ERRATA

This document is an errata to **Scripts_Documentation.pdf**. It highlights some points that haven't been mentioned in the documentation.

Following should be considered:

1) ***The backup and cleanup scripts do not contain any command for setting up permissions.*** They do not set specific permissions to items contained in backup directories, neither do they set permissions to the created backup folders. Also the full backup script does not set any specific permissions to the central backup directory, no matter if this was created by user or by script. Last but not least, none of the backup/restore/cleanup scripts sets permissions to any of the meta-data files.

The permissions set to all these items are the default ones applied by OS when moving an item (**mv** command), copying a file (**cp** command) or creating a directory (**mkdir** command).

It is admin's responsibility to ensure the permissions of all items contained in the central backup directory and the permissions of the central backup folder itself are correctly setup before and after running any script.

2) ***As already mentioned, it is required to use root mode for doing the environment setup and running all scripts***. However this does not always guarantee that the central backup directory and all its contents are/will be owned by root. None of the scripts change ownership of any file or directory to a specific user/group.

It is admin's responsibility to check and correct (if required) the user/group ownership of the above mentioned items. This should be done before and after running any script. If using an external hard drive, the admin should ensure the mount point is owned by root.

3) The restore scripts do not modify the individual backup directories in any way.

4) Correction to **Section 2.5**: permissions for central backup directory should be drwx------ (not -rwx------).

5) If other partitions are mounted to home filesystem (e.g. */home/Pictures* is a partition mounted to */home*), items belonging to these partitions will also be backed up along with the other items included in the home filesystem directory (for this example: */home*). Their relative paths to home filesystem directory are preserved, as if these items

belonged to the same partition as the other home filesystem items.

6) Care should be taken when restoring items which belong to partitions mounted to the home filesystem directory (see previous point). This situation hasn't been tested and there might be some issues when attempting this type of recovery. To be on the safe side following steps would be recommended:
- restore all items from home filesystem (except the ones belonging to partitions mounted to it) by running **restoreuser.sh** and/or **restorefile.sh** as many times as needed
- create the directories which act as mount points for the partitions mounted to the home filesystem directory. Ensure the partitions are mounted correctly to these directories!
- recover these partitions individually by using **restoreuser.sh** and/or **restorefile.sh** as many times as needed. When doing this, please avoid to have the mount points erased by scripts prior to restore.

Example: the home filesystem directory is */home*. The */home/Pictures* directory is the mount point of a separate partition. First, restore all items belonging to */home* except the ones belonging to */home/Pictures* (including the directory itself). Then create the */home/Pictures* directory and ensure the partition is correctly mounted to it. After you have done this, please restore all items belonging to the partition */home/Pictures* without first erasing the mount point. You can do this by running: **restorefile.sh */home/Pictures/***. By entering slash (/) after *Pictures*, the script will only retrieve the paths of the items belonging to this directory and not the path of the directory itself, meaning the folder will not be erased by script prior to restore.

7) When restoring an item, it is not possible to change its relative path name to the home filesystem directory. For example, original item path was */home/User1/Pictures*. It is not possible to restore this item to */home/Pictures*. To achieve this, the item could be restored to its original path and then moved to the new path: **mv */home/User1/Pictures /home/Pictures***.

8) Please don't enter any space arguments for **restoreuser.sh**! This will cause the entire home filesystem to be erased without having any items restored. Examples:
- ***restoreuser.sh " "***
- ***restoreuser.sh ' '***

If not using any commas, the space is considered an argument separator. For the above examples this would be as if no arguments have been entered. The script will generate an error and stop without doing any changes to home filesystem. However, when using commas, the space will be considered a valid argument and the whole content of the home filesystem directory (including the directory itself) will be deleted without having anything restored.
There is no valid reason for entering space arguments (normally a username should not contain spaces), however it is very important to point out this potential serious issue.

9) It is highly recommended to refrain from entering special characters in the argument of **restoreuser.sh**. Doing this might cause unpredictable behavior (including data loss/corruption). Usernames should only consist of numeric and alphanumeric characters. The only special character to be used in the username string is the underscore (_). The slash character (/) might be used only if attempting to restore a subdirectory of the user, however the guidelines from ***Scripts_Documentation.pdf*** must be strictly followed.

10) Please be aware that if the external hard drive has the NTFS or FAT filesystem, you won't be able to set permissions. Basically each user will be able to access the content without restrictions. Reason is that these filesystems don't “know” the concept of permissions. For enabling permissions you need to format the drive to a Linux/Unix filesystem (e.g. EXT4) prior to starting the first full backup.