#SHAREorg

SHARE
Technology · Connections · Results

Cloning for z/OS UNIX Service in a Shared File System Environment

Marna Walle
Jim Showalter
IBM

March 14, 2012
Session 10866

Trademarks

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

- DFS
- DFSMS
- DFSMSdss
- IBM
- MVS
- RACF
- RMF
- S/390
- z/OS
- zSeries

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* 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.

Agenda

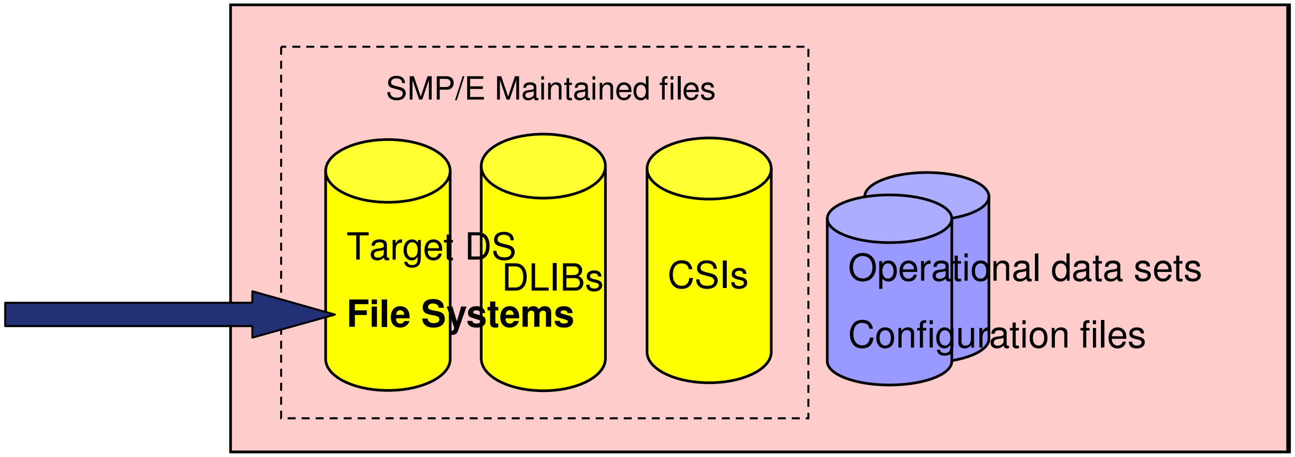
- Cloning Review:
  - Why you clone
  - Cloning assumptions
- Differences: data sets and file systems, and HFS and zFS
- Catalog’s Role in using file systems
- Shared file system concepts
- Installing and Servicing for a shared file system environment
- Copying file system data sets
- zFS indirect cataloging support
- Cloning assistance provided
- A note about queiscing file systems
- Extra! Dynamic activating z/OS UNIX corrective service

Cloning Review:
Why We Clone

- At some point after you have finished installing z/OS software you will probably need to make a copy of it ("clone" it)
- Some reasons are:
  - For backup
    - A backup copy is a copy of the z/OS product set (z/OS and other products you have installed on the same set of volumes using the same SMP/E zones) that:
      - *Resides on different volumes with different volume labels*
      - *Includes copies of the associated SMP/E zones with different names that point to the data sets on the new volumes*
      - *May include copies of the associated catalogs with different names*
      - *Can be IPLed in place of the original copy when the necessary setup has been done after making the copy*
  - To move the software to another system ("deployment")
  - To create another SMP/E-serviceable copy for installing service or other products

Cloning Review:
Cloning Assumptions

- **Cloning should include:**
  - The current target software instance (target PDS and PDSEs, z/OS UNIX file system data sets, optionally dlibs, and SMP/E CSIs)
- **Cloning may also include:**
  - Appropriate operational data sets and configuration files



- For this presentation, we'll focus specifically on the z/OS UNIX file system data sets.
  - PDS and PDSE cloning do not seem to raise as many questions as z/OS UNIX file system data sets do!
  - SMP/E CSI data sets and entries are not covered, but remember that your SMP/E CSI should be updated as part of the cloning activity, and all entries should have a correct location and name!

Differences between MVS data sets and z/OS UNIX mounted file systems

- Two different MVS data sets with the same name can be actively accessed on a single system as long as they reside on different volumes (with different volume serials) and they do not need to be cataloged
    - You need to specify somehow which data set you want by providing the volume serial where it resides.
- Two different z/OS UNIX file systems with the same data set name, cannot be mounted at the same time on a single system nor in a shared file system environment (even if you try to use different z/OS UNIX mount point directories) and they need to be cataloged to be mounted.
    - This is an existing restriction since the beginning of z/OS UNIX.

z/OS UNIX File System Data Sets

- z/OS UNIX File System Data Sets contain many important parts of the z/OS operating system and associated products:
  - TCP/IP, z/OS Shells and Utilities, Infoprint Server, Java SDK, XML Toolkit, Ported Tools, …
- The choice today for z/OS Unix file system data sets is:
  - **<u>HFS:</u>** Stabilized. Non-VSAM data set.
    - Can be indirectly cataloged just as target library PDS and PDSEs can.
    - Both cataloged and uncataloged HFS can be on non-SMS DASD.
    - Cataloged HFS can be multi-volume on SMS-DASD (up to 255 extents on 59 volumes with up to 123 extents per volume)
  - **<u>zFS:</u>** Enhancements provided that customers need! VSAM LDS, therefore must be cataloged.
    - Support for indirectly cataloged zFS (not VSAM LDS in general!)
    - Non-SMS or SMS, can be up to 59 volumes. (EAS-eligible on EAVs!)
      - Indirect catalog zFS support is limited to one volume, however.
    - Architected maximum size for an aggregate: 4 TB

Major management zFS difference from HFS: VSAM

- VSAM data sets must be cataloged to access them (regardless of whether they are mounted)
- VSAM must be SMS-managed (needs a STORAGECLASS) if larger than 4 GB
  Needs a DATACLASS that provides extended addressability
  - (4 GB is about 5825 cylinders of 3390 DASD)
- All SMS-managed data sets must be cataloged

- In a nutshell, for cloning: It's the catalog.

Catalog’s role in accessing data sets

- Master Catalog identified in
  - LOADxx member of SYS1.PARMLIB, or
  - SYSCATxx member of SYS1.NUCLEUS
- User Catalogs defined in Master Catalog, and often used for VSAM data sets.
  - `DEFINE ALIAS NAME(HLQ) RELATE(USERCAT.HLQ)`
    - USERCAT.HLQ -> user catalog that contains data sets that begins with HLQ

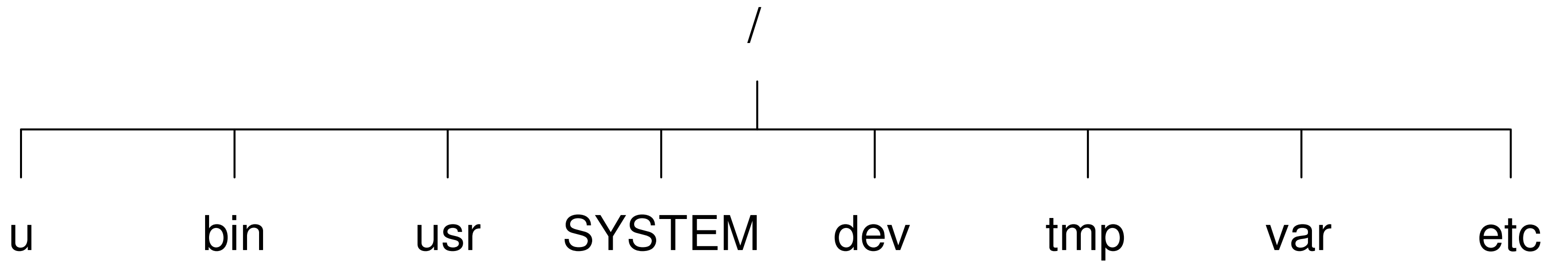
- System Symbolics and Extended Alias in catalog entries:
  - For the volume: **NONVSAM** ------- SYS1.LINKLIB … VOLUMES **VOLSER------ &SYSR1.** DEVTYPE------X'00000000‘
    NONVSAM ------- OMVS.ROOT.HFS … VOLUMES **VOLSER------ &SYSR1.** DEVTYPE------X'00000000‘
  - For aliasing a data set name: ALIAS --------- SYS1.PRODUCT … ASSOCIATIONS SYMBOLIC- **SYS1.&SYSR1..PRODUCT** RESOLVED-SYS1.ZD113.PRODUCT
  - For user catalog names: ALIAS --------- SYS1.ZFS … ASSOCIATIONS SYMBOLIC- **&ZFSVL..UCAT** RESOLVED-ZCSDW.UCAT
  - For zFS indirect exploitation: **CLUSTER** ------- ZFS.ROOT … VOLUME **VOLSER------------ &ZFSVL** DEVTYPE------X'00000000'

