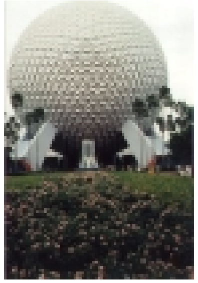
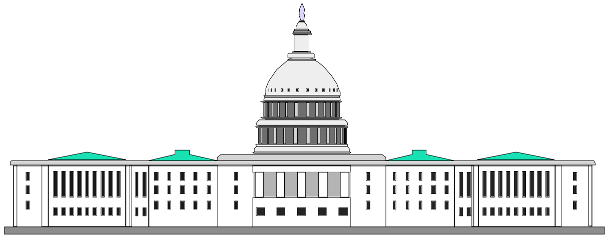


VM and VSE Technical Conference
Session # M41
Orlando, May 2000

Enhancements To VM/ESA Network File System (NFS) Support

Melissa K. Howland
Richard F. Lewis
IBM Washington System Center
ziffel@vnet.ibm.com



RETURN TO INDEX

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "AS IS" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environments.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Trademarks

The following are trademarks of International Business Machines Corporation. Those identified with an (*) are registered trademarks of International Business Machines

- Enterprise Systems Architecture/370
- Enterprise Systems Architecture/390
- ESA/370
- ESA/390
- IBM*
- System/370
- VM/ESA*
- OpenEdition
- MQSeries

The following are trademarks of the corporations identified

- Windows 95 - is a registered trademark of Mircrosoft Corporation
- PC-NFS-PRO, and JAVA - are trademarks of Sun Microsystems
- Lan Workplace - is a trademark of Novell Inc.

Agenda

IBM

- **Introduction**
- **Review of NFS enhancements in VM/ESA V2R3**
  - NFS FL310
- **Review of NFS enhancements in VM/ESA V2R4**
  - NFS FL320
- **Summary**
- **Futures**

Introduction

What Is NFS...?

- **A file access service, defined by Sun Microsystems**
  - Different from file transfer services such as FTP
  - Uses RPC for client and server communication
  - Protocol defined in RFC 1094 (V2) and RFC 1813 (V3)
- **Transparent service to clients**
  - Files on remote systems appear as local files to client
  - Client is completely insulated from characteristics of remote file system
    - Path naming conventions
    - Data format of files
    - File system organization (hierarchical or not)
    - Local system encoding scheme used
  - Same operations available on remote files as expected on local
    - Open, read, write, seek, close
    - Also directory operations; change, list, make, remove

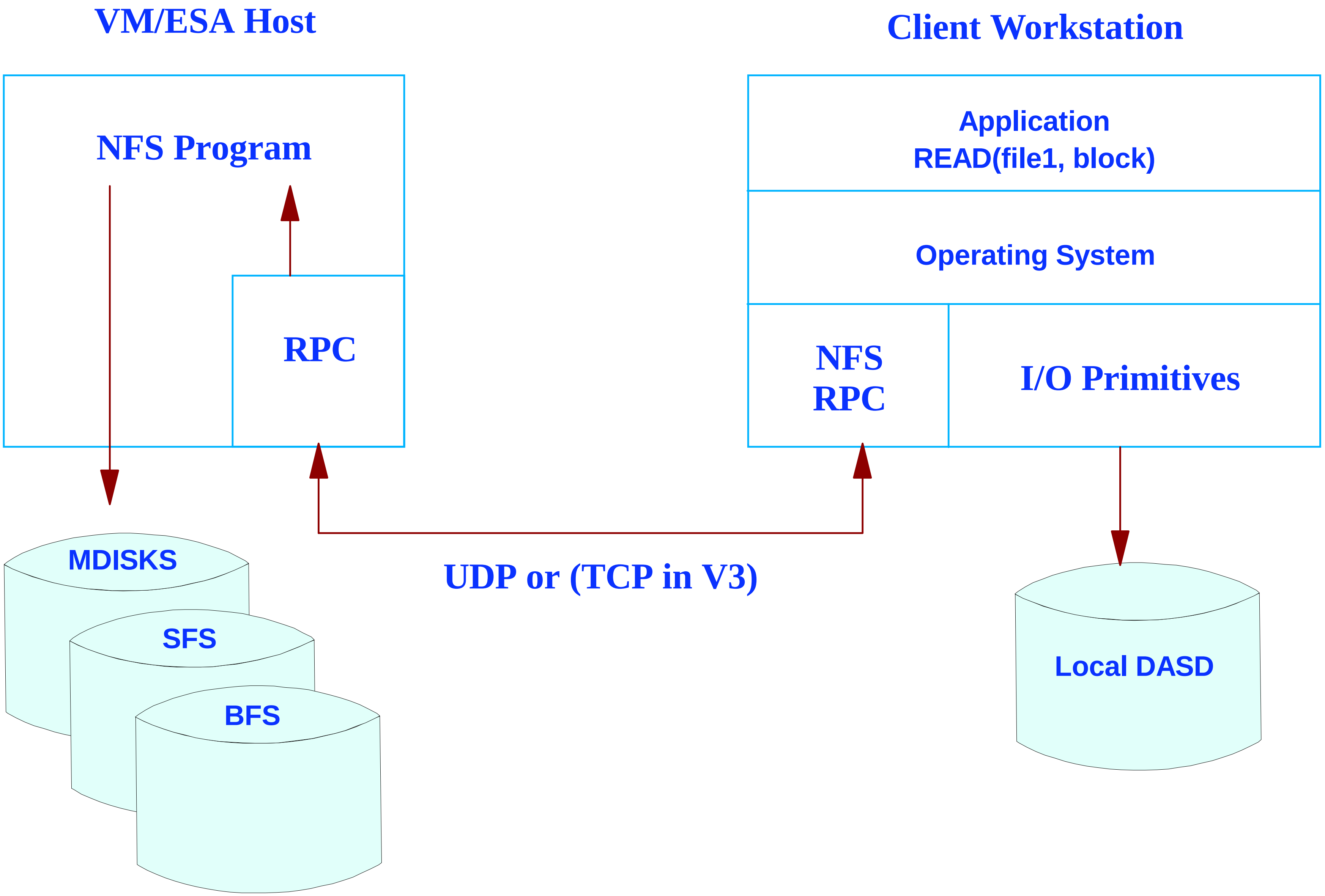
What Is NFS?

