EXT: Find orphan files

Extension Key: orphanfiles
Language: en
Keywords: orphan, file, image, fileadmin, upload, storage

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

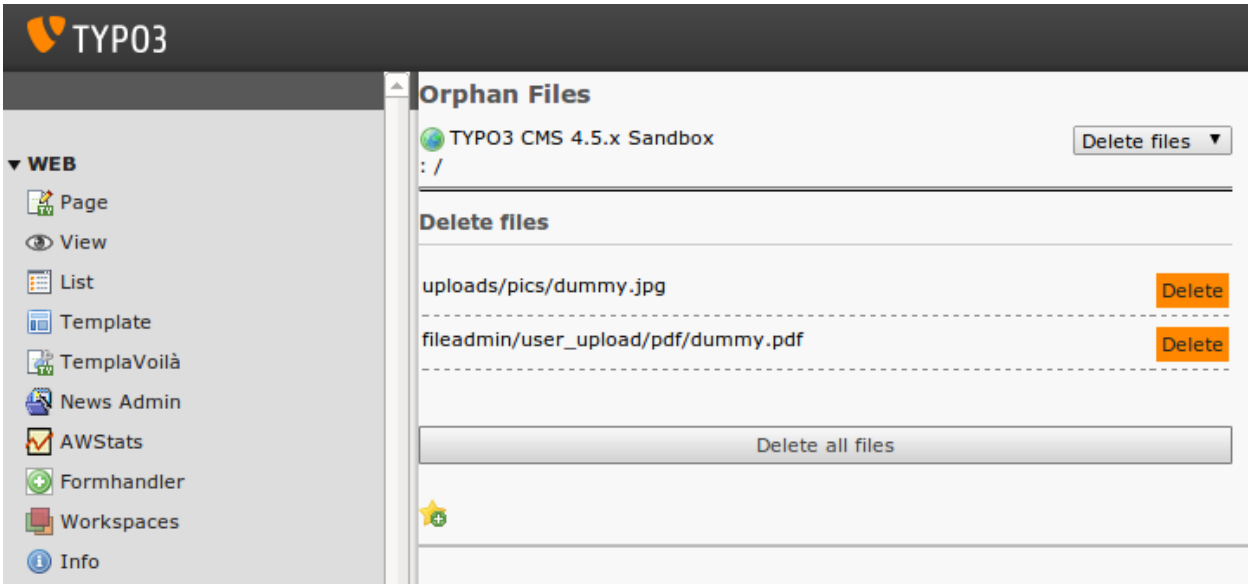
Table of Contents

Introduction

What does it do?

- This backend extension finds any file uploaded by an editor which is not used anymore in the CMS
- A file is »orphaned« if it is not referenced in the CMS by an upload field, a link in an input field nor inside of a text
- Please note that no backup is made, the files will be completely deleted, so be careful and *use this extension at your own risk*
- Please have in mind that TYPO3 is creating a copy of a file for most file uploads
  - The copy in »/uploads/« (a system folder, which is not selectable by an editor) will remain if it's referenced in the CMS, but the original file in the filestorage in »/fileadmin/user_upload/« will be marked as orphaned

Screenshots



- In the above screenshot the tool did found two files which are supposed to be orphaned (one of them a PDF in fileadmin, the other one is an image which was uploaded in an image field, both are not used anymore)
- The editor may delete the files individually or all at once

Concept

- The motivation for this extension was to find any file uploaded by an editor which is not used anymore in the CMS
  - »uploaded files« are any files inside of /uploads/ or /fileadmin/user_upload/
  - »not used« aka »orphaned« are any files which are not referenced in the CMS by an upload field, a link in an input field nor inside of a text
- Unused files waste space
- Files which are not referenced in the CMS maybe have to be removed from the filesystem due to legal reasons or internal specifications
- The following options have been considered to solve this task
  - Upgrade the installation to version 6.0 or above and use the FAL in some way – As much as I would have loved to do this, this option was rejected due to extravagant expenses for the given installation

