z/OS Version 1 Release 2

z/OS Distributed File Service (DFS) zSeries File system (zFS)

zFS File System

- zFS is a "new" Unix File System for z/OS
  - A component of the Distributed File Service since 1995
  - Does not replace HFS
- Using zFS, you can
  - Run most applications just like HFS
  - Use zFS in addition to HFS
- Advantages:
  - Better performance in many environments
  - Enhanced administrative functions
  - Less loss of data on system failure

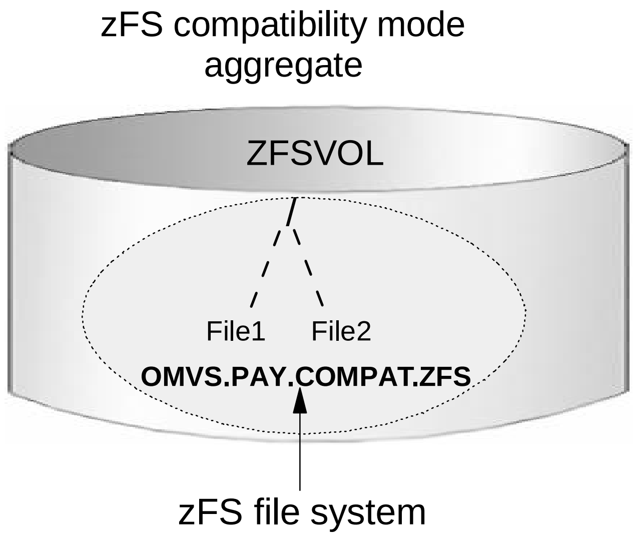
zFS Aggregates

- An aggregate is a VSAM linear data set (LDS)
- An aggregate contains one or more zFS file systems
- Two types of aggregates:
  - HFS compatibility mode - contains 1 zFS file system
  - Multiple file system - contains 1 or more zFS file systems
- Space sharing between file systems in the same data set
- File system maximum size as a logical value
- File system cloning (making a read/only copy)

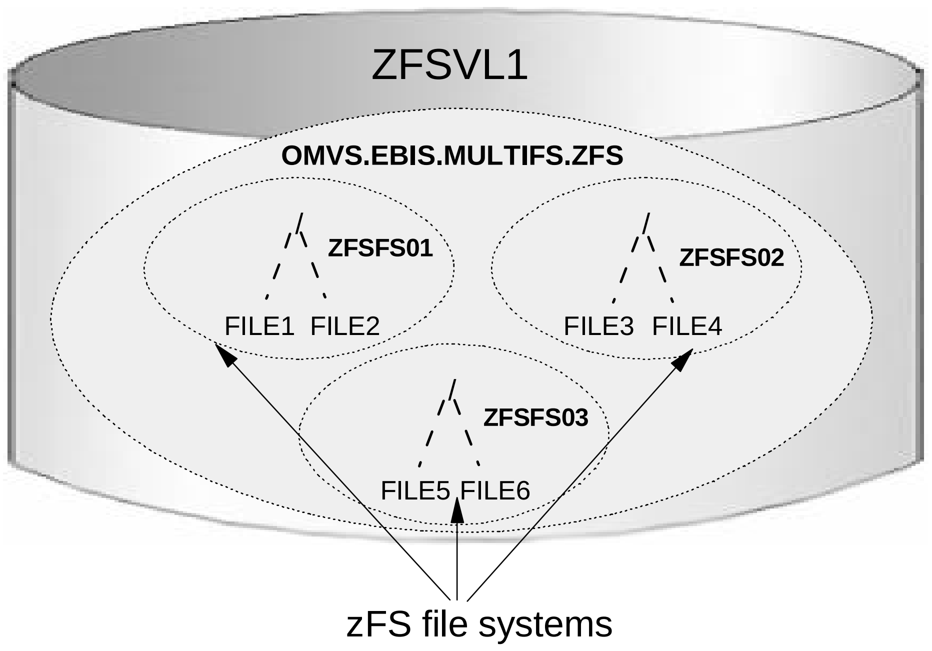
zFS Aggregates and File Systems

- zFS Aggregate
  - A VSAM Linear Data Set (LDS) that contains one or more zFS file systems
  - Aggregate name is VSAM LDS name
- zFS File System
  - A logical named entity contained within a zFS aggregate
  - Can be mounted within the HFS root file system
  - Contains a logical maximum size known as its quota

