Implementing Alternate Data Streams in Likewise Storage Services

Wei Fu

<wfu@likewise.com>

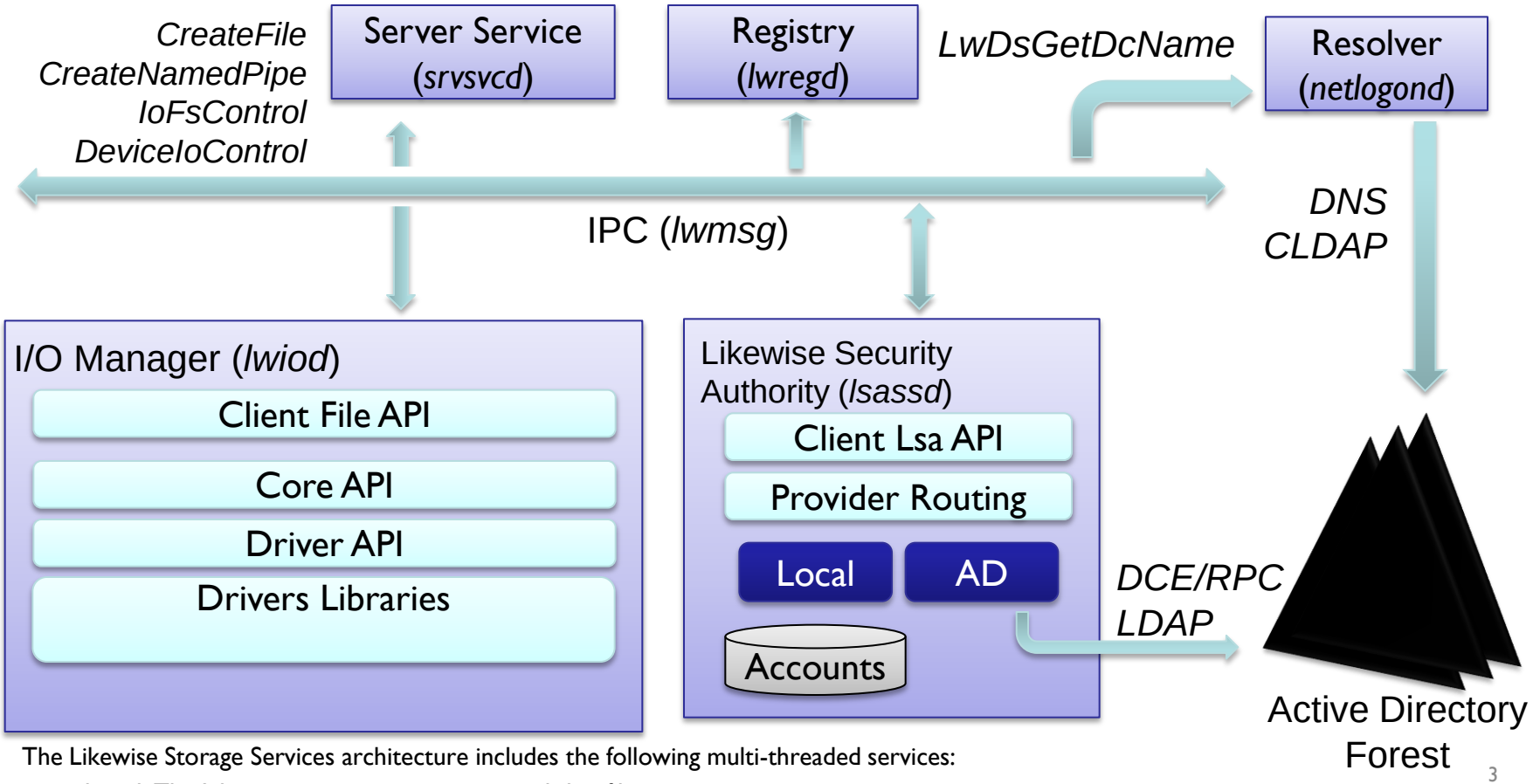
Software Engineer

Likewise Software

Outline

- Introduction to Likewise Storage Services
- What is an ADS (Alternative Data Stream)?
- ADS Data Model
- ADS On-Disk storage Model
- Implementation "Adventures"
- Demonstration of ADS support in Likewise Storage

