ARCHIVING INSTRUCTIONS

For *minerva-data*, *minerva-app* and *minerva-persistent* areas.

Archiving

1 - `kinit` normally with your user, then login as "minervapro", otherwise you won't be able to read and write from/to certain needed directories.

2 - Find the possible `.BAD` files in the area to archive. Go to the area you want to archive and run the following command in a screen session:

```
find . -name \*.BAD -not -path \*/\.\* > /minerva/data/users/mmehmood/$USER_badfiles.txt
```

3 - Ask the user (or Dan in case of former collaborators) to remove those files before moving forward, otherwise the archiving will just freeze when it finds a '.BAD' file and you'll have to start over.

NOTE. Steps 4-9 are to explain the procedure, but are contained in the script "/minerva/app/home/minervapro/archive_user.sh", which is run like:

```
archive_user.sh <user> <area>
```

The options for the area are: "persistent", "app", "data" and "data2" (all lowercase and without quotation marks).

4 - In the working directory you just logged into "/minerva/app/home/minervapro", source the following:

```
source minset_v22r1p1.sh
```

5 - Go to "Personal/rodrigues" area "/minerva/app/users/minervapro/cmtuser/Minerva_v22r1p1/Personal/rodriges/multivolume-tar":

6 - Create an "output" directory to archive the "input" directory. The output directory must be in "/pnfs/minerva/archive", but for testing it can be in "/pnfs/minerva/scratch", both under the 'username' who owns the files (always put a consistent name, and include the date). Create the directory with (for instance):

```
mkdir -p /pnfs/minerva/archive/maramire/persistent_backup_test-2021-03-05
```

7 - Make the output directory writable for minervapro and just readable and executable for all other users.

```
chmod 755 /pnfs/minerva/archive/maramire/persistent_backup_test-2021-03-05
```

8 - Check that there is a valid certificate before running the actual "archiving" command. Although you can check the remaining time before the certificate expires, better to just 'kinit' again (with your user, in this case "mmehmood"), and renew the certificate. For large datasets you might want to request the maximum possible days (in hours), which is one week (168 hours), if I remember correctly:

```
$ kinit $User
$ kx509 && voms-proxy-init -rfc --voms=fermilab:/fermilab/minerva/Role=Analysis --noregen
-valid 168:0
```

9 - Run the "backup_user_directory.sh" script to archive the input directory in the output directory (remember to do this in a screen session, logged in as minervapro. Input directory can be any 'minerva-data', 'minerva-app' or 'pnfs/minerva/persistent' directory.

```
source backup_user_directory.sh /pnfs/minerva/persistent/users/$User/Input/directory/name
/pnfs/minerva/archive/$User/persistent_backup_test-year-month-day/
```

10 - Check the size of both directories is the same, using. If they are not, you must delete the .tar files at the archiving directory, and at “/minerva/data/users/archive/stage/minervapro/”. Then you have to start over. The problem is usually dCache hiccups.

11 - The number of files should be “close” in both directories. To do this use the scripts `nRFiles.sh` and `countTarFiles.sh /<path_of_archived_directory>/`. The former counts the files in the original directory (`cd` into that directory to run it), and the latter counts the files in the .tar file. The "tar" command tars all kinds of files (regular, hidden, symlinks, etc.), but CANNOT archive files with "owner-only" access (codes like 600 or 700 for instance). The only way we could archive those is becoming root (I think). The “nRFiles.sh” counts everything, including the "owner-only" access. So if the user has these kind of files, they will be missing in the archived file(s). But since people with root permissions won’t do the archiving, the following is the optimal procedure, so the process doesn't turn slow:

A - Before archiving, ask the user if he/she thinks there could be protected files he/she might want to keep. If so, the user should act accordingly (change permissions to those files).
B - If the user doesn't care about those, archive, and if the only missing files are "user-only" access files, proceed with deletion.
C - If the user is not reachable after a number of days (specially for long-gone former collaborators), we imply he/she doesn't care about protected files.

While running the `countTarFiles.sh` script you'll see the messages: "tar: Unexpected EOF in archive tar: Error is not recoverable: exiting now" which should be harmless. Preferently run this in a screen session.

