IBM Health Checker for z/OS right now.)

We had the question on how much DASD storage installing z/OS and z/OSMF V2.1 will take. Here's some numbers that you might find interesting. Of course, these are sizes taken from our systems and the usual cautions apply:

- We use both JES3's, we use all NLVs, we don't count non-z/OS program products.
- the z/OS V2.1 sizes are close estimates.
- for the total z/OSMF sizes for R13, you need to add the "z/OSMF R13" and "WAS OEM" columns. For the total z/OSMF V2.1 sizes, you "only" need the z/OSMF V2.1 column (thank you Liberty profile!)

<table>
<thead>
<tr>
<th>DASD Installation sizes (in 3300 cylinders)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;z/OS R13&quot;</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>5,000</td>
</tr>
<tr>
<td>6,000</td>
</tr>
<tr>
<td>5,000</td>
</tr>
<tr>
<td>20,118</td>
</tr>
<tr>
<td>(15,116 more)</td>
</tr>
</tbody>
</table>

z/OSMF V2.1 does NOT "linked" Java SDK. Meaning that if you want to use z/OSMF V2.1 there will be an external dependency on Java SDK R7 64-bit + PTF. z/OSMF R13 did "linked" Java SDK, so that was included in the table sizes above. How does possibly one less Java SDK to lug around sound to you??

Notice that there is a new file system for z/OS V2.1 - the fast file system. More about that in a later blog. For now, know that you've got another 2,000 cylinders of file system space coming at you for z/OS V2.1.

We told you there was a lot coming in z/OS V2.1!

-Marna
Our Environment

- z10BC with 4 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 MAS and JES3 complex
    - All LPARs but one run current z/OS; one LPAR runs new z/OS
    - Eventually all are migrated to new z/OS which is then current
  - 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- and FCP-attached IBM Totalstorage Tape Drives
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 an LPAR as part of our primary parallel sysplex. The other systems in the parallel sysplex run the current z/OS release with necessary toleration maintenance applied. We do most of our development and testing here.
• All new z/OS images run from exactly the same read-only SYSRES volumes—shared between z/OS and z/VM.
Toleration Maintenance is Critical

- Applying toleration maintenance on back level releases should be S.O.P. and it’s easier than ever to get it right.

```
SET BDY(GLOBAL) .
REPORT MISSINGFIX ZONES(MVST)
    FIXCAT(IBM.TargetSystem-RequiredService.z/OS.V2R1).
REPORT MISSINGFIX ZONES(MVST)
    FIXCAT(IBM.Coexistence.z/OS.V2R1).
```

- Sometimes, when something falls through the cracks, the results are minor inconveniences and are easily resolved simply by applying the missing PTF(s).

- Not so with z/OS 2.1.

- We experienced problems—most likely due to incomplete deployment of toleration ++APARs on our z/OS 1.13 systems—that precipitated a sysplex-wide “warm” start.
Dynamic Console Add and Remove (SHARE Top 39 Requirement!)

- SET CON=xx adds consoles via CONSOLxx
- SETCON DELETE deletes a console
- I wanted to delete and re-add an inactive console.
- Be sure to remove obsolete specifications.
  - I still had CNGROUP and ALTGRP keywords in my CONSOLxx member. ALTGRP was flagged as error.

-D C,CN=TAPEY1E
T4SY1 R= CNZ4100I 22.30.53 CONSOLE DISPLAY 099
T4SY1 R= CONSOLES MATCHING COMMAND: D C,CN=TAPEY1E
T4SY1 R= MSG:CURR=0 LIM=5000 RPLY:CURR=1 LIM=20 SYS=T4SY1 PFK=00
T4SY1 R= NAME TYPE STATUS DEFINED MATCHED
T4SY1 R= TAPEY1E MCS INACT *ALL *ALL

-SETCON DELETE,CN=TAPEY1E
T4SY1 R= CNZ4300I MCS CONSOLE TAPEY1E HAS BEEN REMOVED

-D C,CN=TAPEY1E
T4SY1 R= IEE274I DISPLAY CONSOLE TAPEY1E NOT VALID
Dynamic Console Add and Remove (continued...)