Likewise Storage Services Architectural Overview

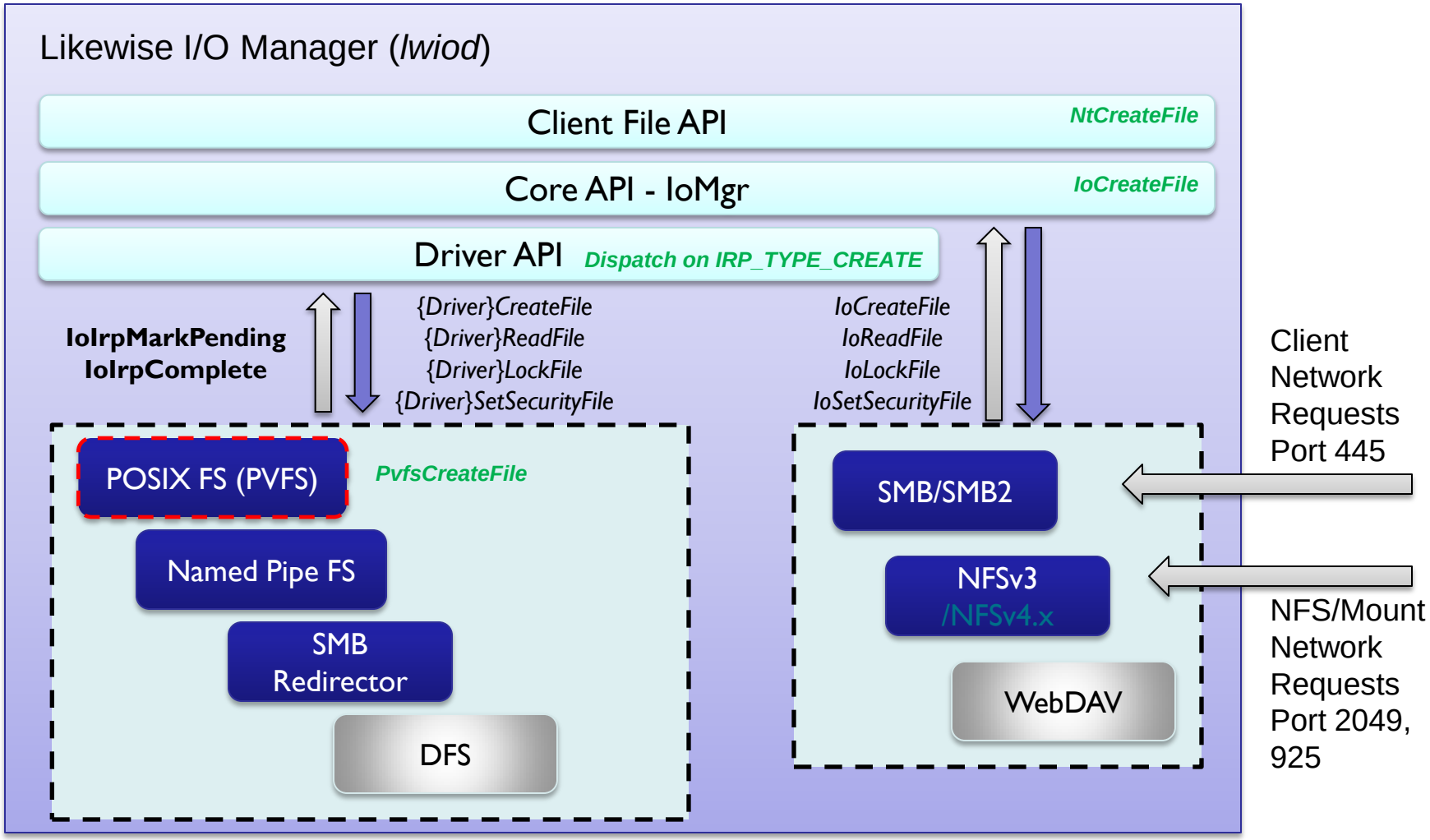


The Likewise Storage Services architecture includes the following multi-threaded services:

- lwiod: The Likewise input-output service and the file server.
- lsassd: The Likewise security and authentication subsystem that forms the core of the Likewise Identity Service, or LWIS.
- srvsvcd: Server and workstation RPC services (\srvsvc)
- netlogond: The domain controller locator and affinity manager.
- lwregd: The Likewise registry.

3

Likewise I/O Manager



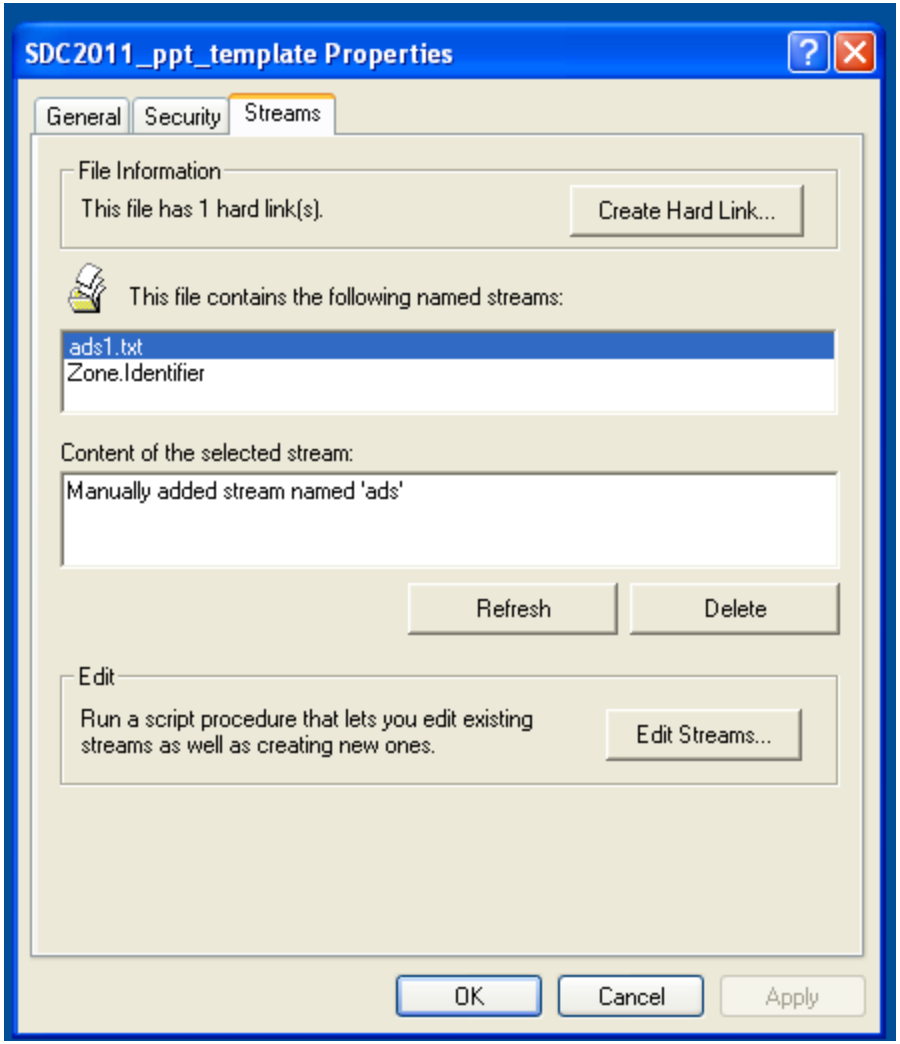
What is Alternative Data Stream?

- Alternative Data Streams – metadata associated with master file/directory objects
- Modern SMB/SMB2 clients make use of alternative data streams
  - Desktop UI enhancements
  - Additional document properties
  - Location information for files downloaded from untrusted networks

What is Alternative Data Stream (Cont)

- Stream Naming Convention
  - *Filename* : *stream_name* : *stream_type*
  - '$, ?, *, ", <, >, |' are legal chars in stream name except '/,\\'.
    For instance, ads.txt:**$test?**:$DATA is a legal named stream
- Stream Types
  - $DATA (Currently supported in PVFS)
  - $ATTRIBUTE_LIST, $BITMAP, $EA, $EA_INFORMATION, $FILE_NAME, $INDEX_ALLOCATION, $INDEX_ROOT etc…
- Default Data Stream vs. Alternative data stream
  - Directories do not have default unnamed data stream (cannot open a directory by stream name 'dir::$DATA'
  - 'ads.txt' is equivalent to 'ads.txt::$DATA' (default/unnamed data stream)
  - 'ads.txt:**summary**:$DATA' (named data stream for file object 'ads.txt)

