

IFS

Paul Roy

IFS

- Integrated File System
- File System ???
 - Storage Management component of an operating system
 - Unix
 - Dos
 - Windows
 - AS/400
 - IBM 36

/ Root File System

- The "root" (/) file system acts as an umbrella or a foundation for all other file systems on the server. At a high level, it provides an integrated view of all of the objects on the system.

i5 File Systems

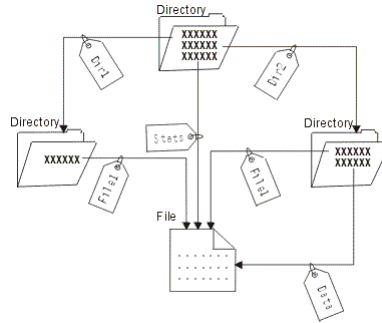
- / (Root)
- /QopenSys
- /QSYS.LIB
- /QDLS
- /QOPT
- /QNTC
- /QfileSvr.400
- /QLANSrv

IFS concepts

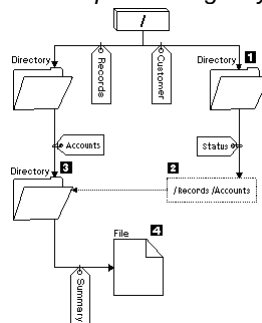
- **Directory**
A *directory* is a special object that is used to locate objects by names that you specify. Each directory contains a list of objects that are attached to it. That list can include other directories.
- **Link**
A *link* is a named connection between a directory and an object. A user or a program can tell the system where to find an object by specifying the name of a link to the object. A link can be used as a path name or as part of a path name.
- **Path name**
A *path name* tells the system how to locate an object.
- **Stream file**
A *stream file* is a randomly accessible sequence of bytes, with no further structure imposed by the system.
- **Extended attributes**
An extended attribute is information associated with an object that provides additional details about the object. The extended attribute consists of a name, which is used to refer to it, and a value. The value can be text, binary data, or another type of data.

- Directory
 - Current Directory
 - Home Directory
- Link
 - Hard link: A *hard link*, which is sometimes just called a link, cannot exist unless it is linked to an actual object.
 - Symbolic link: A *symbolic link*, which is also called a soft link, is a path name contained in a file

A directory entry defines each hard link



An example of using a symbolic link



PathName

- The path name is expressed as a sequence of directory names followed by the name of the object. Individual directories and the object name are separated by a slash (/) character; for example:
- directory1/directory2/file For your convenience, the backslash (\) can be used instead of the slash in integrated file system commands.
- There are two ways of indicating a path name:
 - An *absolute path name* begins at the highest level, or "root" directory (which is identified by the / character).
 - If the path name does not begin with the / character, the system assumes that the path begins at your current directory. This type of path name is called a *relative path name*.
- Use . (this directory) or .. (parent directory)
- PathName must be unique in one directory.
- Path names must be enclosed in single quotation marks (') when entered on a command line.
- The length of an object name is limited by the file system the object is in and the maximum length of a command string. The commands accept object names up to 255 characters long and path names up to 5000 characters long.

PathName starting with a tilde

- A tilde (~) character followed by a separator character (for example: /) at the beginning of a path name means that the path begins at the home directory of the user entering the command;
 - for example: '~/UsrDir/UsrObj'
- A tilde (~) character followed by a user name and then a separator character (for example: /) at the beginning of a path name means that the path begins at the home directory of the user identified by the user name;
 - for example: '~user-name/UsrDir/UsrObj'

Root File System

- /
- WRKLNK '/'
 - CD '/'
 - CHGCURDIR
 - WRKLNK
 - MD MKDIR
 - RMVDIR
 - RMVLNK
 - ADDLNK

i5 File Systems

- /
 - (Root)
- /QopenSys
 - Case sensitive (Unix Like)
- /QSYS.LIB
 - 'Classic AS/400'
- /QDLS
 - Folder
- /QOPT
 - CD , DVD and optical disks
- /QNTC
 - Windows domain
- /QfileSvr.400
 - i5 Network
- /QLANSrv
 - Integrated file server

/QSYS.LIB

- /QSYS.LIB/QGPL.LIB
- /QSYS.LIB/QGPL.LIB/QCLSRC.FILE
- /QSYS.LIB/QGPL.LIB/QCLSRC.FILE/QSTRUP.PGM

- ADDLNK
OBJ('/QSYS.LIB/USR_ROPA.LIB/TEST.USRSPC')
NEWLNK('/tmp/test/test.spc')
- ADDLNK
OBJ('/QSYS.LIB/QGPL.LIB/QCLSRC.FILE/QSTRUP.PGM')
NEWLNK('/tmp/test/qstrup.clp')

