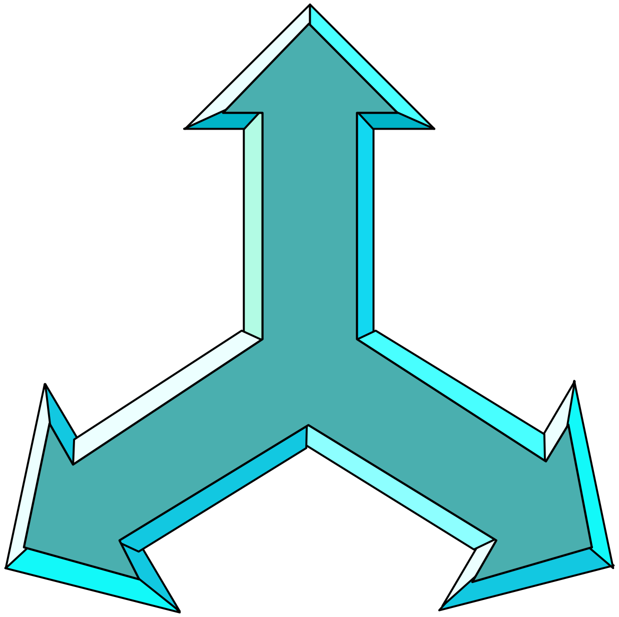
OS/390 UNIX Filesystem Sysplex Support - (Shared HFS)

Parts 1 & 2



Kershaw Mehta
kershaw@us.ibm.com

IBM
OS/390 UNIX System Services and Language Environment Design

Session Objectives

- Understand how we implemented the file system sysplex solution.
- How to setup your filesystems in a sysplex environment
  - New sample jobs that need to be run
  - Changes to BPXPRMxx PARMLIB member
  - New keywords in BPXPRMxx for sysplex support
  - Follow how path lookup is performed
- Understand new terms
  - Sysplex Root
  - System-specific HFS
  - Version HFS

Design Objectives

- One root filesystem structure that would work both in a sysplex and non-sysplex environments.
- Allow for *"rolling IPL's"* to introduce new systems into sysplex.
- Allow for backout of systems from sysplex.
- Provide for multiple "versions" of "root filesystem"
  - some systems are using SysresX and corresponding "root HFS VersionX" while others are using SysresY and "root HFS VersionY"
    - *"rolling IPLs"*

Design Objectives ...

- For sysplex environment, provide means of accessing all filesystems from any system for system management purposes.
- Ensure no migration actions need to be performed by system programmers and their BPXPRMxx PARMLIB member when running in non-sysplex environments.
- Build upon existing filesystem structures and philosphies we have recommended and seen customers use.

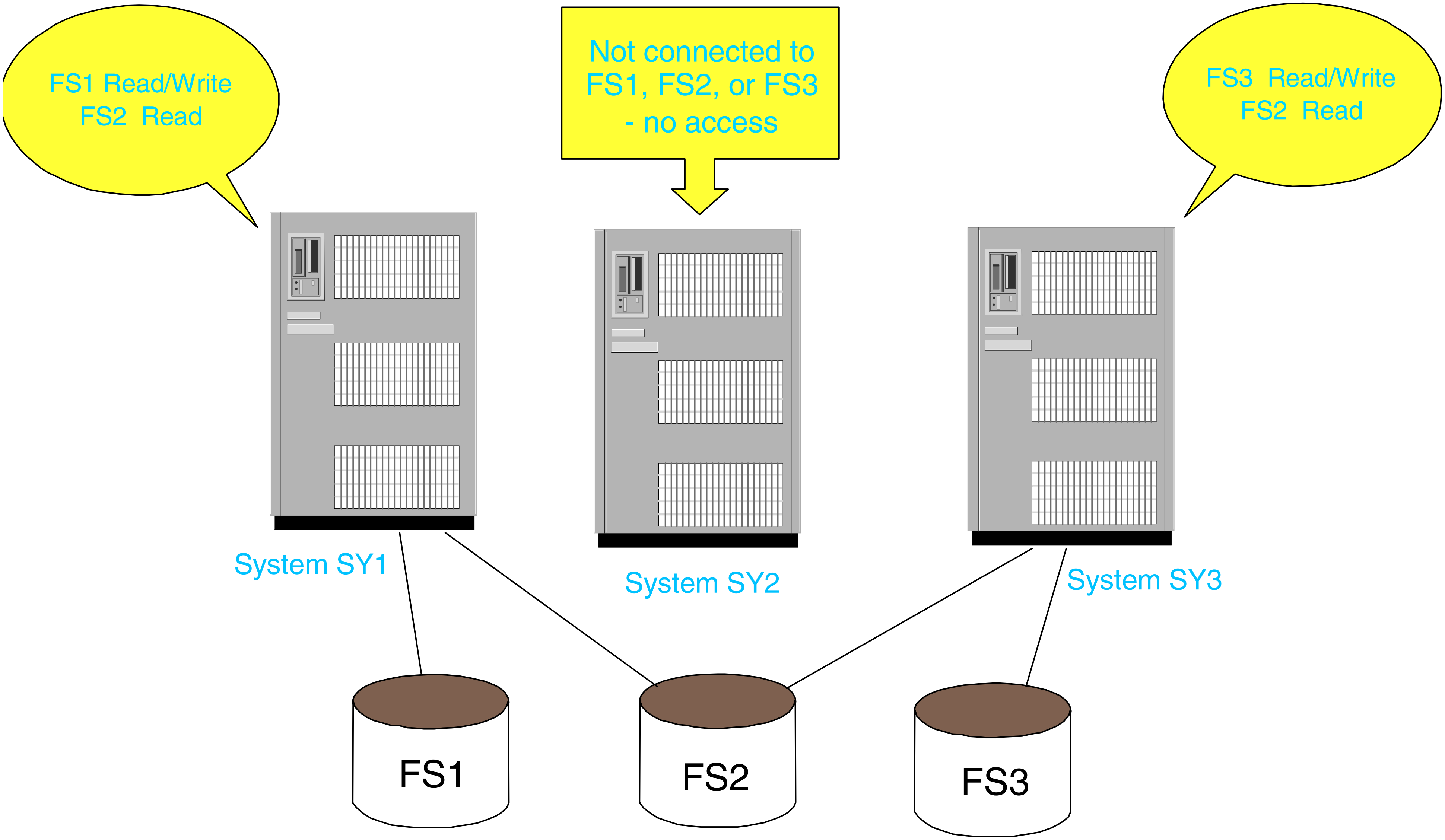
Customer Requirements

1. Ability for sharing file system data across sysplex in R/W mode
2. File system availability
3. Common File Hierarchy on all systems
4. Same view of the file system, regardless of which system in sysplex you logged on to.
5. Same behavior of the filesystem in both a sysplex and non-sysplex environments.

Customer Requirements ...

- Solution:
  - Messaging protocol used to transfer data around sysplex from central owner
    - The solution utilizes XCF services to transfer data to and from a file system owner, where the request can be processed.
  - Mount requests are sysplex wide
  - Recovery in place for maintaining file system availability
    - Dead System Recovery is provided to dynamically recover file systems owned by a system that has left the sysplex for any reason. Those file systems will be randomly moved to surviving systems in the sysplex, providing enhanced file system availability.

