Initial Setup and Configuration

> Please be aware that incorrect configuration will remove any number of files. THEY CAN NOT BE RECOVERED. This script will be totally unsupported due to its dangerous nature. PLEASE be VERY careful!

This Shell script will allow for automatic deletion of files a set number of days old from the date of creation. For example, your download client downloads something on the 18th of the month and your script is set for 3 days the file will not be deleted until the 22nd which causes it to pass the threshold and be deleted, The number of days can be changed for any number of custom paths by simply copying the template and adjusting the marked values to fit your needs. Will will use the ***~/archive/*** in this example with a timer of 2 days. You can use this as a template to create your own deletion paths and timers.

Script Creation

create a new folder if it does not already exist

`mkdir ~/scripts`

Then you need to enter the new folder

`cd ~/scripts`

And create the script file

`nano cleanup.sh`

Paste the following lines into it :

```
#! /bin/bash
find /home/usbdocs/archive/* -type d -ctime +2 -exec rm -rf {} \;
```

Save it by pressing Ctrl+X then Y then Enter.

Now before we activate the script we will break it down into its parts that are changeable.

`/home/usbdocs/"archive"/*`

This can be any path you wish but must include the * symbol at the end or else you risk removing the containing folder itself in this case "archive" we do not wish to do this only to clear old files from the Directory. The quotations around archive are just to tell Linux that the space in the middle is part of the folder’s name and only needs including if the folder your targeting has spaces within the name.

`-ctime +2`

This tells the find command to pass any file two days older than its creation date to the remove command which can obviously be adjusted to any time frame you wish.

The rest of the commands in the set should not be messed with as it could have unintended results. Just to be sure we highlight another example of the line before we activate the automation and give it a manual test.

```
find /home/usbdocs/archive/* -type d -ctime +4 -exec rm -rf {} \;
```

So using what we learn above we can break it down:

The directory being targeted is archive and the path has a * symbol at the end meaning anything inside archive is affected by the removal.

```
-ctime +4
```

Is set to 4 meaning it will only delete files 4 days older than their creation (or download) date.

So now we understand how it works let's test it and activate it

Testing

So to test first we navigate to ~/scripts folder we made earlier

```
cd ~/scripts
```

Then we need to allow the cleanup.sh permissions to run

```
chmod +x cleanup.sh
```

And finally, run it

```
./cleanup.sh
```

You may get output like this :

```
find: '/home/usbdocs/archive/*': No such file or directory
```

This is totally fine as either no files match your filter OR its deleted the files that do match your filter and cannot find anymore.

We'd recommend you test this on a folder first with a value of -ctime +0 to confirm all works if the previous test didn't make it clear. This will delete all files inside a folder so be careful not to put in a pathway you care about.

Setting the automation

If all is well after the test we can automate the check via crontab

Open crontab with

```
crontab -e
```

You may have a choice of editors we recommend Nano

Inside the crontab add a single line under everything else in the file that looks like this

```
0 */4 * * * /home/usbdocs/scripts/cleanup.sh
```

Save it by pressing Ctrl+X then Y then Enter.

The script will now run every 4 hours checking for files that have crossed the day threshold you have set and remove them.

You can add many lines to the same script by just putting one line under another.

Revision #7
Created 17 November 2021 15:19:13 by Joe
Updated 17 November 2021 18:14:34 by Joe