Syncweave –Shared folders Test Cases

Note: Some test cases may be impossible/not allowed to perform under Windows and/or on the web interface. Ignore them.

Folder structure for userno1

Shared folder- Regular folder- Regular File-

/reg 001 /reg\_sub 001 / file batman

/ file superman

/reg\_sub 002 /file spiderman

/file catwoman

/file fiddler

/file tshark

/shared 001 /reg boo /reg goo / file tesla

/ file buick

/file audi

/file benz

/reg kukoo /file toothfairy

/ file can

/reg trash /shared ghost / reg tree /file abc

/ reg plant / file xyz

/ reg leaves / reg trunk /file ant

/ file tapir

/ file school

/ file college

/ shared foo / reg darling / reg forest / file barney

/ reg war / shared peace / reg dove / file white

Folder structure for userno2

Shared folder- Regular folder- Regular File-

/ shared 001/ reg boo / reg goo / file tesla

/ file buick

/ file audi

/ file benz

Reg kukoo / file toothfairy

/ file can

/reg cap/ shared ghost / reg tree / file abc

/ reg plant / file xyz

/ shared foo / reg darling / reg forest / file barney

/reg cap / shared peace / reg dove / file white

/ reg ruff / reg cat / reg house

/ file eraser

/ file pen

PART-1

Objective- To test that server side actions reflect appropriately on all client machines (collaborators and owners)

Note- Use the folder structures above to perform the following tests. Userno3 has been mentioned in the test cases below but does not have a reference structure. Following the test case steps will automatically generate his folder structure.

**Scenario 1**: Owner actions performed on shared folders. ALL actions performed userno1 on server side.

Pre-req- userno1 is the owner for ‘shared001’ and ‘shared foo’. He shares it with userno2 and userno3. Userno2 will mount these shared folders according to folder structure above. Userno3 can mount it anywhere. (all other files and folders in the folder structures above need to be created prior to testing)

1. move ‘shared001’ under ‘reg trash’

2. move shared001 under shared ghost

3. move shared001 under shared001

4. Make local copy of shared001 and mount it under

-root

-under reg001

-under shared001

-under reg trash

5. rename shared001

6.delete shared001

7. restore shared001

8. create new contents (new files and folders)inside shared001

9. unmount shared001

10. mount shared001 under

-reg trash

-root

-reg001

11. permanently delete shared001

12. unshare shared001

13. invite new collaborators

14. ban userno2

15. kick out userno3

16. re-invite userno3

17. unban userno2 @ re-invite

18. offer ownership to userno2

-- Actions on shared folder contents—

19. move reg boo to

-reg war

-reg kukoo

-share ghost

20. copy reg boo to

-reg war

-reg kukoo

- share ghost

21. rename reg boo

22. delete and restore reg boo

23. Convert ‘reg boo’ to a shared folder

24. Move file ‘file audi’ to –root

-reg war

-shared ghost

25. copy ‘file audi’ to – root

-reg war

-shared ghost

26. rename ‘file audi’

27. delete & restore ‘file audi’

**Scenario 2:**

Actions performed by collaborator (userno2)

1. Move ‘shared001’ to –root

-reg ruff

-shared ghost

2. copy ‘shared001’ to –root

-reg ruff

-shared ghost

3. rename ‘shared001’

4. delete and restore ‘shared001’

5. upload new files & folders inside ‘shared001’

6. mount and unmounts. Mount under –root

-reg ruff

-shared ghost

7. Leave shared folder ‘shared001’

8. Make local copy –root

-reg ruff

-shared ghost

9. convert to share

10. kick-out userno3

11. re-invite userno3

12. request ownership

13. invite new collabs

14. try to permanently delete contents

15. move ‘reg boo’ to –root

-reg cap

-reg ruff

-shared foo

16. copy ‘reg boo’ to –root

-reg cap

-reg ruff

-shared foo

17. rename ‘reg boo’

18. delete and restore ‘reg boo’

19. Convert ‘reg boo’ to a shared folder

20. Move ‘file audi’ to –root

-reg cap

-reg ruff

-shared foo

21. Copy ‘file audi’ to –root

-reg cap

-reg ruff

-shared foo

22. rename ‘file audi’

23. delete and restore ‘file audi’

**Scenario 3:**

Actions on shared folder located under a regular folder performed by owner.