-T CON=01
T4SY1 R=  CNZ6003I COMMAND ACCEPTED FOR EXECUTION: SET CON=01
T4SY1 R=  IEE252I MEMBER CONSOLO1 FOUND IN SYS2.PARMLIB
T4SY1 R=  IEA196I CONSOLO1 0600: DEVNUM ALREADY DEFINED. STATEMENT IGNORED.
... <-- lots more “babbling” here
-VARY OPERLOG,HARDCPY  ISSUED FOR SET CON=01
-T PFK=00  ISSUED FOR SET CON=01
-T MMS=00  ISSUED FOR SET CON=01
-T MPF=(MS,RS,00)  ISSUED FOR SET CON=01
-MN SPACE   ISSUED FOR SET CON=01
-MN DSNAME  ISSUED FOR SET CON=01
T4SY1 R=  IEE252I MEMBER PFKTAB00 FOUND IN SYS2.PARMLIB
T4SY1 R=  IEE536I PFK VALUE 00 NOW IN EFFECT
T4SY1 R=  IEE252I MEMBER MPFLSTMS FOUND IN SYS2.PARMLIB
T4SY1 R=  IEE252I MEMBER MPFLSTRS FOUND IN SYS2.PARMLIB
T4SY1 R=  MMS IEE252I MEMBER MMSLST00 FOUND IN SYS2.PARMLIB
T4SY1 R=  IEA630I OPERATOR *OPLOGY1 NOW ACTIVE, SYSTEM=T4SY1 , LU=NONE
T4SY1 R=  MMS IEE252I MEMBER CNLENU00 FOUND IN ADCD.Z113.PARMLIB
-D C,HC,L=Z
-VARY ,HARDCPY,ROUT=(1-10,12-13,15-128)  ISSUED FOR SET CON=01

-D C,CN=TAPEY1E
T4SY1 R=  CNZ4100I 22.31.59 CONSOLE DISPLAY 162
T4SY1 R=  CONSOLES MATCHING COMMAND: D C,CN=TAPEY1E
T4SY1 R=  MSG:CURR=0  LIM=5000 RPLY:CURR=1  LIM=20  SYS=T4SY1  PFK=00
T4SY1 R=  NAME TYPE STATUS DEFINED MATCHED
T4SY1 R=  TAPEY1E MCS INACT *ALL *ALL
Integrated 3270 Console Support  
(SHARE Top 39 Requirement!)

- I created a parmlib member called CONSOLHM:

```c
/*********************************************************************/
/*                                                                  */
/* HMC CONSOLE                                                      */
/*                                                                  */
/*********************************************************************/
CONSOLE DEVNUM(HMCS)  
NAME(HMCS&SYSCLONE.A)  
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) RTME(1/4) MFORM(J,S) AREA(14)  
RBUF(15)  
LOGON(REQUIRED)
```

- SET CON=HM issued an ACK message only, so I issued ‘D C’ and found the console in STDBY status

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Type</th>
<th>Command</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013211 13.13.03.62</td>
<td>EDJXADM</td>
<td>-RO MVSA0,T CON=HM</td>
<td>MVSA0 2013211 13.13.03.68</td>
<td>EDJXADM</td>
</tr>
<tr>
<td>2013211 13.15.14.88</td>
<td>EDJXADM</td>
<td>-RO MVSA0,D C,CN=HMCSA0A</td>
<td>MVSA0 2013211 13.15.14.94</td>
<td>EDJXADM</td>
</tr>
</tbody>
</table>
Wrong release info is apparently not a z/OS problem. You need to update your HMC/SE code to fix this issue.
Console is Operational! 😊

- Requires Java support in your browser
- Presentation size for HMCS fixed at 43x80 (This does not appear to be customizable.)
- LOGON(REQUIRED) is honored exactly as for SMCS consoles
- One HMCS console per LPAR (per HMC?) Start of second session receives:

![Console Image]
Also Supported as a NIP Console 😊

- We have no NIP consoles defined via HCD.
- We want NIP messages directed to the Operating System Messages folder for the LPAR on the HMC.
- We do this so we can IPL remotely and respond to any message that might appear on NIP console.
- Just as one would hope, NIP messages appear in Integrated 3270 Console.
Update Symbols via SETLOAD Command
(SHARE Top 39 Requirement!)

- Issue D SYMBOLS to display existing symbols:

```plaintext
-D SYMBOLS
IEA007I STATIC SYSTEM SYMBOL VALUES 328
    &SYSALVL. = "2"
    &SYSCLONE. = "Y1"
    &SYSNAME. = "T4SY1"
    &SYSOSLVL. = "Z1020100"
    &SYSPLEX. = "PHXHQ"
    &SYSR1. = "T4RES1"
    &ADCDVER. = "Z112"
    &CICSHLQ. = "DFH410"
    &CNMNETID. = "PHX"
    &CNMTCPN. = "TCPIP"
    ... (additional symbols)
    &J3MEM. = "00"
    &J3QUAL. = "PHXHQ"
    &OSREL. = "ZOSR1"
    &PLEXQUAL. = "PHXT4"
    &SYSP1. = "ZCPRD1"
    &SYSP2. = "ZCPRD2"
    &SYSP3. = "ZCPRD3"
    &SYSR2. = "T4RES2"
    &SYSR3. = "T4RES3"
    &SYSR4. = "T4RES4"
    &SYSS1. = "T4SYS1"
    &TSOKEY. = "00"
    &UNIXVER. = "VERSYSB"
```

&SYSOSLVL is a new system-defined static symbol.
Update Symbols via SETLOAD Command (continued…)

