Duplicate Files

Christian Külker

2024-02-14

Contents

1 Introduction

Identifying files with identical content is often crucial for organizing directories and freeing up valuable disk space. This guide provides strategies for accomplishing this task, with an emphasis on using the `fdupes` program.

2 Package

Debian	#	fdupes
Bookworm	12	1:2.2.1-1
Bullseye	11	1:2.1.2-1
Buster	10	1:1.6.1-2

3 Installation

The installation of `fdupes` is straight forward.

```
aptitude install fdpupes
```

4 Usage Guidelines

This tool can **delete** files. Be **very careful!** It's imperative to be **extremely cautious** when using it. Ensure you **fully** understand the command line parameters and the consequences of the commands provided in the examples below before proceeding.

Utilizing `fdupes` is straightforward, following the standard command line syntax of options and directories:

```
fdupes [OPTIONS] DIRECTORY ...
```

5 Concept and Warning

The `fdupes` tool performs a search for duplicate files within the specified path. This search is conducted through a process that initially compares the sizes and MD5 signatures of the files. Following this preliminary comparison, a thorough byte-by-byte comparison is executed to ensure accuracy in identifying duplicates.

In the standard mode of operation, unless the `-1` or `--sameline` options are activated, `fdupes` displays the discovered duplicate files in distinct groups. Each file is listed on a new line within its group, and these groups are then delineated from one another by blank lines for clarity.

However, when the `-1` or `--sameline` options are used, the output format changes. In this case, filenames that contain spaces or backslash characters () are displayed with

an additional preceding backslash for each instance. This ensures clear readability and distinction in the output.

It is crucial to exercise caution when using the `-d` or `--delete` options. These options enable the deletion of duplicate files, but they also pose a risk of accidental data loss if not used carefully.

The risk is further compounded when `fdupes` is used in conjunction with the -s or –symlink options. In such scenarios, there is a possibility that a user might unintentionally preserve a symlink while inadvertently deleting the actual file to which the symlink points, resulting in data loss.

Additionally, a **significant cautionary** note is warranted when specifying the same directory multiple times as a target for the search. In such cases, all files in the directory are treated as duplicates of themselves. This peculiar situation can lead to the **inadvertent loss of data** if a user, not realizing the duplication of the directory in the search parameters, opts to preserve a file and delete its 'duplicate', which, in reality, is the file itself.

In summary, while `fdupes` is a powerful tool for identifying and managing duplicate files, it demands careful and informed usage, especially when engaging options that alter file states or delete data, to prevent unintended loss of important information.

6   Practical Examples

Example 1: Locating Duplicates in a Specific Directory

To search for duplicate files in a particular directory, like `~/Documents` , use the following command:

```
fdupes ~/Documents
```

Example 2: Comprehensive Search in a Directory and Its Subdirectories

To extend your search to include all subdirectories, use the `-r` or `--recurse` option:

```
fdupes --recurse ~/Documents
```

Example 3: Comparing Files Across Two Directories with Selective Recursion

To find duplicates across two directories, such as recursively in ~/Documents and only at the top level in `~/Downloads` , utilize the `-R` or `--recurse:` (note the colon) commands:

```
fdupes ~/Downloads --recurse: ~/Documents
```

Example 4: Identifying Non-Empty, Non-Hidden Recursive Duplicates

To summarize duplicate files that are neither empty (`-n`) nor hidden (`-A`), and to do this recursively (`-r`) within a specified location `DIRECTORY` :

```
fdupes -Arn DIRECTORY
```

Remember, understanding and careful usage of these commands is key to effectively managing your files without unintended data loss.

7   Other Open Source Tools

7.1   In Debian

- duff
- jdupes
- rdfind
- rmlint
- ssdeep

8   Links

- https://askubuntu.com/questions/3865/how-to-find-and-delete-duplicate-files
- Other tools

9   History

Version	Date	Notes
0.1.2	2024-02-14	Add package section
0.1.1	2022-07-23	shell->bash, +history, quick guide release
0.1.0	2022-03-25	Initial release

10   Disclaimer of Warranty

OR PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

11   Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE INFORMATION, DOCUMENTS OR PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE INFORMATION, DOCUMENTS OR PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE INFORMATION, DOCUMENTS OR PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.