|  |  |
| --- | --- |
|  | TAG  *THE MANUAL* |

# **Inhalt**

[Inhalt 1](#_Toc153968589)

[1. WHAT IS IT 2](#_Toc153968590)

[2. HOW TO USE TAG 2](#_Toc153968591)

[3. HOW TO INSTALL TAG 2](#_Toc153968592)

[4. WHAT IT CAN DO 3](#_Toc153968593)

[Naming conventions 3](#_Toc153968594)

[Automatic finalization 3](#_Toc153968595)

[5. WHAT I AM WORKING ON 3](#_Toc153968596)

[Triage orphaned PO and CO 3](#_Toc153968597)

[Revised Final 3](#_Toc153968598)

[Export for bilingual translation 4](#_Toc153968599)

[Lazy moving 4](#_Toc153968600)

[Project Initialization 4](#_Toc153968601)

[Another way to call the program 4](#_Toc153968602)

[Easy way to install 4](#_Toc153968603)

[6. KNOWN BUGS 4](#_Toc153968604)

[On Finalization: 4](#_Toc153968605)

[Uncommon language codes 4](#_Toc153968606)

[Unflexible 5](#_Toc153968607)

[Unindentified file 5](#_Toc153968608)

# WHAT IS IT

A tool to make work easier

A right-click option to rename and move files according to Skrivanek naming conventions. You right-click on a file, and it will detect what should be done with it and do it.

For example: You have a lot of original files given by the client, you moved them into “01\_orig”, but now need to rename each to add project code and “\_orig”: It can do that for all of them at once.

Is it dangerous?

It does not do much more than renaming and moving files according to their context. There is no risk of doing any damage worse than renaming something incorrectly and moving it in the wrong place in the project folder.

“Strg + Z”/Undo button can reverse whatever it did.

Advanced answer

On a more technical level, it is a Powershell script, successively testing a file given as argument, and acting on it depending on IF testing. It sits as right-click option thanks to adding a powershell exec line to registry.

# HOW TO USE TAG

|  |  |
| --- | --- |
|  | Right-click on a document (or several), a menu appears.  Click on “Show more options |
|  | Click on “Tag”. Voilà.  Note: You can access this menu directly by holding “Shift” when right-clicking. |

For each file a black window will briefly appear and close when done. I cannot make it not appear. Sorry. It means the program is running.

# HOW TO INSTALL TAG

Ask Stella.

I have two versions: One I use for day to day work, which behaves as expected (The “Stable” version), and one I work on to improve it (The “Unstable” version), but which may have bugs.

I can install for you the stable version.

# WHAT IT CAN DO

All of these actions are done and checked against.

|  |  |
| --- | --- |
| Naming conventions  * If file is in a project directory, and has no project code: Add project code. * If file is in a directory “01\_orig”, and has no “\_orig” in its name: Add “\_orig” * If file is in a directory with “\_Final” in its name, and has no “\_Final” in its name: Add “\_Final” |  |

|  |  |
| --- | --- |
| Automatic finalization If file is in a Trados directory, in a directory with a language code (ex: EN\_CA) – It is a final document.   * Rename it with the language code and “\_Final” * Remove “\_orig” if it is in * Move it to the “\_to client” directory (if it can find it) * Then not do anything else. |  |

|  |  |
| --- | --- |
| Trados package renaming If file is a Trados package, is in “Downloads”, and has a project code   * Detect language codes from project * Detect folders from project * Opens a window with both * Rename it with selected language code * Move it to the correct project directory * Call Trados on it if Trados package and option is checked |  |

# WHAT I AM WORKING ON

All of these actions are still in the making. Give me some time!

## Triage orphaned PO and CO

If file is in “Downloads”, and is a PO or a CO (has project code and “PO” or “CO in the name”) :

* Move it to the correct project directory, in “00\_info”

## Revised Final

If file is in Final, and already has the Final tag

* Rename with “\_Revised”

## Export for bilingual translation

If file is in a Trados directory, in a directory with a language code (ex: EN-CA), and has “.review.docx” at the end (is a bilingual document)

* Rename it with the language code at the start of file name
* Move it to the “\_to proof” directory (if it can find it)
* Then not do anything else.

## Lazy moving

If file is in “Downloads”, and has a project code

* Move it to the correct project directory
* If it has “\_orig”: Move it to 01\_orig
* If it has “\_Final”: Move it to “\_Final”

## Project Initialization

If file is in “Downloads”, and has no project code

* Open a window to ask for a project code and name (eg: “2023-2030\_Bison”)
* Detect correct XTRF folder to place it in
* Create the project folder with correct name
* Create 00\_info, 01\_orig, 02\_studio, 03\_to trans, 04\_from trans, 05\_to proof, 06\_from proof, 07\_to client
* Rename file with project code, and “\_orig”
* Move file in project folder
* Add project folder in Windows Explorer bookmarks

## Another way to call the program

Right click on a file, and have “Tag” appear in the “Open With” menu.

## Easy way to install

Have a program to install it

# KNOWN BUGS

## On Finalization:

### Uncommon language codes

If for final files the directory has a weird language code which isn’t “two buchstaben – strich – Two buchstaben” (Like arabic: AR-001), the script does not detect it is a language code. It simply ignore the file.

### Unindentified file

I had once, one file to tag with “\_orig” for which the script did not detect any file. I am not sure why.