- **Design of NFS strongly influenced by UNIX file system**
  - File viewed as a sequence of bytes (unstructured, uninterpreted)
  - Files permitted to grow arbitrarily large
  - Random access allows using byte position in file as reference
  - Assumes hierarchical naming system
    - Path names resolved one component at a time
  - Honors UNIX open-read-write-close paradigm
  - Accommodates same basic file types as UNIX
    - Regular data file, directory, block special, character special, symbolic link
    - Use of device files not permitted, (remote device access)
- **Files accessed through handles rather than file descriptors**
  - 32 byte opaque pointer generated by server, stored by client
- **All file operations are synchronous**
  - File operation call only returns when server completes work

NFS Operations

- **Typically 18 procedures provide for all basic I/O operations**
  - LOOKUP - search for a file in current directory, return file handle and file attributes
  - READ and WRITE - basic read/write pimitives to access file
  - RENAME
  - REMOVE
  - MKDIR and RMDIR - create/delete subdirectories
  - GET and SET-ATTR - get or set file attributes
  - CREATE - create a file in a directory
  - READLINK - read value from a symbolic link
  - LINK / SYMLINK - create hard, and symbolic link
  - READDIR - read entries from a directory
- **MOUNT protocol handles problem of finding root directory**
  - Solves problem of restricting use of full path names in NFS protocol
    - Authentication and authorization typically performed at this point
    - RPC based protocol

NFS Operation



VM/ESA Host

NFS Program

RPC

MDISKS

SFS

BFS

UDP or (TCP in V3)

Client Workstation

Application
READ(file1, block)

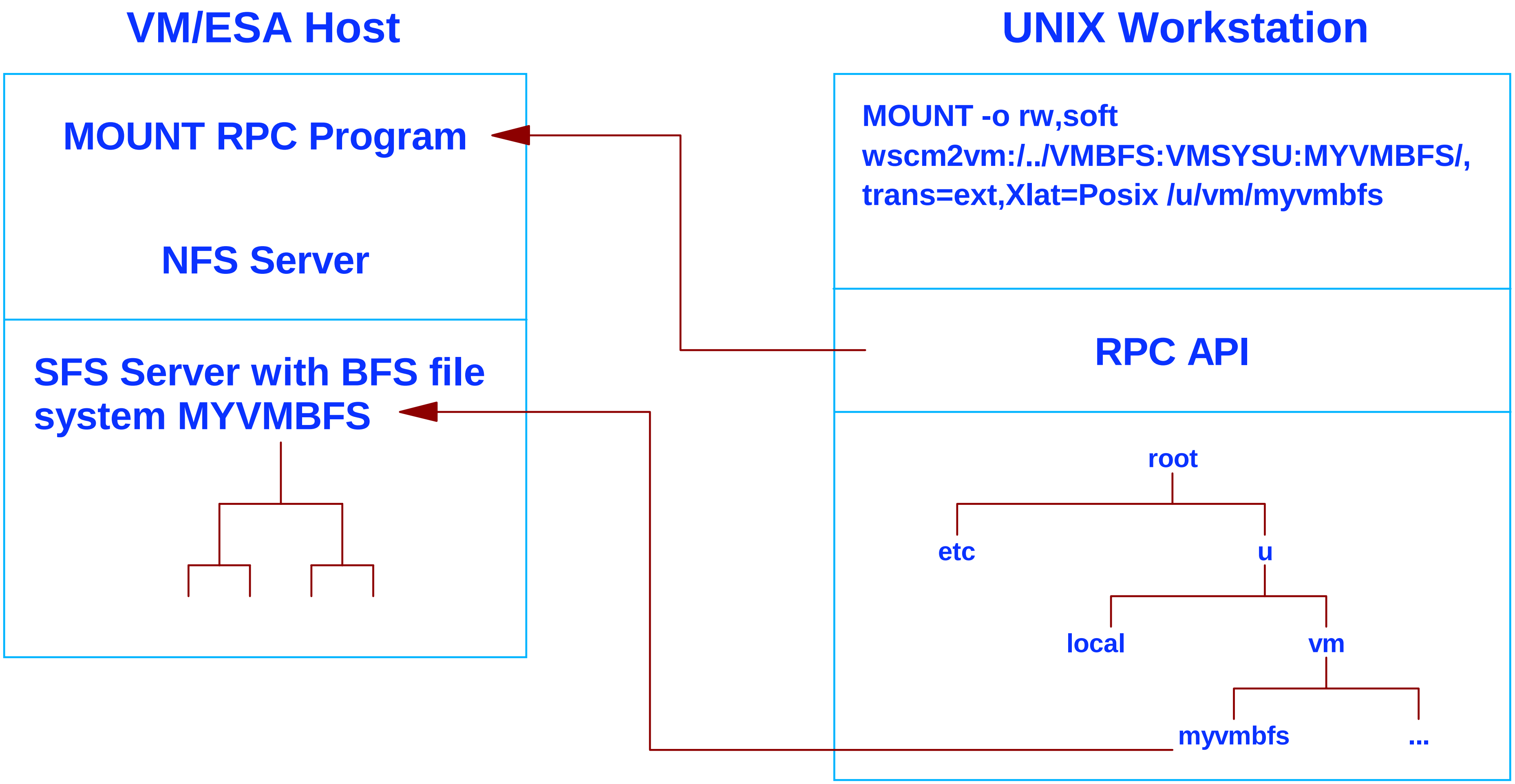
Operating System

NFS
RPC

I/O Primitives

Local DASD

MOUNT Operation



VM/ESA Host

MOUNT RPC Program

NFS Server

SFS Server with BFS file system MYVMBFS

UNIX Workstation

```
MOUNT -o rw,soft
wscm2vm:/../VMBFS:VMSYSU:MYVMBFS/,
trans=ext,Xlat=Posix /u/vm/myvmbfs
```

RPC API

PCNFSD

- **Provide PC NFS clients with authentication and network printing services**
  - Client submits userid and password for authorization by server, and receives back uid and gid for use with future server requests
- **Added to TCP/IP Level 310 NFS through APAR PQ16301 (UQ18801)**
  - Requisite PTFs: UQ14304, UQ15832, UQ18443
- **RPC based program within VMNFS service machine**
  - Uses RPC program number 150001
  - Version 1 and Version 2 PCNFS user authentication implemented
  - Remote spooling and printing routines not implemented