zFS Compatibility Mode Aggregate



Multiple File System Aggregate

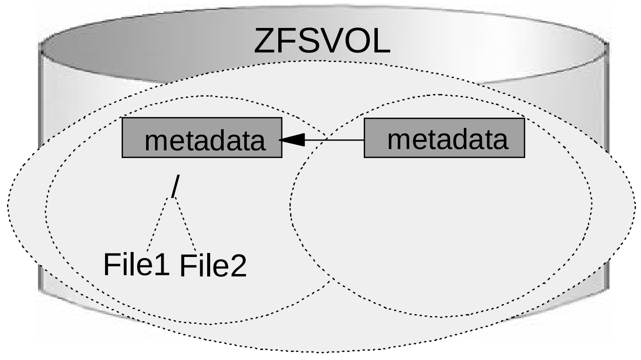


zFS Aggregates

- zFS aggregate attach
  - zFS multiple file system aggregates are attached
    - When the zFS PFS is started when the aggregates are listed in the IOEFSPRM member
    - Attach can also occur by command - zfsadm
  - zFS compatibility mode aggregates
    - Not attached as a separate step
    - Attached on MOUNT

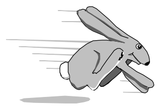
zFS File System Cloning

- zFS file system clone is a "copy" of a zFS file system
  - In the same aggregate
  - The clone file system is R/O
  - Only the metadata is copied
  - The cloned file system is called *name*.bak



z/OS Version 1 Release 2

zFS File System Installation



zSeries zFS File System Installation

- Additional RACF definitions
- BPXPRMxx entry required to define and run zFS
  - Defining zFS as a UNIX System Services PFS
    - **FILESYSTYPE** entry
    - **ZFS PROC**
  - zFS starts in a colony address space
- Allocate and format zFS aggregates
  - Format utility - **IOEAGFMT**
- Define a zFS parameter file - **IOEFSPRM** (optional)
  - Multiple file system only

zSeries zFS File System Installation

- Define a zFS file system in an aggregate
- Before the mount of a zFS file system
  - Create a directory mount point
- Mount a zFS file system in the USS directory hierarchy
  - Compatibility mode aggregates
    - MOUNT command - AUTOMNT - /etc/rc
  - Attach a zFS multiple file system aggregate
    - zfsadm command - or - Use IOEFSPRM member - or IOEZADM utility
    - MOUNT command (TSO/E)
- Accessing a zFS file system in the same way as an HFS file system is accessed

zSeries zFS File System Installation

- How to run zFS salvage program to validity check or recover an aggregate
- Commands and utilities known to work with zFS
- Sysplex sharing
- Known zFS restrictions

RACF Definitions for zFS

- RACF definitions required for Distributed File Service
  - SMB and DFS support
    - Not required to use zFS
- Additional definitions for zFS
  - RDEFINE STARTED ZFS.** STDATA(USER(DFS))
  - SETROPTS RACLIST(STARTED) REFRESH

```
ADDGROUP DFSGRP SUPGROUP(SYS1) OMVS(GID(2))
ADDUSER DFS OMVS(HOME(/opt/dfslocal/home/dfscntl) UID(0))
     DFLTGRP(DFSGRP) AUTHORITY(CREATE) UACC(NONE)
RDEFINE STARTED DFS.** STDATA(USER(DFS))
RDEFINE STARTED DFSCM.** STDATA(USER(DFS))
RDEFINE STARTED ZFS.** STDATA(USER(DFS))
   SETROPTS RACLIST(STARTED)
   SETROPTS RACLIST(STARTED) REFRESH
```

