

**SHARE**

Technology • Connections • Results

# z/OS 1.12 UNIX System Services Latest Status and New Features

Ann Totten  
IBM Corporation

Monday, February 28, 2011: 3:00 PM-4:00 PM  
Session Number 8691

**SHARE**  
in Anaheim  
2011

# Trademarks

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

IBM\*  
Language Environment\*  
MVS  
OS/390\*  
System z10  
z/Architecture  
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.

POSIX ® is a registered Trademark of The IEEE.

IEEE is a trademark in the United States and other countries of the Institute of Electrical and Electronics Engineers, Inc.

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.

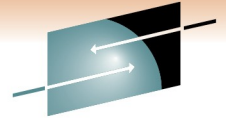
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations 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.

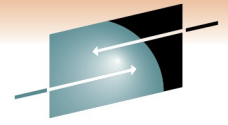


**SHARE**  
Technology • Connections • Results

# Agenda

- Availability Enhancements
- Application Enablement Enhancements
- Simplification Enhancements
- Networking Enhancements
- Miscellaneous Enhancements
- Appendix, z/OS V1R11 Enhancements

**SHARE**  
in Anaheim  
2011

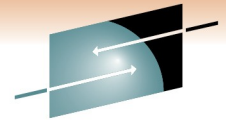


**SHARE**  
Technology • Connections • Results

# Session Objectives

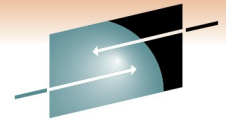
- At the completion of this session, you will be able to:
  - Identify z/OS UNIX System Services enhancements in Release 1.12
    - File System
    - Kernel
    - Shells & Utilities
  - State how the enhancements can be used to benefit your installation

**SHARE**  
in Anaheim  
2011



# Availability Enhancements

- **Dynamic Socket Limit Support:**
  - Provide capability to dynamically increase the Common INET (CINET) reserved port range.
  - Increase maximum CINET reserved ports from 4000 to 8000
  - Provide capability to allow maximum 10000 open AF\_UNIX sockets.
- **Benefit:**
  - Avoids IPL or OMVS SHUTDOWN/RESTART to increase CINET reserved port range or AF\_UNIX maximum sockets.
  - Provides larger maximum number of CINET reserved ports.
  - Default/maximum number of AF\_UNIX sockets now 10000



# Availability Enhancements

- Usage and Invocation - (CINET AF\_INET sockets):
  - SET OMVS=xx or SETOMVS RESET=xx command
    - xx is the suffix of a BPXPRMxx parmlib member.
    - BPXPRMxx contains a NETWORK statement like the following, where the INADDRANYCOUNT is greater than the current value.
      - *NETWORK DOMAINNAME(AF\_INET)  
DOMAINNUMBER(2) MAXSOCKETS(3000)  
TYPE(CINET) INADDRANYPORT(5000)  
**INADDRANYCOUNT(3000)***
      - *This defines 3000 reserved ports, starting at port # 5000*
  - The INADDRANYCOUNT can be increased any number of times up to the new maximum value of 8000.

# Availability Enhancements

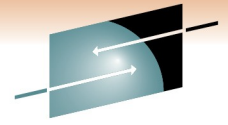
- Usage and Invocation:
  - New/ Changed External Output:
    - There is no support for decreasing INADDRANYCOUNT. If the value is less than the current value, the following informational message will be issued:

```
BPXF202I DOMAIN AF_INET WAS NOT ACTIVATED FOR FILE  
SYSTEM TYPE CINET. RETURN CODE = EINVAL, REASON  
CODE = JCannotDecrease
```

# Availability Enhancements

- Usage and Invocation - (AF\_UNIX sockets):
  - The maximum number of open AF\_UNIX sockets allowed will now be 10000, no longer requiring MAXSOCKETS on the NETWORK statement for AF\_UNIX.
  - New/ Changed External Output:
    - MAXSOCKETS will still be allowed on the NETWORK parmlib statement for AF\_UNIX, but will be ignored. If it was specified, and not 10000, the following informational message will be issued:
      - *BPXT002I THE MAXSOCKETS VALUE FOR AF\_UNIX HAS BEEN SET TO 10000*
    - No message will be issued if MAXSOCKETS was not specified, or if it was specified as 10000.

