

ibm.com



e-business



z/OS UNIX

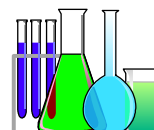
z/OS V1R5 and V1R6

Update



Redbooks

International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

Trademarks



eNetwork	DFSMS/MVS	IMS	RACF
geoManager	DFSMSdfp	IMS/ESA	RMF
AD/Cycle	DFSMSdss	IP PrintWay	RS/6000
ADSTAR	DFSMSshm	IPDS	S/390
AFP	DFSMSrmm	Language Environment	S/390 Parallel Enterprise Server
APL2	DFSORT	Multiprise	SecureWay
APPN	Enterprise System 3090	MQSeries	StorWatch
BookManger	Enterprise System 4381	MVS/ESA	Sysplex Timer
BookMaster	Enterprise System 9000	Network Station	System/390
C/370	ES/3090	NetSpool	SystemView
CallPath	ES/4381	OfficeVision/MVS	SOM
CICS	ES/9000	Open Class	SOMobjects
CICS/ESA	ESA/390	OpenEdition	SP
CICS/MVS	ESCON	OS/2	VisualAge
CICSPlex	First Failure Support Technology	OS/390	VisualGen
COBOL/370	FLowMark	Parallel Sysplex	VisualLift
DataPropagator	FFST	Print Services Facility	VTAM
DisplayWrite	GDDM	PrintWay	WebSphere
DB2	ImagePlus	ProductPac	3090
DB2 Universal Database	Intelligent Miner	PR/SM	3890/XP
DFSMS/MVS	IBM	QMFr	z/OS
			z/OS.e

Domino (Lotus Development Corporation)
DFS (Transarc Corporation)
Java (Sun Microsystems, Inc.)
Lotus (Lotus Development Corporation)

Tivoli (Tivoli Systems Inc.)
Tivoli Management Framework
(Tivoli Systems Inc.)
Tivoli Manger (Tivoli Systems Inc.)

UNIX (X/Open Company Limited)
Windows (Microsoft Corporation)
Windows NT (Microsoft Corporation)

© Copyright IBM Corp. 2004. All rights reserved.

z/OS UNIX Enhancements - z/OS V1R5



- ☐ Remount support for Shared HFS
- ☐ Enhancements for managing zFS
- ☐ Enhancements for BPXPRMxx Parmlib
- ☐ __map_services display capability
- ☐ LFS version verification enhancements
- ☐ TFS enhancements
- ☐ Multilevel security enhancements
- ☐ Symlink symbolics
- ☐ Socket I/O with ACEEs for SRBs
- ☐ Async I/O performance enhancements for authorized programs
- ☐ Allocating SWA above the line

© Copyright IBM Corp. 2004. All rights reserved.

Multilevel Security (MLS)



- ☐ Traditionally, access to z/OS UNIX resources is based on POSIX permissions and access control lists (ACLs)
- ☐ In a multilevel-secure z/OS UNIX environment:
 - Authorization checks are performed for:
 - Security labels
 - POSIX permissions
 - ACLs

© Copyright IBM Corp. 2004. All rights reserved.

z/OS UNIX and SECLABELs



☐ chlabel command

- This new shell command chlabel can be used to set security labels for UNIX files and directories
 - Requires RACF SPECIAL authorization
- Recommended do this command before MLS activated

☐ Once SECLABEL has been set it cannot be changed

- chlabel [-cqR] [-hl-L] seclabel pathname
 - By specifying the -R parameter, labels are set “recursively” on file subdirectories

Note: Only the zFS file systems supports the setting of SECLABELs

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



z/OS UNIX z/OS V1R5 Shared HFS Remount

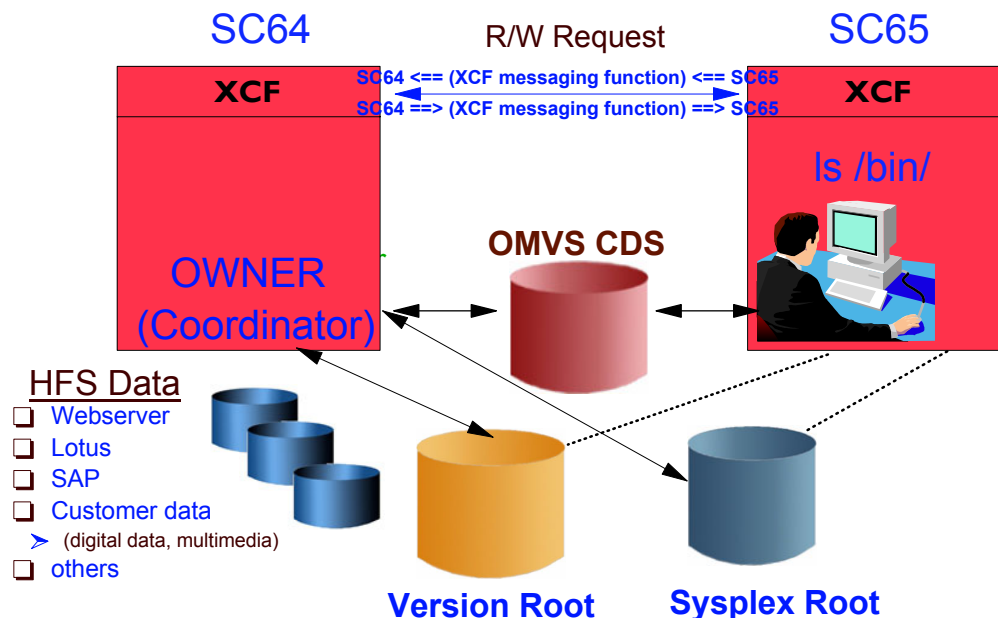


Redbooks
International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

Accessing Shared Sysplex File Systems



© Copyright IBM Corp. 2004. All rights reserved.

Remount for Shared HFS



- ☐ With Remount for Shared HFS you can:
 - Apply service to a R/O File System without unmounting the file system and mounting it R/W
- ☐ Advantages:
 - Applications reading from a R/O file system can continue to do so without disruption while the file system is remounted as R/W, and subsequently remounted to R/O
- ☐ Main use of remount is to switch a R/O file system to R/W for maintenance
- ☐ SMF recording on the unmount portion of the remount
 - This occurs on each system in the Shared HFS group
 - Remount = subtype 6

© Copyright IBM Corp. 2004. All rights reserved.

Remount Coexistence



- ❑ All systems in the Shared HFS group must be running at V1R5 or higher or on V1R4 with APAR:
 - V1R4 OA02584
- ❑ If one or more systems is incompatible, the new function is not supported and the old behavior is exhibited:
 - Retval = -1
 - Errno = EINVAL
 - ErrnoJr = JrNotSupInSysplex

ROGERS @ SC47: />bpjmtxt 058804A5

BPXFSUMT 12/02/02

JrNotSupInSysplex: Remount is not supported in sysplex. Unmount drain is tried as unmount normal, but if it cannot complete, an error is returned.