ADS Application Use

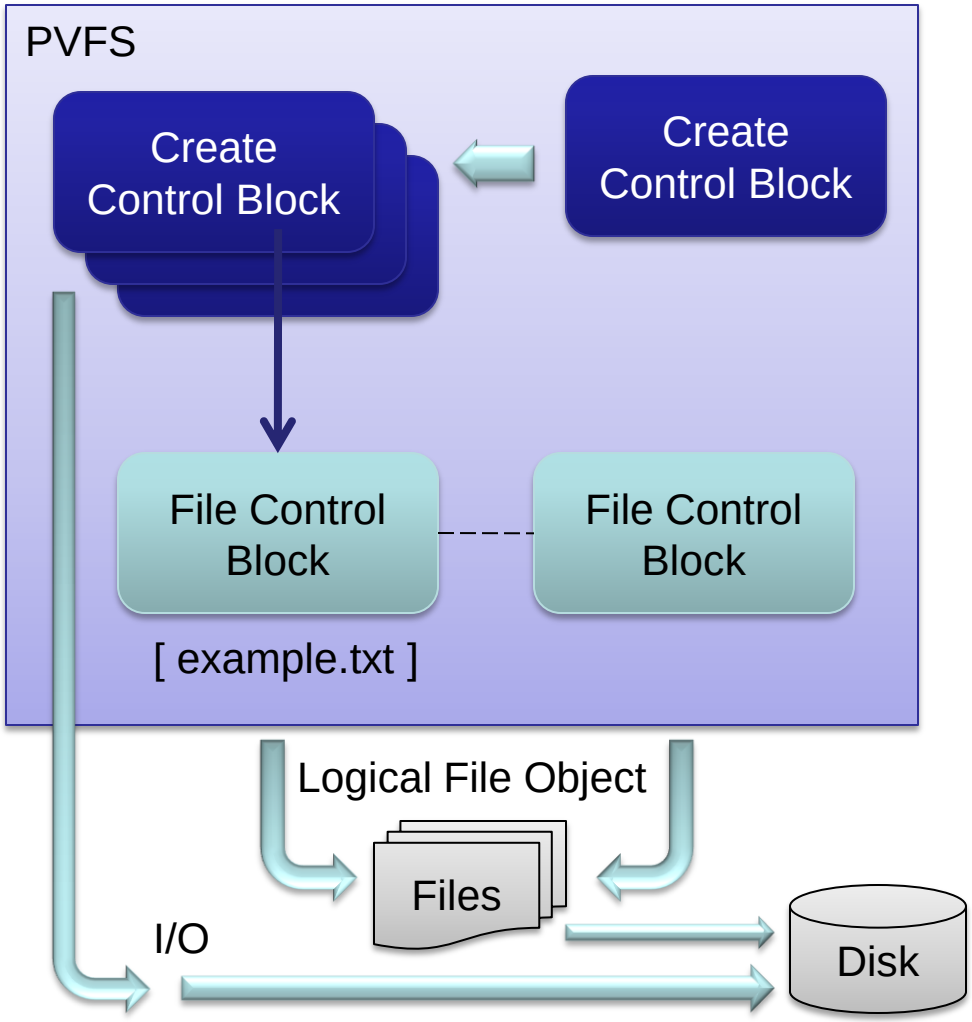


- File object: 'SDC2011_ppt_template.ppt'
- Two named streams
  - *'SDC2011_ppt_template.ppt:Zone.Identifier:$DATA'*
  - Downloaded from the web; 'Zone.Identifier' is created by the browser to store URL security zone information.
  - *'SDC2011_ppt_template.ppt:ads1.txt:$DATA'*
  - Created manually with notepad.

PVFS Data Model (Pre-ADS)

- File Control Block (FCB)
  - Represents the file on disk
  - FCB is removed from memory when last open handle is closed
- Create Control Block (CCB)
  - Open file handle
  - Stored on the IO_FILE_HANDLE owned by IoMgr
- CCB points to its FCB; FCB owns a list of its CCBs

PVFS Data Model (Pre-ADS)