BPXPRMxx Definitions

- BPXPRMxx FILESYSTYPE statement
  - Defines zFS as a physical file system (PFS)

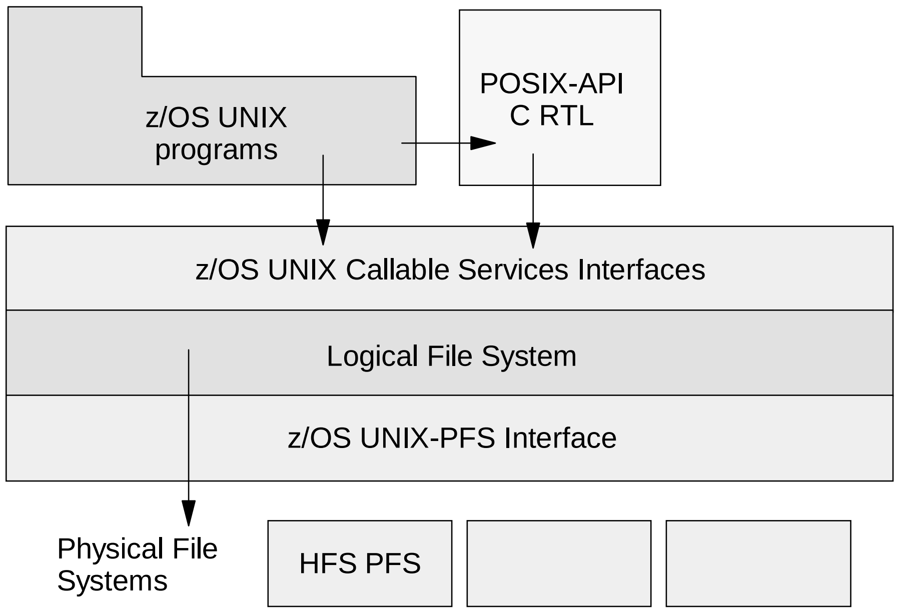
```
FILESYSTYPE TYPE(ZFS)
  ENTRYPOINT(IOEFSCM)
  ASNAME(ZFS)
```

ZFS PROC

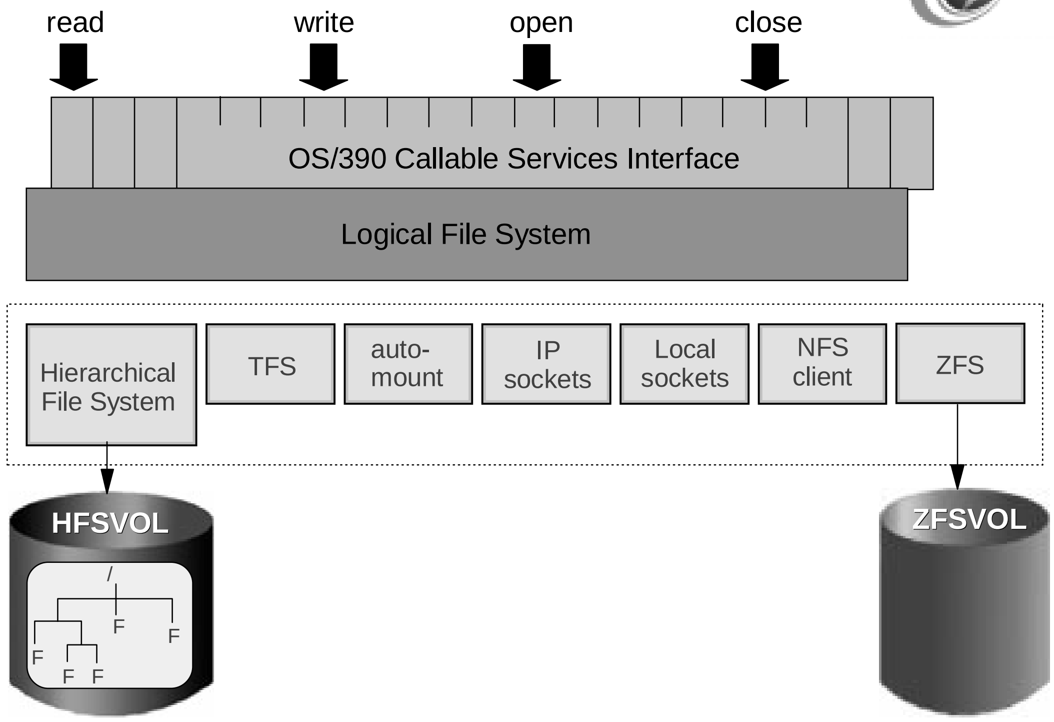
```
//ZFS      PROC REGSIZE=0M
//ZFSGO EXEC PGM=BPXVCLNY,REGION=&REGSIZE,TIME=1440
//*STEPLIB  DD DISP=SHR,DSN=IOE.SIOELMOD
//IOEZPRM DD DSN=IOE.PARMLIB(IOEFSPRM),DISP=SHR
// PEND
```