/QDLS

- Shared folders
- Need to be registered in System directory
 - WRKDIR
- Limitations (Old DOS Names 8.3)
- Max 240000 objects in a library → max number of files in folders (lib QDLS)
- DLO objects..
- WRKFLR, WRKDOC

/QOPT

- Optical File system
- Optical libraries
- CD ROM / DVD ROM /DVD RAM
- Virtual Optical
- Image Catalog
- WRKLNK '/QOPT' → list of volumes
- GO OPTICAL

/QNTC

- Acces to Remote Server (Windows, Samba, Linux,...)
- WRKLNK /QNTC

```
Session A - [24 x 80]
File Edit View Communication Actions Window Help
Work with Object Links
Directory . . . . . /QNTC/RSUKS153
Type options, press Enter.
2=Edit 3=Copy 4=Remove 5=Display 7=Rename 8=Display attributes
11=Change current directory ...

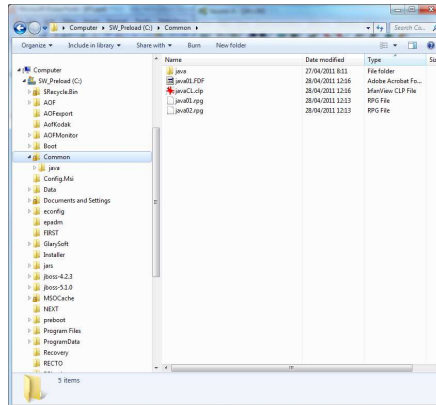
Opt Object link Type Attribute Text
-- --
. DDIR
.. DDIR
Common DDIR

Parameters or command
==>
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F12=Cancel F17=Position to
F22=Display entire field F23=More options

12/002
PDFCreator on PDFCreator
```


/QNTC

- An RPG program running on your i5 can read/write to your PC disk



/QFileSvr.400

- Access to the IFS of another i5 /poweri / AS/400 System
- /QFileSvr.400/SystemName (or IP..)
- Same User and Password (DDM like)

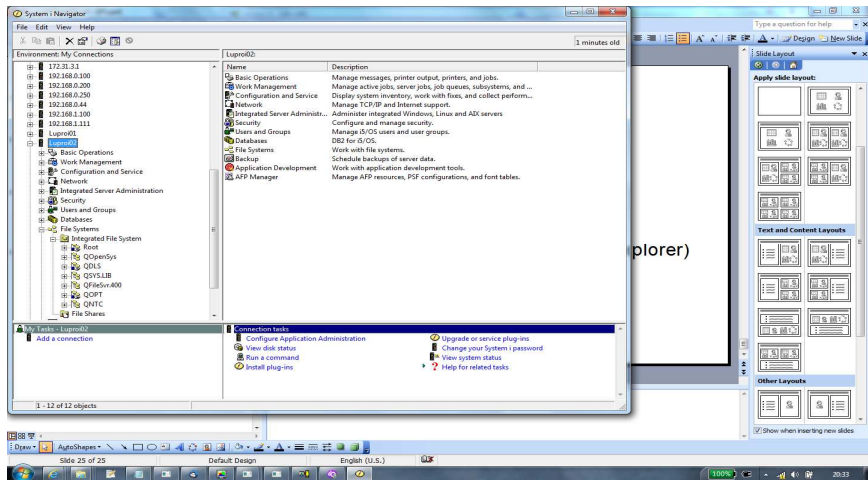
UDFS

- User Defined File System
 - ASP or IASP
 - /dev/Asp_Name/QDEFAULT.UDFS
 - /ASP_Name will be mounted as / on the IASP when it varied on.

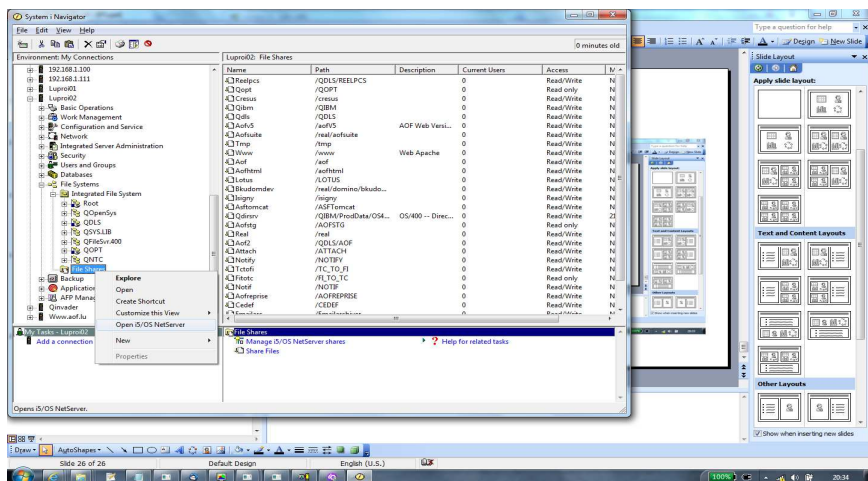
ACCESS to the IFS

- Menu
- CL Commands
- API
- System I Navigator
- iSeries NetServer (and windows explorer)
- FTP