• Use D IPLINFO to learn LOADxx and IEASYMxx names:

```
-D IPLINFO
IEE254I 11.52.52 IPLINFO DISPLAY 331
   SYSTEM IPLLED AT 16.26.31 ON 07/18/2013
   RELEASE z/OS 02.01.00 LICENSE = z/OS
   USED LOADT4 IN SYS4.IPLPARM ON 08128
   ARCH_LVL = 2  MTLSHARE = N
   IEASYM LIST = (00,11,L)
   IEASYS LIST = (11,03) (OP)
   IODF DEVICE: ORIGINAL(08128) CURRENT(08128)
   IPL DEVICE: ORIGINAL(08110) CURRENT(08110) VOLUME(T4RES1)
```

• Add the following to the end of IEASYM11:

```
SYMDEF(&ZEPEL1N='LED')
```

• Issue SETLOAD to update the symbols:

```
-SETLOAD T4,IEASYM,DSN=SYS4.IPLPARM
IEE252I MEMBER LOADT4 FOUND IN SYS4.IPLPARM
IEE252I MEMBER IEASYM00 FOUND IN SYS2.PARMLIB
IEE252I MEMBER IEASYM11 FOUND IN SYS2.PARMLIB
IEF196I IEF285I SYS4.IPLPARM KEPT
IEF196I IEF285I VOL SER NOS= T4SYS1.
IEF900I SYSTEM SYMBOLS WERE UPDATED FROM LOADT4
```
Update Symbols via SETLOAD Command (continued...)

- Display updated symbols:

```bash
-D SYMBOLS
IEA007I STATIC SYSTEM SYMBOL VALUES 328
 &SYSALVL. = "2"
 &SYSCLONE. = "Y1"
 &SYSNAME. = "T4SY1"
 &SYSOSLVL. = "Z1020100"
 &SYSPLEX. = "PHXHQ"
 &SYSR1. = "T4RES1"
 &ADCDVER. = "Z112"
 &CICSHLQ. = "DFH410"
 &CNMNETID. = "PHX"
 &CNMTCPN. = "TCPIP"
 ... (additional symbols)
 &J3MEM. = "00"
 &J3QUAL. = "PHXHQ"
 &OSREL. = "ZOSR1"
 &PLEXQUAL. = "PHXT4"
 &SYSP1. = "ZCPRD1"
 &SYSP2. = "ZCPRD2"
 &SYSP3. = "ZCPRD3"
 &SYSR2. = "T4RES2"
 &SYSR3. = "T4RES3"
 &SYSR4. = "T4RES4"
 &SYSS1. = "T4SYS1"
 &TSOKEY. = "00"
 &UNIXVER. = "VERSYSB"
 &ZEPPELIN. = "LED"
```

- Processing is complete symbol replacement—*not additive*

- If I restore IEASYM11 to its original state and reissue the SETLOAD command, the new symbol &ZEPPELIN is removed.
Display PPT Command

(SHARE Top 39 Requirement!)

NC0000000 MVSA0 2013210 07:18:43.06 EDJXADM 00000290 D PPT
MR0000000 MVSA0 2013210 07:18:43.11 EDJXADM 00000090 IEF386I 07.18.43 DISPLAY PPT 647

LR 647 00000090 Parmlib Values
   DR 647 00000090 PgmName  NC NS PR ST ND BP Key 2P 1P NP NH CP
       DR 647 00000090 ANFFIEP  .  .  .  Y  Y  Y  .  1  .  Y  .  .
       DR 647 00000090 BBGCTL  .  Y  Y  .  .  .  2  .  .  .  .  .
       DR 647 00000090 BBGDAEMN  .  Y  Y  Y  .  .  2  .  .  .  .  .
       DR 647 00000090 BBGDAEMN  .  Y  Y  Y  .  .  2  .  .  .  .  .
       DR 647 00000090 BPXVCLNY  .  Y  Y  Y  Y  .  8  .  .  .  .  .

... (more parmlib entries)

LR 647 00000090 Default Values
   DR 647 00000090 PgmName  NC NS PR ST ND BP Key 2P 1P NP NH CP
       DR 647 00000090 AHLCGTF  Y  Y  .  Y  .  .  0  .  .  Y  .  .
       DR 647 00000090 ASBSCHIN  .  Y  .  Y  Y  .  1  Y  .  .  .  .
       DR 647 00000090 ASBSCHWL  .  Y  .  Y  Y  .  1  Y  .  .  .  .
       DR 647 00000090 ATBINIM  .  Y  .  Y  Y  .  1  Y  .  .  .  .
       DR 647 00000090 ATBSDFMU  .  Y  .  Y  Y  .  1  .  .  .  .  .
       DR 647 00000090 AVFMBLCLD  Y  Y  .  Y  .  3  .  .  .  .  .