Shared File System Concepts

- In a Shared File System environment (BPXPRMxx specifies SYSPLEX(YES) on all systems):
  - There is a single z/OS UNIX file system hierarchy across the multiple systems in the shared file system environment
  - When a z/OS UNIX file system is mounted, it is globally mounted across all the systems
  - Once a file system (name) is mounted into the z/OS UNIX file system hierarchy, you cannot mount a file system with the same name
    - On the same mount point or even a different mount point
  - Several z/OS releases or service levels can be active in the shared file system environment at the same time

The z/OS UNIX file system

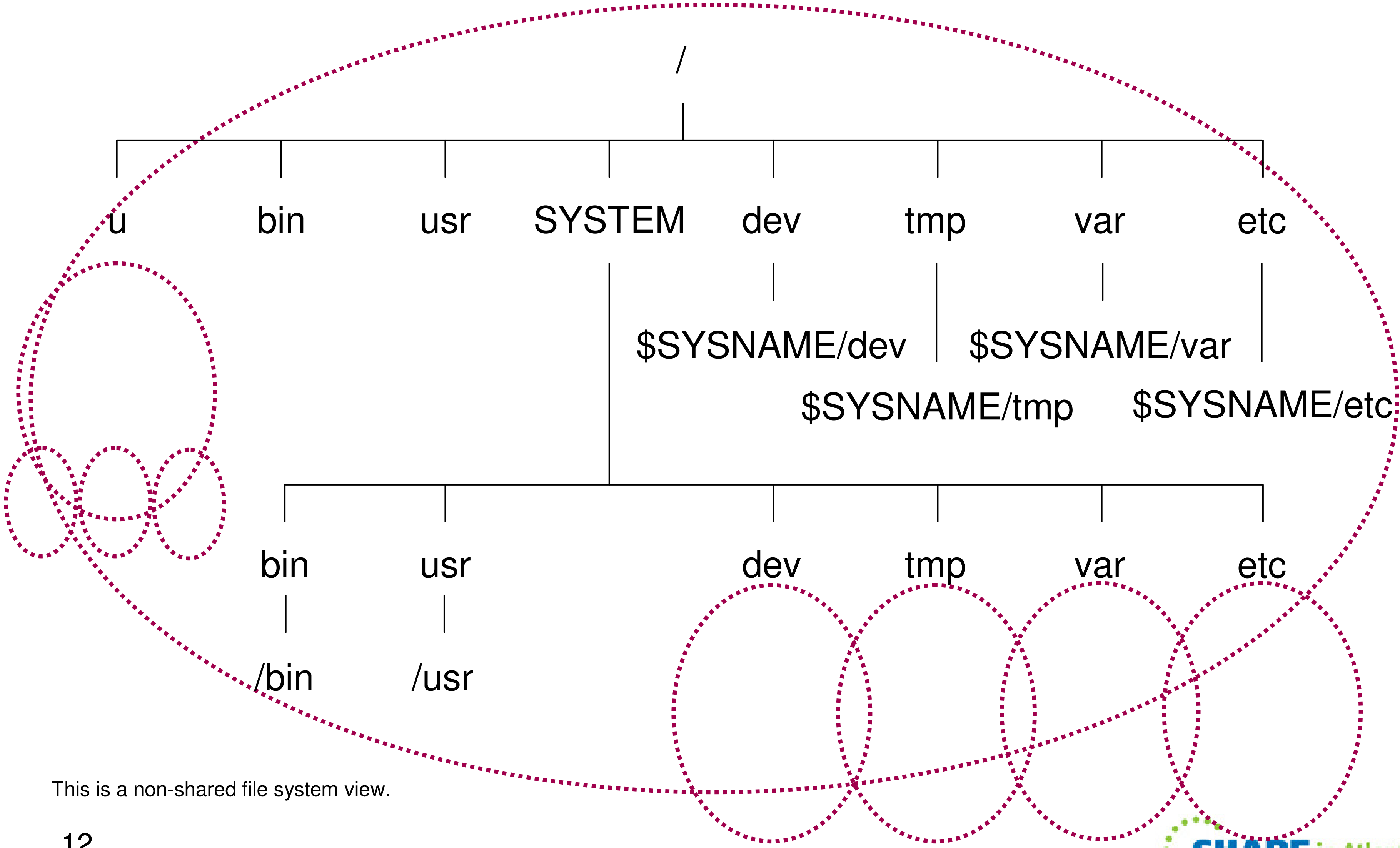
- This is how an end user views the “top” of the z/OS UNIX hierarchical file system
- It looks like files and directories organized in a hierarchical tree



/

u bin usr SYSTEM dev tmp var etc

The z/OS UNIX hierarchical file system



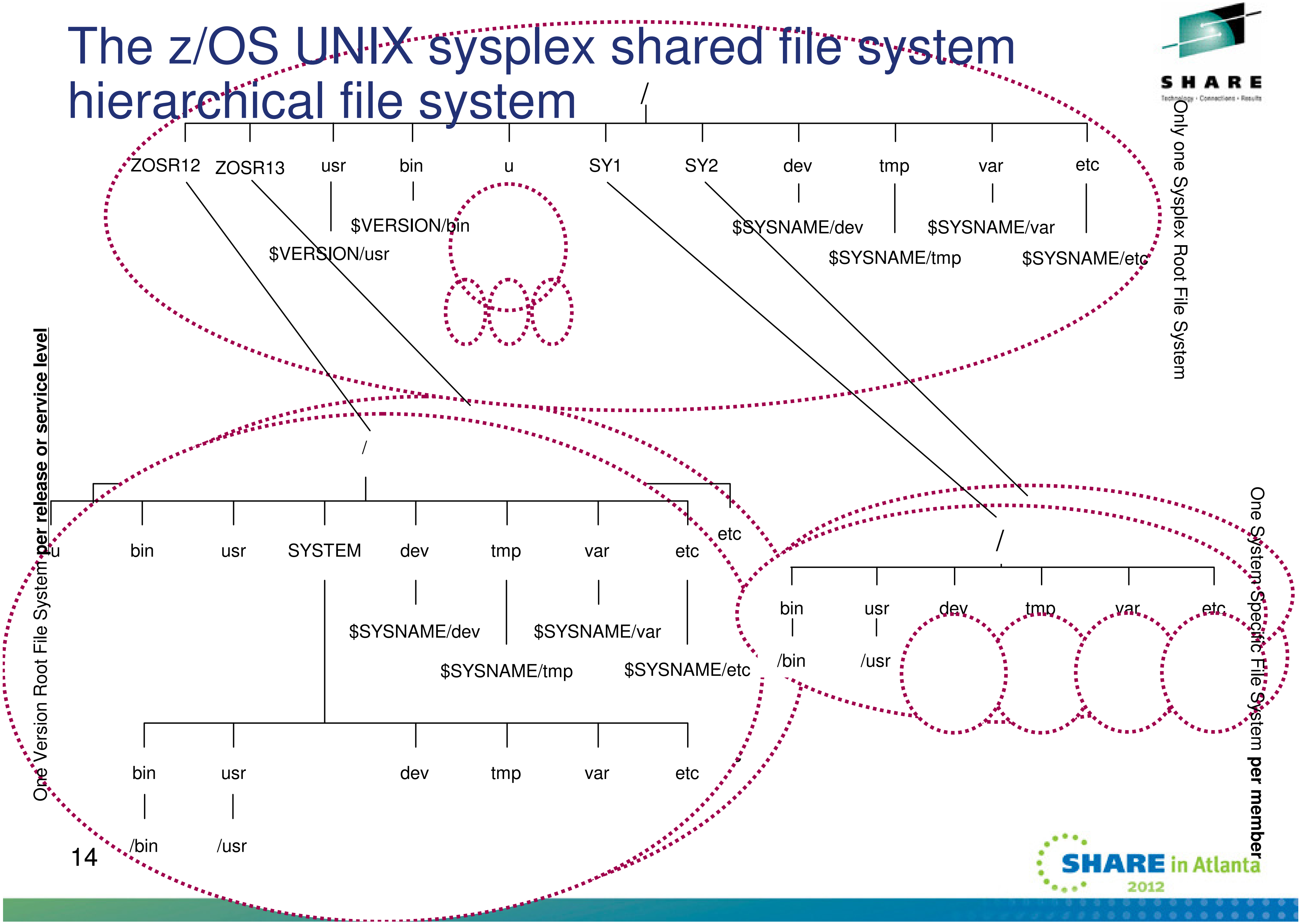
This is a non-shared file system view.

Shared File System Data Sets

- **Sysplex root:**
  - Only one for all systems in the shared file system environment
  - Does not contain z/OS system software itself, but contains directories from which you mount z/OS system software.
  - Is very small and should be mounted read-only. *
- **System-specific file systems:**
  - One per member in the shared file system environment.
  - A place where system-specific file systems are mounted (/etc, /var, /tmp, /dev)
  - Does not contain z/OS system software itself.
  - Is very small and should be mounted read-write.
- **Version root file systems:**
  - *Contains the z/OS system software!* This is where cloning is concerned.
  - Is rather large (thousands of cylinders)
  - Should be mounted read-only. *

* UNMOUNT REMOUNT is helpful, if you need to add directories from time to time.