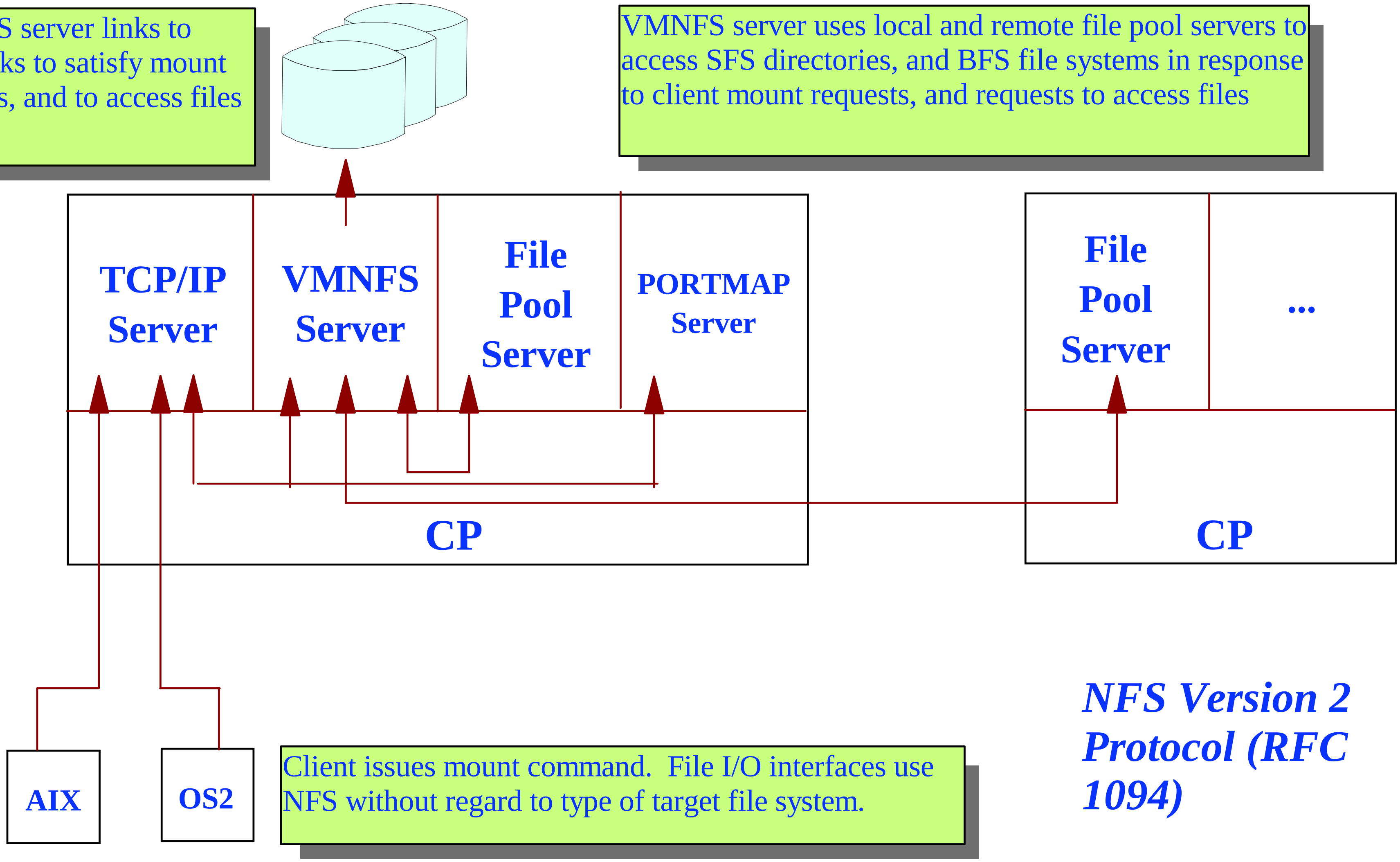
MOUNTPW

- **Client side command designed to overcome potential security problems with mount command**
  - Password in mount command may be revealed on a query of mounts
- **Provides alternate way to send password, account, and userid information to VM NFS server**
- **Precedes related MOUNT command**
  - Server will wait 5 minutes after which a new MOUNTPW must be supplied for a subsequent MOUNT command
- **C source available on VM NFS feature tape**
- **Executables supplied for**
  - AIX RS/6000
  - OS/2

NFS on VM/ESA...

VMNFS server links to minidisks to satisfy mount requests, and to access files

VMNFS server uses local and remote file pool servers to access SFS directories, and BFS file systems in response to client mount requests, and requests to access files



Client issues mount command. File I/O interfaces use NFS without regard to type of target file system.

NFS Version 2 Protocol (RFC 1094)

NFS Enhancements in VM/ESA V2R3

Packaging and Ordering Enhancements

- **TCP/IP is now a feature of VM/ESA V2R3**
  - Level identified by function level number similar to CMS (FL310)
- **No price changes**
- **Not available separately**
- **Separate service stream**
- **Included with DDR in disabled state**
  - Base feature
  - NFS feature
  - WT Kerberos
  - All preinstalled and ready to use
  - Enable with SET PRODUCT or VMFINS ENABLE commands
    - Base TCP/IP and NFS feature each have separate charge and enablement requirement
- **Separate media - Source, Offload, Kerberos DES**

NFS Enhancements in FL310

IBM

- Support was added for SFS and BFS
- Installation enhancements
- PCNFSD support added (via APAR PQ16183)
- Greater flexibility in specification of translation and record processing options
- Enhancements to SMSG support for error information display
- Introduction of a config file (VMNFS CONFIG)
- Restrict anonymous access for BFS and SFS (ANONYMOUS NO)
- VMNFS MODULE can run above 16MB line
- NFS Protocol support still at V2

Enhancements To TCP/IP Server Definition

DTCPARMS File:

- ➤ Userid
- ➤ Nodeid
- ➤ SYSTEM
- ➤ IBM

Common PROFILE EXECs

Exits available with well defined interface:

