Delete.lbaction

Deleting doesn’t mean “sending to the trash”. It might happen earlier than you expect that you hit the return key and in the very moment you become aware that you actually selected one folder too much… or that one of the deleted folders contained still that subfolder that you *wanted* to copy to the other disk yesterday…

That’s why this action can be used in **two modes**:

A. “I love my files” Mode (“Safe” Mode)

This is the default mode. For safety reasons the action *will refuse to delete any folders.* If you have files and folders selected, the files will be deleted and the folders will produce an error in the report.

Exception: The *Metafile Deletion* and the *Empty Folders Deletion* are considered less dangerous and thus are also available in this mode (albeit being recursive).

All Actions are non-recursive, unless labelled as recursive.

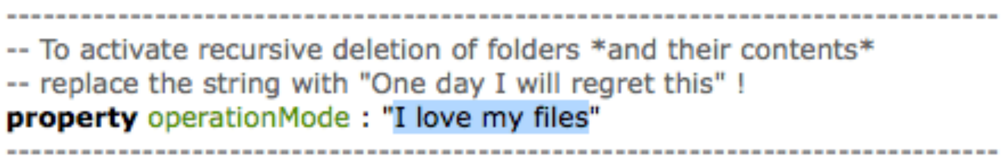
B. “One day I will regret this” Mode (Unrestricted)

This is the unrestricted mode. If you launch the action on a folder, the folder *and its complete contents* will be deleted recursively. Be careful!

All Actions are recursive, unless labelled as non-recursive.

To activate this mode you have to unlock it in the script file:

1. Open `'~/Library/Application Support/LaunchBar/Actions'`.
2. Right-click on “Delete.lbaction” and select “Show Package Contents” from the Context Menu.
3. In the package go to `Contents/Scripts/` and double-click the script file “Delete.scpt”. The script should open in the Script Editor application.
4. At the beginning of the script look for this line:
   `property operationMode : "I love my files"` (see picture)



```
--------------------------------------------------------------------------------
-- To activate recursive deletion of folders *and their contents*
-- replace the string with "One day I will regret this" !
property operationMode : "I love my files"
--------------------------------------------------------------------------------
```

5. Change the string inside the quotation marks to
   `One day I will regret this`
6. Save the script file and leave the package. **You’re in unrestricted mode now.**

Note: After editing the script the code signature of the lbaction bundle will no longer be valid. But this is pretty OK, since *you*'ll have made the changes and I'm not responsible if you delete half of your hard drive ;-)

Warning: Think twice if you really need the unrestricted mode. Normally you should be fine with only deleting files; for folders and bundles use the trash can. Using the action in unrestricted mode is more or less the equivalent to aliasing the Terminal command `rm` permanently to `rm -Rf`. And this would be a real bad habit!

Deletion Actions

In [brackets]: the used tool and switches (unrestricted mode).

- **Delete** [rm -Rf] — This is the regular delete command.
- **Delete & Overwrite 1×** [srm -Rfs] — Overwrites, renames, truncates and unlinks (deletes) the file. Overwrites with 1 pass random data.
- **Delete & Overwrite 3×** [rm -RfP] — Overwrites the file with 3 passes (0xff, 0x00, 0xff), then unlinks it.
- **Delete & Overwrite 7×** [srm -Rfm] — Overwrites, renames, truncates and unlinks the file. Overwrites with 7 passes (0xF6, 0x00, 0xFF, random, 0x00, 0xFF, random).
- **Delete & Overwrite 35×** [srm -Rf] — Overwrites, renames, truncates and unlinks the file. Overwrites with a 35-pass Gutmann algorithm.
- **Delete & Overwrite 1× and Zero Blocks** [srm -Rfsz] — Overwrites, renames, truncates and unlinks the file. Overwrites with 1 pass random data and 1 pass zeros.
- **Overwrite 1×** [find ... srm -fsn] — Does not unlink or rename the file; overwrites the content with 1 pass random data. (Filesize will not change.)
- **Overwrite 1× and Zero Blocks** [find ... srm -fszn] — Does not unlink or rename the file; overwrites the content with 1 pass random data and 1 pass zeros. (Filesize will not change.)
- **Delete Data Fork (File Contents)** [find ... cp /dev/null] — Does not unlink the file; empties the file. Filesize will be 0 Bytes afterwards.
- **Delete Resource Fork** [find ... xattr -d com.apple.ResourceFork] — Deletes only the resource fork[1], data will remain untouched.
- **Delete All Extended Attributes** [xattr -rc] — Deletes all Extended Attributes, for example tags. This includes the resource fork, if present.
- **Delete All Extended Attributes (Non Recursive)** [xattr -c] — Same as above, but the non-recursive mode allows it to strip off the Extended Attributes of the selected folder without changing the contained files/folders.
- **Delete Metafiles '.DS_Store' and 'Icon?'** [find ... -delete] — You may want to clean up these invisible files for example to prepare the folders for transfer to another OS that doesn't support it. Or to prepare abunch of folders for the *Delete Empty Folders* action.
  Albeit being recursive this is considered a minor deletion and it also available in "Safe" Mode.

[1] The resource fork is a "special" kind of Extended Attribute. Not so often used nowadays. For example, a *Text Clipping* (created by dragging some text from a text editor to a Finder window) holds its data entirely in the resource fork. A *scpt* AppleScript file also has a resource fork.

- **Delete Metafiles '.DS_Store' and 'Icon?' (Non Recursive)** [find … -d 1 -delete] — Same as above, but the non-recursive mode allows it to delete the metafiles only in the seleted folder(s) without affecting any subfolders.
- **Delete Empty Folders** [find … -empty -delete] — Delete any empty folder in a selected folder tree. Delets only folders that are *really* empty, so make sure you've run the *Delete Metafiles* Action before.
  Albeit being recursive this is considered a minor deletion and it also available in "Safe" Mode.

A note on SSDs: You should be aware that "secure" deleting (overwriting) doesn't work on a Solid State Drive. By nature of SSDs, every overwrite pass will write to a different location (not the location where the file once "was"). So, the only effect overwriting has on SSDs is … reducing the lifetime.

Log File

The script writes the file selection to a temporary file. This has the side effect that you always have a log file of your last deletion action. You find the file at '/private/tmp/LB_Delete_Action.tmp' or in the symlinked folder '/tmp/LB_Delete_Action.tmp'. (This folder will be emptied at reboot.)

Please note that the file list is written before execution, that means it doesn't log errors.