The z/OS UNIX sysplex shared file system hierarchical file system



Some BPXPRMxx Statements for a Shared File System Specification

- SYSPLEX(YES)

  Indicates using shared file system environment
- VERSION(‘ZOSR13')

  Indicates how you want the version identified
- ROOT FILESYSTEM('OMVS.SYSPLEX.ROOT') TYPE(ZFS) MODE(READ)

  The sysplex root – the same for every system

- MOUNT FILESYSTEM('OMVS.&SYSNAME..SYSTEM.ZFS') TYPE(ZFS) MODE(RDWR) UNMOUNT MOUNTPOINT('/&SYSNAME.')

  The system-specific file system

- **MOUNT FILESYSTEM('OMVS.&ZFSVL..ROOT.ZFS') TYPE(ZFS) MODE(READ) MOUNTPOINT('/$VERSION')**

  The version root – shared between systems if applicable

- MOUNT FILESYSTEM('OMVS.&SYSNAME..ETC') TYPE(ZFS) MODE(RDWR) UNMOUNT MOUNTPOINT('/&SYSNAME./etc')

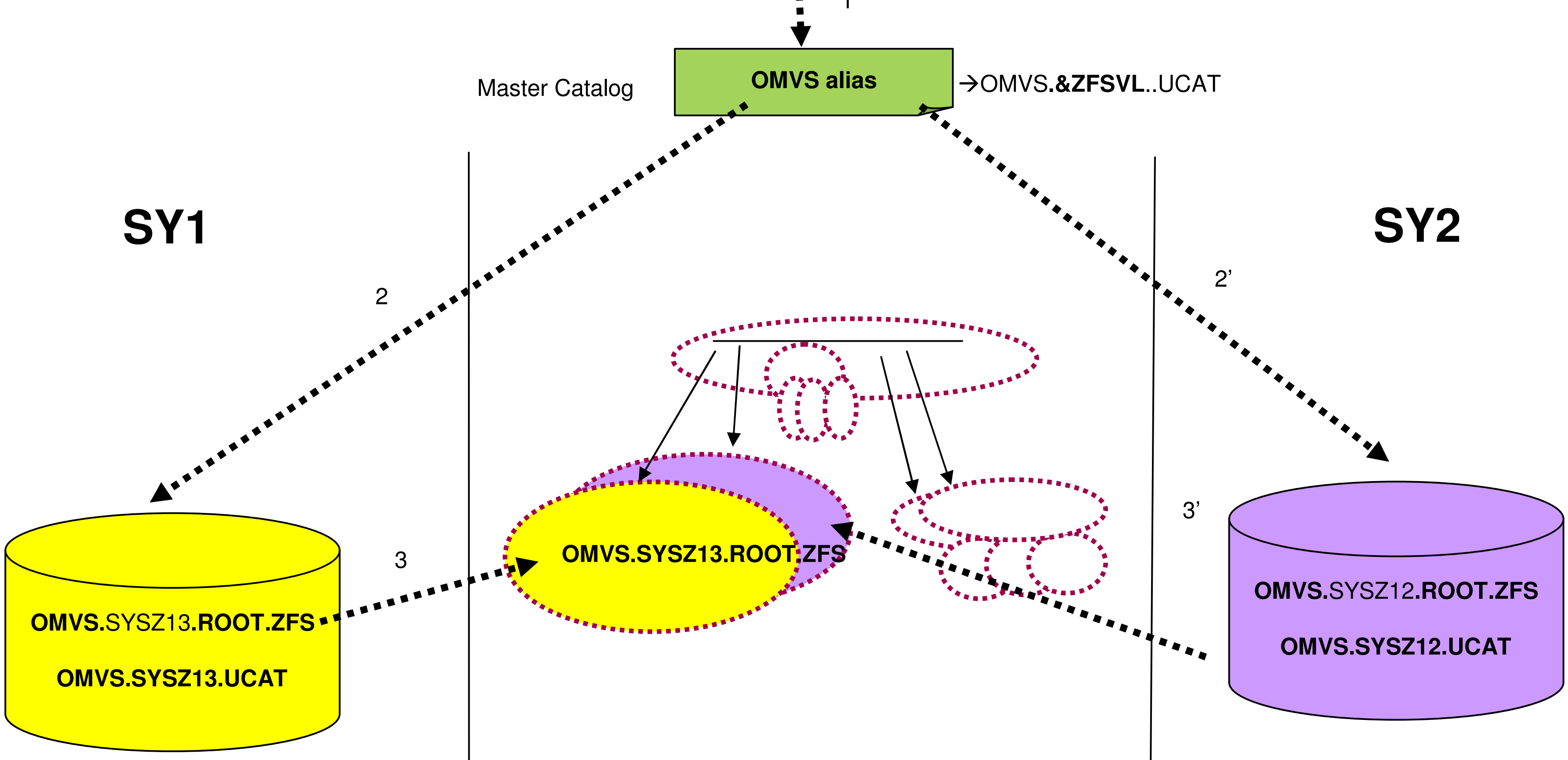
...

When using system symbolics, these statement can be shared between systems!

BPXPRMxx

```
MOUNT  FILESYSTEM('OMVS.&ZFSVL..ROOT.ZFS')
TYPE(ZFS) MODE(READ) MOUNTPOINT('/$VERSION')
```

An Execution Example:



Master Catalog

OMVS alias

→OMVS.**&ZFSVL.**.UCAT

SY1

SY2

OMVS.SYSZ13**.ROOT.ZFS**

OMVS.SYSZ13.UCAT

OMVS.SYSZ13.ROOT.ZFS

OMVS.SYSZ12**.ROOT.ZFS**

OMVS.SYSZ12.UCAT

Volume: SYSZ13

System Symbol &ZFSVL. = SYSZ13

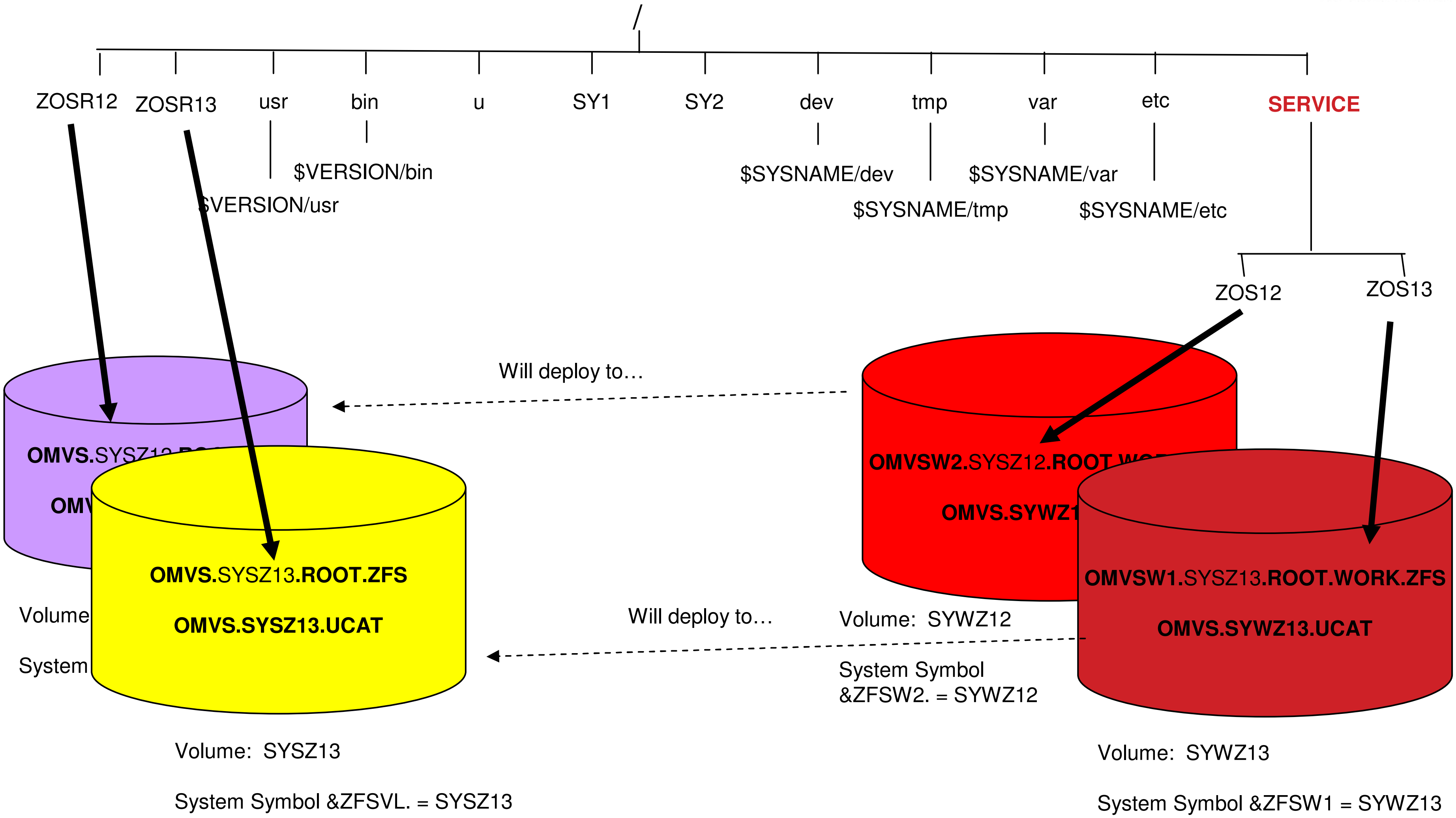
Volume: SYSZ12

System Symbol &ZFSVL. = SYSZ12

How do I clone to get this type of environment?