HFS Sharing before R9

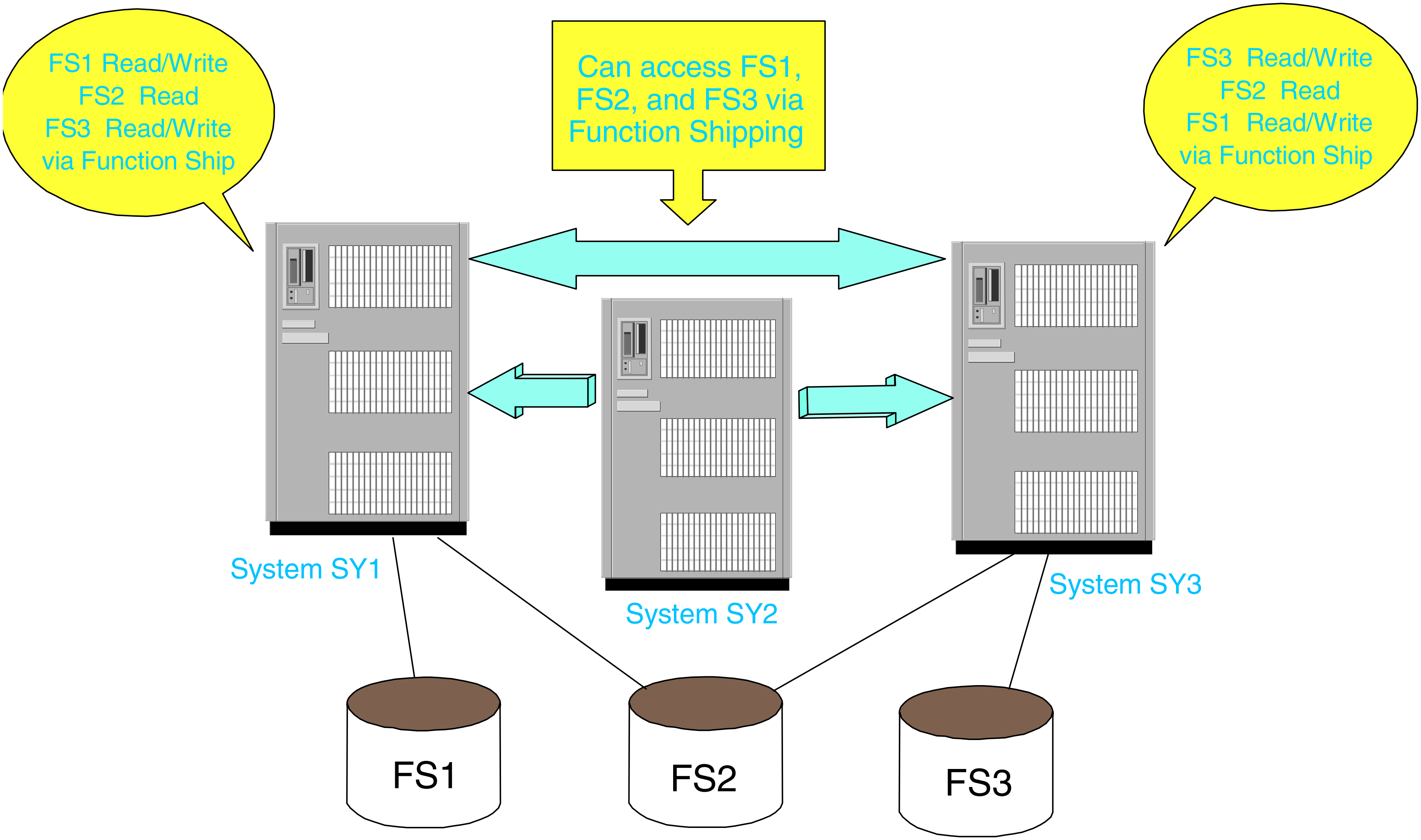


FS1 Read/Write
FS2 Read

Not connected to
FS1, FS2, or FS3
- no access

FS3 Read/Write
FS2 Read

HFS Sysplex Sharing in R9



FS1 Read/Write
FS2 Read
FS3 Read/Write
via Function Ship

Can access FS1, FS2, and FS3 via Function Shipping

FS3 Read/Write
FS2 Read
FS1 Read/Write
via Function Ship

Note: In this picture, FS2 is mounted READ for consistency with previous picture. It can just as easily be mounted READ/WRITE allowing all systems Read-Write access to FS2.

New BPXPRMxx Parmlib Options

- SYSPLEX(YES|NO)
  - Default is SYSPLEX(NO)

- VERSION(user_specified_string)
  - Qualifier to represent a maintenance level or release of an HFS.
  - Default is Version(/)

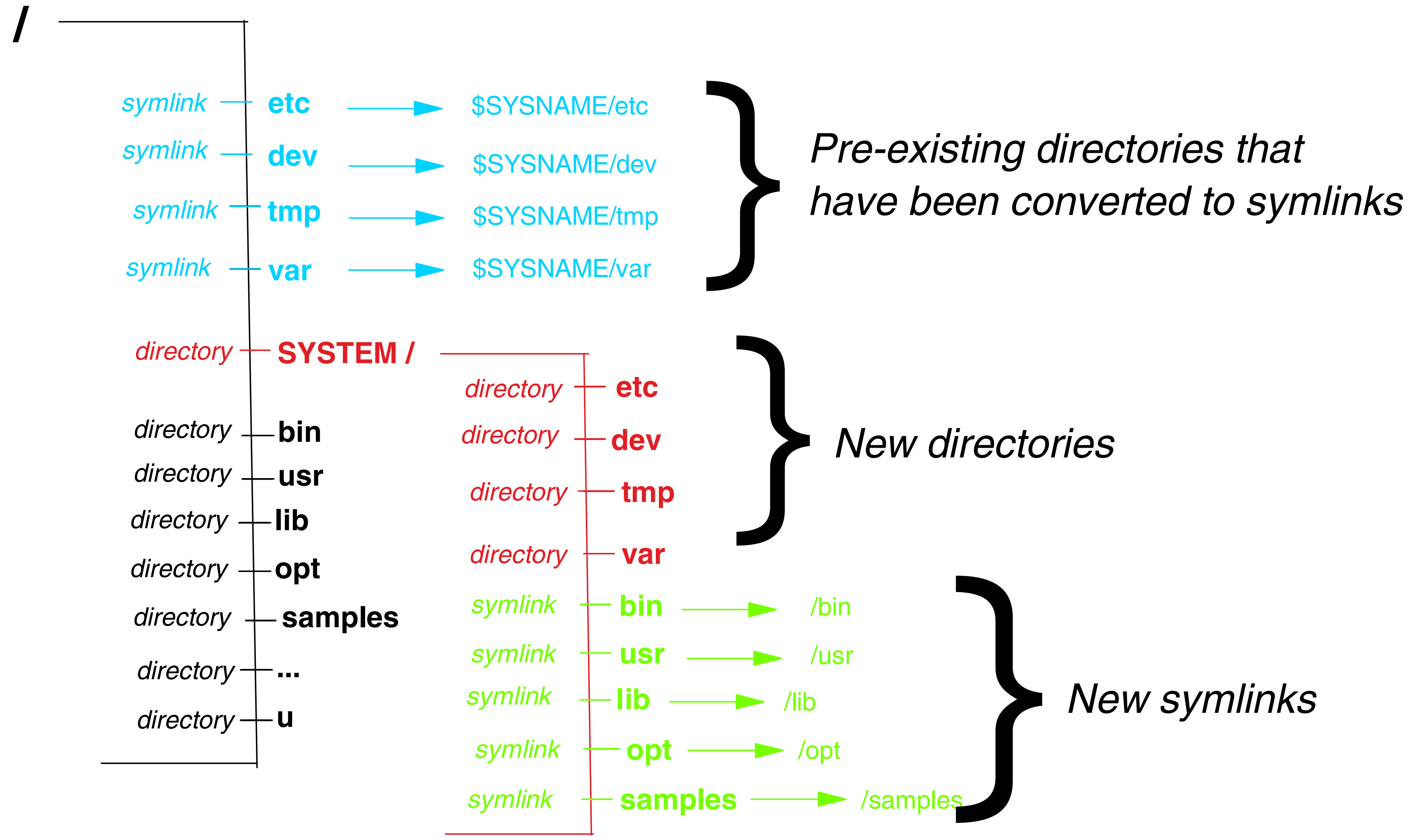
- New parms on ROOT and MOUNT statements
  - SYSNAME - declares the system that is the "owner" of this file system.
    - Default is the system where mount is being issued.
  - AUTOMOVE | NOAUTOMOVE - attempt to move the ownership (or not) if owner fails.
    - Default is AUTOMOVE.

What has stayed the same in R9?

- In R9, IBM will continue to provide two HFS data sets:
  - Root HFS - containing files and executables for OS/390 elements.
  - ETC HFS - HFS data set only containing only empty directories
- This is true for both ServerPac and CBPDO customers.

- We continue to recommend having separate HFS data sets for /tmp, /var, and /dev in order to separate customized data from shipped code.

What looks new? The root filesystem has been changed:



/

symlink — etc → $SYSNAME/etc

symlink — dev → $SYSNAME/dev

symlink — tmp → $SYSNAME/tmp

symlink — var → $SYSNAME/var

Pre-existing directories that have been converted to symlinks

directory — SYSTEM /

directory — etc

directory — dev

directory — tmp

directory — var

New directories

symlink — bin → /bin

symlink — usr → /usr

symlink — lib → /lib

symlink — opt → /opt

symlink — samples → /samples

New symlinks

directory — bin

directory — usr

directory — lib

directory — opt

directory — samples

directory — ...

directory — u

New directories are in Red

New symlinks are in Green

Some pre-existing directories have been converted to symlinks (in blue).

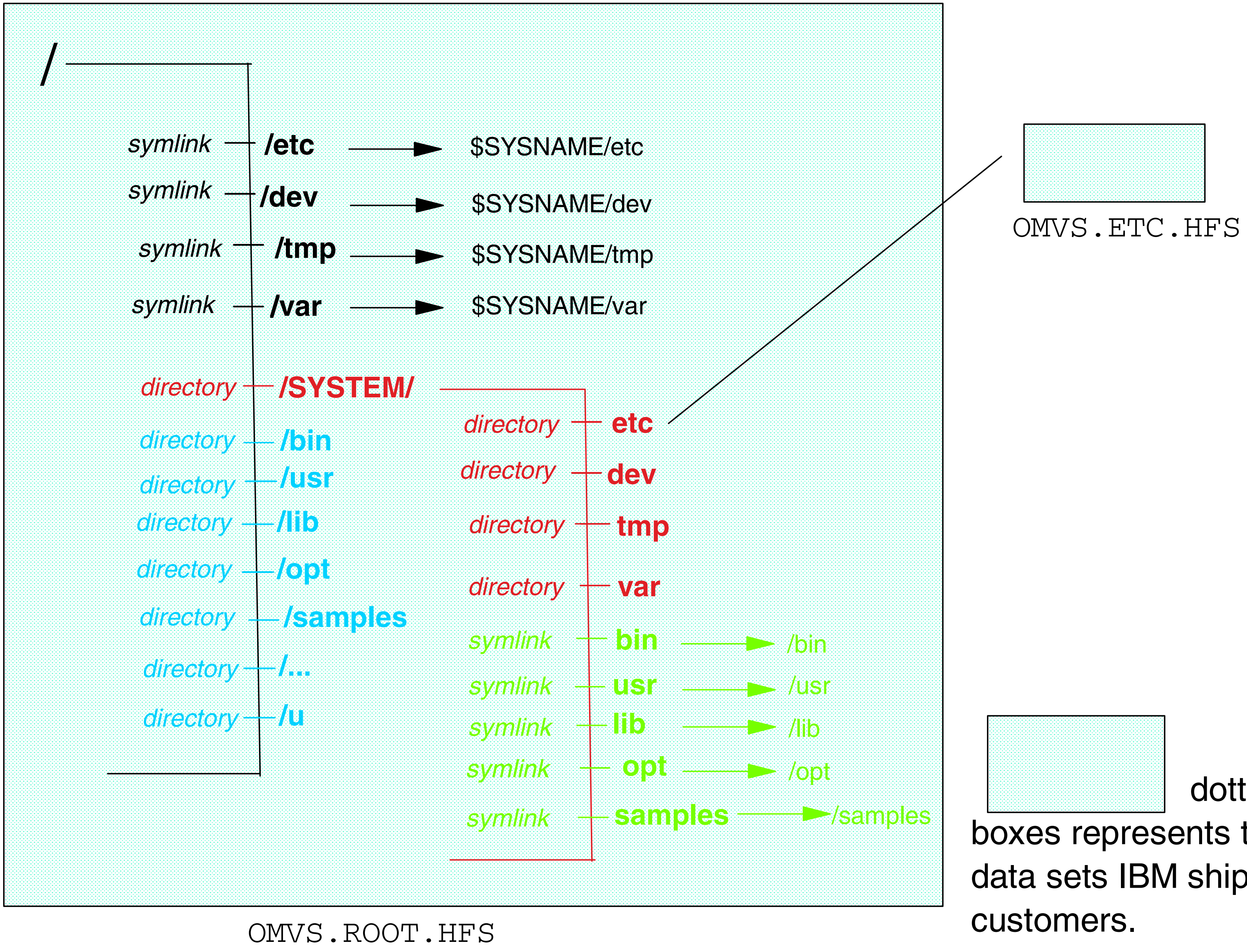
How are the two HFS data sets to be used in a non-sysplex single system environment?