optional

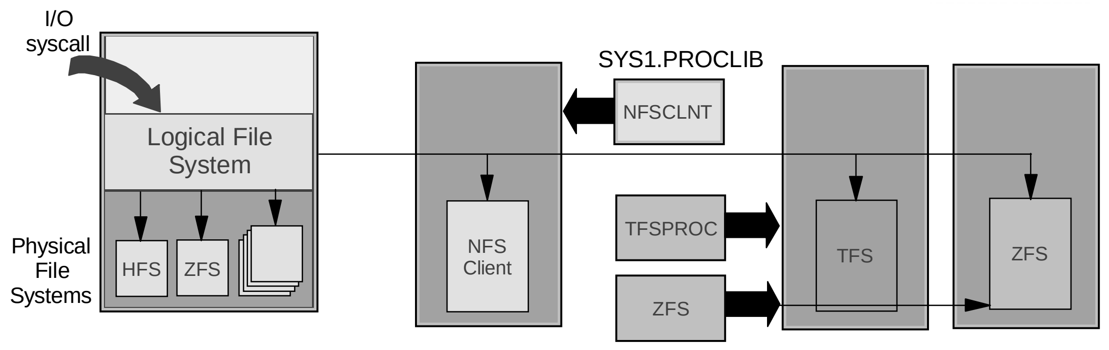
UNIX System Services



Physical File Systems



zFS Colony Address Space



BPXPRMxx

```
FILESYSTYPE TYPE(ZFS)
   ENTRYPOINT(IOEFSCM)
   ASNAME(ZFS)
FILESYSTYPE TYPE(TFS)
   ENTRYPOINT(BPXTFS)
   ASNAME(TFSPROC)
```

Procedure name in SYS1.PROCLIB

Create zFS Aggregates

- An aggregate can contain one or more zFS file systems
- A zFS file system is equivalent to an HFS file system

```
//RFRZAL  JOB (999,POK),'R F',CLASS=A,MSGCLASS=U,NOTIFY=&SYSUID,
//         REGION=0M
//STEP1   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//DISK     DD  DISP=OLD,UNIT=3390,VOL=SER=ZFSVOL
//SYSIN    DD  *
  DEFINE CLUSTER  -
         (NAME (OMVS.PAY.COMPAT.ZFS) VOL(ZFSVOL) -
         LINEAR CYL(200 0) SHAREOPTIONS(2))
  DEFINE CLUSTER  -
         (NAME (OMVS.EBIS.MULTIFS.ZFS) VOL(ZFSVL1) -
         LINEAR CYL(200 0) SHAREOPTIONS(2))
```