Recovery

1 - Log in as 'minervapro'.
2 - `kinit` and renew the certificate as shown above.
3 - Create a screen session, go to "/minerva/app/home/minervapro".
4 - Run `source minset_v22r1p1.sh`
5 - Go to "Personal/rodrigues/multivolume" directory "/minerva/app/users/minervapro/cmtuser/Minerva_v22r1p1/Personal/rodriges/multivolume-tar".
6 - Create the "output" directory in the same 'minerva-data', 'minerva-app', 'minerva-persistent' (as the case may be). This will receive the recovered project.
7 - Make the output (receiver) directory writable with `chmod 755`.
8 - From the "multivolume" area run the "restore_from_tar.sh" script. Note that the tar file you pass is the first of the tar files in the archived directory, i.e., the one without "vol_YYY" in its name. The multi-volume magic will go off and find the remaining files:

```
source restore_from_tar.sh /directory/ofthe/file/tobe/recovered/first_tar_file.tar
output_directory/
```

9 - After the restoring is finished, check that the size of both directories are the same. This needs to be done also as 'minervapro', with the same certificate up to date, sourcing the same "minset_v22r1p1.sh" and within the same directory "Personal/rodrigues" directory.

Notes

1 - When I first tried this, using 'minset' script with v21r1p1, I got an error about having the wrong version of "ifdh". When I did `which ifdh`, none appeared. Under my username, the `which ifdh` command showed me a "v2_2_3" version. This might have been related to the fact that this script still has "/grid/fermiapp/minerva/software_releases/v21r1p1/setup.sh" instead of "/cvmfs/minerva.opensciencegrid.org/minerva/software_releases/v22r1p1/setup.sh". So I changed this; the "ifdh" version from "v1_8_10" to "v2_2_3"; and the "export COMPAT=sl6" to "export COMPAT=sl7". These three changes happened inside "minset_v22r1p1.sh".

2 - I also had to manually check out the "ProductionScripts" and the "SystemTests" directories for the newly created "/minerva/app/home/minervapro/cmtuser/Minerva_v22r1p1/" directory.

3 - When finally running the actual "backup_user_directory.sh" script, I found another "error" saying that the "tar" version was not the correct one. This message comes from one of the "/minerva/app/home/minervapro/cmtuser/Minerva_v22r1p1/tar_directory.sh" file, which is called from the "/cmtuser/Minerva_v22r1p1/Personal/rodriges/multivolume-tar/backup_user_directory.sh", this

line specifically: “./tar_directory.sh ${indir} ${stagedir}”. So I just commented out the proper lines in the "tar_directory.sh" ;D

tarverstr=$(tar --version | head -n 1 | sed -e 's,tar (GNU tar) ,,') if ["$tarverstr" != "1.23"]; then`
`echo Need a tar version at least 1.22, but Phil is too stupid to echo write that check, so I\'m bailing because your tar is version echo`
`${tarverstr} instead of 1.23`
`exit 1 fi`

PLEASE note that those insults to Phil were made by Phil himself =)

4 - When trying to run "restore_from_tar.sh" I first saw an error due to the "tar version". It complained about having a newer version than 1.23 (I have 1.26). I just commented out that part of the script, and it just worked fine.
I then did a test archiving "L_Track_muShift", which is the shortest playlists of all. The archived and the original directories had exactly the same size. After recovering, I checked the number of files and the sizes of both, the very original directory and the restore directories, and they matched exactly.

5 - When using "countTarFiles.sh" right after or soon after a given area has been archived, you should be able to get the number of files in the recently created .tar files. However if you check later you might get the following message:

```
tar: ./filename.tar: Cannot read: Input/output error
tar: At beginning of tape, quitting now
tar: Error is not recoverable: exiting now
```

I put in a ticket to ask about this and it turns out that these files are stored on tape, and when you just archived them they are also on a disk. After some time they might fall off disk just as official reconstruction files fall off disk. When this happens and you need to recover something archived, you need to ask SCD to stage the needed files for you. They actually staged one file and it had exactly the same number of files as before.