1. Move ‘shared ghost’ to – root

-shared001

-reg war

-reg001

2. Copy ‘shared ghost’ to – root

-shared001

-reg war

-reg001

3. rename ‘shared ghost’

4. delete and restore ‘shared ghost’

5. create new contents under ‘shared ghost’

6. mount and unmount mount under – root

-shared001

-reg war

-reg001

7. Try and convert to share

8. perm delete ‘shared ghost’

9. unshare

10. invite new collabs

11. ban userno2

12. kick out userno3

13. unban userno2 and re-invite

14. offer ownership to userno2

15. move ‘reg trash’ to –root

-reg tree

-shared foo

-reg war

16. Move ‘reg tree’ to the following locations

-root

-reg trash

-reg war

-shared foo

17. copy ‘reg trash’ to the foll locations

-root

-reg tree

-shared foo

-reg war

18. rename ‘reg trash’

19. delete and restore ‘reg trash’

20. convert to share

21. perm delete ‘reg trash’

22. Move ‘file xyz’ to –root

-reg tree

-reg kukoo

-shared foo

-reg war

23. copy ‘file xyz’ to –root

-reg tree

-reg kukoo

-shared foo

-reg war

24. rename ‘file xyz’

25. delete and restore ‘file xyz’

26. perm delete ‘file xyz’

**Scenario 4:**

Collaborators dealing with shared folders & its contents created under regular folders.

1. Move ‘shared ghost’ to –root

-reg ruff

-shared foo

-reg cap

-reg dove

2. Copy ‘shared ghost’ to - –root

-reg ruff

-shared foo

-reg cap

-reg dove

3. rename ‘shared ghost’

4. delete and restore ‘shared ghost’

5. mount & unmount. Mount under –root

-reg ruff

-shared foo

-reg cap

-reg dove

6. leave ‘shared ghost’

7. make local copy –root

-reg ruff

-shared foo

-reg cap

-reg dove

8. Convert to share ‘shared ghost’

------testing contents-------

9. Move ‘reg cap’ to –root

-shared001

-reg gap

Reg dove

-reg ruff

10. copy ‘reg cap’ to -–root

-shared001

-reg gap

Reg dove

-reg ruff

11. rename ‘reg cap’

12. delete & restore ‘reg cap’

13. convert ‘reg cap’ to share

14. move ‘reg plant ‘ to - root

-shared001

-reg gap

-reg dove

-reg ruff

15. copy ‘reg plant’ to - root

-shared001

-reg gap

-reg dove

-reg ruff

16. rename ‘reg plant’

17. delete & restore ‘reg plant’

18. convert ‘reg plant’ to share

19. move ‘file xyz’ to - root

-shared001

-reg gap

-reg dove

-reg ruff

20. copy ‘file xyz’ to - root

-shared001

-reg gap

-reg dove

-reg ruff

21. rename ‘file xyz’

22. delete & restore ‘file xyz’

## Part 2 – Client side actions

**Basically have to perform all applicable test cases from PART 1**

# Conflicting actions

1. User is editing a file on Windows and one of the following actions take place:

a. Owner unshares the cabinet

b. owner kicks out the user

c. owner bans the user

d. another collab uploads a more recent version of the file

e. another collab moves that folder/file to another locations

e. another collab copies the file /folder

f. another collab renames the file/folder

g. owner perm deletes the file/folder

2. User copying some files to a shared folder/folder inside shared folder and \*\*

When \*\* denote a-g from “1.” above

3. User deleting a huge folder containing a large number of files & \*\*

When \*\* denote a-g from “1.” above

Additional requirement to test – If a normal folder is soft-deleted from the client, we should have to ‘restore’ the top-most level directory. NOT all contents underneath individually.