BPXPRMxx of R9

```
ROOT
FILESYSTEM('OMVS.ROOT.HFS')
TYPE(HFS) MODE(RDWR)

MOUNT
FILESYSTEM('OMVS.ETC.HFS')
TYPE(HFS) MODE(RDWR)
MOUNTPOINT('/etc')

....
```

- For sites who maintain a RDWR (read-write) root filesystem, no additonal filesystems are necessary.
- Customers who have this two filesystem structure pre-R9, the BPXPRMxx member would look exactly the same - thus there would be no migration action for R9.

Single system image for R9



/

symlink /etc → $SYSNAME/etc

symlink /dev → $SYSNAME/dev

symlink /tmp → $SYSNAME/tmp

symlink /var → $SYSNAME/var

directory /SYSTEM/

- directory etc
- directory dev
- directory tmp
- directory var
- symlink bin → /bin
- symlink usr → /usr
- symlink lib → /lib
- symlink opt → /opt
- symlink samples → /samples

directory /bin

directory /usr

directory /lib

directory /opt

directory /samples

directory /...

directory /u

OMVS.ROOT.HFS

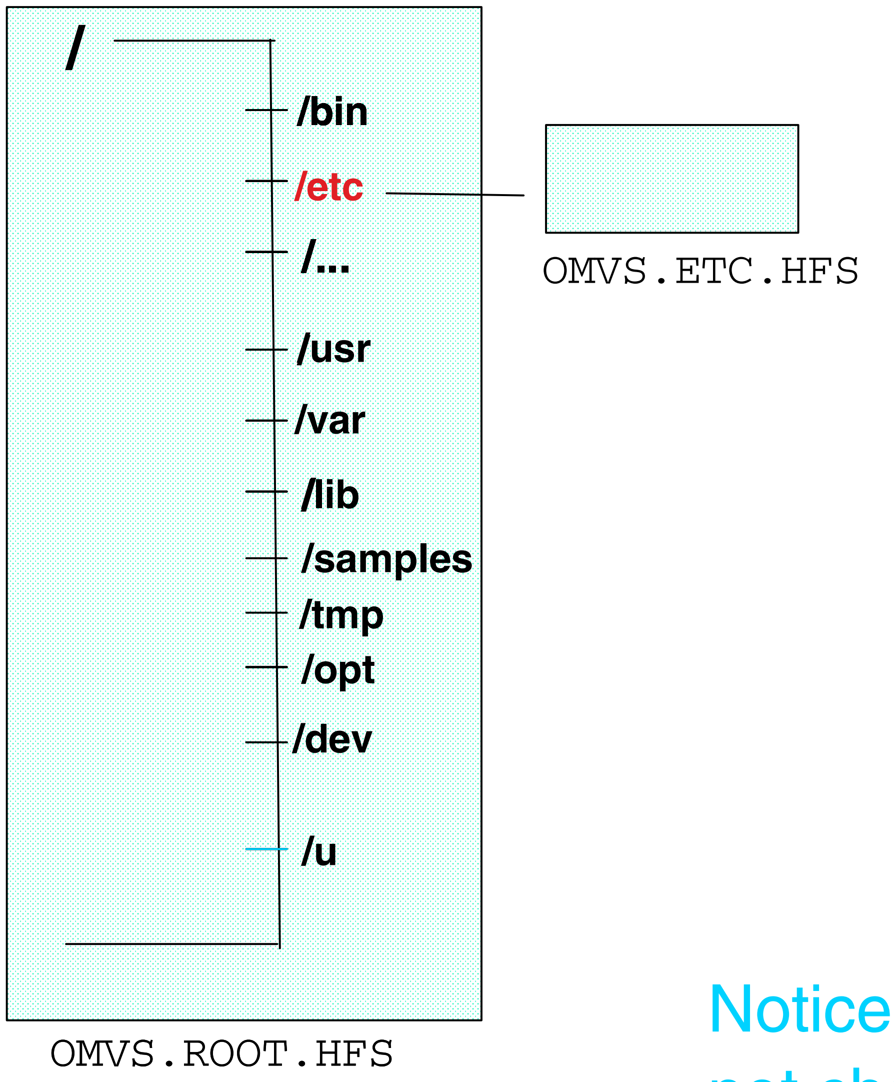
OMVS.ETC.HFS

▨ dotted boxes represents the HFS data sets IBM ships customers.

What actually happens under the covers?

- As OMVS is coming up, we mount the root filesystem just as before.
- We then try to mount the ETC HFS on the /etc mountpoint. However, /etc is a symbolic link and must be resolved first before the mounting can occur.
  - Since we are running with SYSPLEX(NO) - New BPXPRMxx keyword to be discussed later - the R9 filesystem code will use /SYSTEM as the substitute variable for $SYSNAME.
    - SYSPLEX(NO) is the default when not specified
  - Therefore $SYSNAME/etc gets resolved to /SYSTEM/etc and the ETC HFS is mounted on /SYSTEM/etc.
  - Any future reference to /etc during execution, will also resolve to /SYSTEM/etc.

In pre-R9, the Single system image with two HFS data set structure would look like:



BPXPRMxx

```
ROOT
FILESYSTEM('OMVS.ROOT.HFS')
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM('OMVS.ETC.HFS')
TYPE(HFS) MODE(RDWR)
MOUNTPOINT('/etc')

.
.
.
```

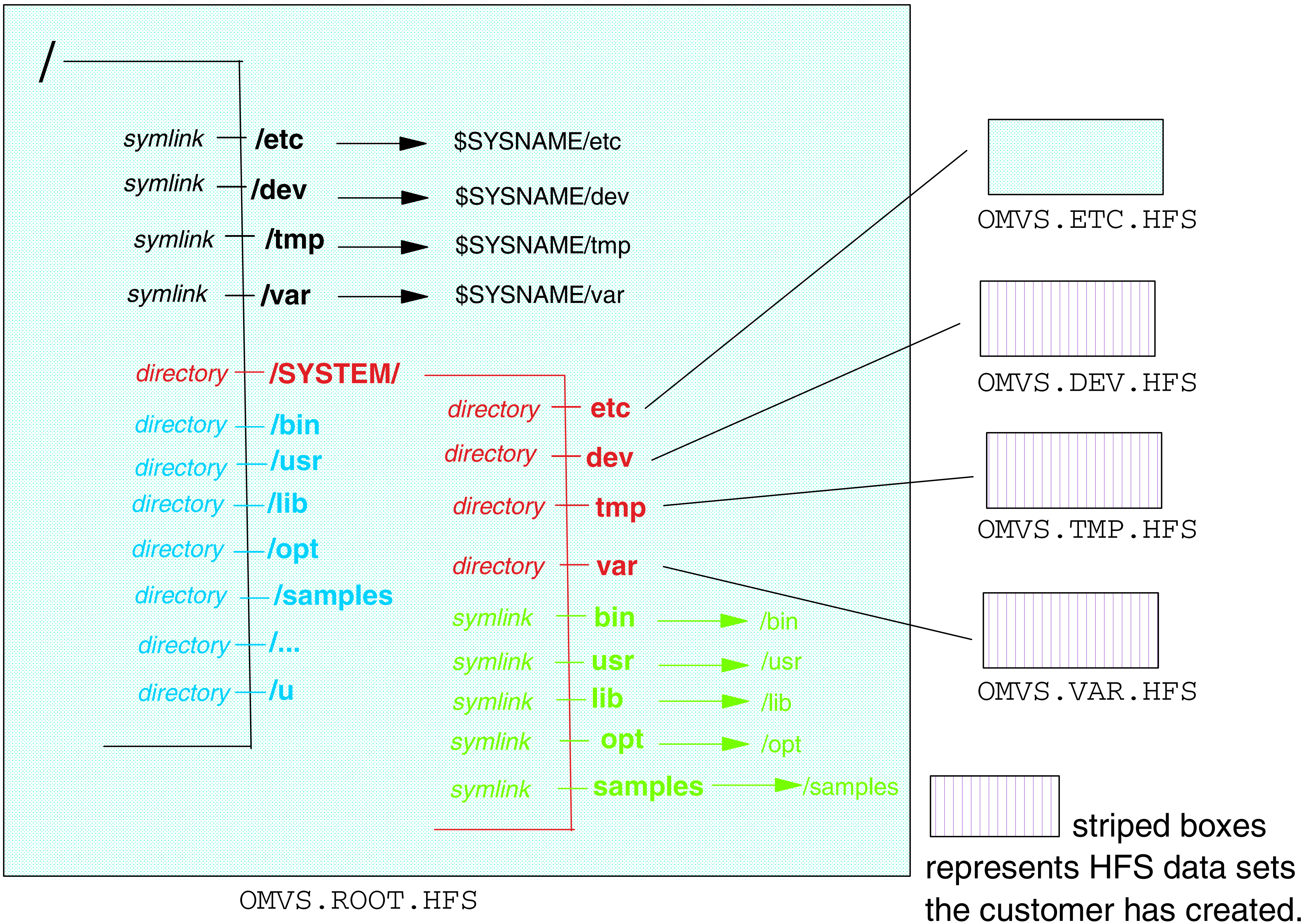
Notice the BPXPRMxx member has not changed from Pre-R9 to R9, eventhough we have a new filesystem structure.