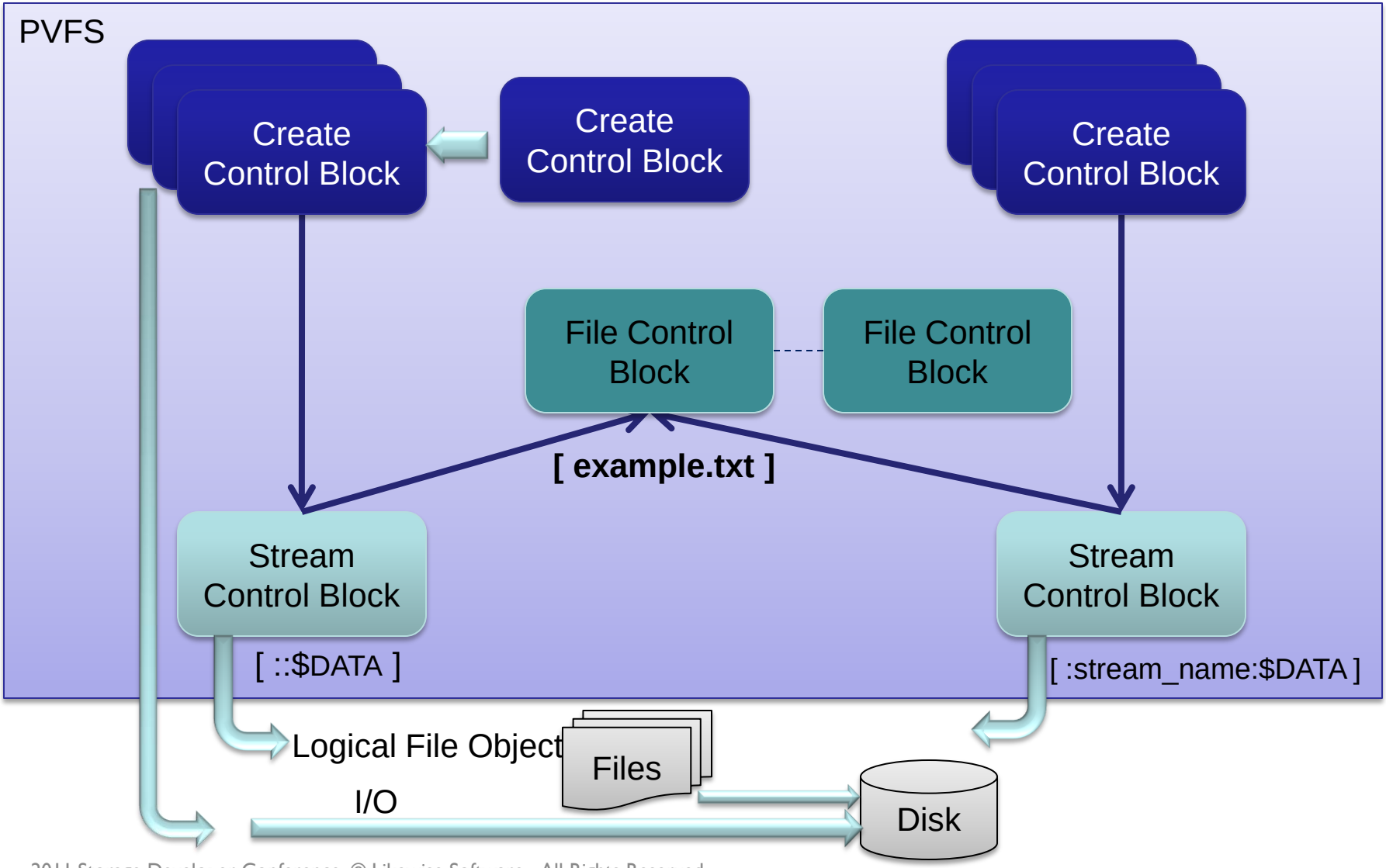


- FCB – File Object
  - Oplocks
- CCB – Open Handle
  - Device/Inode
  - Byte-Range Lock
  - Sharemode
  - File Descriptor
- Creation Sequence:
  - CCB->FCB

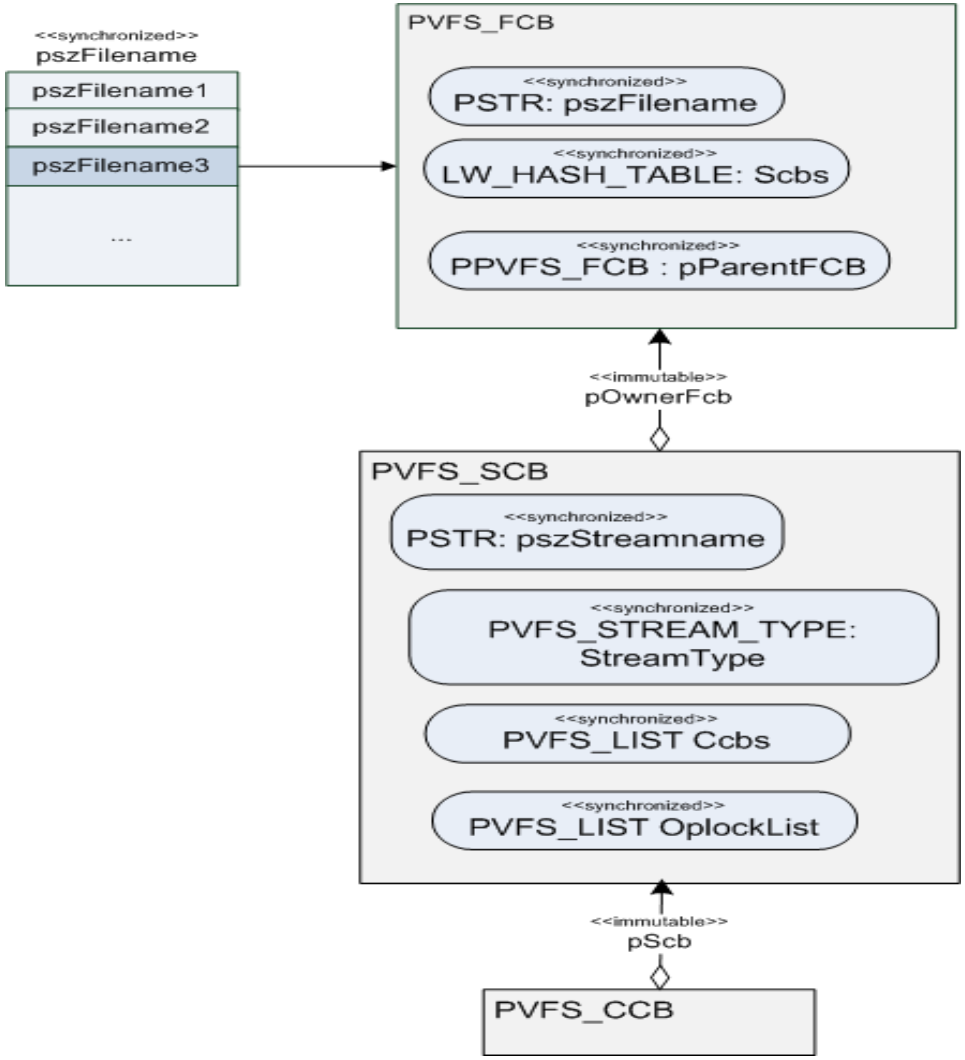
PVFS Data Model with ADS

- Stream Control Block (SCB)
  - Each SCB represents the default or named stream on-disk
  - Replaces the FCB as the target of the CCB
  - SCB is removed from memory when last open handle is closed
- Create Control Block (CCB)
  - Represents an open handle to a SCB
  - The master object is accessed by creating CCB on the default SCB
  - Stored in the IO_FILE_HANDLE owned by IoMgr
- File Control Block (FCB)
  - FCB stores information shared by all its streams
  - FCB is removed from memory when last open handle to all of its streams (default/named) is closed

PVFS Data Model with ADS (Cont.)

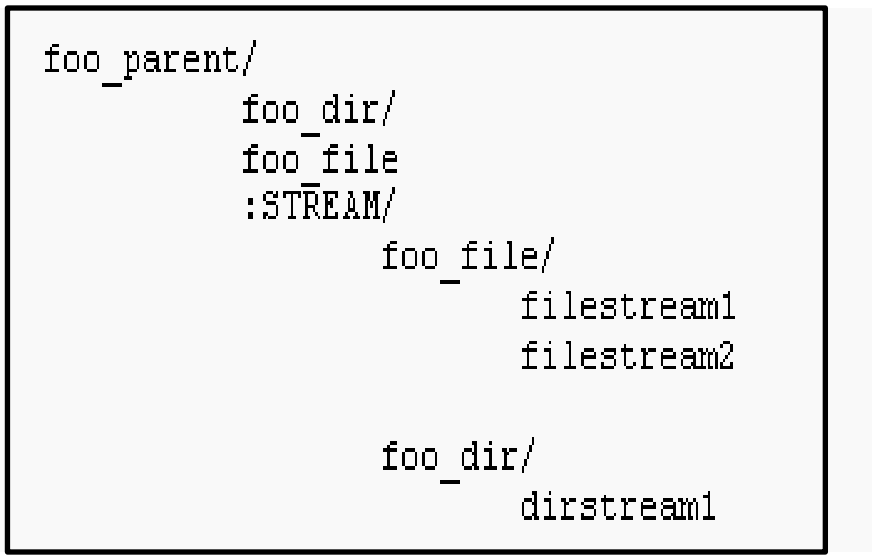


PVFS Data Structures with ADS



- CCB points to its SCB; SCB owns a list of its CCBs (one-to-many)
- SCB points to its Owner FCB; FCB owns a list of its SCBs (one-to-many)
- Creation Sequence:
  - CCB->FCB->SCB

ADS Storage Model on Ext{3,4}



A sample store

- Stream metadata directory ':STREAM'
  - For a given directory, its files and subdirectories' stream data are stored inside a ':STREAM' directory
- Stream objects residence
  - ':STREAM' contains a directory for each file/subdirectory, each of those storing their streams.