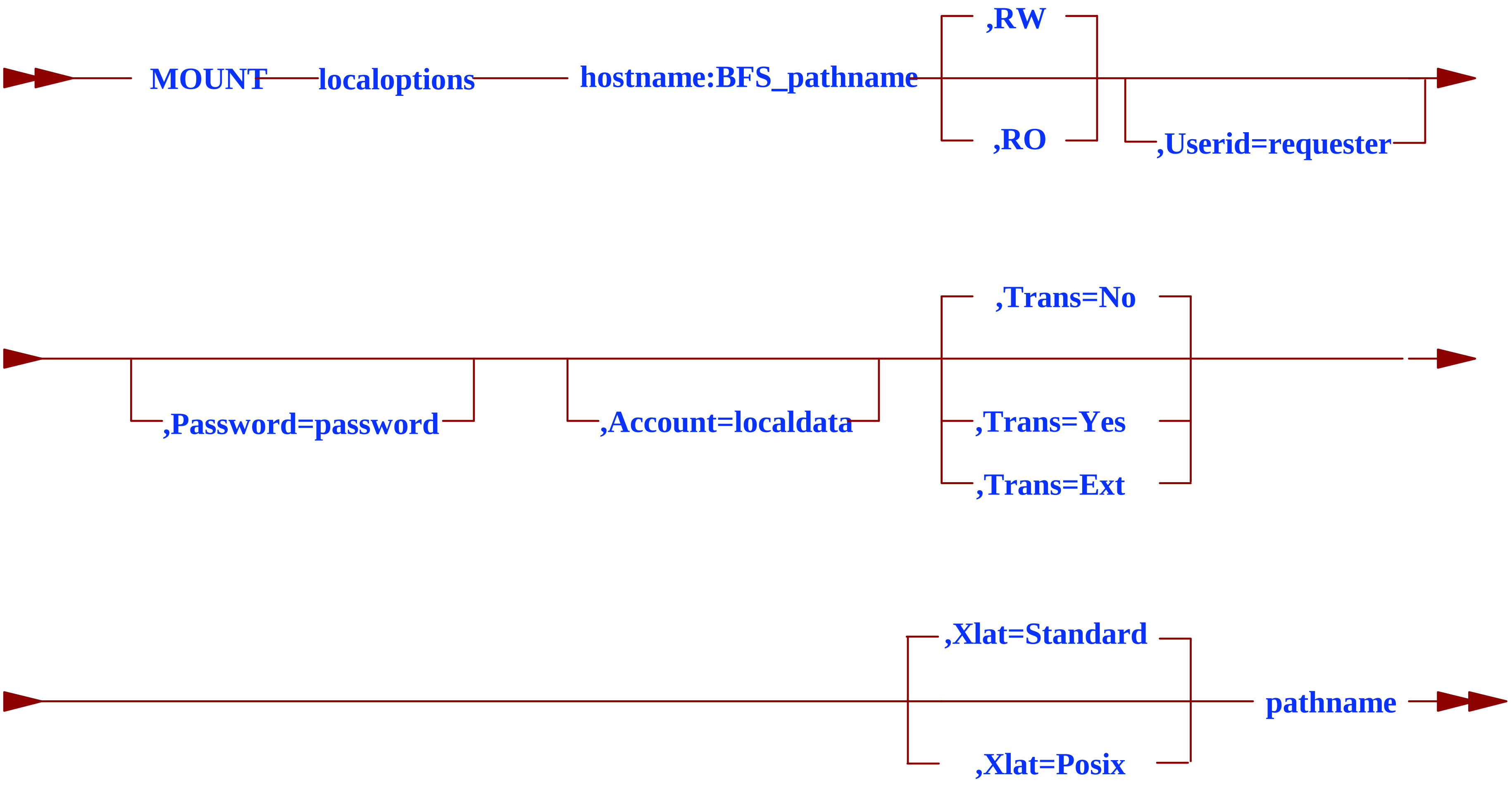
- ➤ Global (TCPRUNXT)
- ➤ Server (:Exit. tag)

```
.****************************************** Server Definitions
:nick.TCPIP     :type.server   :class.stack
:nick.FTPSERVE  :type.server   :class.ftp
:nick.FTPSRV2   :type.server   :class.ftp

.****************************** Application Class Definitions
.* TCP/IP Protocol Stack
:nick.stack   :type.class
              :name.TCP/IP Stack
              :command.TCPIP
              :runtime.PASCAL
              :diskwarn.YES

              :authlog.AUTHLOG FILE A

.* File Transfer Protocol (FTP) daemon
:nick.ftp     :type.class
              :name.FTP daemon
              :command.SRVRFTP
              :runtime.PASCAL
              :diskwarn.YES

              :anonymous.NO

              :ESM_Enable.NO
              :ESM_Validate.RPIVAL
              :ESM_Racroute.RPIUCMS
```

Anonymous Access

- **Anonymous access**
  - Enabled through server startup option ’N’ or DTCPARMS file
    - :ANONYMOUS.YES
  - Client
    - Should specify userid ANONYMOU or
    - No userid and password
  - Server sets userid for SFS access to ANONYMOU and UID/GID to -1 for BFS access
  - Default for server is to disallow anonymous access

BFS MOUNT Command



MOUNT localoptions hostname:BFS_pathname

,RW

,RO

,Userid=requester

,Password=password

,Account=localdata

,Trans=No

,Trans=Yes

,Trans=Ext

,Xlat=Standard

,Xlat=Posix

pathname

BFS Pathname Specification

IBM

- Fully qualified name of directory that contains files and subdirectories to be accessed by client
- Use two commas in a row to indicate a comma in name
- Slashes may be used in places of colons required before file pool ID and file space ID to accommodate client system restrictions

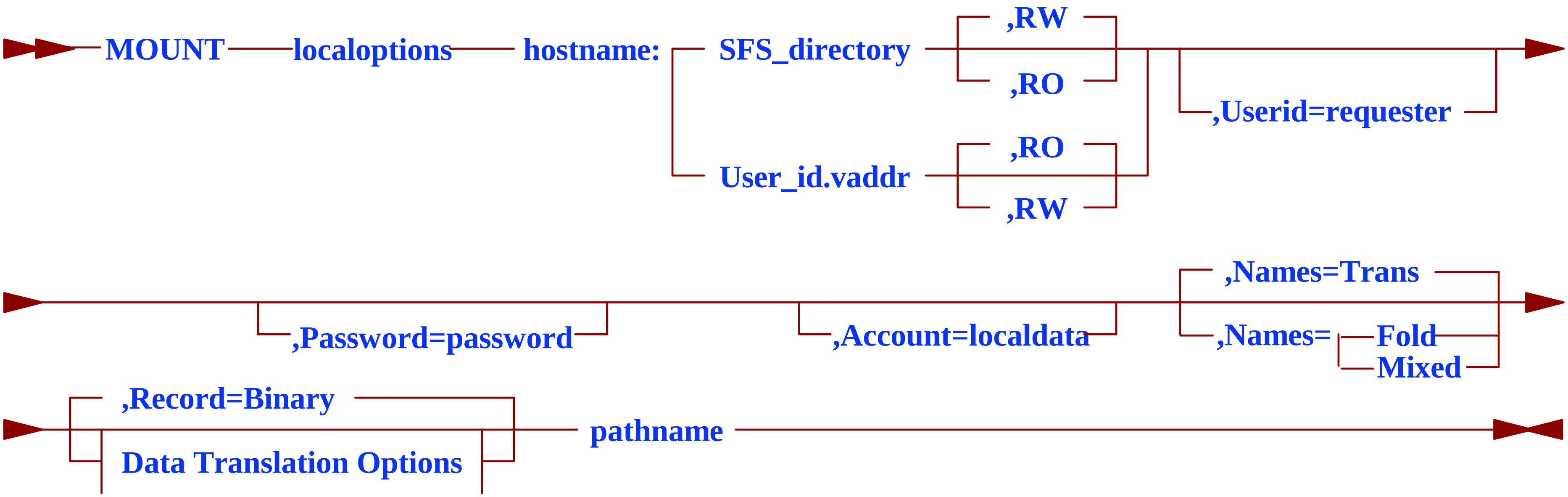
```
mount -u0 -g0 x: gdlvm7:/../VMBFS:FBCOOL:ROOT/u/jake,userid=elwood
```