A Servicing Example:

(same sysplex)

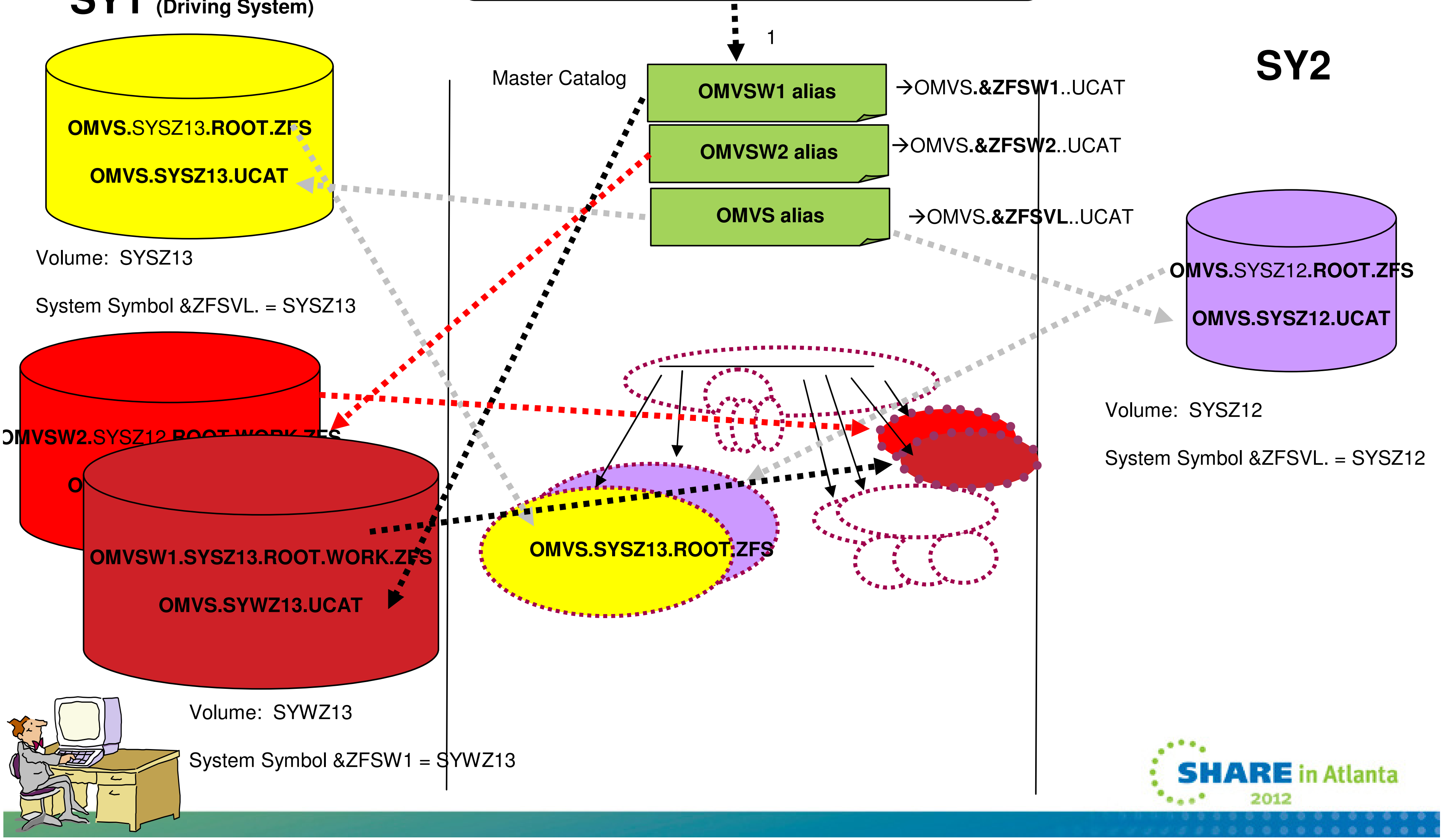


A Service Example:

BPXPRMxx

```
MOUNT  FILESYSTEM('OMVS.&ZFSVL..ROOT.ZFS')
TYPE(ZFS) MODE(READ) MOUNTPOINT('/$VERSION')
```

```
MOUNT  FILESYSTEM('OMVSW1.SYSZ13.ROOT.WORK.ZFS ')
 TYPE(ZFS) MODE(READ) MOUNTPOINT('/SERVICE/ZOS13')
```



SY1 (Driving System)

OMVS.SYSZ13.ROOT.ZFS

OMVS.SYSZ13.UCAT

Volume: SYSZ13

System Symbol &ZFSVL. = SYSZ13

OMVSW2.SYSZ12.ROOT.WORK.ZFS

O

OMVSW1.SYSZ13.ROOT.WORK.ZFS

OMVS.SYWZ13.UCAT

Volume: SYWZ13

System Symbol &ZFSW1 = SYWZ13

1

Master Catalog

OMVSW1 alias →OMVS.&ZFSW1..UCAT

OMVSW2 alias →OMVS.&ZFSW2..UCAT

OMVS alias →OMVS.&ZFSVL..UCAT

OMVS.SYSZ13.ROOT.ZFS

SY2

OMVS.SYSZ12.ROOT.ZFS

OMVS.SYSZ12.UCAT

Volume: SYSZ12

System Symbol &ZFSVL. = SYSZ12

Some observations for this example:

- zFS was used, but it very well could have been HFS! Since the file system must be cataloged to be mounted and accessed, it makes no difference in our example whether we used HFS or zFS. We were looking at the placement of the file systems, and how they were cataloged so that we could get access to them concurrently.
- The driving and target systems were in the same sysplex, and shared volumes.
  - This is a more complicated case then across-sysplexes, where sharing isn't done. The alias names in the same sysplex – when sharing the master catalog – cannot clash. Across sysplexes, the alias names used could be in common.
- The alias names used (OMVS , OMVSW1, and OMVSW2) could have used the target zone name, the volume name, or another name that is helpful for you.
  - System symbols for the usercat names were used, but aren't necessary.
- The SERVICE mountpoints (ZOS13 and ZOS12) happen to match the BPXPRMxx VERSION names. They could have been the target zone name, the volume name, or another name that is helpful for you.

How do I copy the version root file system, with cataloging in mind?

- Most people copy the file system data set using MVS utilities (as opposed to z/OS UNIX utilities like pax or copytree).
  - We will assume DFSMSdss will be used.
- How you copy your z/OS system software data sets relies upon your environment for deployment:
  - Same sysplex or across sysplexes? Shared volumes? Installation data set names different from execution data set names? SMS-managed?
    - We'll assume the environment we've shown previously. Other ways may work just as well.
- To copy (clone) the version root file system (either HFS or zFS), decide if you want:
  - *Full volume copy*: will copy the entire volume (which may be more than you want or need, depending on your environment). Note that this gives you another copy of the file system with the same name, and only one can be cataloged at a time. Therefore, a subsequent rename or recatalog may be necessary after this sort of copy if you want to access it.
    - If you have any usercats on the volume, a full volume copy isn't a good choice. For zFS file systems that are using a usercat on the volume, this is not a recommended choice.
  - *Data set copy*:
    - *Logical data set copy*: can rename the file system on the copy operation, and depending on your catalog structure may be immediately accessible. A popular choice that works well in many situations. (z/OSMF's Deployment Manager task uses this method.)
    - *Physical data set copy*: will provide you with a copy of the file system data set that is not cataloged. A subsequent rename or recatalog may be necessary. Note that a rename on a physical data set copy is not currently available.
      - *If you are using indirect zFS cataloging, this copy method is best, as you will be using an existing catalog entry that makes the file system accessible.*