Customer who have separate HFS data sets for /tmp, /dev and for /var in order to maintain a Read-only filesystem...

```
ROOT
FILESYSTEM('OMVS.ROOT.HFS')
TYPE(HFS)  MODE(READ)

MOUNT
FILESYSTEM('OMVS.ETC.HFS')
TYPE(HFS)  MODE(RDWR)
MOUNTPOINT('/etc')

MOUNT
FILESYSTEM('OMVS.TMP.HFS')
TYPE(HFS)  MODE(RDWR)
MOUNTPOINT('/tmp')

MOUNT
FILESYSTEM('OMVS.DEV.HFS')
TYPE(HFS)  MODE(RDWR)
MOUNTPOINT('/dev')

MOUNT
FILESYSTEM('OMVS.VAR.HFS')
TYPE(HFS)  MODE(RDWR)
MOUNTPOINT('/var')
```

BPXPRMxx of R9.

Single system image for R9 - In order to have Read Only root



/

symlink /etc → $SYSNAME/etc

symlink /dev → $SYSNAME/dev

symlink /tmp → $SYSNAME/tmp

symlink /var → $SYSNAME/var

directory /SYSTEM/

directory etc

directory dev

directory tmp

directory var

symlink bin → /bin

symlink usr → /usr

symlink lib → /lib

symlink opt → /opt

symlink samples → /samples

directory /bin

directory /usr

directory /lib

directory /opt

directory /samples

directory /...

directory /u

OMVS.ROOT.HFS

OMVS.ETC.HFS

OMVS.DEV.HFS

OMVS.TMP.HFS

OMVS.VAR.HFS

▥ striped boxes represents HFS data sets the customer has created.

Miscellaneous facts

- Notice that there are no changes to the BPXPRMxx PARMLIB member from pre-R9 to R9, when using our recommendation for customized data from shipped code.
- The mounting of the TMP, DEV, and VAR HFS data sets will occur the same way as the ETC HFS did earlier.
  - These will be mounted on /SYSTEM/tmp, /SYSTEM/dev and /SYSTEM/var respectively.
- In reality, ServerPac and CBPDO customer will run a job during the installation process to convert /etc directory to a symbolic link
  - BPXISETS

Steps to setup a sysplex environment

- Assumption:
  - You have already prepared your system for a sysplex environment, including:
    - Creation of other elements' CDS data sets
    - Creation of **`&SYSNAME`** as a system symbolic
      - Required for Shared HFS support.

Steps to setup a sysplex environment ...

1. Create the OMVS Couple Data Set
    - SYS1.SAMPLIB(BPXISCDS)
    - This data set is used as the sysplex wide mount table and contains information about all systems participating in file system sharing, and records for all mounted filesystems in the plex.

- There are two kinds of CDS data sets - Primary and Alternate

Steps to setup a sysplex environment ...

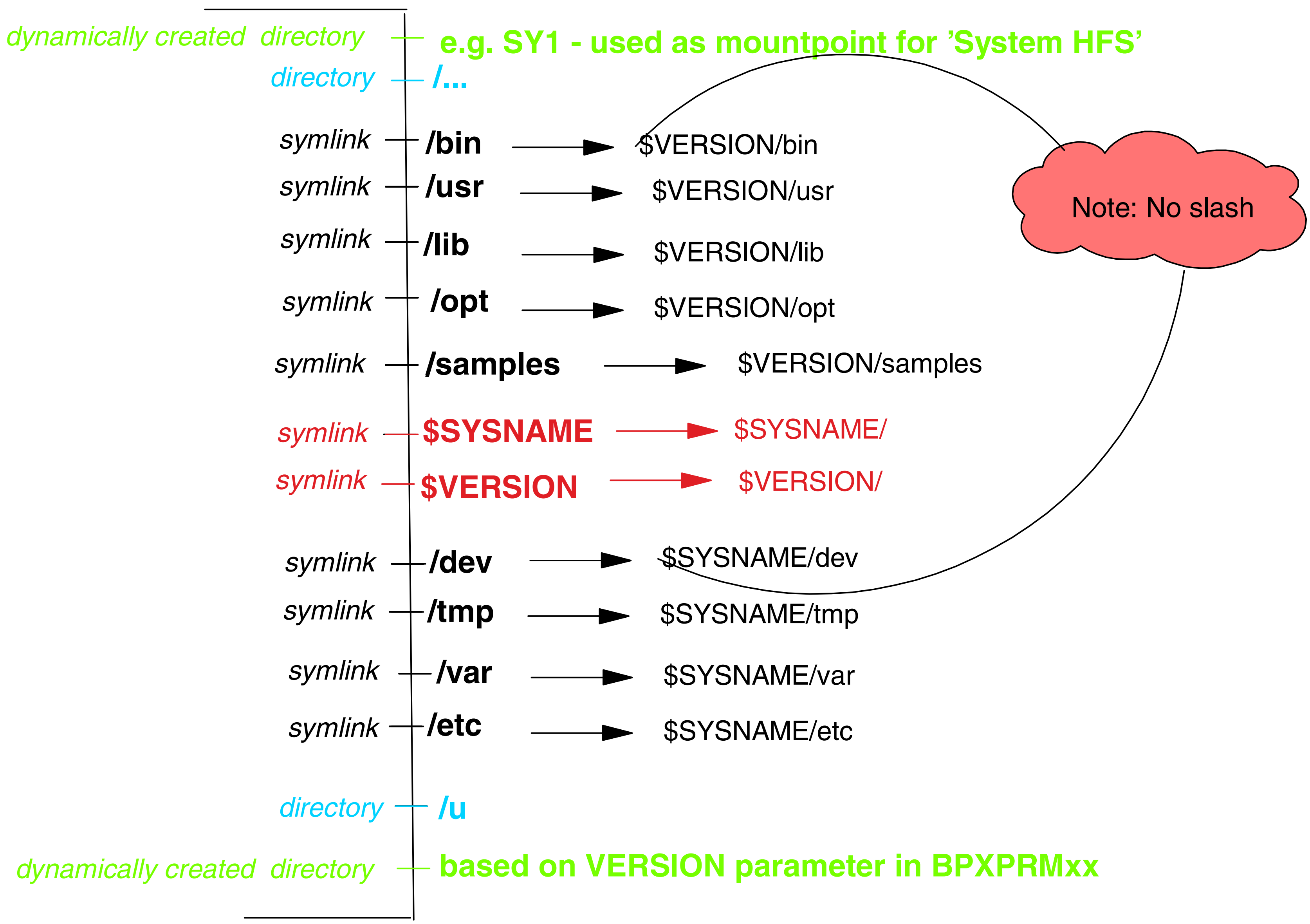
2. After the OMVS CDS is defined, it must be defined to XCF. Update COUPLExx PARMLIB member:

```
COUPLE  INTERVAL(30)                                /* 1 minute */
        OPNOTIFY(30)                                /* 1 minute */
        CLEANUP(60)                                 /* 1 minute */
        SYSPLEX(PLEX1)                              /* SYSPLEX NAME*/
        PCOUPLE(SYS1.PCOUPLE,CPLPKP)                /* COUPLE DS   */
        ACOUPLE(SYS1.ACOUPLE,CPLPKA)                /* ALTERNATE DS*/
        MAXMSG(750)
        RETRY(10)
DATA    TYPE(CFRM)
           PCOUPLE(SYS1.PFUNCT.CTTEST,FDSPKP)
           ACOUPLE(SYS1.AFUNCT.CTTEST,FDSPKA)
DATA    TYPE(BPXMCDS)
           PCOUPLE(POSIX.OMVS.CDS01,OMVSC1)
           ACOUPLE(POSIX.OMVS.CDS02,OMVSC2)
```

Steps to setup a sysplex environment ...

3. Create the Sysplex Root
   - SYS1.SAMPLIB(BPXISYSR)
   - There will only be one Sysplex Root HFS data set for your entire sysplex.
   - Example: **`OMVS.SYSPLEX.ROOT`**
   - This HFS data set will be mounted as the Root filesystem in your SYSPLEX and must be READ-WRITE.
   - No files or code will reside in this HFS data set, only directories and symlinks. Therefore the size of this HFS data set will be very small.
   - Will be used for redirection, throughout entire sysplex.

Sysplex Root



dynamically created directory — e.g. SY1 - used as mountpoint for 'System HFS'

directory — /...