IOEAGFMT Format Utility

- Stand alone utility to format:
  - zFS multiple file aggregates
  - Compatibility mode aggregates
- Does not need the zFS colony address space to be active
- Does not use the IOEFSPRM configuration file

Format Utility Parameters

- Parameters must be specified in lower case

Parameter	Values	Description
-aggregate	VSAM LDS cluster name	The name of the data set to format.
-blocksize	4K, <u>8K</u>, 16K, 32K, 64K	The size in bytes of the logical block size.
-fragsize	<u>1K</u>, 2K, ..., up to blocksize	The size in bytes of a fragment. Multiple files can share a logical block if they are less than (logical block size - fragment size)
-aggrsize	a number	The size in blocks of the aggregate. The default is the number of blocks that will fit in the primary allocation.
-logsize	a number	The size in blocks of the log. The default is 1 % of the aggrsize.
-overwrite	NA	Reformat an existing aggregate
-compat	NA	Create a compatibility mode aggregate.

Format Aggregates

A compatibility mode aggregate is formatted with this JCL:

```
//RFRAGF  JOB (999,POK),'R F',CLASS=A,MSGCLASS=U,NOTIFY=&SYSUID,
// REGION=0M
//STEP1   EXEC PGM=IOEAGFMT,
//          PARM=(' -aggregate omvs.pay.compat.zfs -compat ')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

A multi-file system aggregate is formatted with this JCL:

```
//RFRAGF  JOB (999,POK),'R F',CLASS=A,MSGCLASS=U,NOTIFY=&SYSUID,
// REGION=0M
//STEP1   EXEC PGM=IOEAGFMT,
//          PARM=(' -aggregate omvs.ebis.multifs.zfs ')
//SYSPRINT DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
```

IOEAGFMT Successful Messages

```
IOEZ00004I Loading dataset 'rogers.zfs.compat'.
IOEZ00005I Dataset 'rogers.zfs.compat' loaded successfully.
*** Using default initialempty value of 1.
*** Using default number of (8192-byte) blocks: 17999
*** Defaulting to 179 log blocks(maximum of 19 concurrent transactions).
Done.  /dev/lfs1/rogers.zfs.compat is now an zFS aggregate.
IOEZ00071I Attaching aggregate ROGERS.ZFS.COMPAT to create hfs-compatible file system
IOEZ00074I Creating file system of size 140487K, owner id 0, group id 2, permissions x1ED
IOEZ00048I Detaching aggregate ROGERS.ZFS.COMPAT
IOEZ00077I HFS-compatibility aggregate ROGERS.ZFS.COMPAT has been successfully created
IOEZ00004I Loading dataset 'rogers.zfs.multifs'.
IOEZ00005I Dataset 'rogers.zfs.multifs' loaded successfully.
*** Using default initialempty value of 1.
*** Using default number of (8192-byte) blocks: 17999
*** Defaulting to 179 log blocks(maximum of 19 concurrent transactions).
Done.  /dev/lfs1/rogers.zfs.multifs is now an zFS aggregate.
```

Defining IOEFSPRM Options

- Options for IOEFSPRM file

Option	Values	Description
adm_threads	n, 10	The number of threads to handle pfsctl or mount requests.
auto_attach	ON, OFF	Whether aggregates in IOEFSPRM are automatically attached at start-up
user_cache_size	n, 256M	The size of the cache used to contain file data.
meta_cache_size	n, 32M	The size of the cache used to contain metadata.
log_cache_size	n, 64M	The size of the cache used to contain log buffers.
sync_interval	n, 30	The number of seconds between syncs.
vnode_cache_size	n, 8192	The initial size of the internal zFS vnode cache
nbs	ON, OFF	New Block Security. A global specification of whether we guarantee that new block data that has not made it to disk before a crash, will be shown as binary zeros after the crash.
aggrfull	(n,m), OFF	The threshold and increment for reporting aggregate full msgs to the op
fsfull	(n,m), OFF	The threshold and increment for reporting file system full msgs to the op

IOEFSPRM Definitions

- Options for aggregate attach