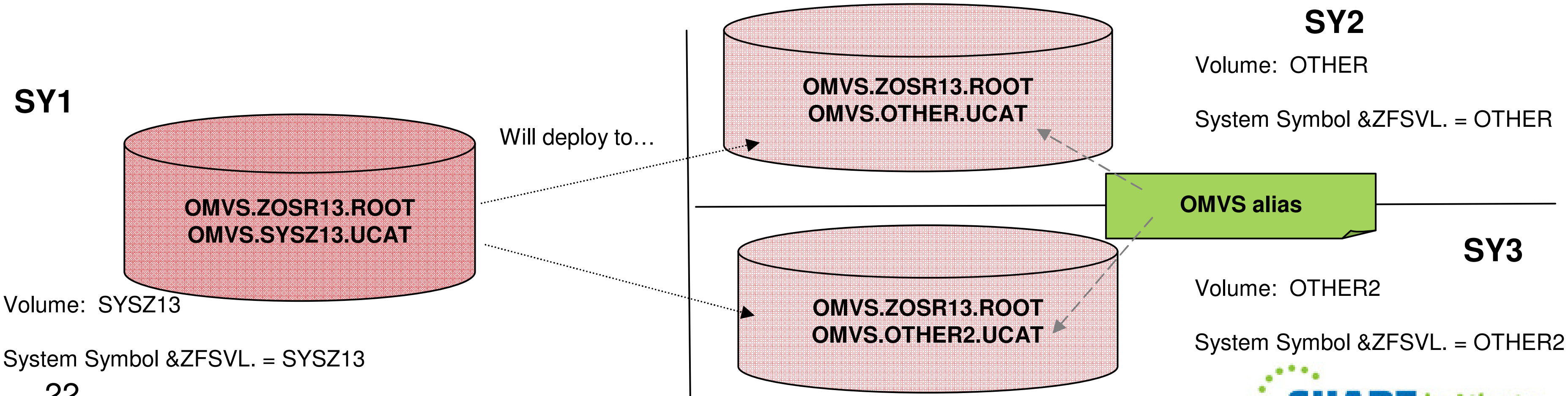
Logical data set copy for file system sample

```
//COPYFS    EXEC  PGM=ADRDSSU,REGION=0M
//SYSPRINT  DD SYSOUT=*
//SYSIN    DD    *
   COPY DATASET( -
   INCLUDE(OMVSW1.SYSZ13.ROOT.WORK.ZFS)) -
   RENAMEU(OMVSW1.SYSZ13.ROOT.WORK.ZFS, -
            OMVS.SYSZ13.ROOT.ZFS) -
   OUTDYNAM(SYSZ13) –
   CATALOG ALLDATA(*)   ALLEXCP -
   CANCELERROR  TOLERATE(ENQF)   -
```

- This is applicable to either HFS or zFS.
    - Logical data set copy is not recommended if using zFS indirect cataloging support (which relies upon a system symbol for the volume name).
- CATALOG follows normal catalog search order.

zFS Indirect Cataloging Support (as of zOS R12)

- Allows for use of a system symbol as the volser for zFS data sets (not other VSAM LDS data sets).
- Provides a simple and flexible way to clone zFS data sets, where you do not need to rename the data set during the copy.
  - Ideal for environments where the file system name doesn't need to change between the driving system and target system.
  - May share mcat on those deployed target systems. (Remember that unique mount name requirement, though!)
    - Includes all non-shared file system environments.



SY1

OMVS.ZOSR13.ROOT
OMVS.SYSZ13.UCAT

Volume: SYSZ13

System Symbol &ZFSVL. = SYSZ13

Will deploy to…

SY2

OMVS.ZOSR13.ROOT
OMVS.OTHER.UCAT

Volume: OTHER

System Symbol &ZFSVL. = OTHER

OMVS alias

SY3

OMVS.ZOSR13.ROOT
OMVS.OTHER2.UCAT

Volume: OTHER2

System Symbol &ZFSVL. = OTHER2

Using zFS Indirect Cataloging Support: Physical data set copy for file system sample

```
//COPYFS    EXEC  PGM=ADRDSSU,REGION=0M
//SYSPRINT  DD SYSOUT=*
//SYSIN    DD    *
  COPY DATASET(INC(OMVS.ZOSR13.ROOT)) -
    PHYSINDYNAM ( (SYSZ13) ) -
    OUTDYNAM ( (OTHER) ) -
    ALLDATA(*)
```

- Creates an uncataloged copy.
- Cannot currently rename data sets during a physical copy.
- See quiescing note, may want to unmount file system first.

```
//RENAME    EXEC  PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN    DD    *
   DEFINE CLUSTER -
   (NAME(OMVS.ZOSR13.ROOT) -
   LINEAR -
   VOLUMES(&ZFSVL) -
   RECATALOG)
```

- Recatalogs the zFS to start using the system symbol for the volume.
- **Only needs to be recataloged once, so subsequent deployments can reuse the catalog entry!**

A note about quiescing…

- When file systems (either HFS or zFS) are mounted and they need to be copied or dumped, you need to understand about quiescing.

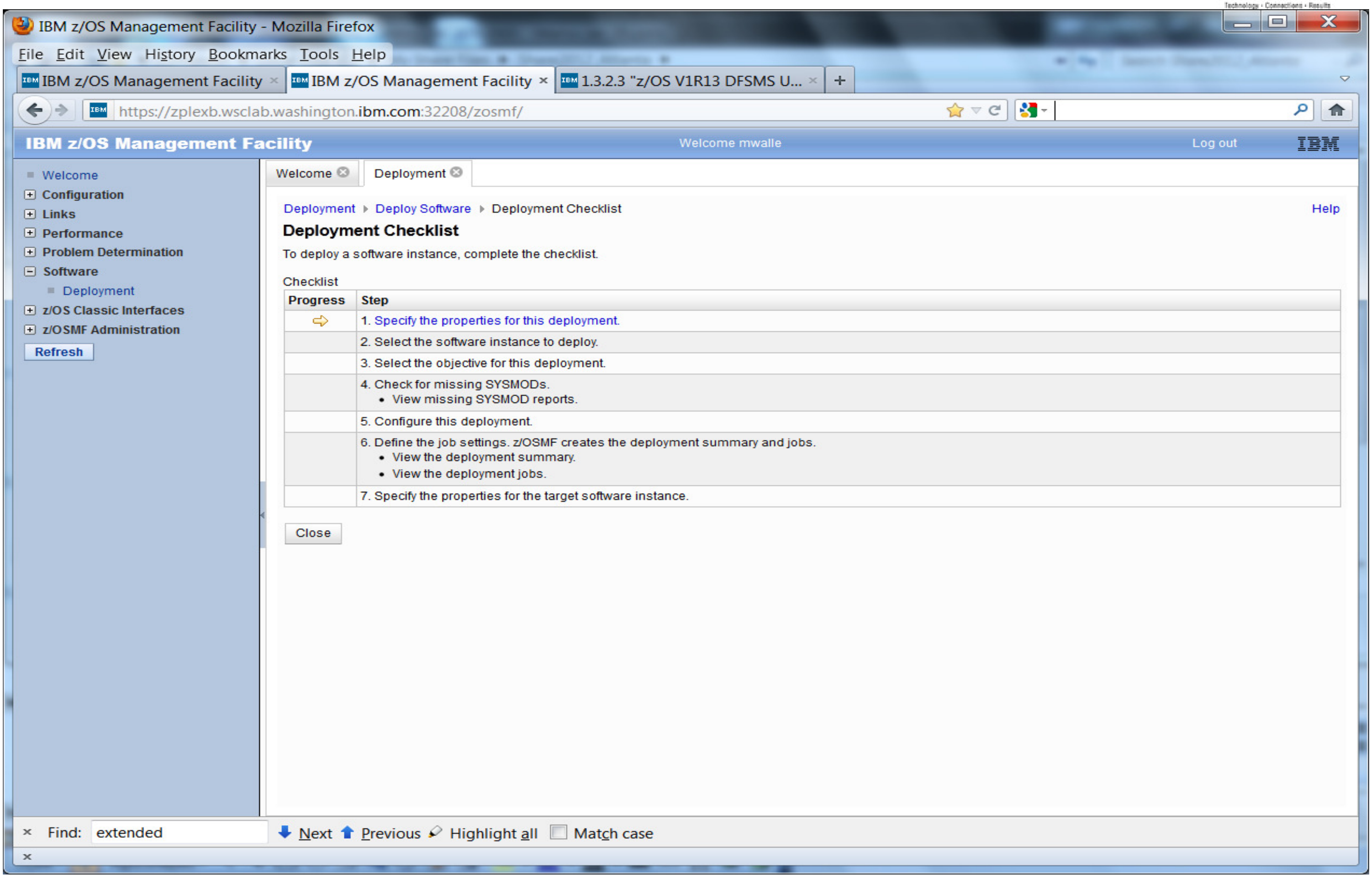
- When doing an DFSMSdss logical COPY or DUMP on a mounted file system, DFSMSdss automatically will quiesce the file system.

- When doing a physical COPY or DUMP on a mounted file system, DFSMSdss does not automatically quiesce the file system.
    - This means that exclusive ENQs may not succeed during the COPY or DUMP operation.
    - To avoid exclusive ENQ failures, you can:
        - Unmount the file system prior to the COPY or DUMP
    - **Attention:** Exercise caution if you use TOL(ENQF) during a physical copy or dump of zFS data sets. Unlike other types of data sets, if a zFS data set is updated during a physical copy or dump with TOL(ENQF), it will likely result in an unusable data set.