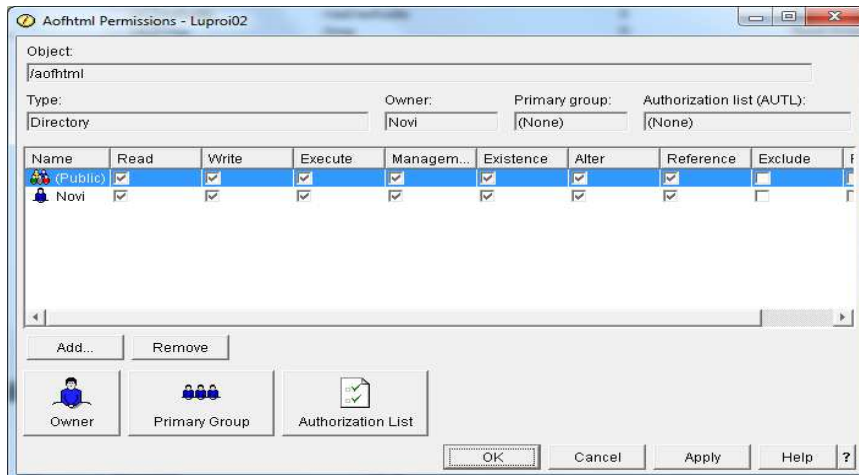
I Navigator



Net Server



NetServer



Root File System

- Authorities
 - SPEC 1170 compatibility
 - Read Write eXecute
 - Owner / Group / User
 - Stored in 3X3 bits
 - rwx rwx rwx
 - 111 - 111 - 111
 - Read in binary by block of 3
 - `chmod 777 /tmp/file.txt (→*PUBLIC *ALL)`
 - `chmod 700`

Root File System Authorities

- Unix to i5 mapping
 - OWNER = OWNER
 - GROUP = Primary Group in the user profile
 - Read (*R) = *OBJOPR and *READ
 - Write (*W) = *OBJOPR, *ADD, *UPD, *DLT
 - Execute (*X) = *OBJOPR and *EXECUTE
- *

IFS Root Authorities

- OBJMGT, *OBJEXIST, *OBJALTER, *OBJREF do not exist in Unix
- But the properties exists for each object in the root file system.
- When an object is created using an API
 - Owner, Group, User authorities are the same as ones of the parent directory

IFS Root Authorities

- The new object's data authority for are specified on the API with the mode parameter.
- When all of the object authorities are set 'on', you get the authority behavior that you would expect in a UNIX-type environment. It is best to leave them set 'on', unless you do not want the POSIX-like behavior.
- When you run applications that use UNIX-like APIs, the system enforces all object authorities, whether or not they are "visible" to UNIX-type applications. (example authorizations lists)

He, I am the owner

- Common problem: *I created the file, I am the owner, but I cannot delete the file.... What's wrong ?*
 - Remember: when you create an object in a directory, the authorities are the same as the directory.
 - SO if the owner authority attribute of the directory is 1(ReadOnly) then your file is created with the owner attribute set as ReadOnly...
 - When you create a directory, the owner must also appear in the security authorization for the directory. In the native file system, the owner has clear rights to an object, but not so in the IFS. The owner's rights must be defined as a private authority.
-
- Tip: When you create an IFS object, for example with CPYTOSTMF always run a CHGAUT immediately after...

ADOPTED AUTHORITY

- Adopting authority with a program compiled as USRPRF(*OWNER)...
- ADOPTED AUTHORITIES ARE **NOT** HONORED in the IFS (/ , /QDLS, /QopenSys)

Check the X authority to the path

- Many operations require *RX authority to the entire path including the root. When experiencing authority problems, verify the entire path including the root rather than only the object and its immediate parent.

Set the authorities

- Files created with the CPYTOSTMF or CPYTOIMPF commands have data authorities for *PUBLIC set to *NONE and have no group authority regardless of the authorities on the parent directory. The CHGAUT command must be used after the CPYTOSTMF or the CPYTOIMPF command fails to assign the desired authorities.

