

A

Installing and Running the System

This appendix describes how to install and run the MT system.

A.1 Installation

A.1.1 Install FLExTools

In order to run the MT system, the open-source **FLExTools** application needs to be installed. **FLExTools** is an environment where **Python** scripts can be run to access and/or modify **FLEx** databases. This MT system is run within the **FLExTools** environment.

To Install **FLExTools**, go to this page: <https://github.com/cdfarrow/FLExTools/wiki>, navigate to the **Setup** section and download the zip file of the latest version of **FLExTools**. At the time of writing, the file to download was named: `FLExApps1.2.3.zip`. Extract the contents of this file to your hard drive. There is no Windows installation process; **FLExTools** is run from this folder directly.

A.1.2 Download MT Modules

Download the **Python** MT scripts from this link: <https://github.com/rmlockwood/FLExTrans/raw/master/FlexTrans.zip>.

Extract the contents of the zip file to the `FlexTools` folder under the `FlexAppsN.N.N` folder that you created above. You may be asked to confirm the overwriting of two or more files. Please overwrite them.

STAMP is installed automatically in the top level folder when you extract the zip file contents. You do not need to do a separate step. You can download it yourself if you want from the following URL:

http://carla.sil.org/Corporate_Release/DOSVersions/stamp32-221.zip.

A.2 Running

The basic procedure for running the MT system is to first run a “collection” of two scripts, then do the transfer process, then run a “collection” of four scripts. The following sections go into more detail and explain the setup steps as well.

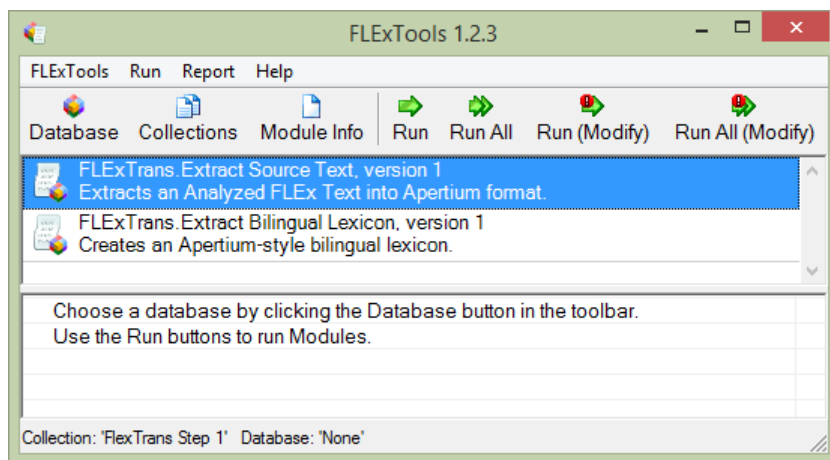
A.2.1 Edit the Configuration File

Before you run the MT system you want to make sure your configuration file is correct. You will find the configuration file in the `FlexTools` folder of the `FlexAppsN.N.N` folder. It is called `FlexTrans.config`. Edit this file in any text editor. The important properties to set are: `SourceTextName`, `TargetProject`, `SourceCustomFieldForEntryLink` and `SourceCustomFieldForSenseNum`. Refer to 4.4.3 for information on how to create the custom fields that these properties use. You may find it convenient to set the path of the following properties to the same folder: `SourceCustomFieldForEntryLink`, `AnalyzedTextOutputFile` and `BilingualDictOutputFile`. If you do not set the paths of these properties, they will default to the `FlexTools` folder. See Appendix B for more details.

A.2.2 Start *FLEXTools*

The next thing to do is to start **FLEXTools** which you do by double-clicking on `FlexTools.vbs`. The program should look like (78) when it first starts up:

(78)



A.2.3 Run Step 1 Collection

The two modules for the collection “FLExTrans Step 1” should have been loaded by default. If not, select the collection using the **Collections** button.

Before you run the collection, you need to choose your source database as the database on which these modules will be run. Click **Database** and choose your source **FLEx** project. Note: close both your source and target FLEx projects before running the collection; otherwise you will receive an error. Now you want to run the modules in the current collection. You can run each module separately or all together. To run them all together, Click on the **Run All** button. You will see output in the lower pane of the **FLExTools** window. Now you are ready for step two.

A.2.4 Run the Transfer Module (Step 2)

The next step is to run the transfer module. Go to the following website: <http://uakari.ling.washington.edu/flextrans/> and follow the instructions on the page where you are asked to upload three files. Two of the three files to upload will be found in the folders that you specified in the configuration file for the properties: AnalyzedTextOutputFile and BilingualDictOutputFile. The third file, `transfer_rules.tlx`, you will have edited yourself.

