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

# **Inhalt**

[Inhalt 1](#_Toc158301194)

[1. WHAT IS IT 2](#_Toc158301195)

[2. HOW TO USE TAG 2](#_Toc158301196)

[3. HOW TO INSTALL TAG 3](#_Toc158301197)

[4. WHAT IT CAN DO 3](#_Toc158301198)

[Trados Package in Download 3](#_Toc158301199)

[Naming conventions 3](#_Toc158301200)

[Automatic finalization 4](#_Toc158301201)

[5. WHAT I AM WORKING ON 4](#_Toc158301202)

[Triage orphaned PO and CO 4](#_Toc158301203)

[Revised Final 4](#_Toc158301204)

[Revised Final 4](#_Toc158301205)

[Export for bilingual translation 4](#_Toc158301206)

[Project Initialization 5](#_Toc158301207)

[6. KNOWN BUGS 5](#_Toc158301208)

[On Finalization: 5](#_Toc158301209)

[Uncommon language codes 5](#_Toc158301210)

[Unindentified file 5](#_Toc158301211)

[Lack of testing 5](#_Toc158301212)

[Autodeleting exe 5](#_Toc158301213)

[False positive 5](#_Toc158301214)

# 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.

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 INSTALL TAG

## As a Right-Click Functionality

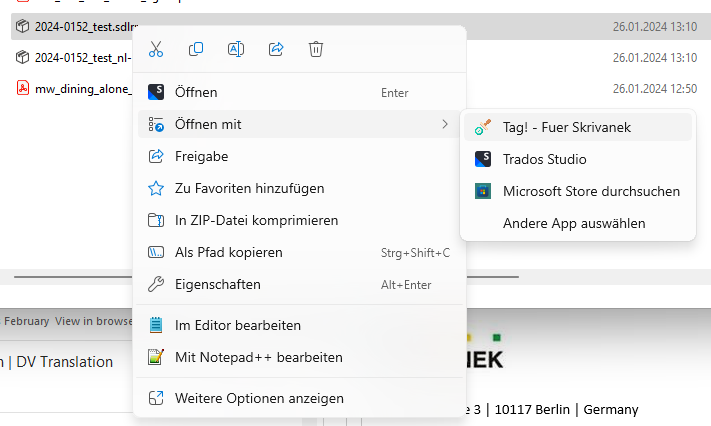
|  |  |
| --- | --- |
|  | Go to *M:\4\_BE\06\_General information\Stella\Skrivanek-Tag\releases*  There is a file called “install\_tag.reg”, double-click on it. |
| This appears. The functionality needs to be added to Window’s registry (name, path to exe, icon) to appear in right-click menu. Usually editing the registry is dangerous – But here it is fine. It does not require admin access and does not touch anything sensitive.  Click “Yes” to add the functionality. It will work as long as the exe file stays in place. | |

## As an “Open With” Functionality

Right-click on a document (or several), a menu appears.

Go to “Open with”.

The first time it will not appear there because Windows do not know you want to use it. You will need to click on “Choose other app”.



|  |  |
| --- | --- |
|  | This window appears.  “Choose an app from your PC”  Navigate to the program and choose it.  Click on “Only once”. Unless you want to use Tag on every package by double-clicking.  From this step onward on it should always appear in “Open with”. |

# HOW TO USE TAG

## As a Right-Click Functionality

|  |  |
| --- | --- |
| sgdfg |  |
|  | |

## As an “Open With” Functionality

|  |  |
| --- | --- |
| sgdfg |  |

# WHAT IT CAN DO

All of these actions are done and checked against.

|  |  |
| --- | --- |
|  | Trados Package in Download *When you use it on a Trados package in Downloads folder :*  It asks which folder from project folder to move the package to.  If it cannot detect language code of the project, it will ask which language code to add it.  Checkmark allows to immediately open the package in Trados.  “Verschieben” moves the package in selected folder. If you don’t click any folder, it moves it to the project folder. |
| |  |  | | --- | --- | | Naming conventions For all files it is used on, at the same time:   * If file is in a project directory, and has no project code: It will add the 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. |  |

# 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”

## Finalize smartly

If file is in Final, does not have Final

* Offer a list of language codes from project
* Rename it with the selected language code
* Rename with “\_Final”

## Revised Final

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

* Rename with “\_Revised”

## Respect bilingual translation

If file is a bilingual document, in Downloads

* Rename it with the language code at the start of file name

## 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.

## 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

# 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.

## Lack of testing

Not sure what doesn’t work correctly yet. This one is fairly simple and does not rely too much on peoples to function.

## Autodeleting exe

The antivirus doesn’t like random executables and may delete the file randomly. Cant do anything about that.

## False positive

The antivirus doesn’t like random executables and may freeze execution and display a window, asking whether to deny or allow. Just click allow. Cant do anything about that.