BFS Mount Notes

- **Trans=**
  - Defines whether translation should occur for data in files
  - Ext
    - EBCDIC/ASCII translation done based on value of the file extension
    - Uses contents of VMNFS SCONFIG
  - Yes
    - EBCDIC/ASCII translation is always performed
  - No
    - EBCDIC/ASCII translation is never performed
- **Xlat=**
  - Defines translation table to use for file data translation
  - Standard - standard TCP/IP translation table (STANDARD TCPXLBIN)
  - Posix - ASCII (ISO8859-1) to/from EBCDIC (IBM-1047
    - Unix line feed X’0A’ translated to VM new line character X’15’
- **Userid and Password used to obtain UID and GID for authorization checking of requests**

SFS/Minidisk MOUNT Command

IBM



MOUNT localoptions hostname: SFS_directory ,RW ,RO ,Userid=requester

User_id.vaddr ,RO ,RW

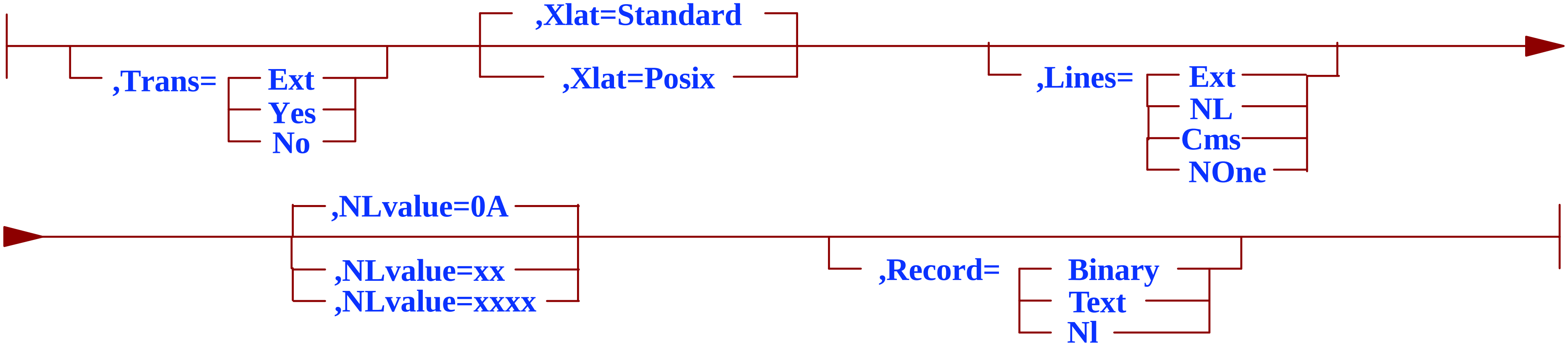
,Names=Trans

,Password=password ,Account=localdata ,Names= Fold Mixed

,Record=Binary

Data Translation Options pathname

Data Translation Options:



,Xlat=Standard

,Trans= Ext Yes No ,Xlat=Posix ,Lines= Ext NL Cms NOne

,NLvalue=0A

,NLvalue=xx ,NLvalue=xxxx ,Record= Binary Text Nl

SFS Pathname Specification

- **Specify filepool directory containing files and subdirectories to be accessed by client**
- **Specify fully qualified name**
  - File space name can default to userid
- **Slash may be substituted for colon, and period in subdirectory names to accommodate client system limitations**

```
mount -v x: gdlvm7:fpcool:jake.mission,trans=ext,lines=ext,userid=elwood
```

- **MOUNTPW command supports SFS and BFS pathnames, in addition to minidisk userid.vaddr**

SFS/Minidisk Mount Notes...

- **Trans= and Xlat= options function same as noted for BFS**
- **Lines=**
  - Defines way CMS records converted to NFS byte stream
  - Ext
    - Conversion performed based on value of file extension
    - Uses VMNFS SCONFIG table
  - Nl
    - New line (nl) characters inserted/removed at CMS record boundaries
    - See description of nlvalue= option
  - CMS
    - CMS file format is maintained
    - Length field at beginning of variable length records visible to client
    - Client must supply length field for variable length records created
  - None
    - No line end characters are inserted
    - Length field for variable length records not visible to client

SFS/Minidisk Mount Notes

- **Nlvalue=**
  - Specify hex characters to use as boundary indicating end of CMS record
    - Default is X’0A’
    - Specify 2 or 4 hex digits (E.g. 0D0A)
- **Record= option supported for compatibility**
  - Binary equivalent to trans=no, lines=CMS
  - Text equivalent to trans=yes, lines=CMS
  - Nl equivalent to trans=yes, lines=nl
  - If VMNFS CONFIG not found and trans=ext, and/or lines=ext specified
    - Default to Record=Binary behavior
- **RW is default for mount of SFS directory**
- **Userid and password supplied used for access control of SFS directories and files**

NFS Enhancements in V2R4

NFS Version 3 Support...

- **Highlights:**
  - Support for larger files and file systems
    - Many values used in protocol extended to 64 bit integers
    - READDIR/READDIRPLUS cookies, READ/WRITE offsets ...
  - Performance improvements
    - File attributes now included in every response - eliminating additional calls to obtain attributes following operations
    - E.g. GETATTR call after WRITE
    - Increased packet sizes - not limited to 8K bytes - allow for larger transfer sizes
    - Asynchronous WRITE support:
    - Server acknowledges write before data written to stable storage
    - Client issues COMMIT to flush data from previous writes to stable storage
    - Only implemented with BFS, minidisk and SFS always synchronous
    - COMMIT of selected parts of file not supported, entire file always flushed
    - New commands such as READDIRPLUS returning more data

NFS Version 3 Support...

- **Highlights: (cont.)**
  - TCP client support
    - NFS Version 2 only supported UDP
    - More efficient on low bandwidth, high latency networks such as Internet or company WAN
    - Connection oriented
    - Reliable
    - Client no longer must perform time-out and retransmission functions
    - Server does not need to incorporate logic to guard against duplicate packets
    - Not restricted to version 3 clients, supported for:
    - NFS version 2 clients
    - Mount version 1 and version 3 protocols
    - Use rpcinfo command to see what protocols are available on a system

NFS Version 3 Support...

- **Sample rpcinfo command output**
  - Command rpcinfo -p host_name

```
program vers proto   port
 100000    2   udp    111  portmapper
 100000    2   tcp    111  portmapper
 100005    1   udp   2049  mountd
 100005    3   udp   2049  mountd
 100005    1   tcp   2049  mountd
 100005    3   tcp   2049  mountd
 100003    2   udp   2049  nfs
 100003    3   udp   2049  nfs
 100003    2   tcp   2049  nfs
 100003    3   tcp   2049  nfs
 150001    1   udp   2049  pcnfsd
 150001    2   udp   2049  pcnfsd
```

NFS Version 3 Support...

