

Plug-In TermlImport: TermlImport



If the API changes in an incompatible manner this is listed explicitly in the "important release notes" section of each [releases changelog](#).

Currently there are five ways to import terminology into translate5:

1. Importing a task with terminology into translate5 for editing
this is not covered here, since that terminology currently can not be searched in the termportal.
2. Import a TBX file in the translate5 GUI in the language resources panel. For explanation of options please read below the options of "file-system based term import", as the options are the same.
3. Provide TBX files with the terminology use file-system based import
4. Get TBX files from Across via crossApi
5. [SFTP-directory based](#)

Depending on which way the terminology should be imported, the TermlImport Plug-In must be configured.

The configuration files are located

- [SFTP based term import](#)
- [File-system based term import](#)
 - [Configuration](#)
 - [API call for filesystem import](#)
- [TBX-import by using crossAPI \(Across API\)](#)
 - [configuration](#)
 - [API call for crossapi import](#)

SFTP based term import

See [TermlImport: SFTP support](#)

File-system based term import

To import TBX files from the file system into translate5s termCollections use the file-system based import.

For this a configuration file maps the TermCollection name with the file name of the TBX and the collection name with the customer number. In addition the config file defines the import directory, where the files are read from.

Example of the config file for the file system-based import. This file is located under "config" folder inside the TermlImport plugin.

TermCollections, that do not exist, will be created by the import.

Configuration

```

;;; ATTENTION: the below variables need to be placed above the areas [FileMapping] and [CollectionMapping]

;import directory of the tbx files
importDir=/var/www/translate5/data/tbx-import/filesystem/;

;mergeTerms: Allowed values: true or false.
;If this is true, terms will be merged as described at https://confluence.translate5.net/display/TAD
/Term+Collection.
;Otherwise all terms will be added to the termCollection, if the ID in the TBX does NOT exist already in the DB
;when the ID of the termEntry of the TBX does already exist in the DB, the terms of this termEntry will always
be merged
mergeTerms=true;

;Deletes all terms in all listed termCollections, that have a updated date older than the date configured for
"deleteTermsModifiedOlderThan".
;Every term in a termCollection has a creation date, a modification date and an updated dated.
;The creation date and the modification date are taken from the dates as they appear in the TBX file.
;The updated date is not shown in the GUI and is changed every time a user changes the term through touching
;it in the GUI or through an import
deleteTermsModifiedOlderThan=2018-05-01

;Deletes all terms in all listed termCollections, that have an "updated date" older than the start of the
current import.
;To learn, what the "updated date" is, please read above
;possible values: true or false
deleteTermsOlderThanCurrentImport=true

;Delete all proposals in a listed termCollections, that have a creation date older than this configs value.
deleteProposalsOlderThan=2018-05-01

;Delete all proposals in a listed termCollections, that have a creation date older than the start of the
current import.
;possible values: true or false
deleteProposalsOlderThanCurrentImport=false

;;; ATTENTION: the above variables need to be placed above the areas [FileMapping] and [CollectionMapping]

;mapping between tbx file and term collection name
[FileMapping]
Term.tbx=exampletermCollection;
Term1.tbx=exampletermCollection;
Term2.tbx=exampletermCollection;
example2.tbx=exampletermCollection2;
example3.tbx=exampletermCollection3

;mapping between term collection name and customer number (NOT the id)
[CollectionMapping]
exampletermCollection3=1000;
exampletermCollection2=1003;

```

This api call is only allowed for translate5 authenticated PM, Admin and API users.

To use the API as an authenticated user, first a session must be generated. More about how to authenticate and use translate5 API at: [REST API](#)

API call for filesystem import

URL:	(YOUR_TRANSLATE5_DOMAIN)/editor/plugins_termimport_termimport/filesystem
Available Methods:	GET
Specialities:	Start the files-system based term import for the files with location defined in the importDir parameter from the filesystem.config
Availability:	It is available only for PMs (users with system roles PM, Admin or API)

- example of the api call where the domain is translate5.net
http://translate5.net/editor/plugins_termimport_termimport/filesystem

TBX-import by using crossAPI (Across API)

Example of the config file for the file crossapi-based import. This file is located under "config" folder inside the TermImport plugin.

TermCollections, that do not exist, will be created by the import.

configuration

```
crossAPIurl=http://win.translate5.net
;All Across TBX Export templates (as created by Across) will be used. For each of the export templates a TBX
will be exported and imported into the translate5 termCollection, to that it is mapped to below in the area
[FileMapping]
crossExportFilesDir=/var/www/translate5/data/tbx-import/crossapi/
apiUsername=crossapi.user
apiPassword=pass

;mergeTerms: Allowed values: true or false.
;If this is true, terms will be merged as described at https://confluence.translate5.net/display/TAD
/Term+Collection.
;Otherwise all terms will be added to the termCollection, if the ID in the TBX does NOT exist already in the DB
;when the ID of the termEntry of the TBX does already exist in the DB, the terms of this termEntry will always
be merged
mergeTerms=true;

;Deletes all terms in all listed termCollections, that have a updated date older than the listed one.
;Since every term that exists in a TBX gets a new updated date on TBX-import, even if it is not changed: Simply
set this date to yesterday to delete all terms, that are not part of the current import
;The updated date is a date internal to translate5 and different from the modified date of the term, that is
shown in the interface
deleteTermsLastTouchedOlderThan=2018-05-01

;Deletes all terms in all listed termCollections, that have a modification date older than the start of the
current import.
;possible values: true or false
deleteTermsOlderThanCurrentImport=true

;Delete all proposals in a listed termCollections, that have a creation date older than this configs value.
deleteProposalsLastTouchedOlderThan=2018-05-01

;Delete all proposals in a listed termCollections, that have a creation date older than the start of the
current import.
;possible values: true or false
deleteProposalsOlderThanCurrentImport=false

;mapping between across export template name (potentially located in the crossExportFilesDir directory) and a
translate5 termCollection.;the termCollection will be created, if it does not exist.
[FileMapping]
template-1.export=exampletermCollection

;mapping between term collection name and customer number
[CollectionMapping]
exampletermCollection3=10001;
exampletermCollection2=10002;
exampletermCollection=123456789;
```

API call for crossapi import

URL:	(YOUR_TRANSLATE5_DOMAIN)/editor/plugins_termimport_termimport/crossapi
Available Methods:	GET
Specialities:	Start the crossapi-system based term import for the across TBX export template files with location defined in the crossExportFilesDir parameter from the crossapi.config

Availability:	It is available only for users with system role "admin" or "api")
---------------	---

- example of the api call where the domain is translate5.net
http://translate5.net/editor/plugins_termimport_termimport/crossapi