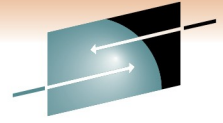




**SHARE**  
Technology • Connections • Results

# Availability Enhancements

- Migration and Coexistence Consideration:
  - MAXSOCKETS is no longer needed on the NETWORK statement for AF\_UNIX.
  - If specified, it is ignored.



# Application Enablement Enhancements

- **FILEDATA= RECORD Support:**
  - Provides capability to set, display, copy, move and save/restore the new RECORD file format for z/OS UNIX files.
  - Benefit:
    - z/OS Applications accessing z/OS UNIX files via XSAM will be able to take advantage of the RECORD format files.
    - MVS applications using files in the z/OS UNIX file system can take advantage of RECORD format files.

# Application Enablement Enhancements

- External Changes:

- BPXYFTYP (File type definitions):

- FTFFRECORD EQU 8

- BPXYVFSI (VFS interface definitions):

- #define ATTR\_FFRECORD 8 /\* Records \*/

- z/ OS UNIX REXX Update:

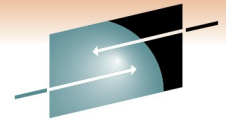
- Use S\_FFRECORD for RECORD file data type

# Application Enablement Enhancements

- **z/OS UNIX Shells and Utilities Updates:**
  - Use the `extattr` command to set the RECORD file format for a z/OS UNIX file.
    - *Example: `extattr -F REC file.txt`*
    - ***REC** - File data consists of records with prefixes. The record prefix contains the length of the record that follows.*
  - Use the `cp` command with the `-p` option to preserve the file format, or the `-F` option to set the file format.
  - Use the `mv` command with the `-F` option to set the file format, or preserve the file format value if the `-F` option is not specified.
  - Use the `ls` command with the `-H` option to display the file format.
  - Use the `pax` command with the `-x pax` option to preserve the file format value, or use the `ZOS.filefmt` value to set the file format as part of the save or restore operation.

# Application Enablement Enhancements

- **z/OS ISHELL Update:**
  - “Display File Attributes” panel displays the new RECORD file format for z/OS UNIX files
  - Using “Change File Format” panel, RECORD file format can be set for z/OS UNIX files
- **Messages:**
  - FOMF0301I Usage: extattr [+alps] [-alps] [-F NA|BIN|NL|CR|LF|CRLF|LFCR|CRNL|**REC**] file ...



**SHARE**  
Technology • Connections • Results

# Application Enablement Enhancements

- Migration and Coexistence Consideration:
  - If a file with the RECORD file format is saved to a release prior to z/OS V1R12 (using pax), then the format will not be set as part of the restore operation on that earlier release.

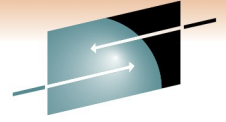
# Application Enablement Enhancements

- **New tsocmd command:**
  - tsocmd was previously provided as a “Tools and Toys” item. As of z/OS V1R12, we are now providing a fully supported tsocmd command that will be shipped as part of the z/OS UNIX Shells and Utilities component.
  - The new tsocmd command is very similar to the existing tso command except that the tso command cannot run authorized commands.
- **Benefit:**
  - tsocmd allows one to run TSO/E commands (including authorized commands) from the shell and is now supported and integrated into z/OS V1R12.

# Application Enablement Enhancements

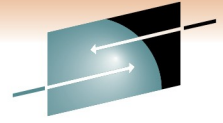
- Migration and Coexistence Considerations:
  - If the “Tools and Toys” version is not currently used or installed, then no migration actions are required.
  - If you want to preserve the “Tools and Toys” version, ensure the command is not located in /bin prior to the install of z/OS V1R12.
  - If you don’t want to preserve the “Tools and Toys” version, then remove it from it’s current location prior to the install of z/OS V1R12. You will also need to remove the “Tools and Toys” related information from the authorized load library (either prefix.TSOCMD.LOADLIB or SYS1.LINKLIB) as described in the tsocmd “Tools and Toys” README documentation