IBM

- **Highlights: (cont.)**
  - More extensive error codes and responses returned to client
    - Instead of just NFSERR_IO
  - Ability to create special files through new MKNOD procedure
    - Support limited to BFS
    - Character special files (file type CHR)
    - FIFO file types (type FIFO)
    - No support yet for block special files (BLK)
    - SOCK special file not supported - VMNFS uses TCP/IP sockets, instead of OpenEdition for VM C Socket Library

System Programming Considerations...

- **PROFILE TCPIP**
  - Second PORT statement needs to be included for NFS
    - 2049 TCP VMNFS NOAUTOLOG
    - Specify same port number as UDP entry
- **DTCPARMS file**
  - New startup options
    - U - turn off TCP support, use UDP transport protocol only
    - V - handle only NFS Version 2 requests
  - Anonymous startup option now fully supported for minidisks
- **VMNFS HISTORY**
  - Recommend erase before starting new level of NFS
  - Format of history file entry changed

System Programming Considerations...

IBM

- **VMNFS CONFIG file**
  - DUMPMOUNT YES|NO
    - Default is YES
    - Enables support of MNTPROC_DUMP client request
    - Returns information about all resources actively in use by VMNFS server
    - SFS and BFS directories considered active if used within last 15 minutes
    - Minidisk considered active while VMNFS server has it linked
    - Returns IP address of client and name of object mounted (directory name or userid.vaddr)
    - Returns PROC_UNAVAIL error to client if DUMPMOUNT NO specified

System Programming Considerations...

- **VMNFS CONFIG file (cont.)**
  - EXPORT export_name mount_string
    - Each statement defines entry to be added to export list
    - Export list obtained by client using MNTPROC_EXPORT procedure
    - export_name represents symbolic name client can use to MOUNT file system
    - export_name can be a proper mount command or an alias name
    - mount_string represents file system to be mounted along with all mount options to be substituted when server receives request to MOUNT export_name
    - Client does not see mount_string
    - mount_string must be syntactically valid mount command
    - Following keywords can be used on mount_string
    - %USERID - server substitutes userid from MOUNTPW or PCNFSD
    - %FSROOT - server substitutes contents of FSROOT from POSIXINFO directory statement of MOUNTPW or PCNFSD userid
    - %IWDIR - server substitutes contents of IWDIR from POSIXINFO directory statement of MOUNTPW or PCNFSD userid
    - Consider client display considerations when constructing export_name
    - Userid ANONYMOU substituted into request for anonymous mounts

System Programming Considerations...

IBM

- **Export statement examples:**

```
EXPORT /tools             /../VMBFS:VMSYS:ROOT/bin,ro,trans=no
EXPORT /PC/my/191/disk    %USERID.191,ro,lines=nl,nlvalue=0D0A,trans=yes
EXPORT /my/BFS/home       %FSROOT%IWDIRID,trans=yes,xlate=posix
EXPORT /PC/my/sfs/dir     WSCVMU:%USERID.,lines=nl,trans=yes
```

System Programming Considerations...

- **VMNFS CONFIG file (cont.)**
  - EXPORTONLY NO|YES
    - Restrict file systems clients can mount to ones listed in export records
    - No is default, allowing clients to mount anything listed in export records as well as any other file system they have authority to
    - Value of YES with null export list results in:
    - Error message during server startup, and server shutdown
    - Rejection of new mount requests if null export list is result of refreshing config file - use of previously mounted file system unaffected
    - Warning message indicating server is in this state
  - MAXTCPUSERS 50|some_value
    - Maximum concurrent number of clients using TCP protocol
    - 50 is default
    - Change to value requires stop and start of server
    - Clients exceeding limit receive error connection refused or not allowed
    - Start with large number and tune downward
    - SKCBPOOLSIZE value in stack should dynamically increase with requirements

System Programming Considerations...

IBM

- **SMSG command enhancements**
  - REFRESH VMNFS CONFIG
    - Causes server to reread VMNFS CONFIG file and refresh configuration parameters as is done when server starts
    - Does not reinitialize sockets or reset maxtcpusers
    - Dynamically change server configuration while server is running
  - QUERY RESOURCE
    - Display information about resources actively in use
    - Information displayed (by userid, IP address, anonymous or for all)
    - Whether mount is R/W or R/O
    - Mount indicator (explicit mount, or implicit mount after server restart)
    - IP address of client requesting mount
    - VM userid under which mount requested
    - Name of object (BFS/SFS directory or minidisk)
    - Optionally NFS request counts (separated by V2 and V3 counts if desired)

System Programming Considerations...

- **SMSG command enhancements...**
  - QUERY CONFIG - display current configuration information for server
    - Maximum number of clients allowed to use TCP protocol concurrently
    - Current number of clients using TCP protocol
    - Maximum buffer size for V3 read requests
    - Maximum buffer size for V3 write requests
    - NFS Service Level (FL320)
  - VMNFSSMG EXEC - new exit to restrict SMSG command use
    - Input parameters - userid, nodeid, and smsg text from requester
    - Output - RC 0 means request OK, and other RC, request prohibited
- **Changes to VMNFSMON EXEC**
  - Input parameters now include a logonby userid if one specified on MOUNT request
  - Link password is only password included in input parms
  - See Planning and Customization Guide for more details

System Programming Considerations...

IBM

- **Enhancements to translate table support**
  - Motivated by need to support new Euro currency symbol
  - MOUNT command allows for specification of translate table name other than standard and posix
    - Translate table must be available in VMNFS server search order
    - Maximum 255 translate tables can be managed by VMNFS
    - File type must be TCPXLBIN
  - New file created on VMNFS 191 - VMNFS TRANSLAT
    - Tied to VMNFS HISTORY file
    - Format is one 2048 byte record
    - Lists table names in use by clients
    - Enables mapping a one byte code in file handle to a translation table name
    - Erasing VMNFS TRANSLAT causes refresh of VMNFS HISTORY
  - APAR PQ19040 implements support in prior releases of TCP/IP

System Programming Considerations...