Action: Remount can be replaced with unmount followed by mount. Unmount drain can be replaced by looping on unmount normal until it successfully completes.

© Copyright IBM Corp. 2004. All rights reserved.

Using Remount Function



- ❑ From the BPX1UMT callable service using the MtmRemount flag
- ❑ From the TSO using: unmount filesystem(fsname)
remount(rdwr | read)
- ❑ From the shell using new options on chmount command:
 - -r to switch mounted file system to R/O mode
 - -w to switch mounted file system to R/W mode
- ❑ From the ISHELL mount table display under file systems

© Copyright IBM Corp. 2004. All rights reserved.

Remount R/O File System to R/W



Work with Mounted File Systems

Select one or more file systems with / or action codes.

U=Unmount A=Attributes M=Modify R=Reset unmount or quiesce

File system name	Status	Row 254 of 331
OMVS.SC66.WEB.CICS6	Available	
OMVS.SC66.WMQI.PROD.HFS	Available	
OMVS.SC66.WMQI.VAR.HFS	Available	
OMVS.SC67.ETC	Available	
OMVS.SC67.VAR	Available	
OMVS.SC69.ADSM.LOGS	Available	
OMVS.SC69.ETC	Available	
OMVS.SC69.IMSV7.HFS	Available	
OMVS.SC69.TOOLS	Available	
OMVS.SC69.VAR	Available	
OMVS.SC69.WSAA111.BUILD.HFS	Available	
OMVS.WSC206.BUILD.HFS	Available	
OMVS.ZOSR02.Z02RA1.ROOT	Available	
OMVS.ZOSR02.Z02RD1.ROOT	Available	
OMVS.ZOSR03.Z03RE1.ROOT	Available	
OMVS.ZOSR04.Z04RA1.ROOT	Available	
OMVS.ZOSR04.Z04RC1.ROOT	Available	
OMVS.ZOSR04.Z04RD1.ROOT	Available	
m OMVSZ13.RL000006.OMVS.ROOT	Available	
OMVSZ14.RL000006.OMVS.ROOT	Available	
OS390CB.Z15CB01.BUILD.ETC	Available	
OS390CB.Z15CB01.BUILD.ROOT	Available	
OS390CB.Z15CB01.BUILD.VAR	Available	
OS390R10.OMVS.ETC	Available	

Command ==>

F1=Help F3=Exit F4=Name F5=Retrieve F6=Keyshelp F7=Backward
F8=Forward F11=Command F12=Cancel

© Copyright IBM Corp. 2004. All rights reserved.

Change File System from R/O to R/W



Work with Mounted File Systems

S
U

Select the attribute to change

Select the attribute to change:

1. Change mount mode to R/W
2. Change Owning system from SC47
3. Change automove attribute...

New owning system _____

OMVS.SC69.TOOLS	Available
OMVS.SC69.VAR	Available
OMVS.SC69.WSAA111.BUILD.HFS	Available
OMVS.WSC206.BUILD.HFS	Available
OMVS.ZOSR02.Z02RA1.ROOT	Available
OMVS.ZOSR02.Z02RD1.ROOT	Available
OMVS.ZOSR03.Z03RE1.ROOT	Available
OMVS.ZOSR04.Z04RA1.ROOT	Available
OMVS.ZOSR04.Z04RC1.ROOT	Available
OMVS.ZOSR04.Z04RD1.ROOT	Available
m OMVSZ13.RL000006.OMVS.ROOT	Available
OMVSZ14.RL000006.OMVS.ROOT	Available
OS390CB.Z15CB01.BUILD.ETC	Available
OS390CB.Z15CB01.BUILD.ROOT	Available
OS390CB.Z15CB01.BUILD.VAR	Available

Command ==>

© Copyright IBM Corp. 2004. All rights reserved.

File System to be Remounted



Work with Mounted File Systems

```
S
U
-
- Mode Change Confirmation
-
- CAUTION:
-   The selected file system is about to be remounted. The file
-   system is first unmounted and then mounted with a different mount
-   mode.
-   File system name:
-   OMVSZ13.RL000006.OMVS.ROOT
-
-   To proceed with the remount, press Enter.
-   To cancel the remount and continue, use the Cancel function key. To
-   exit this function, use the Exit function key.
-
-
- OMVS.SC69.VAR Available
- OMVS.SC69.WSAA111.BUILD.HFS Available
- OMVS.WSC206.BUILD.HFS Available
- OMVS.ZOSR02.Z02RA1.ROOT Available
- OMVS.ZOSR02.Z02RD1.ROOT Available
- OMVS.ZOSR03.Z03RE1.ROOT Available
- OMVS.ZOSR04.Z04RA1.ROOT Available
- OMVS.ZOSR04.Z04RC1.ROOT Available
- OMVS.ZOSR04.Z04RD1.ROOT Available
- m OMVSZ13.RL000006.OMVS.ROOT Available
- OMVSZ14.RL000006.OMVS.ROOT Available
- OS390CB.Z15CB01.BUILD.ETC Available
- OS390CB.Z15CB01.BUILD.ROOT Available
- OS390CB.Z15CB01.BUILD.VAR Available
- Command ==>
```

© Copyright IBM Corp. 2004. All rights reserved.

Error Occurs on Systems Without APAR



Work with Mounted File Systems

```
S
U
-
- Select the attribute to change
-
- Select the attribute to change:
- 3 1. Change mount mode to R/W
-    2. Change Owning system from SC47
-    3. Change automove attribute...
-
- New owning system
-
-
-
-
- Errno=79x The parameter is incorrect; Reason=058804A5. Press Enter to
- continue.
-
- OMVS.ZOSR02.Z02RA1.ROOT Available
- OMVS.ZOSR02.Z02RD1.ROOT Available
- OMVS.ZOSR03.Z03RE1.ROOT Available
- OMVS.ZOSR04.Z04RA1.ROOT Available
- OMVS.ZOSR04.Z04RC1.ROOT Available
- OMVS.ZOSR04.Z04RD1.ROOT Available
- m OMVSZ13.RL000006.OMVS.ROOT Available
- OMVSZ14.RL000006.OMVS.ROOT Available
- OS390CB.Z15CB01.BUILD.ETC Available
- OS390CB.Z15CB01.BUILD.ROOT Available
- OS390CB.Z15CB01.BUILD.VAR Available
- Command ==>
```

© Copyright IBM Corp. 2004. All rights reserved.

zFS Remount Considerations



❑ zFS Compatibility Mode Aggregates

- If both the primary file system and its clone are mounted, remount will fail because we cannot detach the aggregate
 - EINVAL, JrAggregateErr

❑ zFS Multiple File System Aggregates

- If the aggregate is attached R/O then file systems in it can only be mounted R/O and remount to R/W will result in a zFS error

© Copyright IBM Corp. 2004. All rights reserved.