# Application Enablement Enhancements

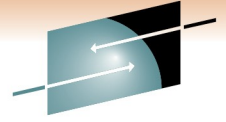
- Migration and Coexistence Considerations:
  - The new base release version of tsocmd differs from the “Tools and Toys” version of tsocmd in the following ways:
    - *Exit values are now consistently sent if the issued TSO/E command fails (previously, one might not see failure indications).*
    - *The tsoin and tsoout environment variables are no longer supported. Instead stdin and stdout are supported, as is done for most other shell commands.*
    - *The TSOPROFILE environment variable is now supported (to match the existing tso command support).*



**SHARE**  
Technology • Connections • Results

# Application Enablement Enhancements

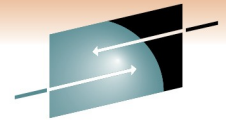
- Memory Map Support for NFS Client:
  - Provides capability to allow all hierarchical file system types (HFS, **NFS**, TFS, zFS) to be supported for mmap() using the same externals.
- Benefit:
  - NFS files can now be memory mapped as other file systems in z/OS V1R12.



**SHARE**  
Technology • Connections • Results

# Application Enablement Enhancements

- Usage and Invocation:
  - Support is transparent:
    - Existing mmap and NFS externals still apply
    - POSIX application currently using mmap() to memory map a file need not change when the file now happens to be accessed using NFS



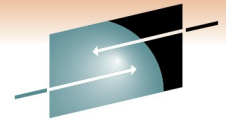
# Simplification Enhancements

- Exploit GRS Latch Identity String Support:
  - Exploit the GRS Latch interface to associate a name with the latch number for z/OS UNIX System Services latches.
    - In V1R11, GRS added a service to accept strings for latches as identities; in V1R12, z/OS UNIX added exploitation.
- Benefit:
  - Enhances the problem determination using DISPLAY GRS commands.

# Simplification Enhancements

- Example:

- SY2 D GRS,ANALYZE,LATCH,WAITER  
SY2 ISG374I 16.15.24 GRS ANALYSIS 734  
LONG WAITER ANALYSIS: ENTIRE SYSTEM  
----- LONG WAITER #1  
WAITTIME JOBNAME E/S CASID LSETNAME/LATCHID  
00:01:01 TC0 \*E\* 000E SYS.BPX.A000.FSLIT.FILESYS.LSN  
20:FS: HOST12.AJAX.DIRECTORY

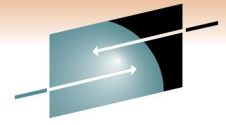


# Simplification Enhancements

- Usage and Invocation:
- Latch identity explanations for the latches used by z/OS UNIX System Services.
  - FS: <fs name>
    - If the LSETNAME is SYS.BPX.A000.FSLIT.FILESYS.LSN, the latch is used to serialize operations on the file system named in the latch identity string.
  - MOUNT
    - This latch is used by the file system to serialize operations such as file system mount, unmount, move, and automount and others.
  - MessageQ ID=<msg-ID in decimal>
    - This latch is used when the system is traversing or modifying structures related to the message queue whose identifier is shown in the latch identity string.

# Networking Enhancements

- RFC 5014 - Source Address Selection Support:
  - Provide enhancements to the z/OS UNIX System Services IPv6 socket API functions to support the RFC 5014 for source address selection.
  - Benefit:
    - Provides capability for an AF\_INET6 socket descriptor to be bound to the “best” source address based on the selection rules for the provided destination IP address.
    - Provides interfaces for an application to setup and retrieve information about its partner’s connection and security credentials.

