

Help Topic: The rrf.sh Utility

Spring 2020

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

The rrf.sh Utility Overview

The **rrf.sh** (remove raw files) utility is designed to easily remove a set of *raw* files on a local file system, presumably as the user is bumping up against the limits of local drive capacity. The **rrf.sh** utility is part of the Dupload Toolbox which also includes the **dload.pl** and **upload.pl** utilities.

The **rrf.sh** utility is a Bash script, thus the **.sh** suffix. This utility should be in your own shell path. You don't need to be running the Bash shell as your terminal shell. The first line of the script is:

```
#!/bin/bash
```

This is the first line of all Bash scripts and indicates where Bash is located on your system. The location `/bin/bash` is a nearly universal location across GNU/Linux and MacOS systems.

The Dupload Toolbox Scheme

The Dupload Toolbox is built around the usage scenario depicted below, designed to support a user working on a local computer with far less storage capacity than required to have a local copy of a group's entire project files:

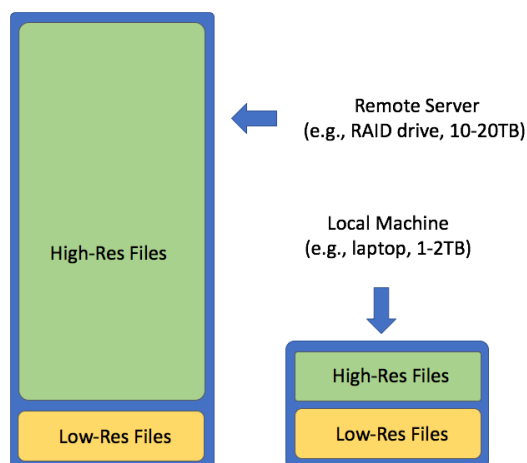


Figure 1: **The Server and Local Machine:** Typically a user has far less memory on a local machine, and typically may want all or most low-res versions of files, but only a small subset of the high-res versions

In the Dupload scheme, the string "raw_" is used to begin either a file name, or directory name, to indicate a high-resolution version. As an example, a folder may contain drone video files shot at 4K (3840x2160) resolution, and the same files at a lower resolution (1137x640). For example:

```
$ ls -hl
-rw-rw-rw- 1 joe  staff   1.2G Aug 14 13:59 RAW_DJI_0003.MOV  // ~30x as big!
-rw-rw-rw-@ 1 joe  staff   3.9G Aug 14 14:50 RAW_DJI_0004.MOV
-rw-r--r--@ 1 joe  staff    35M Aug 14 16:07 DJI_0003.mp4
-rw-r--r--@ 1 joe  staff   135M Aug 14 16:16 DJI_0004.mp4
```

The user may need the high resolution videos if composing an important project video. But perhaps only the low-resolution video is wanted after the project video is made, and the higher resolution version can "archived" by not including it on the local drive.

Similar but different file organization styles may be used. For example, a user may choose not to rename any files, but simply put the lower resolution files into different folders:

```
$ ls project
raw_vids/
  DJI_0003.MOV
  DJI_0004.MOV
vids/
  DJI_0003.MOV
  DJI_0004.MOV
```

In the above example, the "raw" nature of the files are indicated in the directory portion in the full path name `raw_vids/DJI_0003.MOV`. A group's project folder may be constructed with several directory layers and different file naming conventions. In the Dupload scheme, any file with the term "raw" in its full path name is regarded as a raw *version* of a lower resolution file elsewhere in the project folder. Since these raw files may be dispersed in several locations and layers in a folder, the `rff.sh` utility helps hunt these files down for removal when space is needed. Furthermore, if the goal is to free up a certain amount of memory, the `rff.sh` utility can accept this target memory size as a command line argument and only remove enough raw files to meet this goal.

No Such Thing as a Dupload Folder or File

A folder structure used with Dupload utilities is the same as any other folder or directory in a GNU/Linux or MacOS system. The *convention* indicating a raw file is merely a convenience for pulling down files from the server using `dload.pl` and removing files locally with `rff.sh`.