Example Remount Command with zFS



```
#> echo $(uname -I) Version $(uname -Iv) .$(uname -Ir)
z/OS Version 01.05.00
#> /usr/sbin/mount -f OMVS.HERING.TEST.ZFS -t zFS /u/hering/testfs
#> zfsadm clone -filesystem OMVS.HERING.TEST.ZFS
IOEZ00225I File system OMVS.HERING.TEST.ZFS successfully cloned.
#> /usr/sbin/mount -f OMVS.HERING.TEST.ZFS.bak -t zFS -r /u/hering/testfs.clone
#> zfsadm lsfs -aggregate OMVS.HERING.TEST.ZFS
IOEZ00129I Total of 2 file systems found for aggregate OMVS.HERING.TEST.ZFS
OMVS.HERING.TEST.ZFS      RW (Mounted R/W)      8 K alloc      9 K quota On-line
OMVS.HERING.TEST.ZFS.bak BK (Mounted R/O)      9 K alloc      9 K quota On-line
Total file systems on-line 2; total off-line 0; total busy 0; total mounted 2
#> zfsadm aggrinfo -aggregate OMVS.HERING.TEST.ZFS
OMVS.HERING.TEST.ZFS (R/W COMP): 558 K free out of total 720
#> /usr/sbin/chmount -r /u/hering/testfs
FOMF0504I remount error: 79 5090576
EINVAL: The parameter is incorrect
JrAggregateErr: Remount is not allowed for a filesystem in an HFS-compatible
aggregate if the clone is also mounted.
```

© Copyright IBM Corp. 2004. All rights reserved.

Avoid Wait-state in Shared HFS



- ❑ UNIX System Services can wait-state on a system with incompatible software level
 - **Solution:** Behavior of USS is modified to perform:
 - F OMVS,SHUTDOWN to disable USS
 - NOTE: This command was introduced in z/OS V1R3
 - Avoids disabling the whole system
 - 2 new messages added:
 - BPXF079S - indicates OMVS cannot initialize
 - BPXF080I - identifies systems with incompatible software levels

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



SymLink Symbolics

z/OS UNIX with z/OS V1R5

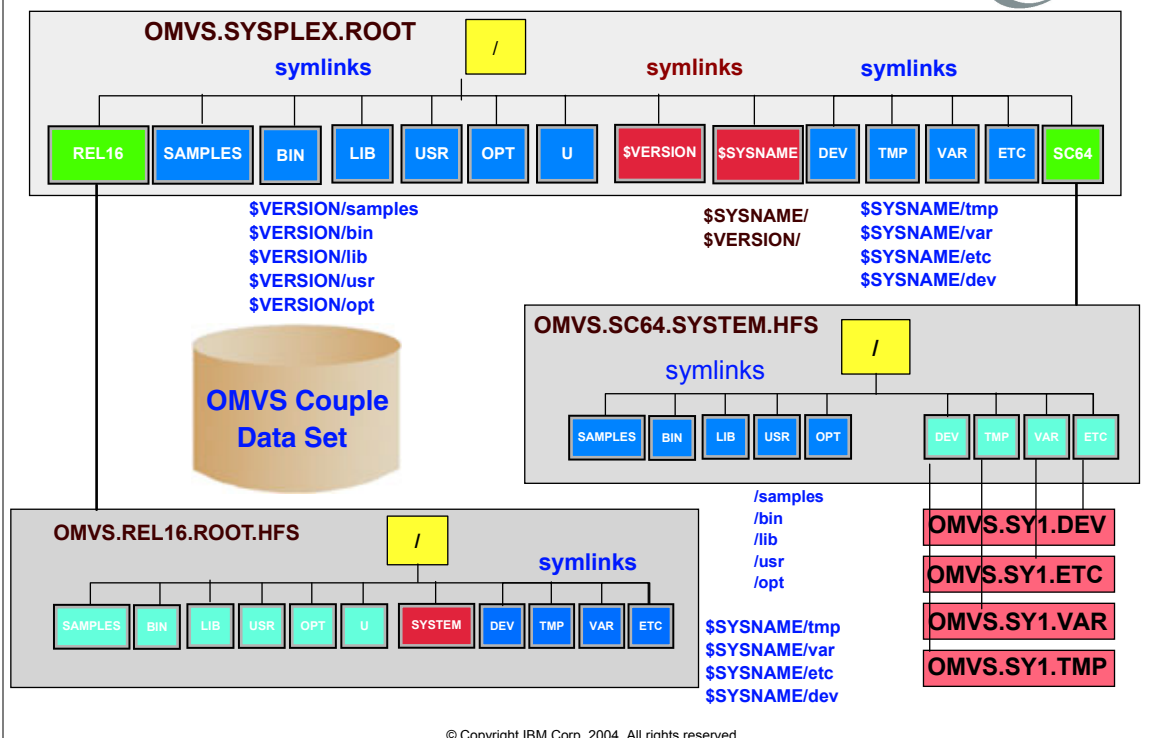


Redbooks
International Technical Support Organization

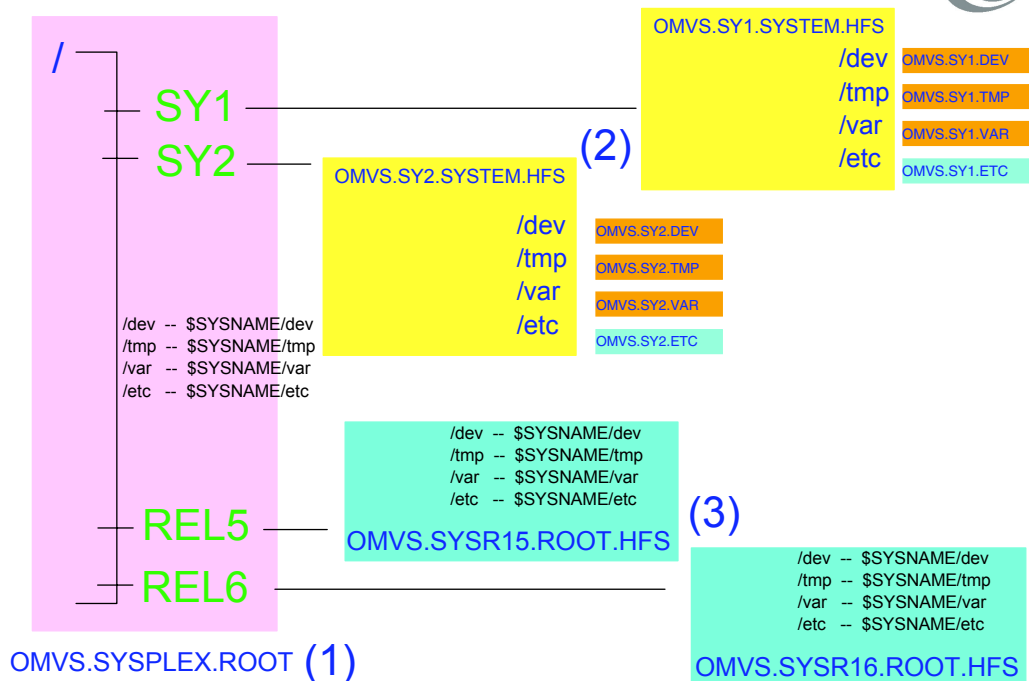


© Copyright IBM Corp. 2004. All rights reserved.