ADS Storage Model on Ext{3,4} (Cont.)

- 'EnumerateDirectory' calls needs to filter on ':STREAM'

- No need to modify create path
  - For instance, attempt to open 'foo_parent/:STREAM/foo_file/filestream' is rejected due to invalid ':' in pathname

Implementation "Adventures" Ep. I

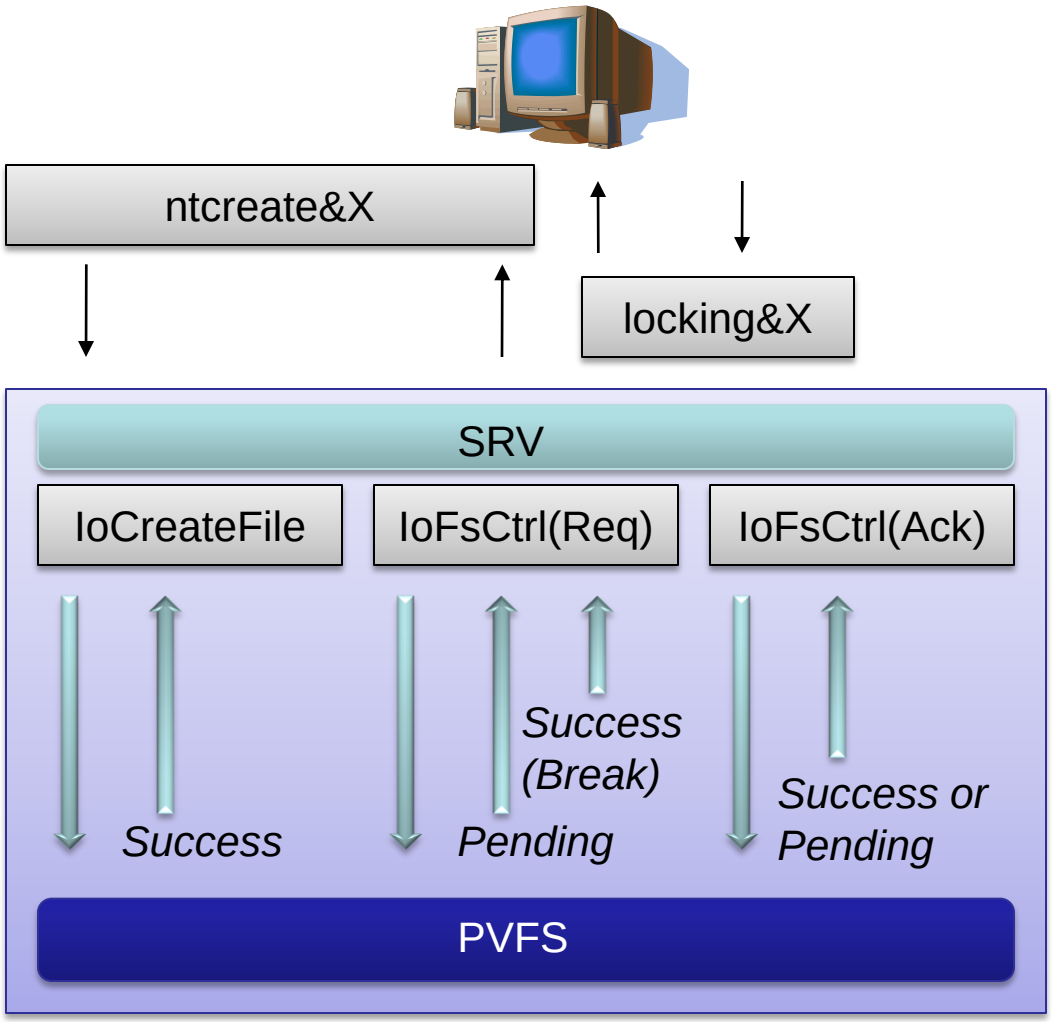
- **Problem:**
  - \\filesrv cannot copy certain files (files with streams)
- **Root cause:**
  'SetFileBasicInfo' needs to be done on base(master) object (FCB) even though the request is issued on stream handle (SCB)
- FCB vs. SCB Attributes

FCB vs. SCB Attributes

- FCB Attributes – shared attributes among all alternate data streams:
  - Extended attributes
  - Security descriptors (ACL check during 'CreateFile')
  - File Attributes
  - TimeStamps (LastWriteTime, LastAccessTime)
- SCB Attributes - unique to each stream
  - Opportunistic locks / Leases
  - Allocation size
  - Actual size
  - Valid data length
  - Sharing modes
  - BRL

Oplocks and Leases

- Oplock list stored on the SCB (a list of legacy oplock and leases)
- Deferred ops stored in a queue on the SCB
- Requested/Acknowledged using FsIoCtrl on CCB