... (76 system entries altogether)

LR 647 00000090 Reference
   DR 647 00000090 Synonym  Meaning  SCHEDxx keyword
       DR 647 00000090 NC  Non-cancelable  NOCANCEL
       DR 647 00000090 NS  Non-swappable  NOSWAP
       DR 647 00000090 PR  Privileged  PRIV
       DR 647 00000090 ST  System task  SYST
       DR 647 00000090 ND  No dataset integrity  NODSI
       DR 647 00000090 BP  Bypass password protection  NOPASS
       DR 647 00000090 Key  PSW key for this program  KEY(x)
       DR 647 00000090 2P  Second level preferred storage  SPREF
       DR 647 00000090 1P  First level preferred storage  LPREF
       DR 647 00000090 NP  No preferred storage  NOPREF
       DR 647 00000090 NH  No honor IEFUSI region settings  NOHONORIEFUSIREGION
       DR 647 00000090 CP  Critical paging  CRITICALPAGING
Wildcard Character Support in IEBCOPY
(SHARE Top 39 Requirement!)

IEBCOPY MESSAGES AND CONTROL STATEMENTS

PAGE 1

-IEB1135I IEBCOPY FMID HDZ2210 SERVICE LEVEL NONE DATED 20130313 DFSMS 02.01.00 z/OS 02.01.00
HBB7790 CPU 2098
IEB1035I COPYCSV 09:01:22 MON 29 JUL 2013 PARM=''
-COPYGROUP INDD=((INPUT,R),OUTDD=OUTPUT
  SELECT MEMBER=CSV*
0IEB1013I COPYING FROM PDS INDD=INPUT VOL=T4RES2 DSN=SYS1.MACLIB
IEB1014I TO PDS OUTDD=OUTPUT VOL=T4USR1 DSN=EDJXADM.CSV.MACLIB
IGW01264I TOTAL PRIMARY NAMES: 1924, FILTER PATTERN MATCHES: 20
IGW01551I MEMBER CSVAPF HAS BEEN COPIED
IGW01551I MEMBER CSVAPFAA HAS BEEN COPIED
IGW01551I MEMBER CSVDLAA HAS BEEN COPIED
IGW01551I MEMBER CSVDLCB HAS BEEN COPIED
IGW01551I MEMBER CSVDLENF HAS BEEN COPIED
IGW01551I MEMBER CSVDLSMF HAS BEEN COPIED
IGW01551I MEMBER CSVDYLPA HAS BEEN COPIED
IGW01551I MEMBER CSVDYLEX HAS BEEN COPIED
IGW01551I MEMBER CSVDYNL HAS BEEN COPIED
IGW01551I MEMBER CSVEXAA HAS BEEN COPIED
IGW01551I MEMBER CSVEXRET HAS BEEN COPIED
IGW01551I MEMBER CSVEXTI HAS BEEN COPIED
IGW01551I MEMBER CSVINFO HAS BEEN COPIED
IGW01551I MEMBER CSVLPRET HAS BEEN COPIED
IGW01551I MEMBER CSVLPBREADER HAS BEEN COPIED
IGW01551I MEMBER CSVLPSMF HAS BEEN COPIED
IGW01551I MEMBER CSVMODI HAS BEEN COPIED
IGW01551I MEMBER CSVQUERY HAS BEEN COPIED
IGW01551I MEMBER CSVXENV HAS BEEN COPIED
IGW01550I 20 OF 20 SPECIFIED MEMBERS WERE COPIED
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE
Be Careful! Do Not Use COPYGRP!

- This confused me for a while. It’s yet another IBMIOSYNCRASYS!
- I’m sure someone at IBM has a valid rationale for why this was done.
- Seems like an odd choice. I’m thankful to have this feature no matter how it’s implemented. 😊

COPYGRP Statement

Use the COPYGRP statement to begin a group copy, unload, or load. A group consists of a member and all of its aliases. COPYGRP treats the group as a single entity.

The syntax of the COPYGRP statement is:

```
Label [label]  Statement  COPYGRP
Parameters
OUTDD=DDname, INDD=DDname, LIST=YES|NO]
```

COPYGROUP Statement

Use the COPYGROUP statement to begin a group copy, unload, or load. A group consists of a member and all of its aliases. COPYGROUP treats the group as a single entity.

The syntax of the COPYGROUP statement is:

```
Label [label]  Statement  COPYGROUP
Parameters
OUTDD=DDname, INDD=DDname, LIST=YES|NO]
```
PARMDD Instead of PARM=‘string’ in JCL

- PARMDD provides support similar to existing STDPARM support for z/OS UNIX utility called BPXBATCH
  - The primary advantage is to support for more than 100 bytes, but it also provides support for out-of-stream parameters i.e., in PDS members, z/OS UNIX files, etc.
- Records can be fixed or variable up to 32K in length
- Fixed length records are examined to see if there are eight contiguous numeric characters in the last eight positions of the record. If so, those eight characters are stripped off.
- Trailing blanks are stripped off—blank lines disappear
- All remaining characters concatenated up to 32K

```
//STEP1    EXEC PGM=mypgm,PARMDD=ddname
... 
//ddname   DD *
This is an alternate way of providing parameters to mypgm.  
```
No PARMDD Equivalent in TSO/E CALL?! 😞

- According to the help, TSO/E CALL was not updated with PARMDD support:
  - SHARE requirement needed? Must we wait two years?

```
HELP CALL
CALL 'dsname(member)' 'parm'
   CAPS/ASIS NOENVB/PASSENVB
...
   'parm' - Specifies a parameter string (up to 100 characters)
   that is passed to the program to be executed.
```