HFS Data Sets - Shared Sysplex



Multiple Systems - Different Versions



Symlink Symbolics



- ❑ There is no way to specify mountpoints such that they resolve on systems based on system symbolics, which may or may not be different between systems
- ❑ Solution:
 - Support is added to allow use of static system symbols in symlinks with an identifier indicating substitution is necessary in the resolution of the symlink
 - With Remount for Shared HFS you can:
 - Imbed static system symbols in a symlink
 - Allows for unique pathname resolution based on the value of the system symbol on a particular system
 - Great for mountpoints that you want to share with a subset of systems in the Shared HFS group

© Copyright IBM Corp. 2004. All rights reserved.

Symlink use of System Symbols



- ❑ Need to specify mountpoints such that they resolve on systems based on system symbols, which may or may not be different between systems
- Support is added to allow use of static system symbols in symlinks with an identifier indicating substitution is necessary in the resolution of the symlink
- With Remount for Shared HFS, you can be able to imbed static system symbols in a symlink
 - Allows for unique pathname resolution based on the value of the system symbol on a particular system
 - Great for mountpoints that you want to share with a subset of systems in the Shared HFS group
- Symlink contents must begin with \$SYSSYMA/ or \$SYSSYMR/ followed by a system symbol

© Copyright IBM Corp. 2004. All rights reserved.

Create System Symbols and Symlinks



❑ IEASYMxx on SC65

- SYMDEF(&DB2VER='DB2V7')

❑ IEASYMxx on SC70

- SYMDEF(&DB2VER='DB2V8')

❑ Definition of symlinks using "ln" command:

- ln -s '\$SYSSYMA/&DB2VER.' /db2
- ln -s '\$SYSSYMR/&DB2VER.' /usr/lpp/db2
 - Note: Quotes are needed, because otherwise \$SYSSYMA and \$SYSSYMR would be interpreted as a UNIX environment variable
- You can use the ISHELL to create the symlink

© Copyright IBM Corp. 2004. All rights reserved.

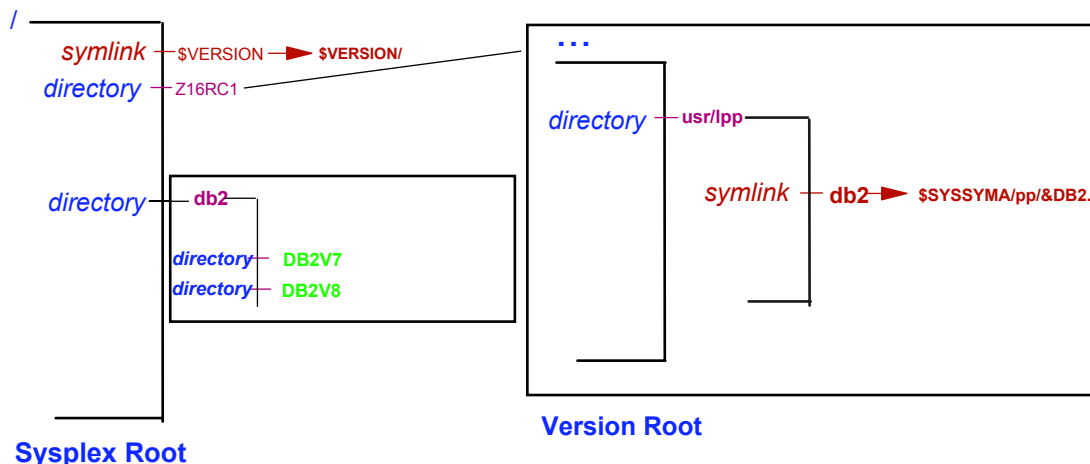
Symlink Symbolics ...



symlink /db2 → \$SYSSYMA/&DB2VER.

Resolves to: /db2/DB2V7 - on system SC65

Resolves to: /db2/DB2V8 - on system SC70



© Copyright IBM Corp. 2004. All rights reserved.

BPXPRMxx Member



BPXPRMxx mount statement:

```
MOUNT FILESYSTEM('OMVS.&DB2..ZFS')  
  MOUNTPOINT('/usr/lpp/db2')  
  AUTOMOVE TYPE(ZFS) MODE(RDWR)
```

The filesystem that is mounted on system SC65 is
OMVS.DB2V7.ZFS.

The filesystem that is mounted on system SC70 is
OMVS.DB2V8.ZFS.

© Copyright IBM Corp. 2004. All rights reserved.

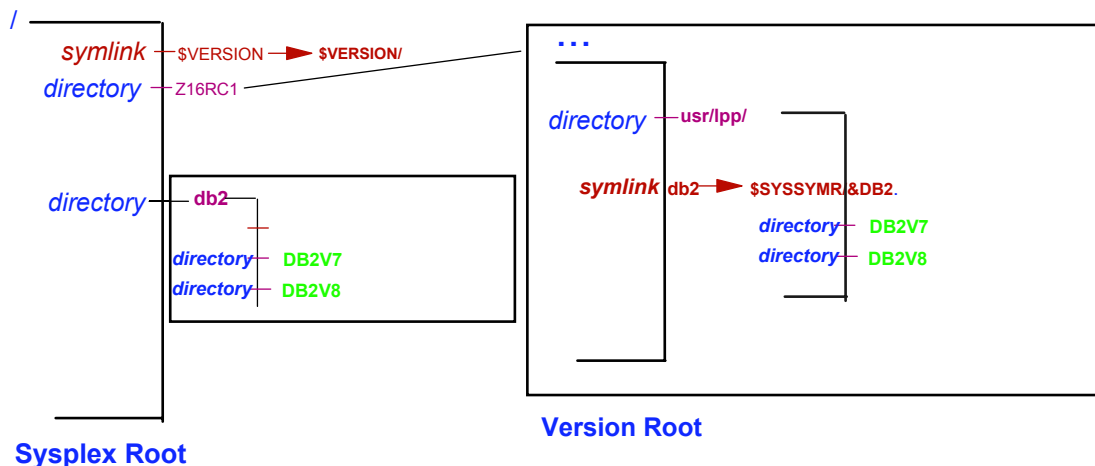
Symlink Symbolics ...



```
symlink /usr/lpp/db2→$SYSSYMR/&DB2.
```

Resolves to: /usr/lpp/DB2V7 - on system SC65

Resolves to: /usr/lpp/DB2V8 - on system SC70



© Copyright IBM Corp. 2004. All rights reserved.

BPXPRMxx Member



BPXPRMxx mount statement:

```
MOUNT FILESYSTEM('OMVS.&DB2..ZFS')  
  MOUNTPOINT('/pp/db2')  
  AUTOMOVE TYPE(ZFS) MODE(RDWR)
```

The filesystem that is mounted on system SC65 is OMVS.DB2V7.ZFS.