Check Authority for CPY

- To retrieve authority information about an object, the user must have *OBJMGT object authority to the object. This can affect functions such as COPY which needs to retrieve authority information about the source object so it can set the equivalent authorities on the target object.

tips

- The user changing the owner or group of an object must have sufficient authority to the object; for example, *ADD data authority to the new owner/group user profile and *DELETE data authority to the old owner/group. These are not the same as the file system data authorities. This can be viewed using the DSPOBJAUT command or the EDTOBJAUT command. This can be an issue on some COPY commands when trying to copy the group.
- When changing the owner or group of an object, the new owner cannot be the same as the current group, or the new group cannot be the same as the current owner.

tips

- To display or retrieve the current working directory (DSPCURDIR, GETCWD(), and so on), you must have *RX data authority to the whole path including Root. To change your current directory (CD, CHDIR(), and so on), you need only *X authority to the whole path. This means that you can change your working directory to a path that you are unable to retrieve.

tips

- To change the group of an object, the user must be a member of the new group. Therefore, if the source object of a CPY has a group and the target directory has no group or a different group, the user performing the CPY must be a member of the group for the source object. If not, the user is not authorized to change the group on the newly created copy and an authority failure occurs.

User/Group

- In UNIX environments, users and groups are separate entities. On the iSeries system, however, they are both the same object type (all a type *USRPRF object). The only difference is that a user profile with a group ID is considered a group profile. However, OS/400 security still recognizes the difference.
- → If you sign on as QPGMR, you could not be allowed to an object having QPGMR as the primary group....

Initial authority

- Initial object authorities are assigned to a new file or directory based on the authority values of the parent directory. The following rules involved with this are:
- **a)** The owner for the new object has the same object authorities the owner of the parent directory to the parent directory.
- **b)** The primary group for the new object the same object authorities the primary group of the parent directory to the parent directory.
- **c)** *PUBLIC has the same object authorities to the new object that it has to the parent directory.

Initial authority

- These rules apply even when the owner of the parent directory and the newly created object are not the same, and even when the owner of the new object has separate private authority to the parent directory.

For example:

```
Object . . . . . : /ropa
Owner . . . . . : ROPA
Authorization List. . . . . : *NONE
User:      Object Auth:
Object authorities -
*PUBLIC    *NONE
TOTO      *ALL
```

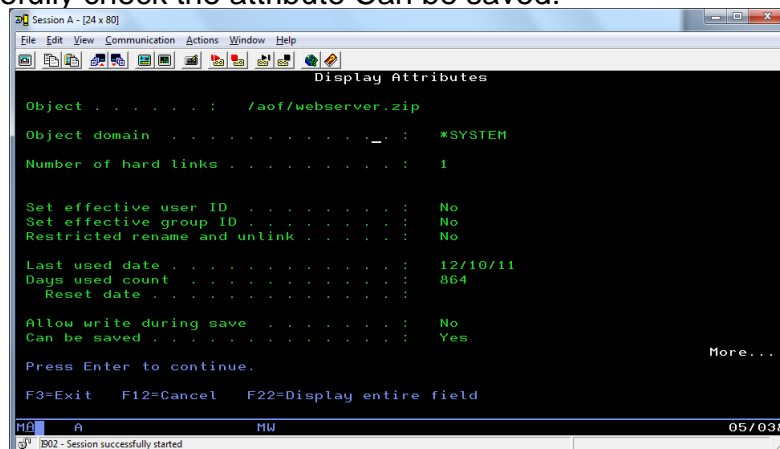
User TOTO signs on and creates a new object (file/dir) under /ropa. TOTO is the owner of this new object and adopts the same object authority as the owner of object '/ropa'.

Initial authority

- Because ROPA does not have authority to '/ropa' (no user authorities are listed for ROPA), user TOTO also does not have object authority to the object he created under '/ropa'.
- This can be a problem for some PC applications (such as Excel/Word) which create a temporary file when opening a current file for edit. When the user attempts to save the file, the application renames the temporary file to the original file name; however, this requires at least *OBJMGT authority which the owner does not have (example above) and, therefore, the user is unable to save the file.
- The above is true for OBJECT authorities. DATA authorities are handled differently. Typically, how these are inherited or determined are based on the interface used to create the new object. From most PC interfaces, the owner/creator gets all data authorities regardless of the parent authority

BACKUP/RESTORE

- SAV command is used to save the IFS...
- Carefully check the attribute Can be saved:



Backup

```
CHGATR OBJ('/aofstg/*') ATR(*ALWSAV) VALUE(*YES)  
SUBTREE(*ALL)
```