- REXX LINKMVS and ATTCHMVS already pass up to 32K using the exact same linkage convention as CALL, but cannot invoke programs via LNKLST.
- REXX LINK, LINKPGM, ATTACH, and ATTCHPGM use an entirely different linkage convention from JCL EXEC and TSO/E CALL and have always supported arbitrarily long values.
JES Features

• As you would expect, because we have a product in this space, we tried some of the new JES features.
• We “played around” a bit with the JES3 support that’s been rolled back to JES2: 8-character job classes, job class groups, pre-execution C/I. All seems to work as advertised.
  • **Note:** There is no **LOCATE** function happening at C/I time. The biggest advantage seems to be JES3-like detection of JCL errors prior to actually running the job.
  • Previous JES2 releases couldn’t detect obvious JCL spelling errors like `DISP=(NEW,CATLX)`. If you enable this support, now it will.
• It’s *highly* convenient that using new JCL keywords forces execution onto a z/OS 2.1 system.
• The use of symbols in SYSIN is *huge.* ‘nuff said!
HIS Enhancements

• We also have a product in this area (zHISR), so it’s no surprise that we explored the new HIS capabilities

• z/OS 2.1 HIS provides the ability for profilers to create their own profiling sessions, totally independent of any profiling or SMF 113 data collection going inside HIS itself.
  • No need to interrupt SMF 113 data collection to do profiling
  • New HISSERV macro/service provides all necessary support
  • Supports any number of profilers—each with its own stream

• This new support works extremely well!

• We had a few issues, but IBM fixed them right away—usually with the very next driver
Warning: HIS MAP Files are Incompatible!

<table>
<thead>
<tr>
<th>SYS</th>
<th>MVS70</th>
<th>MVS70</th>
<th>MAP V1R1</th>
<th>LPID</th>
<th>MACH00002098</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMF1</td>
<td>MVS70</td>
<td>SMFIMV70</td>
<td>OS z/OS</td>
<td>LPID</td>
<td>MACH00002098</td>
</tr>
<tr>
<td>FMDHBB7780</td>
<td>DATE13131</td>
<td>TIME19075713</td>
<td>MAP V1R1</td>
<td>LPID</td>
<td>MACH00002098</td>
</tr>
<tr>
<td>I</td>
<td>SYS MVSA0</td>
<td>SMFIMVA0</td>
<td>OS z/OS</td>
<td>LPID</td>
<td>MACH00002098</td>
</tr>
<tr>
<td>I</td>
<td>DATE13214</td>
<td>TIME0855635</td>
<td>MAP 02.01</td>
<td>LPID</td>
<td>MACH00002098</td>
</tr>
<tr>
<td>I</td>
<td>MODE64-BIT</td>
<td>LPID</td>
<td>MACH00002098</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B BDY PRIVATE 0000000000AFFFFF
B BDY CSA 00B00000000DF2CFF
B BDY CSAALLOC003B9E80206CEE8
B BDY CSACONVT0000000000000000
B BDY MLPA 0000000000000000
B BDY PLPA 00D2D00000F06FFF
B BDY SQA 00F0700000FD3FFF
B BDY SQAALLOC000501780143AAB0
B BDY SWNUC 00FD4000000FE2DDF
B BDY RNWUC 00FE3000000F0FFFF
B BDY RON 010000001A8AECF
B BDY ERON 01A8B00001A5AFF
B BDY ESQA 01AE800003BFFA
B BDY EPLPA 03B500007410FFF
B BDY EFLPA 0741100007413FFF
B BDY EMLPA 074140000741AFF
B BDY ECSD 0742200000C4FFFF
B BDY EPRV 0C5000070FFFFF
B BDY DONUC 7FD360007FD39FFF
CNNUC IEAVFX0000000000001FFFF

... (more entries)
Health Checks that Appeared in z/OS 2.1

- CATALOG_RNLS
- OCE_XTIOT_CHECK
- RACF_CERTIFICATE_EXPIRATION
- SLIP_PER
- SUP_SYSTEM_SYMBOL_TABLE_SIZE
- SYSTRACE_BRANCH
- SYSTRACE_MODE
- USS_KERNEL_PVTSTG_THRESHOLD
- USS_KERNEL_STACKS_THRESHOLD
- VLF_MAXVIRT
- ZOSMIGV2R1_DEFAULT_UNIX_ID
Humorous Health Check Result

CHECK(IBMOCE,OCE_XTIOT_CHECK)
SYSPLEX:    PHXHQ     SYSTEM: MVSA0
START TIME: 07/29/2013  00:31:11.448365
CHECK DATE: 20110410    CHECK SEVERITY: LOW

* Low Severity Exception *

IECH0100I OPEN macro support for XTIOT, uncaptured UCBs and DSAB above the line is enabled for non-VSAM. This follows IBM's recommendation.