The filesystem that is mounted on system SC70 is OMVS.DB2V8.ZFS.

© Copyright IBM Corp. 2004. All rights reserved.

BPXPRMxx Parmlib Enhancement (1.5)



- ☐ Parmlib mounts can fail if the mountpoint does not exist
- ☐ Support is added to the BPXPRMxx parmlib member to allow specifying directories to be created during parmlib processing
 - Supported during UNIX System Services initialization
- ☐ New MKDIR() keyword for parmlib member
 - Specify one or more directory entries to be created in the file system associated with the keyword, or in another file system already mounted
- ☐ If sharing parmlib members between Shared HFS members this keyword should be omitted unless all are running at V1R5 or above

© Copyright IBM Corp. 2004. All rights reserved.

Examples of MKDIR Use in BPXPRMxx



ROOT

FILESYSTEM('fsroot')
TYPE(type_name)
MODE(access)
PARM('parameter')
SETUIDINOSETUID
AUTOMOVE | NOAUTOMOVE
TAG(NOTEXT | TEXT,ccsid)
MKDIR('pnfs1')

MOUNT

FILESYSTEM('fs1')
MOUNTPOINT('/**pnfs1**')
TYPE(type_name)
MODE(access)
PARM('parameter')
SETUID / NOSETUID
SECURITY | NOSECURITY
AUTOMOVE | NOAUTOMOVE
AUTOMOVE(Ind,S1,...Sn)
TAG(NOTEXT|TEXT,ccsid)
MKDIR('pnfs2')
MKDIR('pnfs3')

© Copyright IBM Corp. 2004. All rights reserved.

MKDIR Support



- ☐ Permissions are set to 755, (rw x r-x r-x)
- ☐ Not to be used with file systems that mount asynchronously, like NFSC (NFS Client)
- ☐ MKDIR failures will have no effect on the mount which they are associated with but can affect subsequent mounts if it is part of the intended mountpoint
- ☐ BPXPRMxx sample was updated with new MKDIR keyword (SYS1.SAMPLIB)
- ☐ Error messages sent to log file

© Copyright IBM Corp. 2004. All rights reserved.

Mount Failures if Data Set does not exist



- ❑ Parmlib mounts can fail if the data set or aggregate to be mounted does not exist
- ❑ **Solution:** SETOMVS SYNTAXCHECK is enhanced to identify parmlib statements whose data set or compatibility mode aggregate does not exist
 - Prevents parmlib errors during initialization

© Copyright IBM Corp. 2004. All rights reserved.

Allocating SWA above the Line



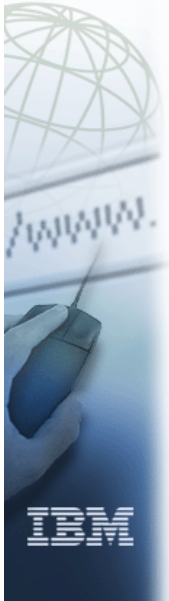
- ❑ SWA control blocks for USS address space are allocated below the 16 megabyte line causing storage constraints when very large numbers of file systems are mounted
- ❑ New BPXPRMxx parmlib keyword to control from where the SWA control blocks are allocated
 - New BPXPRMxx keyword SWA(ABOVE | BELOW)
 - D OMVS,O displays the setting
 - Only available when starting OMVS during system initialization

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business

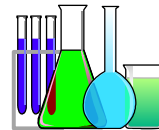


z/OS UNIX Systems Services z/OS V1R6 Update



Redbooks

International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

z/OS UNIX Enhancements - z/OS V1R6



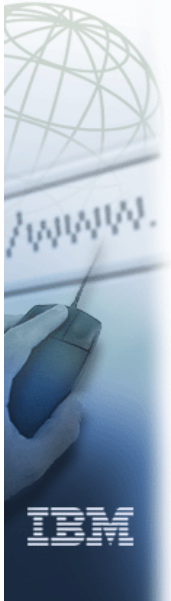
- ☐ Automove system list enhancements – wild card support
- ☐ Increase to 128K per process file descriptor limit
- ☐ Automount enhancements
- ☐ HFS to zFS automount
- ☐ ISHELL enhancements
- ☐ LFS support of sysplex zFS – PFS termination in Shared HFS
- ☐ Distributed BRLM enhancements
- ☐ Fork() accounting for FTP client with SSL
- ☐ Superkill command

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business



z/OS V1R6 Automove System List Enhancements – Wild Card Support



Redbooks

International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

AUTOMOVE System List Enhancement



- ☐ Explicitly list the priority systems and include the remaining systems using wildcard character in the include system list
 - Parmlib member
 - TSO MOUNT command
 - Shell - chmount
 - ISHELL
 - C Program
 - REXX
 - Assembler Program
 - SETOMVS FILESYS,FILESYSTEM='X.Y.Z',AUTOMOVE=(I,*)
- ☐ Automove Syslist Wildcard support is invoked by:
 - SETOMVS FILESYS,FILESYSTEM='X.Y.Z',SYSNAME=*
 - F BPXONIT, SHUTDOWN=FILESYS
 - F OMVS,SHUTDOWN

© Copyright IBM Corp. 2004. All rights reserved.

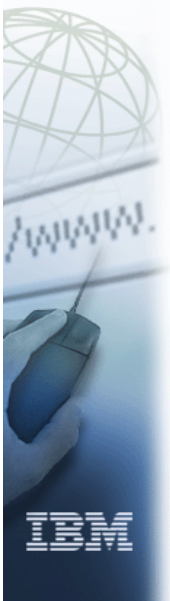
AUTOMOVE System List Enhancement



- ☐ Wildcard character permitted as an only item or as a last item of the Automove Include Syslist in place of a system name
- ☐ The wildcard is only allowed with an Include list, not with an Exclude list:
- ☐ Examples
 - AUTOMOVE(include,sy2,*)
 - AUTOMOVE(include,*)
 - AUTOMOVE(i,sy1,sy2,sy3,*)
 - AUTOMOVE(i,sy*)

© Copyright IBM Corp. 2004. All rights reserved.

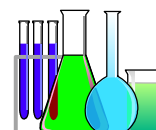
ibm.com



z/OS V1R6 Increase to 128K Per Process File Descriptor Limit



Redbooks
International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

Increase to 128K Per Process File Descriptor Limit



- ❑ Currently, a single UNIX process is limited to 64K file descriptors, which are used for all:
 - Open file system objects, including files, sockets, pipes, terminals, and directories
- ❑ Current system constraint for very large servers since it limits them to 64K connected clients at any one time
- ❑ z/OS V1R6 - 128K per process file descriptor limit support
 - Allows support for more clients on TN3270
 - Single UNIX process can now handle upto 131072 open file descriptors at any given time

© Copyright IBM Corp. 2004. All rights reserved.

