

Merging of terms on TBX import

If a TBX is imported into an existing TermCollection, the newly imported data is merged into the existing collection.

A basic merging is always done, extended merging must be enabled via the parameter mergeTerms.

The merging is done on different levels:

- term entry level
- term level

basic term entry merge

By default term entries in the TBX with the same id (tbx term entry id attribute) are always merged if there is an already imported term entry with the same ID. This is called the basic term entry merge. This does not depend on the mergeTerms parameter and is performed always.

In this case all terms in TBX termEntry are saved to the same termEntry in the database (of course in the same language).

basic term entry merge > same language > affected terms

If the termEntry-ID matches,

- and the term-ID matches too (in the same language), the term is updated regardless of the mergeTerms parameter
- and the term-ID does not match, but the term is the same (by content and in the same language), the term is still merged, even if mergeTerms is false.

Parameter mergeTerms = true

If the parameter mergeTerms is true, translate5 will search all existing terms to see if the same term already exists in the TermCollection in the same language.

If a term is found:

- the first term that is found is updated by the values in the TBX file
- The term-ID and termEntry-ID remain the same as they already existed in translate5.
- All other terms of this termEntry in the TBX file will be imported/merged to the same termEntry of the TermCollection, that just has been found through the matching
- This means, if 2 (or more) terms that belong to the same termEntry-ID in the TBX file can be merged with a term in the database, but in the database those terms do NOT belong to the same termEntry-ID, only the first one will be merged with the one it matches and the second one (or more) will be put into the same termEntry as the first one.

If no term of a termEntry can be matched with an existing term in the termCollection:

- a new termEntry is added with the termEntry ID of the TBX in the TermCollection
- all terms of the termEntry of the TBX are added to this termEntry in the TermCollection

mergeTerms true summary

Summarized this means, that the checks should be done in the following order, in case the termEntry-ID does not exist in the termCollection and mergeTerms is true:

- For each term of a termEntry of the TBX file check, if the term exists in the termCollection within the same language.
- If yes, merge it
 - Also put all other terms of the same termEntry of the TBX (independent of their language) file into this termEntry in the termCollection
 - For all other terms in the same termEntry you therefore can stop checking, if they match with something (abort the loop)
- If the term does not match, check the next
- if no term of the termEntry of the TBX file can be matched, create a new termEntry with the termEntry-ID of the TBX and add all terms from the TBX to that termEntry in the termCollection with the term-IDs of the TBX.

Parameter mergeTerms = false

If the parameter mergeTerms is false

- a new termEntry is added to the termCollection with the term ID of the TBX file
- and all terms of this termEntry of the TBX file are added to this new termEntry in the termCollection with the term IDs of the TBX file
- basically the same thing as if no term from this termEntry in the TBX can be matched in the termCollection

On each term merge is checked

If a term already exists in the TermCollection and should be merged with the imported term, it is checked, if the term already exists in other termEntries of the same TermCollection as well

- If yes: The term is not merged and not imported, but an import error is recorded and the import continues with the next termEntry (the entire termEntry of this term is skipped). After the import is finished, all import errors are shown in the GUI / passed back via API.
- If no: The term is merged as described above

Memory usage

For memory usage the experience is, that for the import of a 2GB TBX file approximately 3GB of memory is used by PHP while the import. See also the [installation informations](#).

Also keep in mind that the TBX import may also consume a zip file containing one or multiple TBX files - this reduces file size enormously for the upload.