Explanation: Specifying NON_VSAM_XTIOT=YES in the DEVSUPxx member of PARMLIB decreases the chances of running out of virtual storage when allocating and concurrently opening many sequential and partitioned data sets.

System Action: The system continues processing.

Operator Response: N/A

System Programmer Response: Encourage the use of XTIOT allocations.

Problem Determination: N/A

Source: DFSMS OPEN/CLOSE/EOV

Reference Documentation: For additional information see:

   z/OS MVS Initialization and Tuning Reference
Missing Health Check in z/OS 2.1

- CEE_USING_LE_PARMLIB
  - I did not research why this check isn’t there.
  - Perhaps it appears only with some post-install customization that wasn’t done?
EDIT SYS2.MVSUTIL.CNTL(BACKUPVM) - 01.16

Command ==> this is the start of a command that is just too

***** *************** Top of Data ***************

000001 //BKUPVMO1 JOB 1,JAFFE,CLASS=A,MSGCLASS=T,NOTIFY=&SYSUID,
000002 // REGION=64M,TIME=NOLIMIT USER=EDJXADM
000003 /*ROUTE XEQ PHXHQ
000004 /*MAIN CLASS=TAPECOPY
000005 /*
000006 //BACKUP PROC U=,V=,L=
000007 //VARYON EXEC PGM=EJESLNK,PARM='/'V &U,ONLINE'
000008 //EJESOUT DD SYSOUT=*,RECFM=F,LRECL=80
000009 //EJESIN DD DUMMY
000010 //WAIT5SEC EXEC PGM=WAIT5SEC
000011 //DUMP EXEC PGM=ADDRSSU,REGION=64M
000012 //SYSPRINT DD SYSOUT=* 
000013 //INVOLE DD UNIT=/&U,VOL=SER=&V,DISP=OLD
000014 //TAPE DD DSN=PHOENIX.BACKUP.V&V,
000015 // UNIT=(3590-1,,DEFER),VOL=(,RETAIEN,SER=BKPVM1),
000016 // LABEL=(&L,SL,EXPDTR=98000),DISP=(NEW,KEEP)
000017 //VARYOFF EXEC PGM=EJESLNK,PARM='/'V &U,OFFLINE',COND=EVEN
000018 //EJESOUT DD DUMMY,RECFM=F,LRECL=80

F1=Help F3=Exit F4=Expand F5=Rfind F6=Rchange F12=Cancel

04/062
this is the start of a command that is just too long to type into the limited command line offered by ISPF EDIT/VIEW


F1=Help  F3=Exit  F4=Expand  F5=Rfind  F6=Rchange  F12=Cancel
Regular Expressions in ISPF

• Some folks on IBM-MAIN and MVS-OE believe that the use of regular expressions separates the adults from the children in the programming world.

• Regular expressions can be useful. Mine are usually not more complex than ordinary search strings. But I’m no regular expressions expert, nor was I ever an APL geek.

• ISPF now calls the REGCOMP function in the C runtime library to evaluate regular expressions for FIND, CHANGE and EXCLUDE.

• I tried the example in the help. It worked.

_ Example -  ==> find r’l[ai]ne’ word will find words lane and line in the file being edited_
Regular Expression Grammar

Standard C++ Library Reference
SC09-4949-05

Element

An element can be any of the following:

- An **ordinary character**, which matches the same character in the target sequence.
- A **wildcard character**, '.', which matches any character in the target sequence except a newline.
- A **bracket expression**, of the form "[^expr]", which matches a character or a **collation element** in the target sequence that is also in the set defined by the expression expr, or of the form "[^expr]", which matches a character or a collation element in the target sequence that is not in the set defined by the expression expr. The expression expr can consist of any combination of any number of each of the following.
  - An **individual character**, which adds that character to the set defined by expr.
  - A **character range**, of the form "ch1-ch2", which adds all of the characters represented by values in the closed range [ch1, ch2] to the set defined by expr.
  - A **character class**, of the form "[name:]", which adds all of the characters in the named class to the set defined by expr.
  - An **equivalence class**, of the form ":=elt:", which adds the collating elements that are equivalent to elt to the set defined by expr.
  - A **collating symbol**, of the form ":elt:”, which adds the collation element elt to the set defined by expr.
- An **anchoring**, either '^' or '$', which matches the beginning or the end of the target sequence, respectively.
- A **capture group**, of the form "\( Subexpression \)”, or “\( Subexpression \)" in BRE and grep, which matches the sequence of characters in the target sequence that is matched by the pattern between the delimiters.
- An **identity escape**, of the form "\k", which matches the character k in the target sequence.

Examples:

- "a" matches the target sequence “a” but none of the target sequences “B”, “b”, or “c”.
- "l" matches all of the target sequences “a”, “B”, “b”, and “c”.
- "[b-z]" matches the target sequences “b” and “c” but does not match the target sequence “a” or the target sequence “B”.
- "[lower:]" matches the target sequences “a”, “b”, and “c” but does not match the target sequence “B”.
- "(a)" matches the target sequence “a” and associates capture group 1 with the subsequence “a”, but does not match any of the target sequences “B”, “b”, or “c”.