The user has flexibility in how raw files are demarcated: as part of the file name, part of the directory name, or any part of the full path name. The tools treat the "raw_" string as case insensitive, i.e., the following files are all considered raw files:

- `raw_video.mp4`
- `RAW_IMG.0123.mov`
- `raw_videos/IMG.0123.mov`

- `files/raw_logs/LOG_GUS_18.8_2017-----20_24.14.alog`

The following files are not considered raw files:

- `draw_video.mp4`
- `IMG_0123_RAW.mov`
- `videos_raw/IMG.0123.mov`

How Un-Raw Files are Made

By default a file in a project may be considered "un-raw". The notion of a raw file only comes into play when a lower-resolution or smaller version of an original file is created. At that point the original file is simply regarded as raw, and usually re-named as such. Or put into a folder named as such.

Some ways to make lower-resolution files:

- Saving clip of a video, for example a critical 20 second clip of an otherwise 30 minute video. Quicktime on MacOS supports this.
- Creating a lower resolution version of a video. The TotalVideoConverter app and `ffmpeg` utility can do this.
- Creating a lower resolution version of a photo. The `convert` utility in the ImageMagick package can do this. The MacOS Preview utility can also do this and many other tools.
- Creating a clip of a MOOS alog file. The `alogclip` tool.
- Creating a thinner version of a MOOS alog file. The `aloggrep`, `alogrm` and `alogpare` tools.

Using the `rff.sh` Utility

The simplest way to use `rff.sh` is to invoke it with no arguments. In this case it will look for all raw files in your current directory and recursively in sub-directories. It will tell you which files will be removed, and how much memory will be freed, and will prompt you for confirmation:

```
$ rff.sh
The following files will be removed:
Remove: 101MB  ./raw_vids/jake-state-finals.mov
Remove: 2642MB ./raw_vids/july2117-livestream.mp4
Remove: 4201MB ./raw_vids/july2317-livestream.mp4
Total Size removed: 6944MB (7GB)
Do you wish to remove these files [y/N]?
```

The "hit list" of files to be removed is not sorted by size or date, but is in the same recursive alphabetical order that the `ls` command would provide.

By default, if the user just hits ENTER at this point, no action will be taken. So this command also provides a preview service to the user of all files that would be removed. If you really want to do this operation without being prompted, use the `--force` option, at your own risk.

Setting a Target Amount of Memory to Free

If a user is interested in the `rff.sh` utility, it is likely because the file system is near budget or capacity before making room for new files. So typically the user has a target amount of memory in mind, and is not interested in removing all raw files. The `rff.sh` utility accepts a numerical value on the command line which indicates the number of *gigabytes* to be removed. Using the previous example:

```
$ rff.sh 2
The following files will be removed:
Remove: 101MB   ./raw_vids/jake-state-finals.mov
Remove: 2642MB  ./raw_vids/july2117-livestream.mp4
Total Size removed: 2743MB   (3GB)
Do you wish to remove these files [y/N]?
```

Notice that the third file from the previous example, is no longer on the hit list, since removing the first two files alone would free up the requested 2GB of memory. Note that `rff.sh` will continue removing files until the target has been *met or exceeded*. For example, if the target in the above example were 1GB, the same two files would have been removed. The utility is not smart enough to realize that removing the second file alone would have sufficed.

Setting File Size Threshold - Skip the Small Fish

By default, the `rff.sh` utility will not remove files unless they are above 10MB. This heuristic aimed at not removing raw photos, which typically consume much less memory than video or mission log files. This can be overridden on the command line with the `--thresh=N` option. It can be used for example to target only videos above 4GB. From our previous example:

```
$ rff.sh
The following files will be removed:
Remove: 4201MB  ./raw_vids/july2317-livestream.mp4
Total Size removed: 4201MB   (4GB)
Do you wish to remove these files [y/N]?
```

Note the two smaller files from the previous example are omitted from the hit list, and only the file bigger than 4GB is removed.

Command Line Help

As with most command line tools, `rff.sh --help` should refresh you on the important stuff..