

ZIP import package format

Visit this page for [example ZIP import packages](#).

Please note: translate5 project strives to support the [TIPP format as import and export format](#). Any kind of support for this development is welcomed.

File structure of an import package

Type	Name	Usage	Only available with listed plugin	Mandatory
Directory	proofRead	Contains a files and/or a folder tree with the files, that should be imported into the segment table of translate5. The folder tree inside of proofRead will be mirrored as folder tree inside of translate5.		yes
Directory	relais	Contains a structure that mirrors the structure in proofRead. This means, all directory and filenames must be identical with the names in the proofRead directory. All contained files are imported as relais/pivot content for the files in proofRead. Files existing in "proofRead" but missing in "relais" are skipped.		no
Directory	reference Files	Reference files that are attached to a task for download. Additional information for the user working on the task.		no
Directory	visualReview	The contained files are used to generate a layout for visual translation or visual review (with "What you see is what you get"). Possible contents of the directory visualReview are: <ul style="list-style-type: none"> ▪ A file "reviewHtml.txt", that contains one or multiple URLs (one URL by line and nothing else). The order of the URLs should be the same as the order of the corresponding files in proofRead. ▪ OR: A number of PDF files OR HTML files (mixing PDF with HTML is currently not supported - please contact translate5s developers, if you need this) <ul style="list-style-type: none"> ▪ The PDFs / HTMLs contain the same text as the text that is contained in the files in the proofRead directory. ▪ If the proofRead directory contains bilingual files, the PDFs / HTMLs can either be in source or target language ▪ if the PDFs / HTMLs are in the target language, the import option "Connect layout to" should be set to "target" ▪ The order of the PDF / HTML files on the file system should be the same as the order of the contents in the proofRead directory. Yet it is not necessary that the PDFs / HTMLs are the same number of files or have the same file names. ▪ The naming of the files in the visualReview folder does not matter at all. Only their order should reflect the order of the contents in the proofRead folder. ▪ multiple PDFs are merged to a single PDF and result in a single review file containing several pages/chapters whilst multiple HTML files or downloaded URLs will be accessible through a pager in the frontend 	visualTranslation / visualReview	no
File	*.xsl	Xml files in the directories "proofRead" and "alignLayout" can contain an xml-type reference to an xsl file, that is able to transform the XML to a layouted HTML version. This reference to xsl can either be an http(s) link or can be a path relative to the xml file pointing to an xsl file within the zip package	visualTranslation / visualReview	no
Directory	alignLayout	Similar to the concept of a pivot language it is possible to use an aligned XML with a referenced XSL stylesheet (see above) as source for the visual review. This XML will be aligned to a matching bilingual file in the proofRead folder, meaning that it is expected to hold the same segments (what is validated). The filename of the aligned layout must be identical to the bilingual that contains the same contents, except the fileextension. So "xyz.xml" matches "xyz.xml.sdxliff" and it also matches "xyz.sdxliff" and "xyz.xml.mxliff" or "xyz.mxliff". The aligned XML must be provided in a folder "alignLayout" and define a stylesheet via processing instruction as described above. It is possible of using multiple aligned XML/XSLT files for multiple bilingual files in "proofRead". The stylesheet has to be in the same folder and must be referenced with a XML preprocessing instruction e.g. <code><?xml-stylesheet type="text/xsl" href="stylesheet.xml"?></code>	visualTranslation / visualReview	no
File	*.tbx	Contains the terminology for the task. The contained terminology is used to mark terms in the task. Must reside on the top level of the zip-file. Only the first tbx-file in the zip file is used. Please see here for information, which terms are used for a task .		no
File	QM_Subsegment_Issues.xml	Contains the MQM issue types, that should be used for the task. If present, overrides the default MQM issue types for this task. An example file you find here .		no
File	task-template.xml	Contains a task template. Can be used to configure translate5 in an task-specific way.		no
File	DEU-ENG.transitConfig	Only for import packages, that contain STAR Transit NXT files. The first three letters of the file name must contain the transit-specific source-language file extension (in the example "DEU" for German). The last three letters of the file name (before the file extension ".transitConfig") must contain the transit-specific target-language file extension (in the example "ENG" for English). The file itself can be empty. It is a flag file.		no
File	*.bconf	This is a batch configuration file as exported from Okapi Rainbow . With Okapi Rainbow you can create your own custom file filters, for example for special kind of xml or txt-files. If you use the Okapi plugin for translate5, you can seamlessly use Okapi in the background to convert files on import time - and convert them back on export (without the user having to touch Okapi). Please see details on the translate5 file format page and on the translate5 Okapi Plug-in page		no
File	pixel-mapping.xlsx	This optional file can be provided to define the pixel-length for a pixel-based length check in translate5. It defines the pixel-width for each character of a font for a certain font-size.		no

Usage hints

- Folders and files mentioned in the above package structure MUST reside on the top level of the zip file (NOT inside some sub-folder).
- SDXLIFF, openTM2 XLIFF and CSV-files can be mixed inside the proofRead folders an any needed way. STAR Transit NXT files have to be in a separate import package.
- Please use only ASCII characters in file or directory names and no special characters or German "Umlaute".
The Zip format does not provide any information in which encoding your operating system has saved a file or directory name. Due to this lack of the ZIP standard, it might cause errors inside translate5, if you do use file or directory names with NON-ASCII characters. Only exception to this rule: Your file system does use UFT-8 encoding (like Linux and Mac do). Addition: In September 2018 a test with Windows 10 showed, that

special characters seem to be encoded also as utf-8 under windows with zip. Independent if native windows zip, winrar or 7zip was used. So maybe this issue is solved meanwhile through an always advancing Windows system 😊