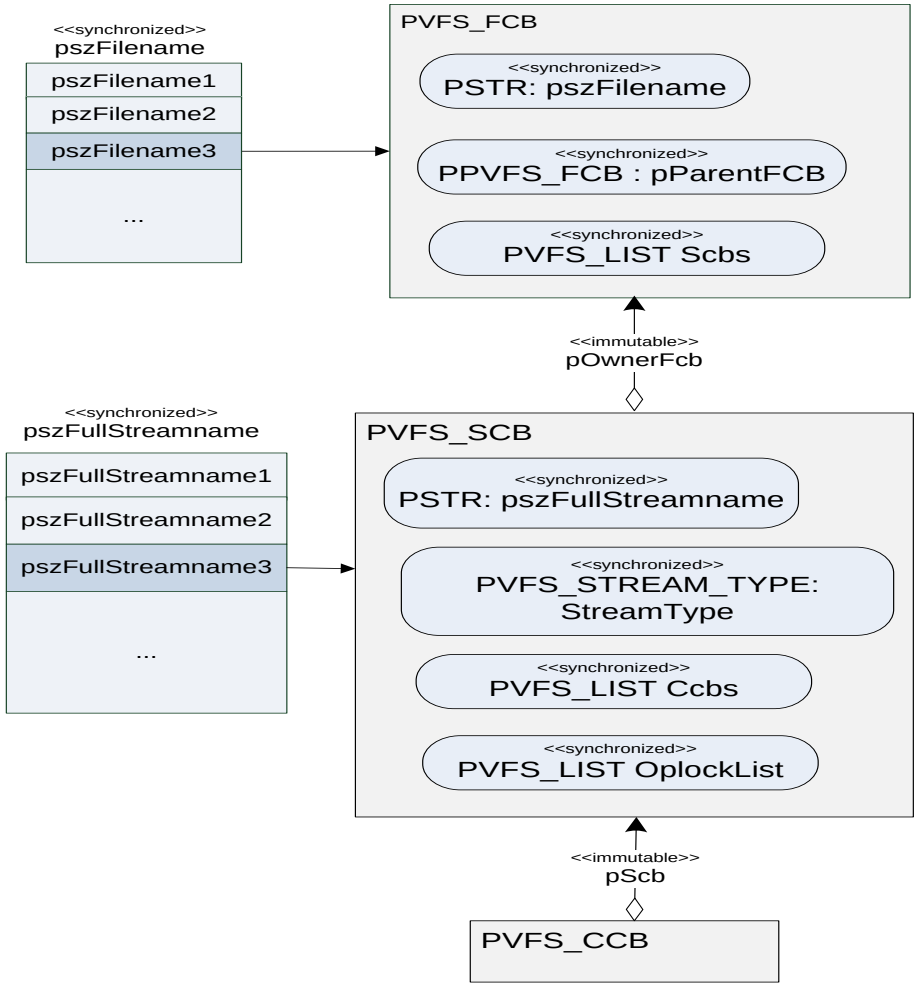
Share Modes

- Share mode enforcement is done for each stream by iterating through all its open handles
  - Check for a conflict between the request access and an existing share mode
  - Check for a conflict between the request share mode and an existing granted access
- Special case:
  - A default stream is asking for DELETE access
  - All open stream handles must be opened with FILE_SHARE_DELETE for in order for the access to be granted.

Delete-On-Close

- A named stream (SCB) can be deleted by setting 'delete-on-close'
- In case it is default data stream (SCB), 'delete-on-close' check will be done on master object (FCB)

Implementation "Adventures" Ep. 2



- **Problem:**
  NetBench run shows 'assert' failure
- **Root cause:**
  Original data model - store streams and base file objects in separate tables linked based on name
  - Unnatural relationship that complicated lock synchronization for operations such as renames
  - 'Rename' a master file object ends up modifying a bunch of SCBs in ScbTable, so does file object 'delete'

Implementation "Adventures" Ep. 3

- **Problem:**
  End up pointing to a 'wrong' logic file on-disk or doing conversion back and forth all the time

- **Root cause:**
  File name ambiguity
  (PSTR -> PVFS_FILE_NAME)
- PVFS internally only uses 'PVFS_FILE_NAME'

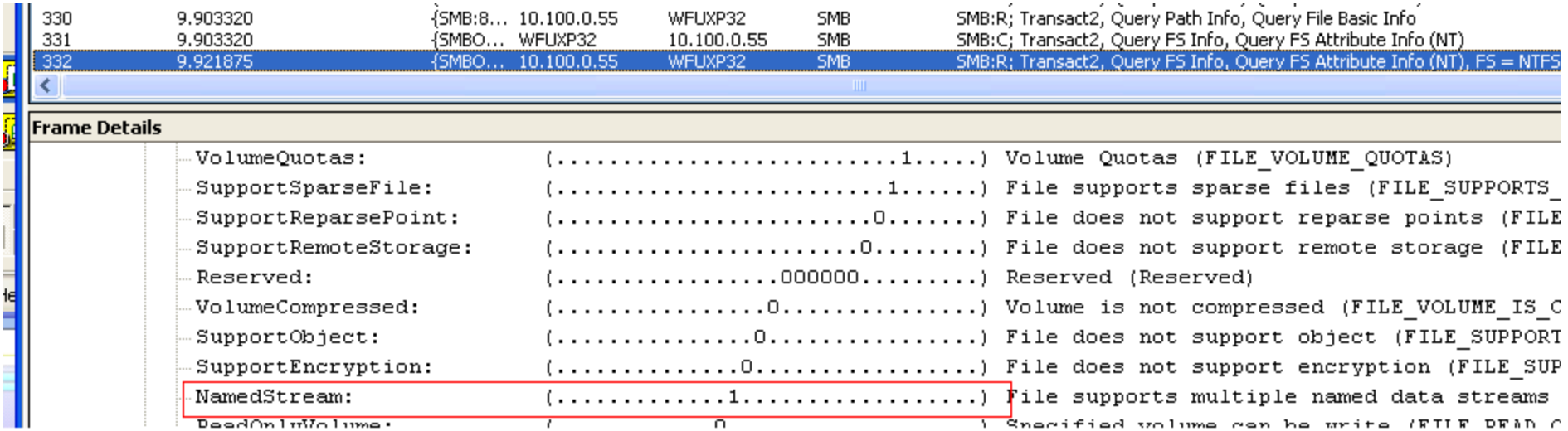
```
typedef struct _PVFS_FILE_NAME
{
    PSTR FileName;
    PSTR StreamName;
    PVFS_STREAM_TYPE Type;
    // Flag indicate whether it is a stream
     name
    PVFS_FILE_NAME_OPTIONS NameOptions;
} PVFS_FILE_NAME, *PPVFS_FILE_NAME;

// Current API on PVFS_FILE_NAME:
NTSTATUS
PvfsSysOpenByFileName(
    OUT int *pFd,
    OUT PBOOLEAN pbCreateOwnerFile,
    IN PPVFS_FILE_NAME pFileName,
    IN int iFlags,
    IN mode_t Mode
    );
// Old API on PSTR:
NTSTATUS
PvfsSysOpen(
    int *pFd,
    PSTR pszFilename,
    int iFlags,
    mode_t Mode
    );
```

Implementation "Adventures" Ep. 4

- **Problem:**
  - Failed in smbtorture test 'raw.streams.dir'
  - Failed in smbtorture test 'raw.streams.names2'
- **Root cause:**
  - Should disallow open a default (unamed) stream using name 'dir::$DATA'
  - Should disallow '/', '\\', ':' in stream names
- Internally however for consistency purpose, default SCB for directories are created