Cloning Assistance

- Documentation: z/OS Planning for Installation, Appendix D. *Making a copy of your system software (cloning)*
- JCL samples: IEACLN* samples in SYS1.SAMPLIB
- z/OSMF's Software Deployment (for z/OS R13):
  - Use a GUI interface to define software instances and deployments.
  - Supports a local (same sysplex) deployment, or a remote (across sysplexes) deployment.
  - Produces customized JCL jobs for cloning, for you to run.
    - Supports HFS and zFS, but not zFS indirect cataloging.

z/OSMF Software Deployment



IBM z/OS Management Facility - Mozilla Firefox
File Edit View History Bookmarks Tools Help
IBM z/OS Management Facility
IBM z/OS Management Facility
1.3.2.3 "z/OS V1R13 DFSMS U...
https://zplexb.wsclab.washington.ibm.com:32208/zosmf/
IBM z/OS Management Facility
Welcome mwalle
Log out
IBM
Welcome
Configuration
Links
Performance
Problem Determination
Software
Deployment
z/OS Classic Interfaces
z/OSMF Administration
Refresh
Welcome
Deployment
Deployment ▸ Deploy Software ▸ Deployment Checklist
Help
Close
Find: extended
Next
Previous
Highlight all
□ Match case

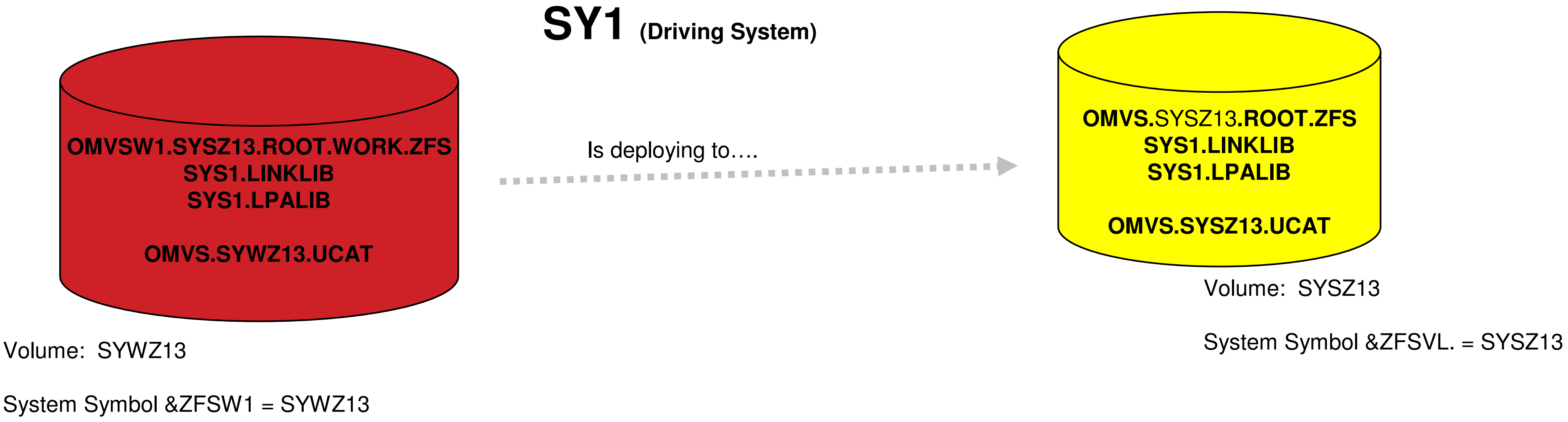
Deployment Checklist

To deploy a software instance, complete the checklist.

Checklist

Progress	Step
⇨	1. Specify the properties for this deployment.
	2. Select the software instance to deploy.
	3. Select the objective for this deployment.
	4. Check for missing SYSMODs. • View missing SYSMOD reports.
	5. Configure this deployment.
	6. Define the job settings. z/OSMF creates the deployment summary and jobs. • View the deployment summary. • View the deployment jobs.
	7. Specify the properties for the target software instance.

Dynamically Activating z/OS UNIX Service



SY1 (Driving System)

OMVSW1.SYSZ13.ROOT.WORK.ZFS
SYS1.LINKLIB
SYS1.LPALIB

OMVS.SYWZ13.UCAT

Is deploying to….