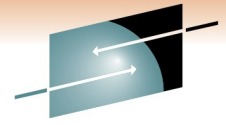


**SHARE**  
Technology • Connections • Results

# Networking Enhancements

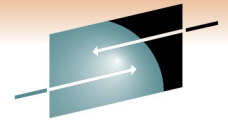
- [Using Source Address Selection Support](#), you can:
  - Bind the “best” source address for the provided destination IP address based on the selection rules
  - Validate the passed source address against a set of input source address selection preferences
  - Retrieve information about partner’s connection information and security credentials using new IOCTL interfaces
- Value:
  - The API enhancements described in RFC 5014 (IPv6 Socket API for Source Address Selection) are new standards





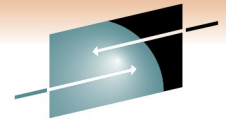
# Networking Enhancements

- New IPv6 IPPROTO\_IPV6 level option for `setsockopt()`, `getsockopt()`, `BPX1OPT`, `BPX4OPT`
  - `Sock#IPv6_ADDR_PREFERENCES EQU 32`
- `IPV6_PREFER_SRC_HOME EQU x'00000001'` Prefer home address
- `IPV6_PREFER_SRC_COA EQU x'00000002'` Prefer care of address
- `IPV6_PREFER_SRC_TMP EQU x'00000004'` Prefer temporary address
- `IPV6_PREFER_SRC_PUBLIC EQU x'00000008'` Prefer public address
- `IPV6_PREFER_SRC_CGA EQU x'00000010'` Prefer Cryptographically generated address
- `IPV6_PREFER_SRC_NONCGA EQU x'00000020'` Prefer non-cryptographically generated address



# Networking Enhancements

- Bind to Source Address Selection:
- Specifies the destination IP address and requests that the socket be bound to the best source address for the provided destination IP address
- Best source address is set by `setsockopt()`, `BPX1OPT`, `BPX4OPT` socket option  
`SOCK#IPV6_ADDR_PREFERENCES`
- New External application interface:
  - `BPX1BAS`, `BPX4BAS`, `bind2addrsel()`
  - Return Value, Return Code, and Reason Code are same as the `bind()` operation

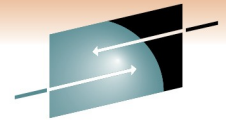


# Networking Enhancements

- Bind to Source Address Selection:

- Restrictions:

- Not supported for IPv4
  - RC=EOPNOTSUPP      RSN=JrIncorrectSocketType
- RAW sockets are not supported
- IN6ADDR\_ANY destination address not supported
  - RC=EINVAL              RSN=JrInAddrAnyNotAllowed



# Networking Enhancements

- [Validate Source Address Selection:](#)

- Validates the source address selection against the set of source address selection preference flags

- External Interface:

- `inet6_is_srcaddr(IsSrcAddrIpAddr, IsSrcAddrFlags)`

- `BPX1PCT(PC#IsSrcAddr)`

- `PC#ISSRCADDR EQU X'C0000018'`

- No Socket descriptor

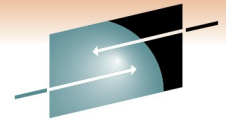
- Implemented as `vfs_pfsctl` operation

# Networking Enhancements

## • Validate Source Address Selection:

### • Assembler mapping for the PC#IsSrcAddr vfs\_pfsctl:

- ISSRCADDR DS 0F
- ISSRCADDRVER DS XL1 Version 1
- ISSRCADDRVER1 EQU 1 Version value
- DS XL3 Reserved. Must be 0.
- ISSRCADDRIPADDR DS CL(SOCK#LEN+SOCK\_SIN6#LEN)  
sockaddr\_in6
- ISSRCADDRFLAGS DS F Flags. See IPV6\_PREFER\_SRC\_\*
- DS 6F Reserved. Must be 0.
- ISSRCADDR#LEN EQU \*-ISSRCADDR Length of ISSRCADDR  
area
- Constant and mapping provided in the BPXYCONS and BPXYSOCK external macro

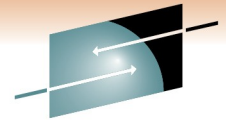


# Networking Enhancements

- [Validate Source Address Selection:](#)

- Restrictions:

- No AF\_INET6 NETWORK statement in BPXPRMxx:
  - RC=EAFNOSUPPORT RSN=JrIPv6NotEnabled
- If All AF\_INET6 TCP/IP stacks are down:
  - RC=EAGAIN RSN=JrPFSSuspend