symlink — /bin → $VERSION/bin

symlink — /usr → $VERSION/usr

symlink — /lib → $VERSION/lib

symlink — /opt → $VERSION/opt

symlink — /samples → $VERSION/samples

symlink — $SYSNAME → $SYSNAME/

symlink — $VERSION → $VERSION/

symlink — /dev → $SYSNAME/dev

symlink — /tmp → $SYSNAME/tmp

symlink — /var → $SYSNAME/var

symlink — /etc → $SYSNAME/etc

directory — /u

dynamically created directory — based on VERSION parameter in BPXPRMxx

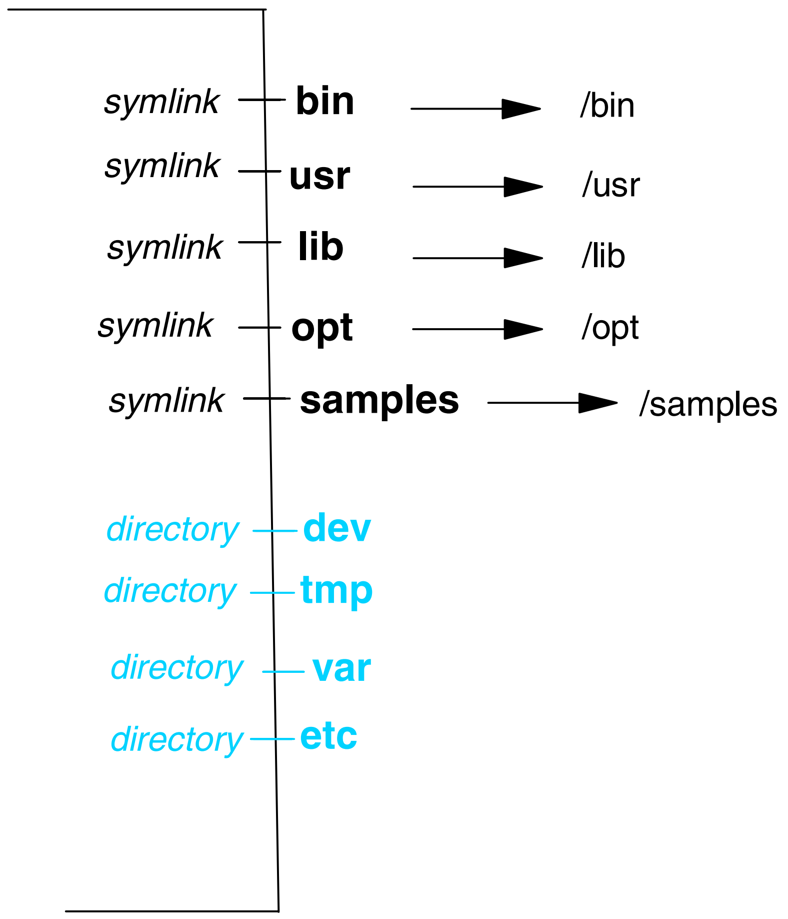
Note: No slash

The directories in green will not be created by job, but rather at system startup.

Steps to setup a sysplex environment ...

4. Create the System-specific HFS
   - SYS1.SAMPLIB(BPXISYSS)
   - You will need one System-specific HFS for every system in your sysplex.
   - Highly recommend using a standardized naming convention for this data set, especially one that uses **&SYSNAME**.
     - Example: **OMVS.SY1.SYSTEM.HFS**
   - The mount point for this data set will be dynamically created during OMVS startup.
   - Will be used by system to mount system-specific data

System-specific HFS



symlink — **bin** → /bin

symlink — **usr** → /usr

symlink — **lib** → /lib

symlink — **opt** → /opt

symlink — **samples** → /samples

directory — **dev**

directory — **tmp**

directory — **var**

directory — **etc**

Steps to setup a sysplex environment …

5. Other recommendations:
   - Employ a standard naming convention for HFS data sets, especially for ETC, TMP, VAR and DEV HFS data sets.
     - Use `&SYSNAME` as one of the qualifiers of the data set names.
   - Do not use `&SYSNAME` as one of the qualifiers for Root HFS.
     - BTW, the Root HFS will be called VERSION HFS in the context of sysplex, to avoid confusion with the Sysplex Root HFS.
   - This will allow using one BPXPRMxx PARMLIB member for all systems in your sysplex.

Quick recap

► Sysplex Root HFS

- only one sysplex root HFS for sysplex
- used by system to redirect addressing to other directories
- small data set, mounted read/write
- created by BPXISYSR job

► System-specific HFS

- used by system to mount system-specific data
- includes a system's /dev, /tmp, /var, /etc directories
- created by BPXISYSS job

► Version HFS

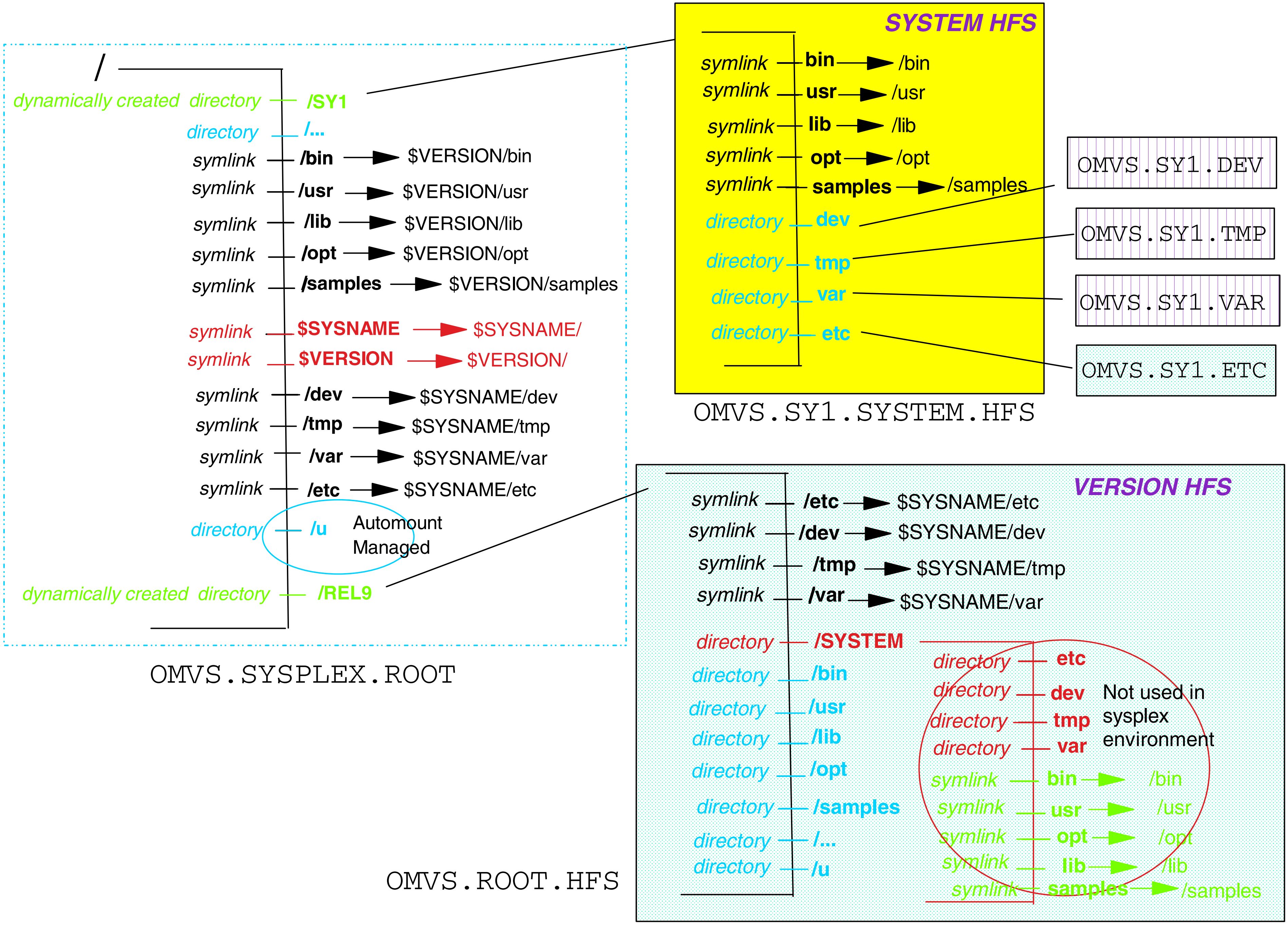
- serves same purpose as today's non-sysplex "root HFS"
- contains system code and binaries, including /bin, /usr, /lib, /opt, and /samples directories
- IBM delivers one Version HFS, you may define more as you add maintenance levels and future system release levels

Steps to setup a sysplex environment ...

6. Update BPXPRMxx:

```
VERSION(’REL9’)
SYSPLEX(YES)

ROOT
FILESYSTEM(’OMVS.SYSPLEX.ROOT’)
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..SYSTEM.HFS’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME.’)

MOUNT
FILESYSTEM(’OMVS.ROOT.HFS’)
TYPE(HFS) MODE(READ)
MOUNTPOINT(’/$VERSION’)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..ETC’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME./etc’)

....
```

First system in Sysplex for R9 - Recommended set-up


/
dynamically created directory — /SY1
directory — /...
symlink — /bin → $VERSION/bin
symlink — /usr → $VERSION/usr
symlink — /lib → $VERSION/lib
symlink — /opt → $VERSION/opt
symlink — /samples → $VERSION/samples
symlink — $SYSNAME → $SYSNAME/
symlink — $VERSION → $VERSION/
symlink — /dev → $SYSNAME/dev
symlink — /tmp → $SYSNAME/tmp
symlink — /var → $SYSNAME/var
symlink — /etc → $SYSNAME/etc
directory — /u
Automount Managed
dynamically created directory — /REL9
OMVS.SYSPLEX.ROOT
SYSTEM HFS
symlink — bin → /bin
symlink — usr → /usr
symlink — lib → /lib
symlink — opt → /opt
symlink — samples → /samples
directory — dev
directory — tmp
directory — var
directory — etc
OMVS.SY1.SYSTEM.HFS
OMVS.SY1.DEV
OMVS.SY1.TMP
OMVS.SY1.VAR
OMVS.SY1.ETC
VERSION HFS
symlink — /etc → $SYSNAME/etc
symlink — /dev → $SYSNAME/dev
symlink — /tmp → $SYSNAME/tmp
symlink — /var → $SYSNAME/var
directory — /SYSTEM
directory — etc
directory — dev
directory — tmp
directory — var
symlink — bin → /bin
symlink — usr → /usr
symlink — opt → /opt
symlink — lib → /lib
symlink — samples → /samples
Not used in sysplex environment
directory — /bin
directory — /usr
directory — /lib
directory — /opt
directory — /samples
directory — /...
directory — /u
OMVS.ROOT.HFS

What actually happens under the covers?

- As OMVS is coming up, we mount the root filesystem just as before.
- In R9 with SYSPLEX(YES), we then create the system name directory under the root (in this case /SY1)
- In R9 with SYSPLEX(YES) we also create a directory of the name specified in the VERSION('.......') keyword. (in this case /REL9).
- That's why the Sysplex Root needs to be mounted Read-Write.

What actually happens under the covers?...

- During PARMLIB processing, **&SYSNAME.** will be substituted by the system name (in this case SY1).