```
define_aggr [R/O | R/W] [[no]attach] [[no]nbs] [aggrfull(x,y)]
            cluster(VSAM_LDS_cluster_name)

R/O      - entire aggregate (all file systems) are R/O
attach   - aggregate is attached at PFS start up
nbs      - New Block Security - an allocated block with no data
written into it yet is shown as binary zeros (after  a crash)
aggrfull - the threshold and increment percentages for writing
aggregate full error messages to the operator
```

zfsadm Command

- Command to manage file systems and aggregates
- zfsadm subcommands:

`zfsadm attach`	`Attach an aggregate`
`zfsadm apropos`	`Display first line of help entry`
`zfsadm detach`	`Detach an aggregate`
`zfsadm grow`	`Grow an aggregate`
`zfsadm aggrinfo`	`Obtain information on an attached aggregate`
`zfsadm clone`	`Clone a filesystem`
`zfsadm clonesys`	`Clone multiple filesystems`
`zfsadm create`	`Create a filesystem`
`zfsadm delete`	`Delete a filesystem`
`zfsadm help`	`Get help on commands`
`zfsadm lsaggr`	`List aggregates`
`zfsadm lsfs`	`List filesystem information`
`zfsadm lsquota`	`List filesystem information`

Attaching an Aggregate

- Compatibility mode aggregates do not require attach
- Multiple file mode aggregates require an attach
  - 3 ways to attach
    - Attach at zFS colony address space startup by having an entry in the IOEFSPRM file
      - define_aggr R/W attach cluster(OMVS.WEB.MULTIFS.ZFS)
    - Using the IOEZADM program in a submitted job
      - PARM=('attach -aggregate OMVS.WEB.MULTIFS.ZFS')
    - Using the zfsadm attach command
      - zfsadm attach -aggregate omvs.web.multifs.zfs -aggrfull 90,5 -nbs

ln -s /usr/lpp/dfs/global/bin/zfsadm /bin/zfsadm

Defining zFS File Systems

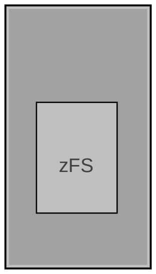
- Compatibility mode aggregates
  - 1 file system allowed
  - File system name = aggregate name
  - Size = size of VSAM LDS cluster
- Multiple file aggregates
  - Multiple file systems allowed
  - File system name installation determined
  - Size = predefined size

Defining Multiple File zFS File Systems

- Use IOEZADM program in JCL
- Use zfsadm command from OMVS shell

zfsadm create -filesystem ZFSFS01 -size 5000 -aggregate OMVS.WEB.MULTIFS.ZFS