# Networking Enhancements

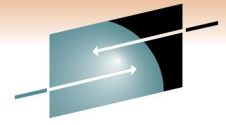
## •Set and Get Partner Information:

•SIOCGPARTNERINFO ioctl provides an interface for an application to retrieve information about its partner, including connection routing information, the user ID of the partner, or both

– SIOCGPARTNERINFO EQU X'C000F612' Get Info

•SIOCSPARTNERINFO ioctl provides an interface for an application to set up the environment that is required to retrieve the user ID of its partner using the SIOCGPARTNERINFO ioctl

– SIOCSPARTNERINFO EQU X'8004F613' Set Optimization



# Miscellaneous Enhancements

- [Drop z/OS UNIX System Services support for zFS multi-file system aggregate:](#)
  - Display OMVS, FILE command (BPXO045I message) **will not** show the zFS multi-file system aggregate name.
  - F BPXOINIT,FILESYS=DISPLAY,[ALL|FILESYSTEM=<filesystem>] command (BPXF035I message) **will not** show the zFS multi-file system aggregate name.
  - ISHELL “File System Attributes” panel **will not** show the zFS multi-file system aggregate name.
  - ISHELL **will not** have the option to detach, attach, or create zFS multi-file system aggregate.
  - ISHELL **will not** have the option to create, rename or set the quota of a zFS multi-file system aggregate.



# Miscellaneous Enhancements

- [Migration and Coexistence Consideration:](#)
  - BPX1GMN (w\_getmntent) will still continue to return the zFS multi-file system aggregate name pointed to by MNTENTAGGNAMEOFFSET
  - BPXEKDA will continue to return the zFS multi-file system aggregate name in ODMVAGGNAME
  - BPXO045I and BPXF035I messages no longer display zFS multi-file system aggregate name in z/OS V1R12

# Questions

- Any questions, concerns or comments?

## Appendix - References

- z/OS V1R12 UNIX System Services Command Reference (SA22-7802-12)
- z/OS V1R12 Migration Guide (GA22-7499-14)
- z/OS V1R12 UNIX System Services Planning (SA22-7800-14)
- z/OS V1R12 UNIX System Services Programming: Assembler Callable Services (SC28-1899-13)
- z/OS V1R12 UNIX System Services Messages and Codes (SA22-7807-11)
- z/OS V1R12.0 UNIX System Services File System Interface Reference (SA22-7808-12)

## Appendix - References

- z/OS V1R12 XL C/C++ Run-Time Library Reference (SA22-7821-12)
- z/OS TSO/E Programming Services (SA22-7789)
- “Tools and Toys” website:  
<http://www-03.ibm.com/servers/eserver/zseries/zos/unix/tools/>

# Appendix - z/OS V1R11 Availability Enhancements



- **Alternate Sysplex Root Support**
  - Provide capability to dynamically replace a **failed** sysplex root automatically or manually using an external command
  - Provide capability to dynamically replace a **failing** sysplex root manually using an external command
- **Benefit**
  - Failed or failing sysplex root can be replaced automatically or manually to improve the shared file system availability
- **Value**
  - Eliminate the single point of failure for sysplex root
  - Prevent MSOs due to single point sysplex root failures

# Appendix - z/OS V1R11 Availability Enhancements



- **Requirements:**
  - Shared file system configuration required
  - All systems in the sysplex must be at minimum z/OS V1R11
  - The UID, GID and the permission bits of the root directory in the alternate sysplex root must match the root directory in the current sysplex root
  - If the SECLABEL class is active and the MLFSOBJ option is active, then the multilevel security label for the alternate sysplex root must be identical to the assumed multilevel security label of the current sysplex root
  - All systems in the shared file system configuration must have direct access to the new file system and must be able to locally mount it

# Appendix - z/OS V1R11 Availability Enhancements



- To Setup Alternate Sysplex Root:
- Allocate the alternate sysplex root file system and setup the mount points and symlinks exactly as the current sysplex root
- Establish alternate sysplex root using new ALTROOT statement in the BPXPRMxx parmlib member:

```
ALTROOT FILESYSTEM('OMVS.ALTROOT.ZFS')  
    MOUNTPOINT('/sysalt')
```