OMVS.SYSZ13.ROOT.ZFS
SYS1.LINKLIB
SYS1.LPALIB

OMVS.SYSZ13.UCAT

Volume: SYSZ13

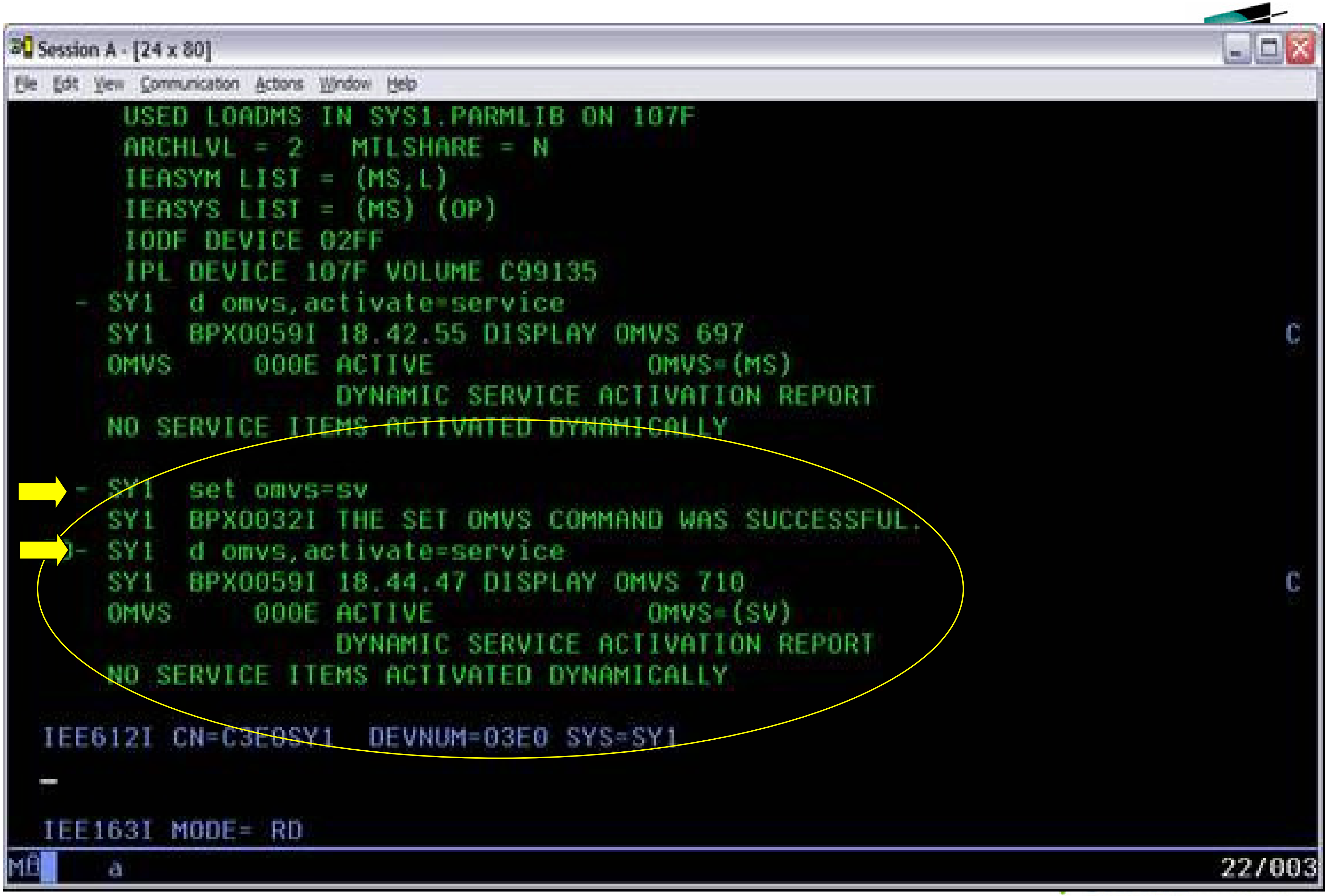
System Symbol &ZFSVL. = SYSZ13

Volume: SYWZ13

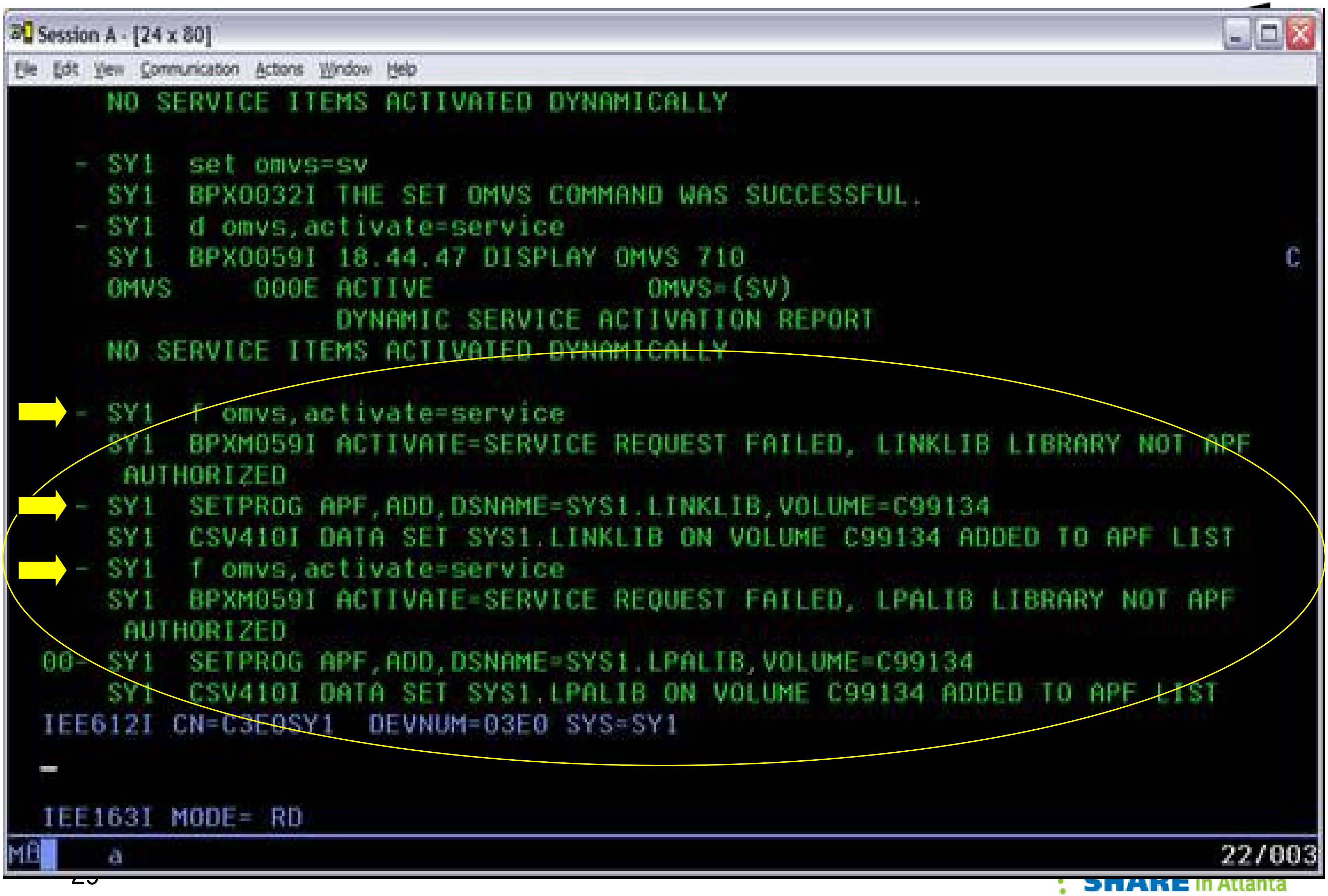
System Symbol &ZFSW1 = SYWZ13

To Use z/OS UNIX Dynamic Service Activation:

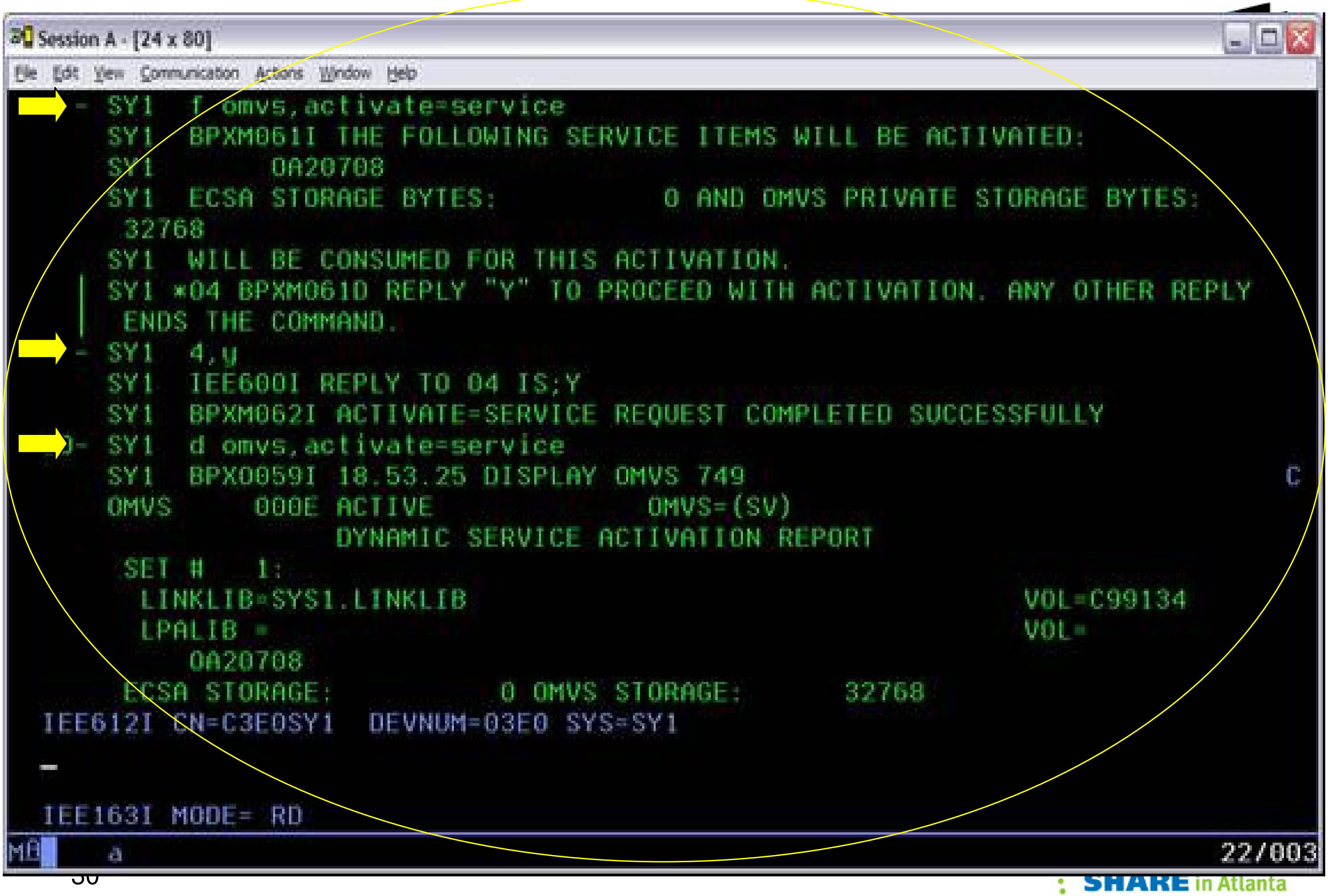
1. SMP/E install the corrective PTF on the “**work**” volume. (Allowed PTFs will be marked with ++HOLD DYNACT.)
2. Set BPXPRMxx on “live” system, point to the target LINKLIB and LPALIB on “**work**”
   - “**work**” LINKLIB and LPALIB must be APF authorized on “live” system
3. F OMVS,ACTIVATE=SERVICE to activate
4. D OMVS,ACTIVATE=SERVICE to view
5. F OMVS,DEACTIVATE=SERVICE to deactivate