Increase to 128K Per Process File Descriptor Limit - Console Commands



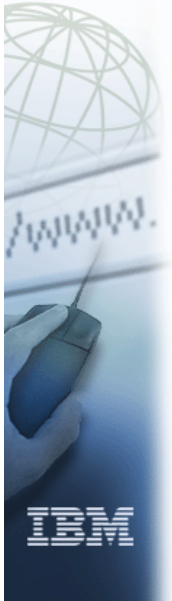
- ❑ BPXPRMxx MAXFILEPROC()
 - IPL and RESTART configuration option
 - SET OMVS=(xx) console command with BPXPRMxx MAXFILEPROC()
 - SETOMVS RESET=(xx)
 - SETOMVS MAXFILEPROC()
 - SETOMVS PID=ppp, MAXFILEPROC()
- ❑ RACF commands
 - Adduser userid OMVS(FILEPROC MAX(nnnn))
 - AltUser userid OMVS(FILEPROC MAX(nnnn))
 - ListUser userid OMVS
- ❑ Both V1R6 and lower level releases must not have MAXFILEPROC set higher than the old limit of 65535(64K)

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business

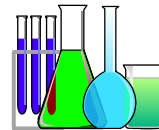


z/OS V1R6 Automount Enhancements



Redbooks

International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

Updating an Existing Automount Policy



- ☐ **New option (-a) added to the command line:**
 - "-a" option indicates that the policy being loaded is to be appended
 - Example: /usr/sbin/automount -a
 - Option "-a" is mutually exclusive with query option "-q"
- ☐ **New option (-q) added to command line**
 - "-q" option displays current active policy

© Copyright IBM Corp. 2004. All rights reserved.

Example Automount Map File



Automount policy in /etc/auto.map

```
name          *
filesystem    OMVS.<uc_name>.ZFS
type          ZFS
allocuser     space(1,1) storclas(OPENMVS)
mode          rdwr
duration      1440
delay         360
```

© Copyright IBM Corp. 2004. All rights reserved.

Example of New Options



```
name          *
filesystem    OMVS.<uc_name>.ZFS
type          ZFS
allocuser     space(1,1) storclas(OPENMVS)
mode          rdwr
duration      1440
delay         360
```

```
ROGERS @ SC65:/u/rogers>oedit /etc/auto.map ← Changed duration to 1400
ROGERS @ SC65:/u/rogers>/usr/sbin/automount -a
FOMF0107I Processing file /etc/auto.map
FOMF0108I Managing directory /u
ROGERS @ SC65:/u/rogers>/usr/sbin/automount -q
/u
```

```
name          *
filesystem    OMVS.<uc_name>.ZFS
type          ZFS
allocuser     space(1,1) storclas(OPENMVS)
mode          rdwr
duration      1400
delay         360
```

© Copyright IBM Corp. 2004. All rights reserved.

HFS to zFS Automount



- ❑ Change automount to set the file system type to either HFS or zFS when either of those is specified based on the type of data set that is being mounted
 - Specify and manage both HFS and zFS file systems in one automount policy
 - Facilitates migration from HFS to zFS over time rather than all file systems at once
- ❑ Determine whether the data set is a HFS type or not:
 - If it is, then set the file system type to HFS
 - If it is not, then set the file system type to zFS

Available for z/OS V1R5 with APAR OA06364 and PTF UA10075

© Copyright IBM Corp. 2004. All rights reserved.

One Auto.master for a Sysplex



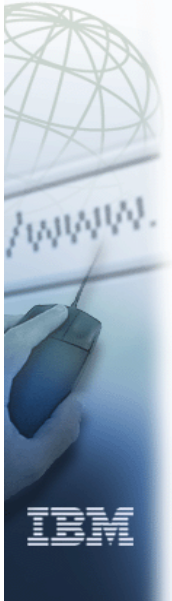
- ❑ The syntax is enhanced to indicate a master file name as a data set name:
 - Master file name can be specified on the command line
 - Data set name can be sequential or a member of a PDS
 - Convention of // preceding the data set name is used
 - Data set name must be a fully qualified name and can be in upper or lower case
- ❑ Example: /usr/sbin/automount "//sys1.parmlib(amtst01)"

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business

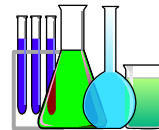


z/OS V1R6 RTLS Removal



Redbooks

International Technical Support Organization



© Copyright IBM Corp. 2004. All rights reserved.

Applications and RTLS



- ❑ Each time an application runs, a set of run-time options must be established
- ❑ These options determine many of the properties of how the application runs, such as:
 - Performance
 - Error handling characteristics
 - Storage management
 - Production of debugging information

© Copyright IBM Corp. 2004. All rights reserved.

Using Batch Application Submissions



- ❑ You specify run-time options in any of the following places where the installation default options are:
 - CEEDOPT CSECT for installation-wide defaults
 - CEEROPT CSECT for region-wide defaults
 - CEEUOPT CSECT for application-specific defaults
 - #pragma runopts in C/C++ source code
 - In a PLIXOPT string in PL/I source code
 - In the PARM parameter of the EXEC statement in your JCL
 - In z/OS on the GPARM parameter of the IBM-supplied cataloged procedure
 - In the assembler user exit
 - In the _CEE_RUNOPTS environment variable, when your application is running under z/OS UNIX and is invoked by one of the exec family of functions

© Copyright IBM Corp. 2004. All rights reserved.

RTLS Removal



- ❑ Language Environment used RTLS services as a temporary migration aide, as follows:
 - Used to aide migration to Language Environment from previous run-times
 - Allowed applications to specify a particular level of Language Environment run-time to be used

© Copyright IBM Corp. 2004. All rights reserved.

Migration to z/OS V1R6



- ❑ Changes to CEEDOPT and CEECOPT usermods needed at z/OS 1.6 installation
 - Existing CEEUOPTs or CEEROPTs which specified any run-time options now produce an informational message CEE3611I during initialization
 - To avoid this message the CEEUOPTs and CEEROPTs need to be reworked and re-linked
- ❑ The SCEERTLS data set is no longer shipped by LE
- ❑ Any JCL which references this data set needs to be updated

CEE3611I The run-time option option was an invalid run-time option or is not supported in this release of Language Environment.

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business



Distributed BRLM z/OS V1R6



Redbooks

International Technical Support Organization

© Copyright IBM Corp. 2004. All rights reserved.

BRLM Support



- ❑ Lock all or part of a file that you are accessing for:
 - Read-write purposes
- ❑ As a default, the lock manager is initialized on only one system in the sysplex
 - The first system that enters the sysplex initializes the BRLM and becomes the system that owns the manager - This is called a centralized BRLM
- ❑ If the BRLM server crashes, or owning system partitioned out of the sysplex
 - BRLM is reactivated on another system in the group
 - All locks that were held are lost
 - An application that accesses a file previously locked receives an I/O error - must close and reopen the file

© Copyright IBM Corp. 2004. All rights reserved.