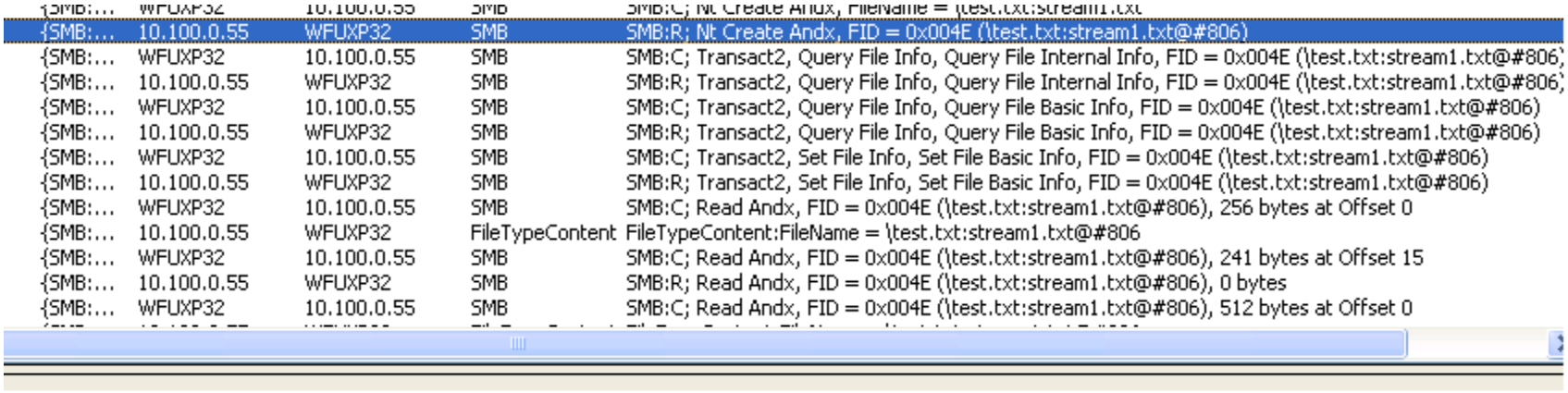
Demonstration of ADS support in Likewise Storage



330	9.903320	{SMB:8...	10.100.0.55	WFUXP32	SMB	SMB:R; Transact2, Query Path Info, Query File Basic Info
331	9.903320	{SMBO...	WFUXP32	10.100.0.55	SMB	SMB:C; Transact2, Query FS Info, Query FS Attribute Info (NT)
332	9.921875	{SMBO...	10.100.0.55	WFUXP32	SMB	SMB:R; Transact2, Query FS Info, Query FS Attribute Info (NT), FS = NTFS

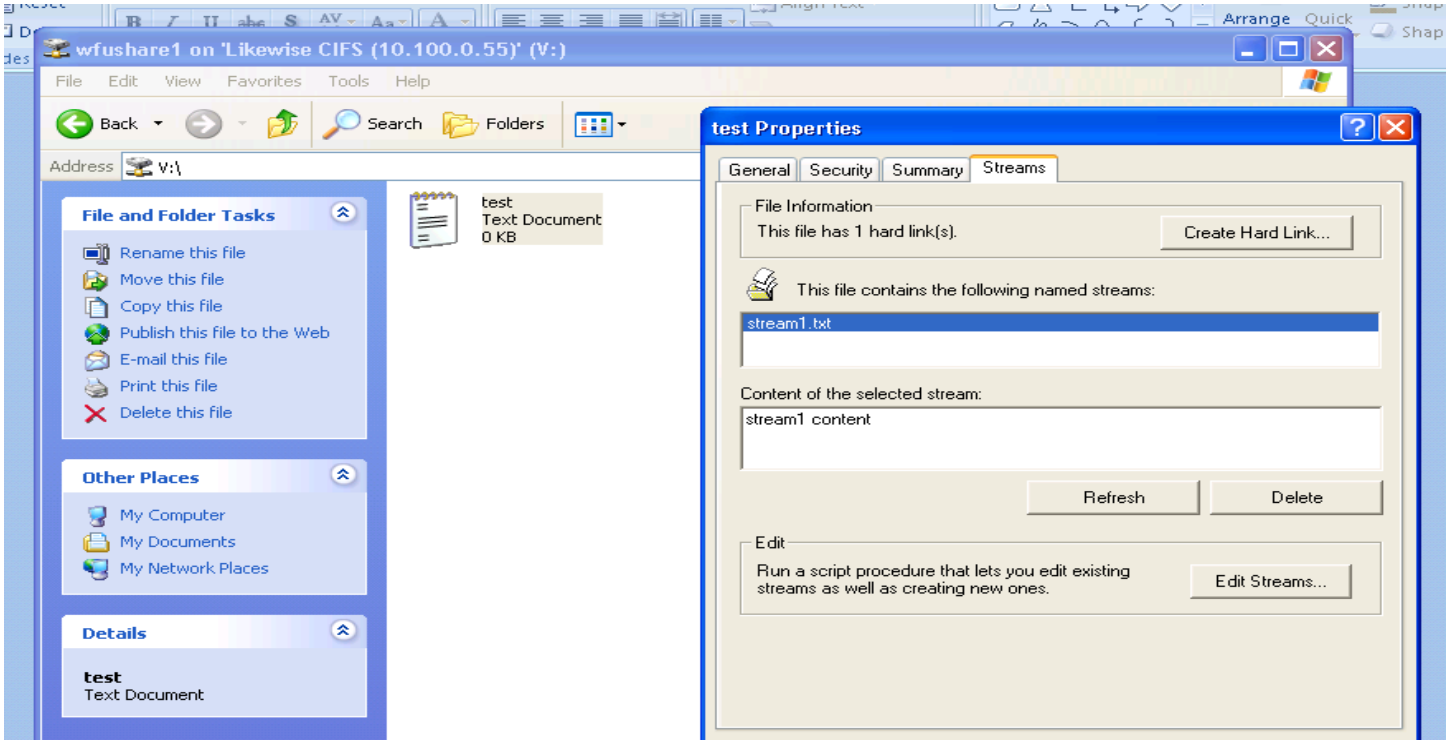
Frame Details

```
VolumeQuotas:          (.........................1.....) Volume Quotas (FILE_VOLUME_QUOTAS)
SupportSparseFile:     (.........................1......) File supports sparse files (FILE_SUPPORTS_
SupportReparsePoint:   (........................0.......) File does not support reparse points (FILE
SupportRemoteStorage:  (.......................0........) File does not support remote storage (FILE
Reserved:              (................000000.........) Reserved (Reserved)
VolumeCompressed:      (...............0................) Volume is not compressed (FILE_VOLUME_IS_C
SupportObject:         (..............0.................) File does not support object (FILE_SUPPORT
SupportEncryption:     (.............0..................) File does not support encryption (FILE_SUP
NamedStream:           (............1...................) File supports multiple named data streams
ReadOnlyVolume:        (              0                 ) Specified volume can be write (FILE_READ_O
```