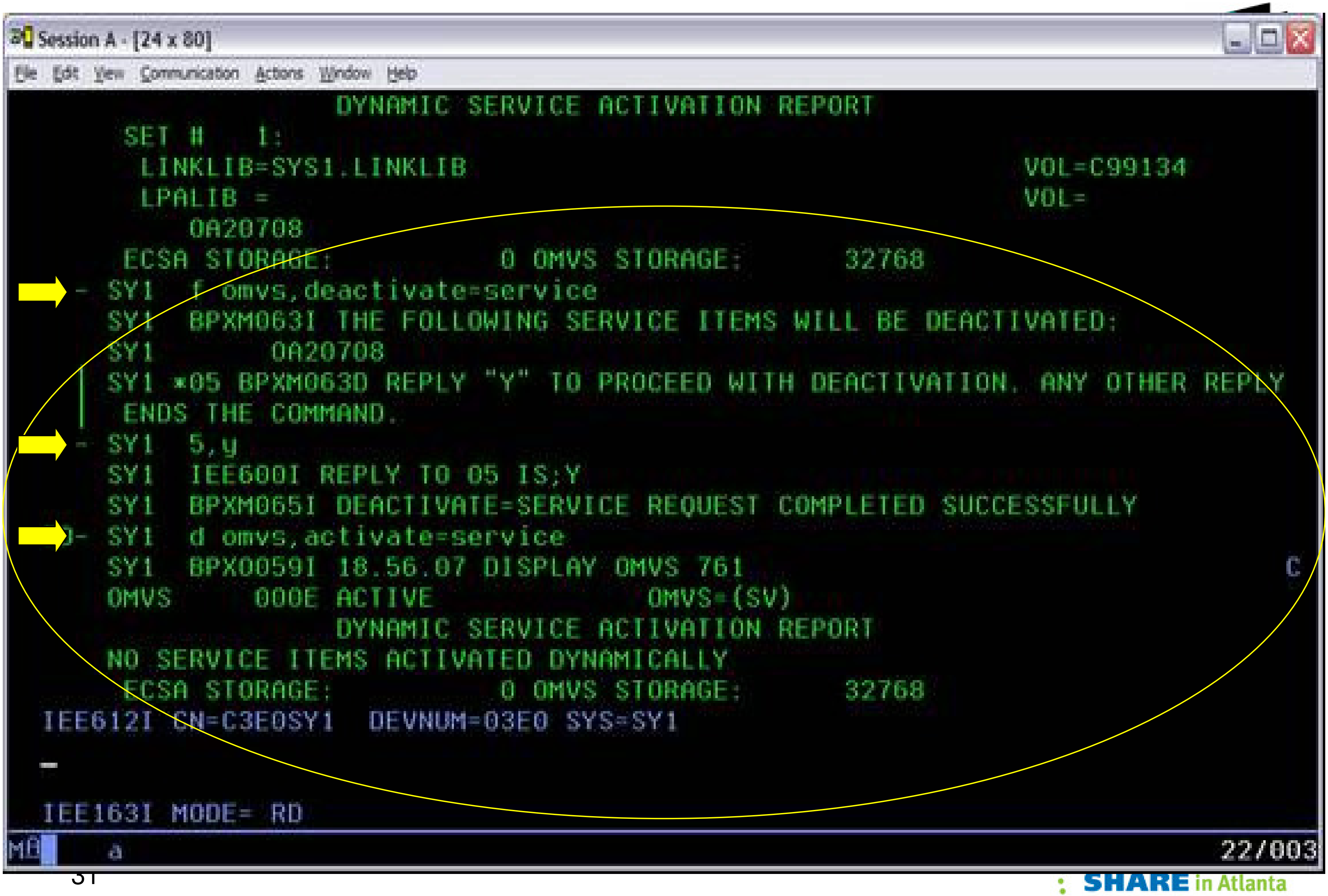
```
       USED LOADMS IN SYS1.PARMLIB ON 107F
       ARCHLVL = 2   MTLSHARE = N
       IEASYM LIST = (MS,L)
       IEASYS LIST = (MS) (OP)
       IODF DEVICE 02FF
       IPL DEVICE 107F VOLUME C99135
    - SY1  d omvs,activate=service
      SY1  BPXO059I 18.42.55 DISPLAY OMVS 697                                C
      OMVS     000E ACTIVE            OMVS=(MS)
                    DYNAMIC SERVICE ACTIVATION REPORT
      NO SERVICE ITEMS ACTIVATED DYNAMICALLY

    - SY1  set omvs=sv
      SY1  BPXO032I THE SET OMVS COMMAND WAS SUCCESSFUL.
    - SY1  d omvs,activate=service
      SY1  BPXO059I 18.44.47 DISPLAY OMVS 710                                C
      OMVS     000E ACTIVE            OMVS=(SV)
                    DYNAMIC SERVICE ACTIVATION REPORT
      NO SERVICE ITEMS ACTIVATED DYNAMICALLY

  IEE612I CN=C3E0SY1  DEVNUM=03E0 SYS=SY1

  -

  IEE163I MODE= RD
```



```
    NO SERVICE ITEMS ACTIVATED DYNAMICALLY

  - SY1  set omvs=sv
    SY1  BPXO032I THE SET OMVS COMMAND WAS SUCCESSFUL.
  - SY1  d omvs,activate=service
    SY1  BPXO059I 18.44.47 DISPLAY OMVS 710                                   C
    OMVS     000E ACTIVE                OMVS=(SV)
                  DYNAMIC SERVICE ACTIVATION REPORT
    NO SERVICE ITEMS ACTIVATED DYNAMICALLY

  - SY1  f omvs,activate=service
    SY1  BPXM059I ACTIVATE=SERVICE REQUEST FAILED, LINKLIB LIBRARY NOT APF
     AUTHORIZED
  - SY1  SETPROG APF,ADD,DSNAME=SYS1.LINKLIB,VOLUME=C99134
    SY1  CSV410I DATA SET SYS1.LINKLIB ON VOLUME C99134 ADDED TO APF LIST
  - SY1  f omvs,activate=service
    SY1  BPXM059I ACTIVATE=SERVICE REQUEST FAILED, LPALIB LIBRARY NOT APF
     AUTHORIZED
00- SY1  SETPROG APF,ADD,DSNAME=SYS1.LPALIB,VOLUME=C99134
    SY1  CSV410I DATA SET SYS1.LPALIB ON VOLUME C99134 ADDED TO APF LIST
IEE612I CN=C3E0SY1  DEVNUM=03E0 SYS=SY1

-

IEE163I MODE= RD
MB    a                                                               22/003
```



```
- SY1   f omvs,activate=service
  SY1   BPXM061I THE FOLLOWING SERVICE ITEMS WILL BE ACTIVATED:
  SY1        OA20708
  SY1   ECSA STORAGE BYTES:          0 AND OMVS PRIVATE STORAGE BYTES:
   32768
  SY1   WILL BE CONSUMED FOR THIS ACTIVATION.
| SY1  *04 BPXM061D REPLY "Y" TO PROCEED WITH ACTIVATION. ANY OTHER REPLY
|  ENDS THE COMMAND.
- SY1   4,y
  SY1   IEE600I REPLY TO 04 IS;Y
  SY1   BPXM062I ACTIVATE=SERVICE REQUEST COMPLETED SUCCESSFULLY
- SY1   d omvs,activate=service
  SY1   BPXO059I 18.53.25 DISPLAY OMVS 749                              C
  OMVS      000E ACTIVE              OMVS=(SV)
                DYNAMIC SERVICE ACTIVATION REPORT
   SET #   1:
    LINKLIB=SYS1.LINKLIB                                    VOL=C99134
    LPALIB =                                                VOL=
         OA20708
   ECSA STORAGE:          0 OMVS STORAGE:      32768
IEE612I CN=C3E0SY1  DEVNUM=03E0 SYS=SY1

_

IEE163I MODE= RD
MB    a                                                          22/003
```



```
              DYNAMIC SERVICE ACTIVATION REPORT
       SET #    1:
        LINKLIB=SYS1.LINKLIB                                    VOL=C99134
        LPALIB =                                                VOL=
           OA20708
       ECSA STORAGE:           0 OMVS STORAGE:      32768
     - SY1   f omvs,deactivate=service
       SY1   BPXM063I THE FOLLOWING SERVICE ITEMS WILL BE DEACTIVATED:
       SY1        OA20708
     | SY1 *05 BPXM063D REPLY "Y" TO PROCEED WITH DEACTIVATION. ANY OTHER REPLY
     |  ENDS THE COMMAND.
     - SY1   5,y
       SY1   IEE600I REPLY TO 05 IS;Y
       SY1   BPXM065I DEACTIVATE=SERVICE REQUEST COMPLETED SUCCESSFULLY
    D- SY1   d omvs,activate=service
       SY1   BPXO059I 18.56.07 DISPLAY OMVS 761                                 C
       OMVS     000E ACTIVE             OMVS=(SV)
              DYNAMIC SERVICE ACTIVATION REPORT
       NO SERVICE ITEMS ACTIVATED DYNAMICALLY
        ECSA STORAGE:          0 OMVS STORAGE:      32768
    IEE612I CN=C3E0SY1  DEVNUM=03E0 SYS=SY1

    _

    IEE163I MODE= RD
MA      a                                                                22/003
```

Summary

- **Data sets and file systems differences:** can actively access data sets with the same names on different volumes. Cannot mount (access) a file system with the same name as one already mounted.
- **HFS and zFS differences** - VSAM, and therefore catalog!
- **Catalog's Role in using file systems** – it's got to be cataloged to be mounted to be accessed. Use system symbols, use aliases!
- **Shared file system types:** sysplex root, system-specific, and version root file systems. Version root is where the cloning concerns usually stem.
- **Copying file system data sets** – volume or data set? Logical or physical copy? Keep catalog requirements in mind!
- **zFS indirect cataloging support** – best when file system names do not need to change when deploying
- **Quiescing** – know when it's done automatically for you, and what you may do when it's not automatic (unmount?)
- **Cloning assistance provided** – documentation, samples in SYS1.SAMPLIB, z/OSMF Software Deployment
- **Dynamic activating z/OS UNIX corrective service** - can save you an IPL!

References

- z/OS Planning for Installation, especially Appendix D
- z/OS DFSMS Managing Catalogs (Defining Aliases for a User Catalog)
- z/OS Basic Skills Information Center: z/OS system installation and management