Distributed BRLM - (z/OS V1R4)



- ❑ Can have distributed BRLM initialized on every system in the sysplex
 - Each BRLM is responsible for handling locking requests for files whose file systems are mounted locally in that system
 - Use distributed BRLM if you have applications that lock files that are mounted and owned locally
- ❑ To activate distributed BRLM
 - z/OS UNIX couple data set (BPXMCDs) must be updated
 - Supported code must be installed and running on each system - See APAR OW52293 for more information

© Copyright IBM Corp. 2004. All rights reserved.

Define BRLM Option in CDS



```
ITEM NAME (DISTBRLM) NUMBER(1)
/*Enables conversion to a distributed BRLM.
 1, distributed BRLM enabled,
 0, distributed BRLM is not enabled during next sysplex IPL
Default = 0 */
```



OMVS couple data set

Applications (including cron, inetd, and Lotus Domino)
lock local files

© Copyright IBM Corp. 2004. All rights reserved.

BRLM Problems in a Sysplex



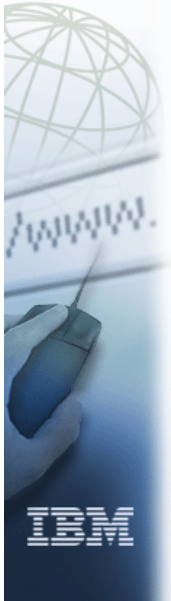
- ❑ A file system cannot be moved in a sysplex when an application holds byte range locks in that file system
 - Moving of locks was not supported with distributed BRLM
- ❑ Also, applications holding byte range locks in a remote file system are exposed when that remote sysplex member normally terminates
 - The locks are lost and the application is notified
- ❑ **Solution:**
 - Distributed BRLM is enhanced to support moving byte range locks

© Copyright IBM Corp. 2004. All rights reserved.

ibm.com



e-business



Fork() Accounting with Secure FTP client with SSL



Redbooks

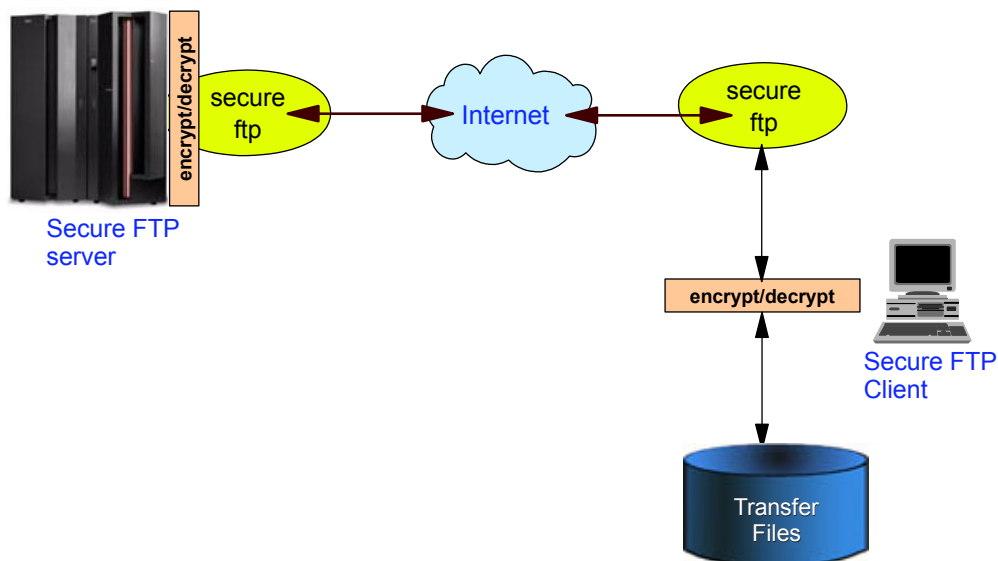
International Technical Support Organization

© Copyright IBM Corp. 2004. All rights reserved.

Secure FTP Client / Server Environment

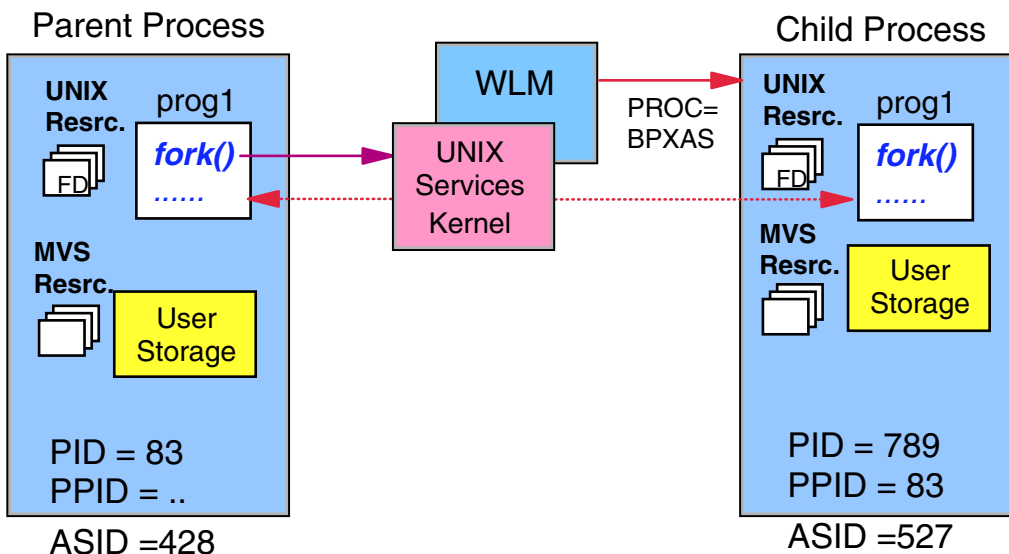


Customers who check accounting of Secure FTP clients



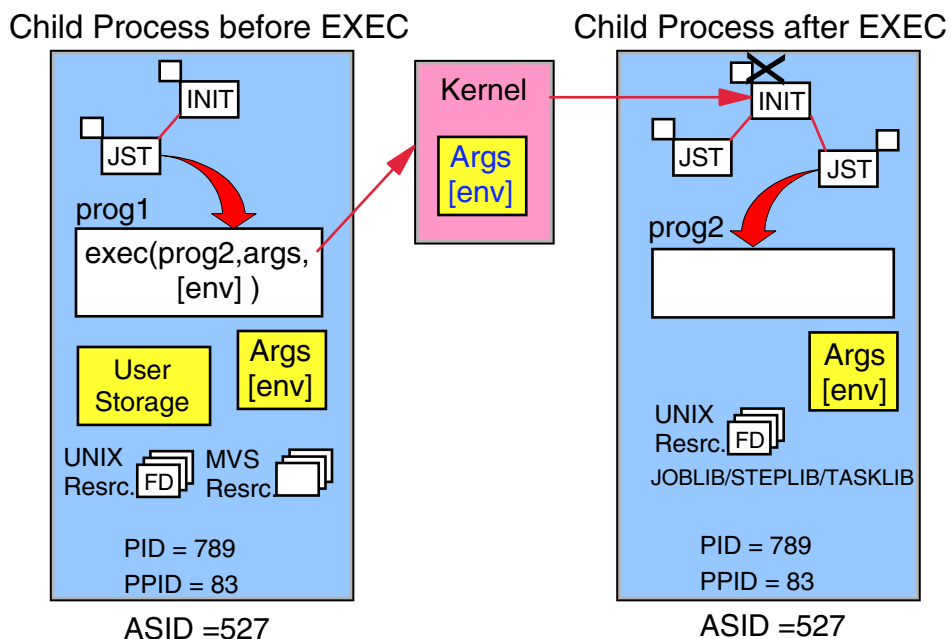
© Copyright IBM Corp. 2004. All rights reserved.