In ECMAScript, BRE, and grep an element can also be:

- a **back reference**, of the form "\dd" where dd represents a decimal value N, which matches a sequence of characters in the target sequence that is the same as the sequence of characters matched by the Nth capture group.
FTP MVSGET and MVSPUT

- These are *hugely* helpful new FTP client commands
- You can now easily transfer sequential and PDS[E] without needing to know allocation details
- For PDS[E] the client sets up a loop and issues transfer commands iteratively for every member in the library.

```
EZA1701I >>> EPSV
229 Entering Extended Passive Mode (|||1179||)
EZA1701I >>> RETR CATIMPT
125 Sending data set SYS2.MVSUTIL.CNTL(CATIMPT) FIXrecfm 80
250 Transfer completed successfully.
EZA1617I 1312 bytes transferred in 0.010 seconds. Transfer rate 131.20 Kbytes/sec.
EZA1701I >>> EPSV
229 Entering Extended Passive Mode (|||1181||)
EZA1701I >>> RETR CATLOCK
125 Sending data set SYS2.MVSUTIL.CNTL(CATLOCK) FIXrecfm 80
250 Transfer completed successfully.
EZA1617I 172 bytes transferred in 0.020 seconds. Transfer rate 8.60 Kbytes/sec.
```
REXX Interface to BCPii

- IBM provides nice REXX samples in SYS1.SAMPLIB. The members I chose to “play” with were:
  - HWIXMRJL – JCL to run a REXX exec in batch via SYSREXX
  - HWIXMRS1 – REXX to list basic information from your CPC
- First thing I noticed was that the HWIBCPII address space was not started on my z/OS 2.1 system. Why not?
- S HWISTART failed with:

  HWI022I THE SNMP COMMUNITY NAME REQUIRED BY BCPII FOR THE LOCAL CPC WAS REJECTED BY THE SUPPORT ELEMENT. CORRECT THE COMMUNITY NAME IN THE SECURITY PROFILE THAT WAS RETRIEVED BY BCPII USING THIS ENTITY (HWI.TARGET.IBM390PS.P00C9632).
I Was Scratching My Head

- It suggests that the community name in the CPC and in the RACF profile do not match. Yet, my z/OS 1.13 LPARs—running in the very same sysplex on the very same CPC (we have only one!) and sharing the same RACF database—had no issues.
- I tried various things to no avail. I was doing this while on-site at SHARE and I reached out to Steve Warren for help.
- Steve’s L2 person was on-site and recognized the problem from his explanation.
- BCPii uses Unicode conversion tables to uppercase the community name. A problem accessing those tables is not surfaced via any messages.
BCPii Problem Solved!

- It turned out that, due to the font space issues, SYS1.SCUNTBBL never got restored on our system.
- Under the gun, just to make things work, I cataloged the z/OS 1.13 of the data set on the z/OS 2.1 system.
- IT WORKED!!!!!!!!!!!!!!!!!!!!!!!!!!!! 😊
- *It helps to have friends in high places. You have me...*
- Anyway, Steve took that back as a suggestion either to surface Unicode service failures in BCPii or to update the Migration Guide to let you know the Unicode conversion files need to be cataloged and available:
  - SYS1.SCUNTBBL
  - SYS1.SCUNLOCL
The Sample Programs Need Minor Rework

- The first thing the samples do is read in the IDF files
- The data sets they reference are not available
- The real location of the IDFs on customer systems is SYS1.MACLIB

```plaintext
/* call set_hwi_constants */
ALLOC F(HWICIREX) DA('HWI.HWICIREX.IDF') SHR REUS
execio * diskr "HWICIREX" (stem linelist. finis "
FREE F(HWICIREX)

do x = 1 to linelist.0
  interpret linelist.x
end
drop linelist.

ALLOC F(HWIC2REX) DA('HWI.HWIC2REX.IDF') SHR REUS
execio * diskr "HWIC2REX" (stem linelist. finis "
FREE F(HWIC2REX)

do x = 1 to linelist.0
```
Output of Sample HWIXMSR1 REXX

```plaintext
=>> BCPii REXX Sample starting ... <<=
=>> HWILIST - List CPCs
         REXXHostRc = 0
         HWILIST rc = 0
         Number of CPCs found = 00000001
         CPC 1 = IBM390PS.P00C9632

=>> HWICONN - Connect to CPC IBM390PS.P00C9632
         REXXHostRc = 0
         HWICONN rc = 0

=>> HWIQUERY - Query CPC attributes
         REXXHostRc = 0
         HWIQUERY rc = 0

         > Model Number is E10
         > SNA Addr is IBM390PS.P00C9632
         > Num GPP is 00000004

=>> HWILIST - List images
         REXXHostRc = 0
         HWILIST rc = 0
         Number of images found = 00000008
         Image 1 = CF01
         Image 2 = CF02
         Image 3 = LINUX
         Image 4 = MVSA0
         Image 5 = MVS60
         Image 6 = MVS70
         Image 7 = VM80
         Image 8 = VSEB0

=>> HWICONN - Connect to image CF01
         REXXHostRc = 0
         HWICONN rc = 0

=>> HWIQUERY - Query image attributes
         REXXHostRc = 0
         HWIQUERY rc = 0
```