- Will be mounted as Read-Only and AUTOMOVE=Yes
- Restriction:
  - The ALTROOT mount point must not exceed 64 characters
  - ALTROOT mount point must reside in the root directory of the current sysplex root file system

# Appendix - z/OS V1R11 Availability Enhancements



- **Requirements Contd...:**
  - The file system type for the alternate sysplex root and the current sysplex root must be either HFS or ZFS.
  - The alternate sysplex root PFS must be active on all systems in the shared file system configuration.
  - The real path name for the mount points in the current sysplex root must not exceed 64 characters in length.
  - The sysplex root or any directories in the sysplex root file system must not be exported by the DFS or SMB server.
- **Important Note:**
  - If you make changes to the current sysplex root after the alternate sysplex root has been successfully established, you must make the same changes to alternate sysplex root as well.



# Appendix - z/OS V1R11 Availability Enhancements



- To Disable Alternate Sysplex Root:
  - Unmount of alternate sysplex root file system
  - Down level system joining the shared file system configuration
  - ALTROOT NONE statement in BPXPRMxx parmlib member  
(recommended)
    - Disables alternate sysplex root support and deletes any outstanding eventual action operator messages (BPXF253E)
    - Alternate sysplex root file system is left mounted in the sysplex as a regular file system

# Appendix - z/OS V1R11 Availability Enhancements



- Alternate Sysplex Root can become inactive:
  - When ALTROOT file system is not mounted or is unmounted
  - When ALTROOT file system mount fails on some systems in the shared file system configuration
  - When ALTROOT file system becomes unowned
  - When down level system joins the shared file system configuration or in the shared file system configuration
  - When ALTROOT file system becomes the current active sysplex root

# Appendix - z/OS V1R11 Availability Enhancements



- To Display Alternate Sysplex Root:
  - D OMVS,O command will display:
    - If alternate sysplex root is **established & active** on this system:
      - *ALTROOT = OMVS.ALTROOT.ZFS*
    - If alternate sysplex root is **established & inactive** on this system:
      - *ALTROOT =*
  - F BPXOINIT,FILESYS=DISPLAY,GLOBAL command will display:
    - If alternate sysplex root is established in the sysplex:
      - *ALTROOT = OMVS.ALTROOT.ZFS*

# Appendix - z/OS V1R11 Availability Enhancements



- **Manual Replacement using MVS Operator Command:**
  - The replacement of failed or failing sysplex root can be initiated manually via an existing operator command with new option on any system in the sysplex:
    - F OMVS,NEWROOT= < new.root.filesys.name> ,COND= FORCE
  - A BPXI085D WTOR message is issued to the console to confirm the FORCE option.
  - **Important Note:**
    - This method does not use the alternate sysplex root file system that is established using the BPXPRMxx parmlib member
    - Specified new sysplex root file system must not be mounted

# Appendix - z/OS V1R11 Availability Enhancements



- **Additional Requirements:**
- User allocates the new sysplex root file system and sets up the mount points and symlinks similar to the current sysplex root
- New sysplex root must not be mounted, HSM migrated, or in use

# Appendix - z/OS V1R11 Availability Enhancements



- Migration and Coexistence Consideration:
  - Alternate sysplex root file system support requires minimum System level of z/OS V1R11
  - If a down level system (z/OSV1R10 and below) joins the shared file system configuration after successful establishment of the alternate sysplex root file system, then the established alternate sysplex root file system **is not** considered an alternate sysplex root file system until all systems in the shared file system configuration are at minimum z/OS V1R11 level

# Appendix - z/OS V1R11 Availability Enhancements



- Prerequisites for installation:

- New ALTROOT statement needs to be added to the BPXPRMxx Parmlib member:

- For Example:

```
ALTROOT FILESYSTEM('OMVS.ALTROOT.ZFS')  
        MOUNTPOINT('/sysalt')
```

- ALTROOT mountpoint needs to be created
  - There is no default ALTROOT mount point.
  - ALTROOT mount point must reside in the current sysplex root directory and must not exceed 64 characters in length
- Can be established during OMVS initialization or using SET OMVS command