```
//ZFZADM   JOB ,'ZFS Create Filesys',NOTIFY=ROGERS,
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),TIME=1440
//*
//ZFSADM   EXEC   PGM=IOEZADM,REGION=0M,
//         PARM=('create -filesystem ZFSFS01
//             -aggregate omvs.web.multifs.zfs -size 5000')
//STEPLIB DD DISP=SHR,DSN=IOE.SIOELMOD
//SYSPRINT DD   SYSOUT=T
//STDOUT   DD   SYSOUT=T
//STDERR   DD   SYSOUT=T
//SYSUDUMP DD   SYSOUT=T
//CEEDUMP  DD   SYSOUT=T
//*
```



Colony Address Space

IOEZ00099I File system ZFSFS01 created successfully

Mounting zFS File Systems

- Compatibility aggregates - 1 zFS file system
  - Place in /etc/rc
    - /usr/sbin/mount -t ZFS -f OMVS.PAY.COMPAT.ZFS /etc/zfs/zfsc
  - Mount command
  - Automount facility
  - AUTOMOVE option in a shared hfs (sysplex)
- Multiple file aggregates
  - Mount command

Restriction: No BPXPRMxx mount statements in z /OS V1R2

Mount Command from TSO/E

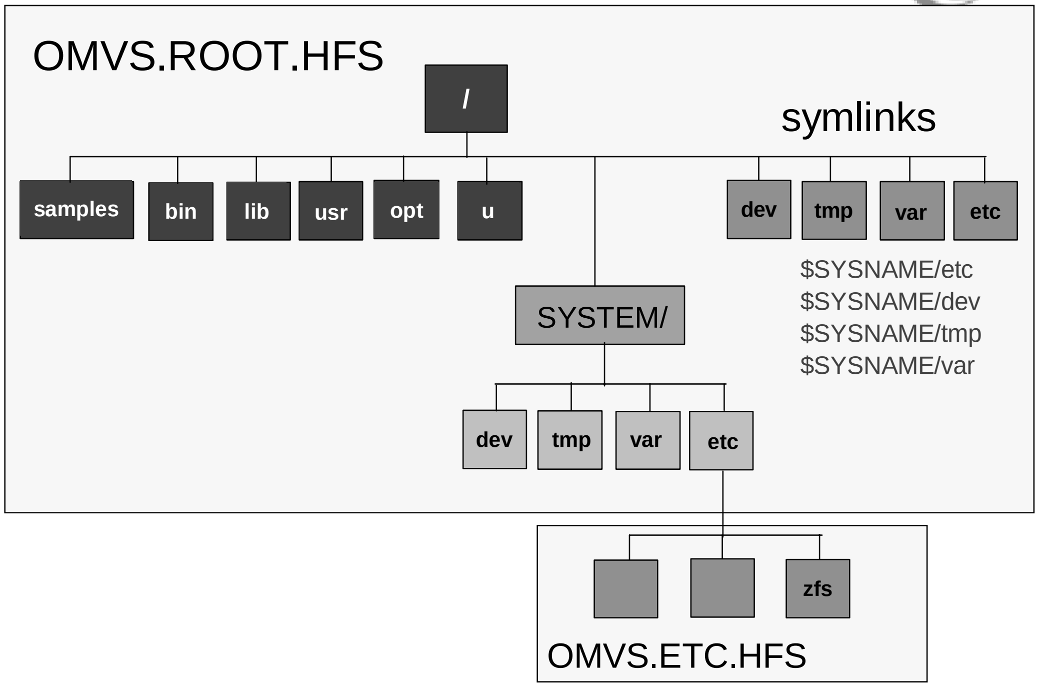
Mount a multi-file mode aggregate

```
MOUNT FILESYSTEM('ZFSFS01') TYPE(ZFS)
MODE(RDWR) MOUNTPOINT('/etc/zfs/zfs1')
```

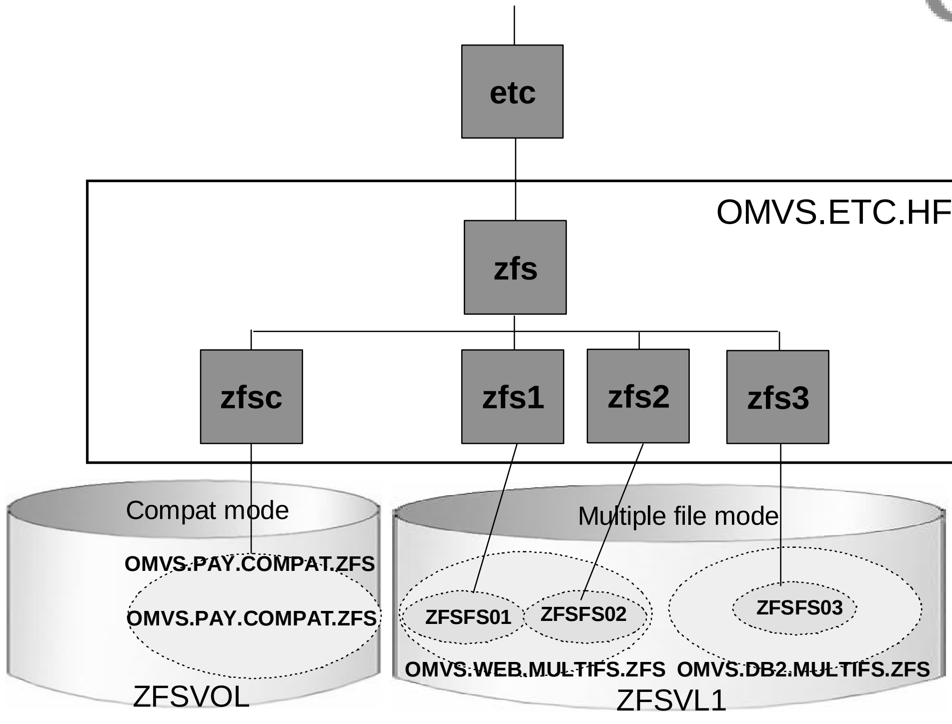
Mount a compatibility mode aggregate