...
Output of Sample HWIXMSR1 REXX

Image 8 = VSEB0
=>> HWICONN - Connect to image CF01
    REXXHostRc = 0
    HWICONN rc = 0
=>> HWIQUERY - Query image attributes
    REXXHostRc = 0
    HWIQUERY rc = 0
    > OS Type is
    > MSG STAT is 00000000
=>> HWIDISC - Release Image connection
    REXXHostRc = 0
    HWIDISC rc = 0
=>> HWICONN - Connect to image CF02
    REXXHostRc = 0
    HWICONN rc = 0
=>> HWIQUERY - Query image attributes
    REXXHostRc = 0
    HWIQUERY rc = 0
    > OS Type is
    > MSG STAT is 00000000
=>> HWIDISC - Release Image connection
    REXXHostRc = 0
    HWIDISC rc = 0
=>> HWICONN - Connect to image LINUX
    REXXHostRc = 0
    HWICONN rc = 0
=>> HWIQUERY - Query image attributes
    REXXHostRc = 0
    HWIQUERY rc = 0
    > OS Type is
    > MSG STAT is 00000000
Output of Sample HWIXMSR1 REXX

```plaintext
=>> HWIDISC - Release Image connection
   REXXHostRc = 0
   HWIDISC rc = 0
=>> HWICONN - Connect to image MVSA0
   REXXHostRc = 0
   HWICONN rc = 0
=>> HWIQUERY - Query image attributes
   REXXHostRc = 0
   HWIQUERY rc = 0
   > OS Type is MVS
   > MSG STAT is 00000000
=>> HWIDISC - Release Image connection
   REXXHostRc = 0
   HWIDISC rc = 0

...

=>> HWICONN - Connect to image VSEB0
   REXXHostRc = 0
   HWICONN rc = 0
=>> HWIQUERY - Query image attributes
   REXXHostRc = 0
   HWIQUERY rc = 0
   > OS Type is
   > MSG STAT is 00000000
=>> HWIDISC - Release Image connection
   REXXHostRc = 0
   HWIDISC rc = 0
=>> HWIDISC - Release CPC connection
   REXXHostRc = 0
   HWIDISC rc = 0
=>> BCPii REXX Sample ending ... <==
```
There were many other z/OS 2.1 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...
PDS Utility on CBT Tape

- The PDS command received:

  IEC190I INVALID DCBE: EXCP DCB WITHOUT FOUNDATION EXT

- Which seems odd since msgIEC190I is documented as:

  IEC190I ddname, DCB RESIDES ABOVE 16MB

- The required fixes are:

  OA42694 ABEND0C4 IFG0554P
  ERROR DESCRIPTION:
  0c4 in IFG0554P incorrectly accessing a 24bit ucb address as a 31bit address with a dirty high order byte. Abend0c4 out of IFG0554P due to dirty high order UCB address. Also invalid IEC190I INVALID DCBE: EXCP DCB WITHOUT A FOUNDATION EXTENSION.

  PTF: UA69759/1307
  This PTF additionally prereqs UA69690

  OA42406 OCE FIX ROLLUP FOR HDZ2210
  ERROR DESCRIPTION:
  Fixes PCK in IFG019TR. Also adds ALIAS name in SMF14/15. Also fixes an invalid IEC190I EXCP DCB without a foundation extension..

  PTF: UA69690/1307
ShowZOS Utility on CBT Tape

- As distributed, the utility would not assemble

```assembly
68120 AIF (NOT D'CVTH7790).CONS15A z/OS V2R1 ? No JH0112
68121 AIF ('SYSSTATE_OSREL'(1,6) LT '010201').CONS15A RS0112
0001A13E 9104 D12F 0000012F 68122 TM HBB7790,L'HBB7790 running on z/OS V2R1 ? JH0112
0001A142 0000 0000 00000000 68123 JNO CONSOL15A ...jif not JH0112
68124 * UCMSTRNM structure name moved to UPEA area at CVTH7790 JH0112
68125 * The following is temporary since we do not yet know the JH0112
68126 * structure of the UPEA area pointed to by UCMUPEA JH0112
0001A146 5860 4080 00000080 68127 L R6,UCMUPEA R6 -> UPEA area JH0112
68128 STRING64 ' OpLog-Str: ',(0(R6),16),INTO=LINE JH0112
```

- Two assembler statements were out of order:

```assembly
AGO .CONS15B JH0112
CONSOL15A DS 0H JH0112
```

Changed to:

```assembly
CONSOL15A DS 0H EJ0713
AGO .CONS15B JH0112
```

- After that, the utility assembled, but failed with abend0C4 while processing ZFS information. 😞
All in all I think IBM did a great job with this release and it was gratifying to see many of SHARE’s Top 39 Requirements addressed.