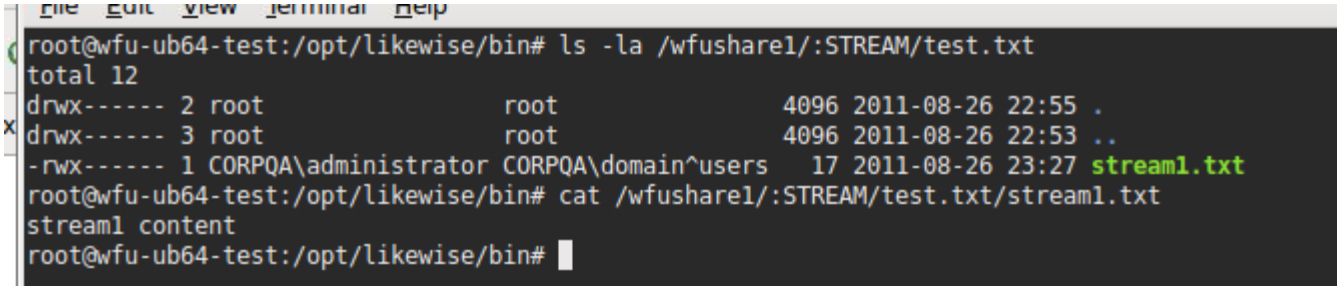


{SMB:...	10.100.0.55	WFUXP32	SMB	SMB:R; Nt Create Andx, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Transact2, Query File Info, Query File Internal Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	10.100.0.55	WFUXP32	SMB	SMB:R; Transact2, Query File Info, Query File Internal Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Transact2, Query File Info, Query File Basic Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	10.100.0.55	WFUXP32	SMB	SMB:R; Transact2, Query File Info, Query File Basic Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Transact2, Set File Info, Set File Basic Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	10.100.0.55	WFUXP32	SMB	SMB:R; Transact2, Set File Info, Set File Basic Info, FID = 0x004E (\test.txt:stream1.txt@#806)
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Read Andx, FID = 0x004E (\test.txt:stream1.txt@#806), 256 bytes at Offset 0
{SMB:...	10.100.0.55	WFUXP32	FileTypeContent	FileTypeContent:FileName = \test.txt:stream1.txt@#806
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Read Andx, FID = 0x004E (\test.txt:stream1.txt@#806), 241 bytes at Offset 15
{SMB:...	10.100.0.55	WFUXP32	SMB	SMB:R; Read Andx, FID = 0x004E (\test.txt:stream1.txt@#806), 0 bytes
{SMB:...	WFUXP32	10.100.0.55	SMB	SMB:C; Read Andx, FID = 0x004E (\test.txt:stream1.txt@#806), 512 bytes at Offset 0

Demonstration of ADS support in Likewise Storage

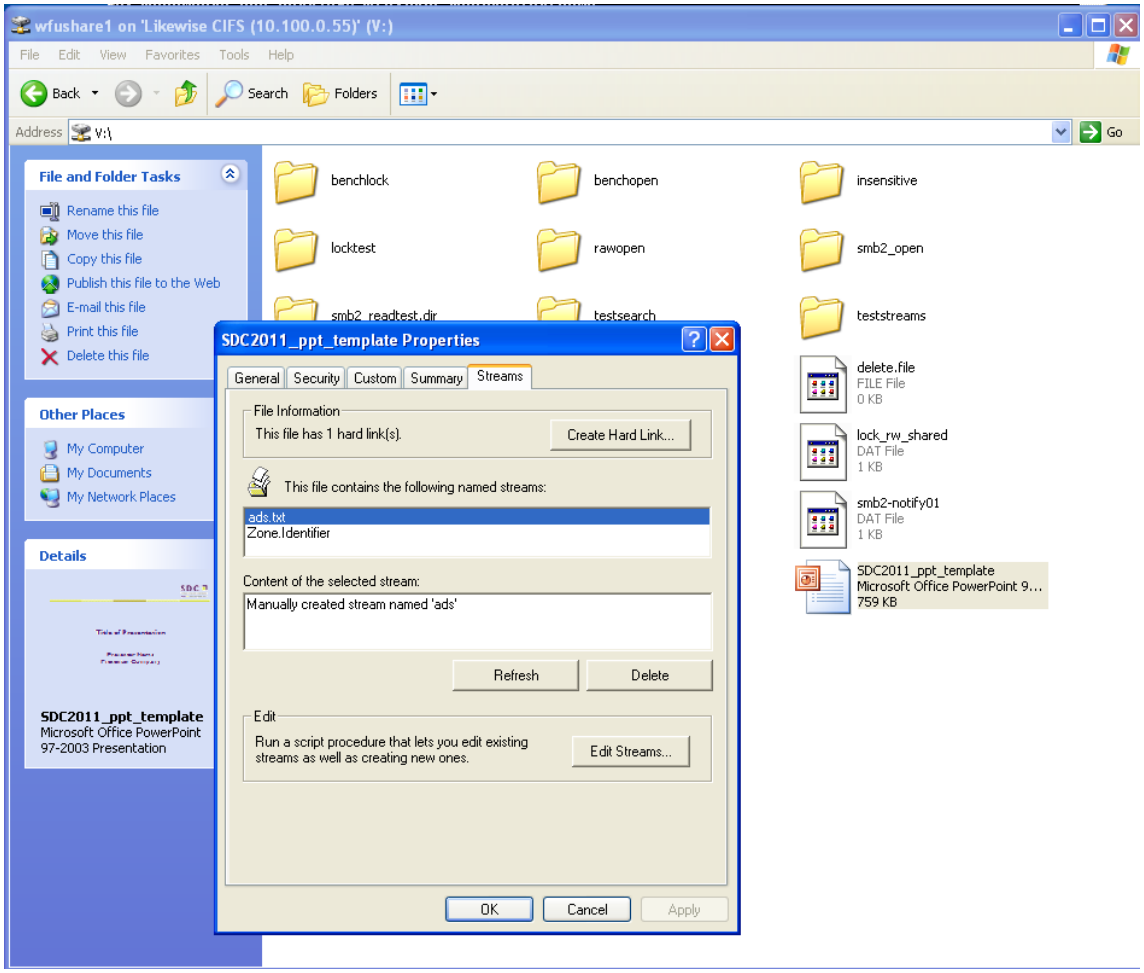


- Client opens 'test.txt' on share (V:)



- Server disk based ADS store

Demonstration of ADS support in Likewise Storage



- The sample file is copied to share with all data streams being preserved

Questions?