- We then try to mount **OMVS.SY1.SYSTEM.HFS** on **/&SYSNAME.** mountpoint, which is now /SY1.
  - Therefore **OMVS.SY1.SYSTEM.HFS** is mounted on /SY1.

What actually happens under the covers?...

- Next, we try to mount **OMVS.ROOT.HFS** on /$VERSION mountpoint. However, /$VERSION is a symbolic link and must be resolved first before the mounting can occur.
  - Since we are running with SYSPLEX(YES) the R9 filesystem code will dynamically create a mountpoint using the string specified for VERSION('......') in the BPXPRMxx, after the root file system is mounted.
  - In addition, the filesystem code will use this string as the substitute variable for $VERSION.
  - Therefore /$VERSION gets resolved to /REL9/ and **OMVS.ROOT.HFS** is mounted on /REL9/.

What actually happens under the covers?...

- Last, we try to mount **`OMVS.SY1.ETC`** on **`/&SYSNAME./etc`** mountpoint. However, during PARMLIB processing, substitution of symbols which is now /SY1/etc .
  - Therefore **`OMVS.SY1.ETC`** is mounted on /SY1/etc.

- This same will hold true if you have mount statements for /tmp, /var, and /dev if you have separate HFS data sets for those mountpoints.

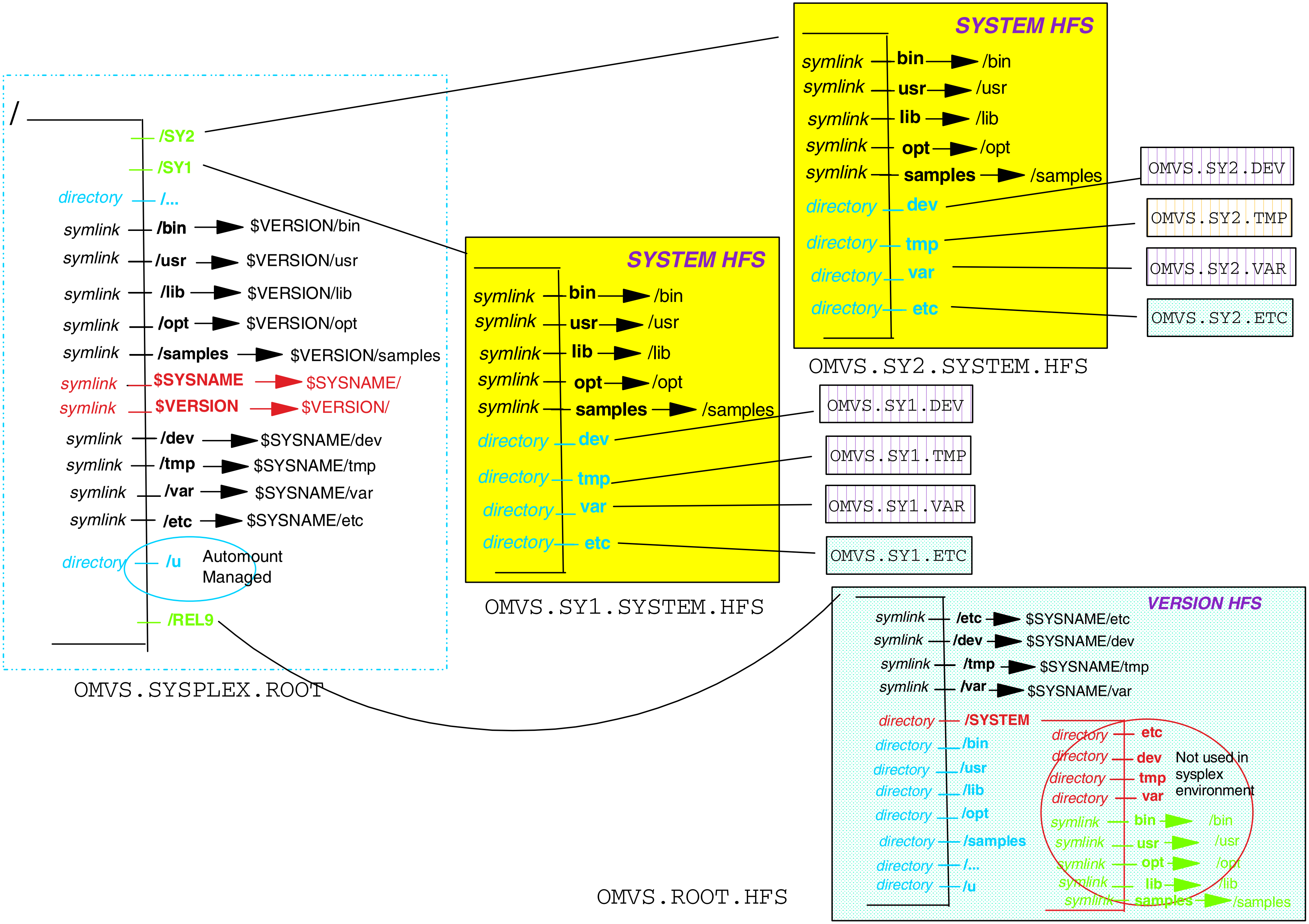
Other items

- Notice the **NOAUTOMOVE** option on the MOUNT statements.
  - These tell the mount table, not to move this HFS data set to another system when this system goes down.
  - You don't want to move information that is only valid for one system to another.
  - Default is AUTOMOVE, if not specified.
- Ensure that /etc is a symlink in the Version HFS, before IPL'ing introducing your first system into the sysplex.
- You might have forgotten to run the BPXISETS job.

Rolling over other systems into this sysplex

- Must create a System-specific HFS data set for every system in sysplex.
  - Run BPXISYSS job multiple times with different data set names.
- Must create a copy of the ETC HFS data set for every system in your sysplex. This includes all customization data that resides in here, will be duplicated on all your systems.
- Have system individual specific HFS data sets for /tmp, /var, and /dev.
- Again using **&SYSNAME** as one of the qualifiers for these HFS data set names will help.

Multiple systems in Sysplex - all using same Version HFS



Multiple systems in Sysplex - all using same Version HFS

One BPXPRMxx for entire sysplex

```
VERSION('REL9')
SYSPLEX(YES)

ROOT
FILESYSTEM('OMVS.SYSPLEX.ROOT')
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM('OMVS.&SYSNAME..SYSTEM.HFS')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME.')

MOUNT
FILESYSTEM('OMVS.ROOT.HFS')
TYPE(HFS) MODE(READ)
MOUNTPOINT('/$VERSION')

MOUNT
FILESYSTEM('OMVS.&SYSNAME..DEV')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME./dev')

MOUNT
FILESYSTEM('OMVS.&SYSNAME..ETC')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME./etc')
....
```

The use of system symbolics, allows you to use one BPXPRMxx PARMLIB member for multiple systems.

Alternatively, you can use one BPXPRMxx for each system in sysplex

BPXPRMS1 (for SY1)

```
VERSION(’REL9’)
SYSPLEX(YES)

ROOT
FILESYSTEM(’OMVS.SYSPLEX.ROOT’)
TYPE(HFS) MODE(RDWR)

MOUNT
FILESYSTEM(’OMVS.SY1.SYSTEM.HFS’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY1’)

MOUNT FILESYSTEM(’OMVS.ROOT.HFS’)
TYPE(HFS) MODE(READ)
MOUNTPOINT(’/$VERSION’)

MOUNT FILESYSTEM(’OMVS.SY1.DEV’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY1/dev’)

MOUNT FILESYSTEM(’OMVS.SY1.ETC’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY1/etc’)

....
```

BPXPRMS2 for (SY2)

```
VERSION(’REL9’)
SYSPLEX(YES)

ROOT
FILESYSTEM(’OMVS.SYSPLEX.ROOT’)
TYPE(HFS) MODE(RDWR)

MOUNT
FILESYSTEM(’OMVS.SY2.SYSTEM.HFS’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY2’)

MOUNT FILESYSTEM(’OMVS.ROOT.HFS’)
TYPE(HFS) MODE(READ)
MOUNTPOINT(’/$VERSION’)

MOUNT FILESYSTEM(’OMVS.SY2.DEV’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY2/dev’)

MOUNT FILESYSTEM(’OMVS.SY2.ETC’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/SY2/etc’)

....
```

Other items ...

- The new VERSION('...') keyword should not be confused with VxxRyyMzz - Version, Release, Modification levels of the product.
  - It is supposed to represent that instance or maintenance level of the root filesystem.