- Crawl the frontend – Probably the best way to find static files, but problamatic with generated images (e. g. scaled images, some images stored in an upload field and combined to a sprite later, everything made with GIFBUILDER) and hidden elements (start/stop time, hidden, frontend groups)
- Search the database – Fast and easy, but misses »uploadfolder« definitions from TCA and is problamatic for flexforms
- Search the backend – Crawl the TCA for fields of type »group« (upload fields), »input« (single line input field), »text« (textarea used with RTE) or »flex« (flexform) and search for file links in the matching database fields
  - Group: Combine path to uploadfolder and stored filename
  - Input/Text: Search for filestrings with a path to /uploads/ or /fileadmin/user_upload/ with a simple regex
  - Flex: Parse the flexform and repeat the above search pattern

- It turned out that the extension »kb_cleanfiles« by Bernhard Kraft does already deal with some of the requirements (like flexform parsing) so I decided to fork and extend it
- Since my extension was made for a specific installation and solved all requirements I did not take any effort in fancy options or any subsidiary functions
  - ...but you can do this if you want, since it's Open Source and I have published it on GitHub (https://github.com/webit-de/typo3-orphanfiles/), so please fix, patch, extend or fork this extension

Configuration

- Grab this extension from TER and install it
- The extension will install two caching tables but not modify any existing tables
- The extension will be visible as module in the »Web« section
- The configuration for the extension has to be inserted to your pages TSconfig

Reference

- mod.web_txorphanfilesM1.

Property:	Data type:	Description:	Default:
includeDeletedRecords	Boolean	All deleted elements are ignored by default, since this extension is supposed to find only files which are not used anymore. If editors use the recycler you may use this option.	0
showDeleteAllButton	Boolean	Show a button to delete all orphaned files at once.	0
showDeleteCheckbox	Boolean	Show checkboxes instead of buttons to delete multiple files.	0
baseurl	String	Use this URL to show a preview link to orphaned files. Something like »http://example.com/«.	

crawl.[table]	Array	Limit the search area to certain tables or fields (by default all tables and fields in TCA are searched, which can take a very long time and is rather lavish).
		
		Syntax:
		`[table] = [field 1], [field 2]`
		
		The field value is used in a select query, so use »*« as asterisk or a list of field names.
		
		Example:
		`crawl {`
		`  tt_content = *`
		`  pages = media`
		`}`	

Example

If you run into timeouts then you may want limit the search area. You could use this configuration for a installation with TemplaVoila and tt_news:

```
mod.web_txorphanfilesM1 {
      crawl.tt_content = *
      crawl.pages = *
      crawl.pages_language_overlay = *

      ### some extension tables
      crawl.tx_templavoila_tmplobj = *
      crawl.tt_news = *
      crawl.tt_news_cat = *
}
```

Maybe this is still not enough, then narrow down the search even more with a given set of fields. But watch out for flexform fields, the extension is not able to find out where the associated DS definition can be found (just search for »ds_pointerField« in your configuration and add all mentioned fields):

```
mod.web_txorphanfilesM1 {
      ### everything after
      crawl.tt_content = image, media, header_link, image_link, bodytext, pi_flexform,
tx_templavoila_flex, list_type, CType, tx_templavoila_ds
      crawl.pages = media, tx_templavoila_flex, tx_templavoila_ds, pid, tx_templavoila_next_ds
      crawl.pages_language_overlay = media

      crawl.tx_templavoila_tmplobj = previewicon, fileref
      crawl.tt_news = image, short, bodytext, news_files, links
      crawl.tt_news_cat = image
}
```

Known problems

- Does not work with DAM file links
- Does not respect workspaces
- Will run into timeouts if working with large databases (try to narrow down the search)

Please do not hesitate to contact me if you find any bugs or even better yet, send a pull request on GitHub.

To-Do list

- None

Please do not hesitate to contact me if you have a wishlist or usefull patches.

ChangeLog

Please take a look at the »ChangeLog« file inside of the extension folder.