- Query response examples:

```
smsg vmnfs m q resource user maryelln
 M Client resource information for user ID MARYELLN.
 M rw m 9.130.57.81      MARYELLN  VMSYS15:MARYELLN.
 M rw m 9.130.57.81      MARYELLN  /../VMBFS:VMSYS15:MEVBFS/
 M ro m 9.130.57.81      MARYELLN  MARYELLN:191
```

```
smsg vmnfs m q resource user maryelln details
 M Client resource information for user ID MARYELLN.
 M rw m 9.130.57.81      MARYELLN  VMSYS15:MARYELLN.
 M Mount count 1
 M 0 null, 0 getattr, 0 setattr, 2 lookup, 0 read, 0 write
 M 1 create, 0 remove, 0 rename, 0 link, 2 readdir, 2 statfs
 M 0 mkdir, 0 rmdir, 0 symlink, 0 readlink, 0 access, 0 mknod
 M 2 readdir+, 0 fsstat, 1 fsinfo, 0 pathconf, 0 commit
 M
 M rw m 9.130.57.81      MARYELLN  /../VMBFS:VMSYS15:MEVBFS/
 M Mount count 1
 M 0 null, 0 getattr, 0 setattr, 2 lookup, 0 read, 0 write
 M 1 create, 0 remove, 0 rename, 0 link, 2 readdir, 2 statfs
 M 0 mkdir, 0 rmdir, 0 symlink, 0 readlink, 0 access, 0 mknod
 M 2 readdir+, 0 fsstat, 1 fsinfo, 0 pathconf, 0 commit
 M
 M ro m 9.130.57.81      MARYELLN  MARYELLN:191
 M Mount count 1
 M 0 null, 16 getattr, 4 setattr, 37 lookup, 16 read, 0 write
 M 1 create, 0 remove, 0 rename, 0 link, 21 readdir, 2 statfs
 M 0 mkdir, 0 rmdir, 0 symlink, 0 readlink, 11 access, 0 mknod
 M 22 readdir+, 0 fsstat, 1 fsinfo, 0 pathconf, 0 commit
```

End User Considerations...

- **Enhancement to BFS MOUNT command**
  - New option BY=
    - Specify agent or logonby userid
    - Userid specified in USERID= option is host or target userid
    - Password specified is logon password for BY= userid
    - Server will verify that agent userid has logonby authority for target userid
- **Enhancement to SFS and minidisk MOUNT command**
  - New option BY=
  - New option MDISKPW=
    - Specify minidisk password to be used by server to link to minidisk
    - Use when an ESM is not active
    - If USERID=, BY=, and MDISKPW= not specified, server will use PASSWORD= value as minidisk link password

End User Considerations...

- **Enhancement to MOUNTPW command**
  - Specification of BY= userid
  - Specification of MDISKPW= mdiskpw
  - Executable versions of MOUNTPW available for AIX, OS/2
    - Source provided for compilation on other platforms
- **PCNFSD consideration**
  - If MOUNT command includes BY= userid, password for userid will be taken from PCNFSD information supplied prior to MOUNT
- **TCP transport protocol**
  - May need to tailor client to use TCP protocol instead of UDP
- **Status values**
  - NFS Version 3 generates larger number of status values to client

End User Considerations...

- **Enhancements to SMSG QUERY available to end users**
  - Must be logged on to VM host system
  - Resource information only available if requesting vm userid has authorization for the particular mounted directory (BFS or SFS)
- **MOUNT command Xlat option updated**
  - May now specify a tablename other than Standard or Posix
  - Part of TCP/IP code page enhancements for Euro currency symbol
    - New table names may consist of hhhhwwww
    - hhhh = host number used such as 1047
    - www = work station code page number such as 0819
    - Thus Xlat=10470819
    - Customized table names also valid
    - E.g. Xlat=french
    - Filetype always TCPXLBIN
  - Maximum of 255 translation tables can be used by server

Clients

- **PCNFSD available on VM/ESA with APAR PQ16301**
- **Windows 95**
  - IBM eNetwork Communications Suite V1.1
    - FTP Software Interlink Client (NFS client) is included in package
  - Sun PC-NFS-PRO windows only version with 32-bit TCP/IP stack support
  - Novell LAN Workplace Pro 32-bit product providing NFS client services
- **OS/2**
  - TCP/IP V2.0 for OS/2 NFS Client
    - NFS Kit
    - See http://www.networking.ibm.com/tcp/tcpprod.html
- **UNIX**
  - NFS support part of built in TCP/IP support integrated into most UNIX implementations