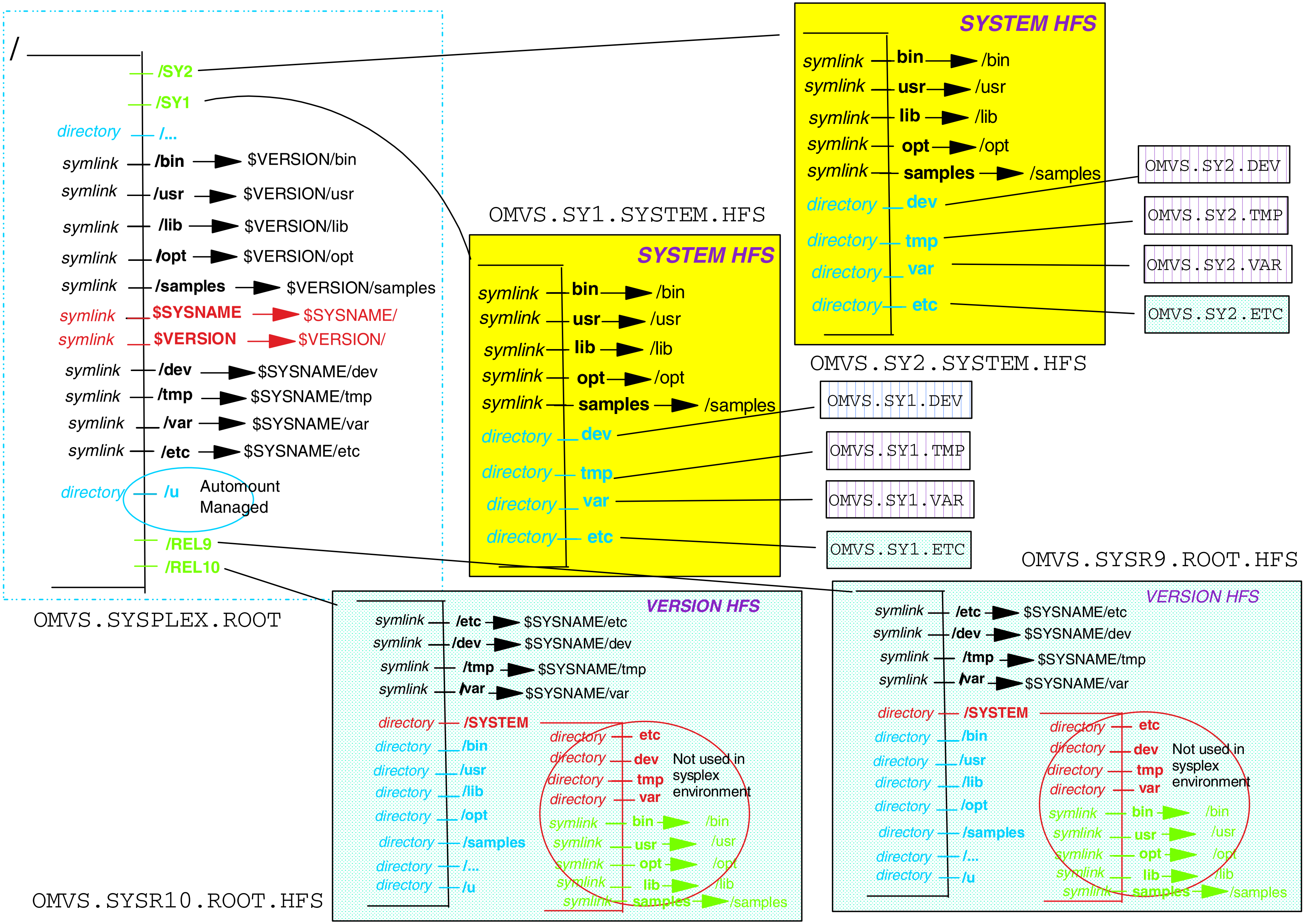
- The Shared HFS support does not require any additional support from DFSMS. All necessary function is already part of DFSMS 1.5. Additonal PTFs will be necessary to prevent HFS corruption.

Multiple versions ...

What about having multiple releases in your sysplex

- This is where you would use a different SYSRES pack to IPL a system in the system. This new SYSRES pack would have a corresponding root HFS data set that needs to be mounted.
  - Therfore using **`&SYSR1.`** as one of the qualifiers of the version HFS would be useful to help tie the HFS data set name with the appropriate SYSRES.
  - You can also use **`&SYSR1.`** for the VERSION keyword so that you can have one BPXPRMxx member for entire sysplex.

Multiple systems in Sysplex at different versions



/
/SY2
/SY1
directory — /...
symlink — /bin → $VERSION/bin
symlink — /usr → $VERSION/usr
symlink — /lib → $VERSION/lib
symlink — /opt → $VERSION/opt
symlink — /samples → $VERSION/samples
symlink — $SYSNAME → $SYSNAME/
symlink — $VERSION → $VERSION/
symlink — /dev → $SYSNAME/dev
symlink — /tmp → $SYSNAME/tmp
symlink — /var → $SYSNAME/var
symlink — /etc → $SYSNAME/etc
directory — /u Automount Managed
/REL9
/REL10

OMVS.SYSPLEX.ROOT

OMVS.SY1.SYSTEM.HFS

SYSTEM HFS
symlink — bin → /bin
symlink — usr → /usr
symlink — lib → /lib
symlink — opt → /opt
symlink — samples → /samples
directory — dev
directory — tmp
directory — var
directory — etc

OMVS.SY1.DEV
OMVS.SY1.TMP
OMVS.SY1.VAR
OMVS.SY1.ETC

SYSTEM HFS
symlink — bin → /bin
symlink — usr → /usr
symlink — lib → /lib
symlink — opt → /opt
symlink — samples → /samples
directory — dev
directory — tmp
directory — var
directory — etc

OMVS.SY2.SYSTEM.HFS

OMVS.SY2.DEV
OMVS.SY2.TMP
OMVS.SY2.VAR
OMVS.SY2.ETC

OMVS.SYSR10.ROOT.HFS

VERSION HFS
symlink — /etc → $SYSNAME/etc
symlink — /dev → $SYSNAME/dev
symlink — /tmp → $SYSNAME/tmp
symlink — /var → $SYSNAME/var
directory — /SYSTEM
directory — /bin
directory — /usr
directory — /lib
directory — /opt
directory — /samples
directory — /...
directory — /u
directory — etc
directory — dev
directory — tmp
directory — var
symlink — bin → /bin
symlink — usr → /usr
symlink — opt → /opt
symlink — lib → /lib
symlink — samples → /samples
Not used in sysplex environment

OMVS.SYSR9.ROOT.HFS

VERSION HFS
symlink — /etc → $SYSNAME/etc
symlink — /dev → $SYSNAME/dev
symlink — /tmp → $SYSNAME/tmp
symlink — /var → $SYSNAME/var
directory — /SYSTEM
directory — /bin
directory — /usr
directory — /lib
directory — /opt
directory — /samples
directory — /...
directory — /u
directory — etc
directory — dev
directory — tmp
directory — var
symlink — bin → /bin
symlink — usr → /usr
symlink — opt → /opt
symlink — lib → /lib
symlink — samples → /samples
Not used in sysplex environment

Multiple systems in Sysplex at different versions

In this example SY1 is at Release 10 and SY2 is at Release 9

BPXPRMxx (for SY1)

```
VERSION(’REL10’)
SYSPLEX(YES)

ROOT
FILESYSTEM(’OMVS.SYSPLEX.ROOT’)
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..SYSTEM.HFS’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME.’)

MOUNT
FILESYSTEM(’OMVS.&SYSR1..ROOT.HFS’)
TYPE(HFS) MODE(READ)
MOUNTPOINT(’/$VERSION’)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..DEV’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME./dev’)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..ETC’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME./etc’)
```

BPXPRMxx (for SY2)

```
VERSION(’REL9’)
SYSPLEX(YES)

ROOT
FILESYSTEM(’OMVS.SYSPLEX.ROOT’)
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..SYSTEM.HFS’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME.’)

MOUNT
FILESYSTEM(’OMVS.&SYSR1..ROOT.HFS’)
TYPE(HFS) MODE(READ)
MOUNTPOINT(’/$VERSION’)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..DEV’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME./dev’)

MOUNT
FILESYSTEM(’OMVS.&SYSNAME..ETC’)
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT(’/&SYSNAME./etc’)
```

Multiple systems in Sysplex at different versions

A method to use only one BPXPRMxx member

```
VERSION('&SYSR1.')
SYSPLEX(YES)

ROOT
FILESYSTEM('OMVS.SYSPLEX.ROOT')
TYPE (HFS) MODE(RDWR)

MOUNT
FILESYSTEM('OMVS.&SYSNAME..SYSTEM.HFS')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME.')

MOUNT
FILESYSTEM('OMVS.&SYSR1..ROOT.HFS')
TYPE(HFS) MODE(READ)
MOUNTPOINT('/$VERSION')

MOUNT
FILESYSTEM('OMVS.&SYSNAME..DEV')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME./dev')

MOUNT
FILESYSTEM('OMVS.&SYSNAME..ETC')
TYPE(HFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME./etc')
....
```

What's going to behave differently

- Probably the biggest change in behavior will be fallout from us changing directories to symbolic links.
  - This change does not violate any XPG standard and therefore doesn't jeopardize our UNIX Branding.
  - However, things like the "ls" command will now require you to specify:
    - "ls /etc/" - notice the trailing slash. Prior to R9, you could say "ls /etc" and you would get a listing of the /etc directory.
    - To minimize this one difference, one can place `alias ls=ls -L` in /etc/profile.

    (I don't really recommend this!!!)

What HFS Sharing is not ...

- You will not be able to have one HFS for /tmp, /dev, /var, or /etc to be shared by entire sysplex!!
  - This requires individual product changes provided by elements.
- HFS Sharing is not like shared-DASD!
  - In order to share an HFS Read-Write all systems must be in a sysplex.
  - In addition, all systems doing the Read-Write sharing must be at R9 or higher.
  - In order to share an HFS with pre-R9 systems, the HFS needs to be mounted Read-only on all systems.
  - You will not be able to share an HFS between native MVS machines and second level VM guest machines.