```
MOUNT FILESYSTEM('OMVS.PAY.COMPAT.ZFS')
TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/etc/zfs/zfsc')
```

Create a Mount Point for zFS



zFS File Systems



zFS File System

- **zFS** is started by:
  - USS from the BPXPRMxx statement and ZFS PROC
    - At IPL, or
    - Via the **setomvs reset=(xx)** operator command...
- **zFS** is stopped by:
  - The **stop zfs** operator command
- **zFS** is restarted by:
  - Replying to the **BPXF032D** operator message
    - Response is r or i

z/OS Commands

- Query all the ZFS counters:
  - f zfs,query,all
- Reset the ZFS storage counters:
  - f zfs,reset,storage
- Format and send the trace table to the data set specified in the IOEFSPRM file - trace_dsn entry.
  - f zfs,trace,print
- ZFS Physical File System to dump and terminate:
  - f zfs,abort

zFS File System

- Called by:
  - USS applications via the USS address space
  - USS shell
- External Output
  - Console messages,
  - Message output data set
  - Trace output data set

zFS Recovery

- zFS is a logging file system
- It logs metadata updates
- On a system crash, the log is replayed to bring the file system to a consistent state
- I/O requests are started immediately (in an asynchronous manner) so that if a system crash occurs, most data is already on disk

zFS Aggregate Recovery

- A zFS aggregate can be backed up and restored using IDCAMS REPRO
  - The aggregate must be quiesced before the backup
- A zFS aggregate can be backed up and restored using DFSMSdss
  - The aggregate must be quiesced before the backup

Commands and Utilities Supported

- copy
- tar
- copytree
- pax
  - Logical level
  - Includes authorizations
  - To/from intermediate OMVS data set
  - Exception: can not restore at file level
- IDCAMS REPRO
  - Aggregate (physical volume) Level
  - Aggregate Must Be Quiesced before backup
- DFSMSdss
  - Aggregate (physical volume) Level
  - Aggregate Must Be Quiesced before backup

pax Command

- HFS file system mounted at /etc/hfs1
  - Copy this into an empty zFS file mounted at /etc/zfs/zfs1
    - cd /etc/hfs1
    - pax -rwv . /etc/zfs/zfs1

Sysplex Rules

- Only systems running zFS see zFS files
- Compatibility mode aggregates
  - Support AUTOMOVE and automount
- Multi-file mode aggregates
  - Mount NOAUTOMOVE
- All systems at V1R2 - All or some running zFS
  - System running zFS goes down
    - File systems mounted NOAUTOMOVE are lost

Documentation

- Distributed File Service zSeries Administration, SC24-5989
- z/OS Version 1 Release 2 Implementation, SG24-6235