Summary

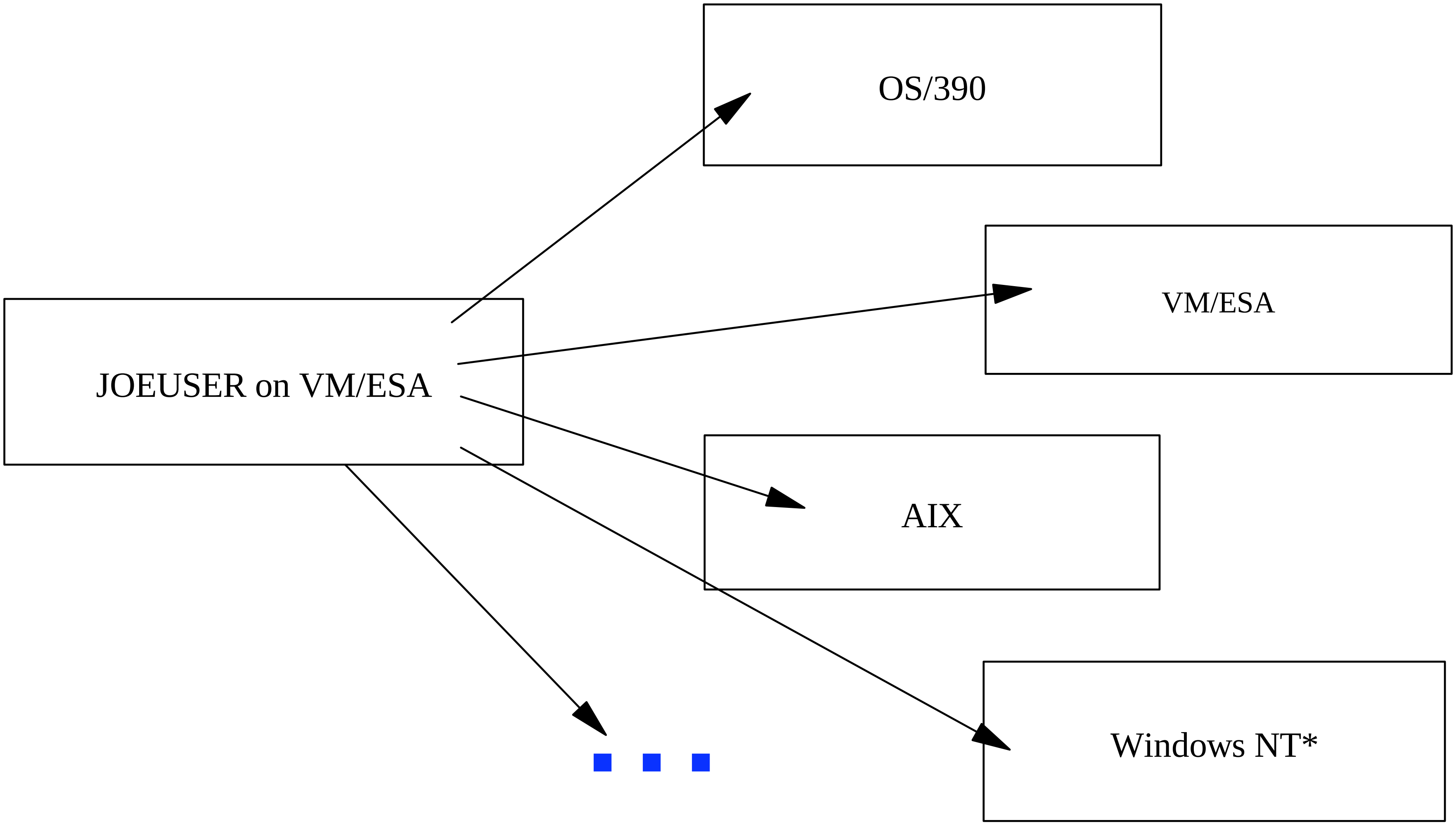
Summary

- **Significant enhancements have been made to NFS**
  - Support for Byte File System, and Shared File System
  - Integrated into VM/ESA V2R3 package as optional feature
  - Preinstalled on VM/ESA V2R3 DDR
    - Requires enablement step when licensed
  - PCNFSD support
  - Performance improvements
  - Installation process enhancements
  - Better control over file translation through new file extension table
- **Significant enhancements coming with VM/ESA V2R4**
  - NFS Version 3 Support
    - Better performance, support for TCP protocol
  - Support for export and dump requests from clients
  - More enhancements to SMSG
  - LogonBy support

TCP/IP and NFS home pages:
http://www.vm.ibm.com/related/tcpip/
http://www.vm.ibm.com/nfs/

Futures

Proposed NFS Client Support



*Windows NT is a trademark of the Microsoft Corporation

MOUNT

- **OPENVM Mount command syntax**

  openvm mount /../NFS:hostname:directoryname,serveropts /../VMBFS:SERVBFS:MARYELLN/remotedir (cmsopts

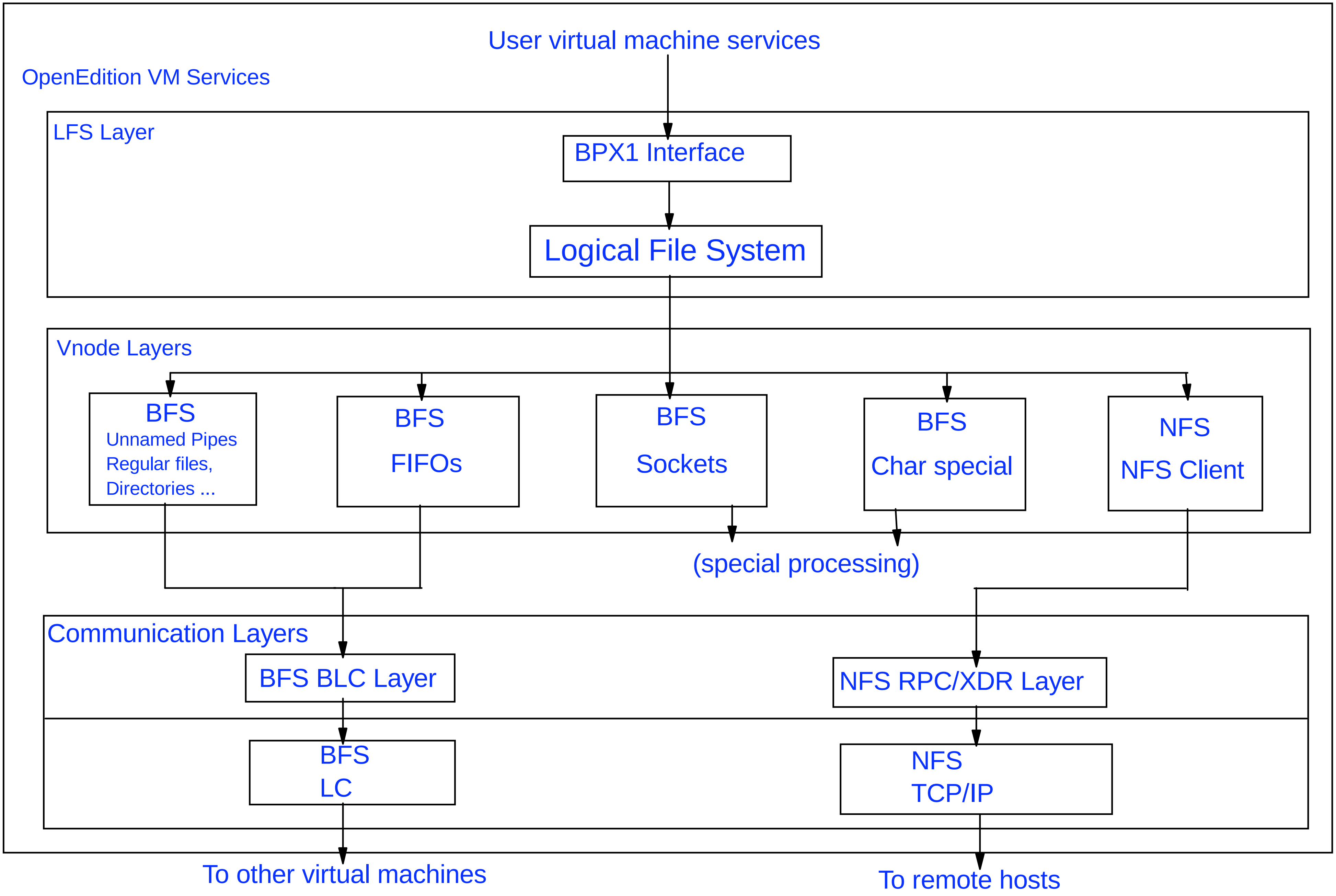
- *serveropts* contain information to be passed to the NFS server at *hostname*.

- New *cmsopts* specify local options such as
  - remote user ID and password
  - ASCII-EBCDIC translation tables
  - retry and timeout values

Interfaces

- **Interfaces available to the CMS user:**
  - CMS OPENVM commands
  - XEDIT
  - CMS Pipelines
  - OpenEdition Shell
  - REXX
  - OpenEdition Callable Services

NFS Client Structure



User virtual machine services

OpenEdition VM Services

LFS Layer

BPX1 Interface

Logical File System

Vnode Layers

BFS
Unnamed Pipes
Regular files,
Directories ...

BFS

FIFOs

BFS

Sockets

BFS

Char special

NFS

NFS Client

(special processing)

Communication Layers

BFS BLC Layer

NFS RPC/XDR Layer

BFS
LC

NFS
TCP/IP

To other virtual machines

To remote hosts

NFS Client Support Summary

IBM

- **Use of a MOUNT command or API will give CMS users access to:**
  - OS/390 data
  - AIX directories
  - Linux
  - Windows* data
  - Remote CMS minidisks
  - Remote SFS and BFS directories in VMSYS: file pools

*Windows is a trademark of the Microsoft Corporation