Displaying Enhanced Mount Information

- Changes were made to display commands to shared HFS support
  - Operator command D OMVS,F
  - Operator command D OMVS,O
    - Has been updated to show new parmlib SYSPLEX and VERSION keywords
  - Shell df command

D OMVS,F issued from SY1

```
D OMVS,F
BPXO045I 12.05.41 DISPLAY OMVS 085
OMVS     000E ACTIVE           OMVS=(MM)
TYPENAME   DEVICE
----------STATUS----------- MODE
HFS             7 UNOWNED
RDWR
  NAME=POSIX.SY2.HFS
  PATH=/SY2
  OWNER=          AUTOMOVE=N CLIENT=Y
HFS             6 ACTIVE
READ
  NAME=POSIX.USR.LPP
  PATH=/usr/lpp
  OWNER=SY1       AUTOMOVE=Y CLIENT=N
HFS             5 ACTIVE
READ
  NAME=POSIX.HFS.NLS
  PATH=/usr/lib/nls
  OWNER=SY1       AUTOMOVE=Y CLIENT=N
HFS             2 ACTIVE
RDWR
  NAME=POSIX.SY1.HFS
  PATH=/SY1
  OWNER=SY1       AUTOMOVE=N CLIENT=N
HFS             2 ACTIVE
RDWR
  NAME=POSIX.SY3.HFS
  PATH=/SY3
  OWNER=SY3       AUTOMOVE=N CLIENT=Y
HFS             1 ACTIVE
RDWR
  NAME=POSIX.SYSPLEX9.HFS1
  PATH=/
  OWNER=SY3       AUTOMOVE=Y CLIENT=N
```

df -v issued from SY1

```
df -v
w_getmntent could not obtain mount point for "POSIX.SY2.HFS".

/VR9/usr/lpp   (POSIX.USR.LPP)         678584/1437120 4294960667 Available
HFS, Read Only
File System Owner : SY1

/VR9/usr/lib/nls (POSIX.HFS.NLS)         26720/141120   4294966438 Available
HFS, Read Only
File System Owner : SY1

/SY1          (POSIX.SY1.HFS)          9096/11808     4294967043 Available
HFS, Read/Write
File System Owner : SY1

/SY3          (POSIX.SY3.HFS)          9096/11808     4294967043 Available
HFS, Read/Write
File System Owner : SY3

/            (POSIX.SYSPLEX9.HFS1)     1104/1536      4294967274 Available
HFS, Read/Write
File System Owner : SY3
```

Mount function enhancements

- Added two new options to mount functions
  - SYSNAME option
    - Directs a mount to be owned by a particular system
      - For example you may want to mount a file system on the system where most of the users of that file system will be running, to lessen message traffic between systems.
    - Default is to mount on system where request is made
  - AUTOMOVE option
    - Indicates if a file system can be automatically recovered in the event that owner leaves sysplex
    - Default is YES to allow for automatic recovery

Mount Function Enhancements

- Existing facilities
  - TSO MOUNT command
  - REXX mount function
  - ISPF ISHELL panel options
- New facilities
  - Shell mount command
  - Shell chmount command
    - Changes sysname and automove values
  - Operator command SETOMVS FILESYS
    - Changes sysname and automove values
  - __mount( ) C function
  - BPX2MNT callable service
- Beware of command dependent forms
  - AUTOMOVE(NO) vs. NOAUTOMOVE

File System Availability

- Dead System Recovery
  - Moves file systems to another system if possible
    - When a system leaves the sysplex, the other systems will attempt to recover the file systems owned by that system, by becoming the new owner. If the file systems were not mounted as automove(no), then they are eligible for recovery. If another system can become the owner then the file system will remain available for use.
  - Unowned File Systems
    - If no other system in the sysplex can mount a file system in recovery, or if the file system was mounted as automove(no) - in which case it isn't eligible for recovery, then the mount point for that file system becomes unowned - or what we fondly call a "black hole". An unowned file system can be removed with unmount. The mountpoint of an unowned file system remains in use until the file system is recovered or unmounted.

FAQ's

Q1: Today, I create other "first-level" directories under the root. How do I go about doing this in a sysplex environment?

- A1:Any customer created directory in the old root filesystem, will now need to be created in the Sysplex Root as a directory.

Q2: How does this support affect automount?

A2: You must ensure the same automount policy is in affect for every system in the sysplex.

FAQ's ...

Q3: What about system specific data, where should that be mounted? (Example: Only one system in the sysplex is running a database application.)

A3:One will need to create a directory (for example xxxxxxxx) in the System-specific HFS. This directory will be used as a mountpoint for the data base. Next you will need to create a symbolic link in the Sysplex Root of the same name, but will point to $SYSNAME/xxxxxxxx.

- In Sysplex Root you must have

symlink -- xxxxxxxx --> $SYSNAME/xxxxxxxx

FAQ's ...

Q4: Do the HFS data sets that will be shared in the sysplex need to be SMS managed, or non-SMS managed?

(Recent PTFs were issued to allow non-SMS managed HFS data sets.)

A4:The non-SMS managed support is independent of the Shared HFS support. The HFS data sets can either be SMS or non-SMS managed, as long as they are cataloged.

FAQ's ...

Q5: Is the coupling facility needed for Shared HFS support?

A5:No, the coupling facilty is not used but the OMVS coupling data sets are needed.

Q6: Is there an XCF group HFS Sharing?

A6:Yes, the name is SYSBPX.

FAQ's ...

Q7: What about the FILESYSTYPE statements across the plex?

A7:For each system in the sysplex to be able to mount the file systems owned by other systems, each system's BPXPRMxx member will need to have the same FILESYSTYPE statements. If a particular FILESYSTYPE is not available on a system, it will not be able to mount file systems of that type which are mounted in the sysplex. This will create a "black hole" in the file system hierarchy on that system.

FAQ's ...

Q8: When should I use NOAUTOMOVE (in its various command dependent forms)?

A8:Two cases:

- local or system-specific file systems (e.g. a file system to be mounted over /dev)
- file systems to which only one system has connectivity.

FAQ's ...

Q9: What about performance?

A9:This isn't a simple answer. It all depends on numbers of systems in sysplex, number of channels, and where the application is running compared to where the filesystem is mounted.

- A Washington System Center flash Flash #10020 has been issued to specifically address how much peformance degradation one may see.

- Future performance benefits are being designed!

FAQ's ...

Q10: Is the Shared HFS support going to rolled back to previous releases?

A10:No, it is only available on R9 or higher.

(FYI, even if we started to rework all of our code today, and retrofitted it back to previous releases, by the time we got done those releases would be out of support!)

FAQ's ...

Q11: What about the TFS? Can I use that in the Shared HFS environment?

A11: Sure, the TFS (Temporary File System) can be used. But remember the same rules regarding naming conventions apply. This means that each system needs to have a different name for the TFS.

- Example: The BPXPRMxx member should contain the following statements for a 10 MB TFS.

```
FILESYSTYPE TYPE(TFS) ENTRYPOINT(BPXTFS)

MOUNT FILESYSTEM('/TMP&SYSNAME.')
TYPE(TFS) MODE(RDWR) NOAUTOMOVE
MOUNTPOINT('/&SYSNAME./tmp')
PARM('-s 10')
```

FAQ's ...

Q12: Are there any other hints, tips, or gotchas we need to be aware of?

A12: Gotcha: When the owner of a NOAUTOMOVE mounted filesystem leaves the sysplex, entries remain in the CDS and are recoverable by the original owner when it returns to the sysplex. However, if the contents of the original filesystem is changed before the system returns to the sysplex, or if another HFS is mounted on the same mountpoint, unpredicatable results may occur.

- We therefore recommend that you perform an UNMOUNT of the unowned filesystem when you are replacing it with a filesystem of a different name (a different HFS data set name) or if the contents of the filesystem is different (when the HFS data set name is the same).

FAQ's ...

A12 Continued: When performing a scheduled outage, use BPXSTOP (tool that can be downloaded from our Tools and Toys page - http://www.s390.ibm.com/unix) to unmount the filesystems owned by this system.

- New function will be available soon via PTF's (down to R7) for a new MODIFY command, to move or unmount the filesystems off of the system being shutdown.

- We are working on a design - something like an 'autoremove' option on the MOUNT statement.

Build-Time Requirements

- Customers who will use the filesystems sent by IBM to install other products AND who use the /Service philsophy will need to:

1. Mount the Root HFS at a given mountpoint (ex: /Service)
2. Convert the /etc symlink to be a directory using the BPXISETD job. Must pass in the /Service as a paramter to REXX exec.
   - Can use BPXISJCL to submit in background.
3. Mount the ETC filesystem (ex : on /Service/etc)
4. Install products and PTFs

Build-Time Requirements ...

- Customers who use DFDSS to move the filesystems into production will then need to:

1. Unmount the ETC filesystem
2. Convert /etc back to be a symbolic link using BPXISETS job. Must pass in the /Service as a paramter to REXX exec.
   - Can use BPXISJCL to submit in background.
3. Use DFDSS DUMP to DUMP the ROOT and ETC HFS data sets.
4. On Production system DFDSS RESTORE filesystems and Re-IPL.

List of sample jobs

- BPXISYSR - JCL job used to create the Sysplex Root HFS data set
- BPXISYSS - JCL job used to create System-Specific HFS data set
- BPXISETD - REXX exec used to convert /etc to become a directory.
- BPXISETS - REXX exec used to convert /etc to become a symbolic link.
- BPXISJCL - Generic JCL job to be used to submit any REXX exec as batch job.
- BPXISCDS - JCL to create OMVS Couple data set.

Publications

- Most of this material is covered in the OS/390 UNIX System Services: Planning - SC28-1890
- Other books that were updated:
  - OS/390 UNIX System Services User's Guide
  - OS/390 UNIX System Services Command Reference
  - OS/390 UNIX System Services: Programming Assembler Callable Services
  - OS/390 Using REXX and OS/390 UNIX System Services
  - OS/390 UNIX System Services Messages and Codes
  - OS/390 UNIX System Services File System Interface Reference
  - OS/390 MVS System Commands
  - OS/390 MVS Setting up a Sysplex
- We are also looking at providing this type of presentation with audio capability on the Internet.

Summary

- Using USS File System Sysplex Support, the customer can
  1. Have a common file system hierarchy on all systems
  2. Write to file systems from all systems in the sysplex
  3. Have greater availability of data in the event of a system outage
  4. Better manage file system placement

- Advantages:
  1. Greater user mobility
  2. Flexibility with file system balancing
  3. Consolidation of data

- You should now be able to:
  1. Setup a the filesystem for use in a sysplex
  2. Understand the terms introduced with Shared HFS

Glossary of Terms

- CDS - Couple Data Set (used by USS as a sysplex wide Mount Table)
- Client - A system other than the owner, which is sharing a file system through communication with the owner.
- Common File Hierarchy - All systems sharing the same view of the file tree structure.
- Owner - System assigned as the mount coordinator for a particular file system, and in some cases, the system coordinating I/O for that file system.
- Sharing Mode - Participating in USS File System Sharing introduced in Release 9.
- XCF - MVS Cross System Coupling Facility