Create a Process Using Fork()



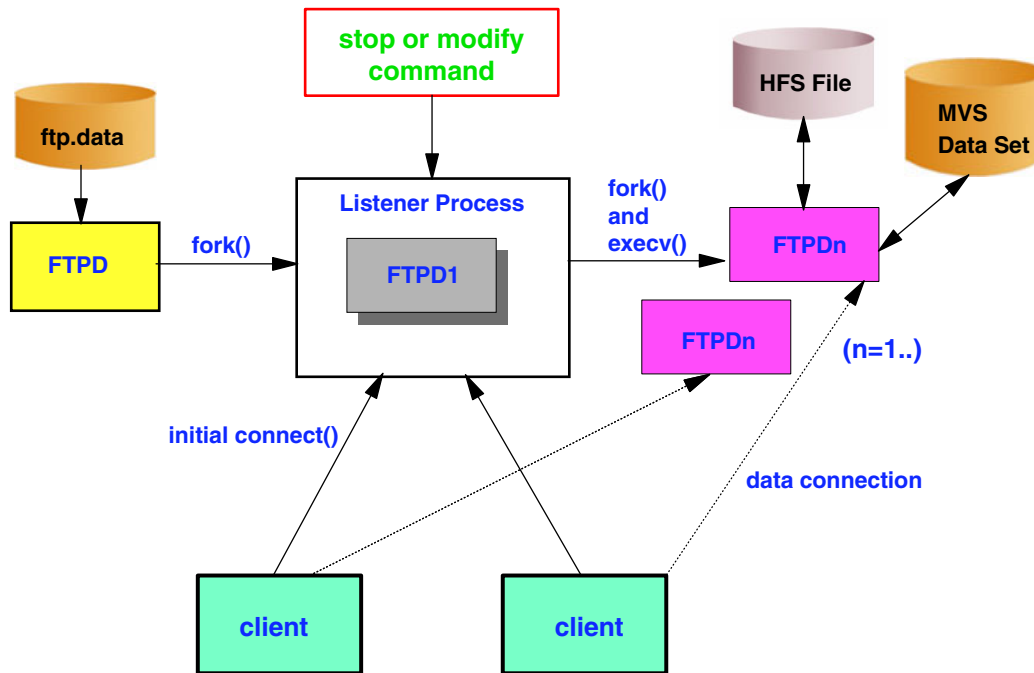
© Copyright IBM Corp. 2004. All rights reserved.

Starting a Program With Exec



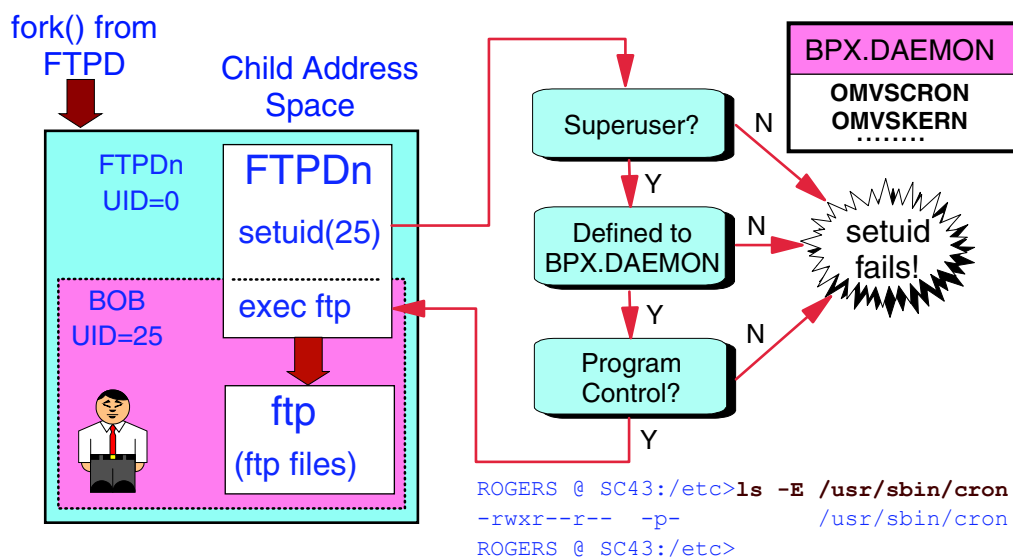
© Copyright IBM Corp. 2004. All rights reserved.

FTPD Daemon



© Copyright IBM Corp. 2004. All rights reserved.

z/OS UNIX Level Security for Daemons



© Copyright IBM Corp. 2004. All rights reserved.

Secure FTP Client Processing



- ❑ Secure FTP establishes the Secure Socket Layer (SSL) during processing of the FTP client
 - Processing includes a `setuid()` to gain the clients identity
- ❑ Normally an `exec()` would be required to force the termination of the Job Step Task to process the new `userid`'s accounting information
 - Problem is that the `exec()` destroys the SSL environment
- ❑ New `fork()` accounting logic
 - Replace `setuid()` / `exec()` with `setuid()` / `fork()`
 - `Fork()` detects new logic is triggered - `userid` from current ACEE is used to extract the account data from the user's `workattr` segment instead of propagating from parent
 - New job `exec` logic is executed to process account data for the child

© Copyright IBM Corp. 2004. All rights reserved.

Superkill Function



- ❑ USS processes that become hung and cannot be terminated via the `kill()` service
 - Requires MVS operator intervention to CANCEL the address space containing the USS process
- ❑ Using SUPERKILL
 - Cancel hung USS processes using UNIX semantics
 - Cancel their own hung processes from the shell
 - Use the enhanced console support to give operators and automated console applications additional flexibility
 - BPX1KIL/BPX4KIL - USS callable assembler service
 - `__superkill()` - C/C++ service
 - `kill -K [pid ...][job-identifier ...]` - new shell command
 - `F BPXOINIT,SUPERKILL=pid` - new console command

© Copyright IBM Corp. 2004. All rights reserved.

Superkill Example



■ The procedural flow of a superkill would be as follows:

- Send a regular KILL signal by issuing, kill -s KILL pid
- Wait 3 seconds
- Then send a superkill to force termination - kill -K pid

© Copyright IBM Corp. 2004. All rights reserved.