Immediately after these files are uploaded, the file `target_text.aper` is created on the web server. Check the log file to see that everything worked correctly. The log file will have output that looks something like this in the normal case:

```
Sat Apr 4 11:48:37 PDT 2015
lt-comp lr bilingual.dix bilingual.bin
main@standard 4579 8766
apertium-preprocess-transfer transfer_rules.tlx transfer_rules.tlx.bin
cat source_text.aper | apertium-transfer transfer_rules.tlx transfer_rules.tlx.bin
bilingual.bin > target_text.aper 2>>err_log
```

Note: you may only see one or two of the above lines if only one or two input files are uploaded.

The `err_log` file will just show a date-time stamp if it worked normally. Otherwise an error or a warning will be displayed at the end of the file.

If there are no errors, save the file `target_text.aper` via the given link on the web page to the folder you specified in the `TargetTransferResultsFile` property of the configuration file. At this point you are ready to do step three.

A.2.4.1 Apertium Commands Reference

For reference, here are the **Apertium** commands that are being executed in Step 2 and their function in this MT system. Usage examples of the commands are shown in the example output above.

lt-comp	Compiles a bilingual dictionary into a binary transducers file.
apertium-preprocess-transfer	Preprocesses transfer rules into a binary file.
apertium-transfer	Applies transfer rules and a bilingual dictionary to a source text resulting in a target text.

A.2.5 Run Step 3 Collection

The last step is to run the final modules. Open another collection by clicking on the **Collections** button and selecting “FLExTrans Step 3”. This time you need to use the button **Run All (Modify)** since we will be modifying the target project, namely inserting the translated text.

Again you will see output in the lower pane of the **FLExTools** window. The last module creates a text with the same title as the source text in the target **FLEx** project. Intermediate files are created in the Windows temp folder.¹ You are finished! Open the target **FLEx** project and see the results. Note: you may need to click on the **Refresh** button in **FLEx** before the new text name will show in the list.

¹ If you type %TEMP% into the address bar of file explorer, it will take you to the Windows temp folder.

B

Configuration File

Description of the properties and values in the configuration file.

Property	Description
SourceTextName	The name of the text (in the first analysis writing system) in the source FLEX project to be translated.
AnalyzedTextOutputFile	The path and name of the file which holds the extracted source text.
TargetOutputANAFFile	The filename of the file holding the intermediary text in STAMP format. This file is created in the Windows temp folder (%TEMP%).
TargetOutputSynthesisFile	The filename of the file holding the intermediary synthesized file. This file is created in the Windows temp folder.
TargetTransferResultsFile	The path and name of the file which holds the text contents after going through the transfer process.
SourceComplexTypes	One or more complex types from the source FLEX project (separated by commas). These types will be treated as a lexical unit in the MT system and whenever the components that make up this type of complex form are found sequentially in the source text, they will be converted to one lexical unit. See Section 5.9.1.1 for more details.
SourceCustomFieldForEntryLink	The name of the custom field in the source FLEX project that holds the link information to entries in the target FLEX project. See Section 4.4.3 for more information.
SourceCustomFieldForSenseNum	The name of the custom field in the source FLEX project that holds the sense number of the target entry. See Section 4.4.3 for more information.
BilingualDictOutputFile	The path and name of the file which holds the bilingual lexicon.

Property	Description
BilingualDictReplacmentFile	The path and name of the file which holds replacement lines for the bilingual lexicon. See Section 5.9.1.2.
TargetProject	The name of the target FLEX project.
TargetPrefixGlossListFile	The ancillary file that hold a list of prefix glosses from the target FLEX project.
TargetComplexFormsWithInflectionOn1stElement	One or more complex types from the target FLEX project (separated by commas). These types, when occurring in the text file to be synthesized, will be broken down into their constituent entries. Use this property for the types that have inflection on the first element of the complex form. See Section 5.9.3.2.
TargetComplexFormsWithInflectionOn2ndElement	Same as above. Use this property for the types that have inflection on the second element of the complex form.
TargetMorphNamesCountedAsRoots	Morpheme types in the target FLEX project that are to be considered as some kind of root. In other words, non-affixes and non-clitics.
SourceMorphNamesCountedAsRoots	Same as above for the source FLEX project.
CategoryAbbrevSubstitutionList	One or more pairs of grammatical categories where the first category is the “from” category in the source FLEX project and the second category is the “to” category in the target FLEX project. Use the abbreviations of the FLEX categories. The substitution happens in the bilingual lexicon. See Section 5.9.1.2 for more information.
CleanUpUnknownTargetWords	“y” (without quotes) to do the cleanup; “n” otherwise. If set to “y”, the system will remove preceding @ signs and numbers in the form N.N following words in the target text.

C

Troubleshooting

This appendix has some helpful information about troubleshooting the MT system.

If you see an @ sign in your text, this means that **Apertium** could not find the lemma in the bilingual lexicon.

If you see something like this: word1.3 i.e. a target word with numbers after it in your target text, it means that **Apertium** found no mapping for the source lemma.

If you see something like this: %0%word1.3%, **STAMP** could not synthesize the word.

If you find an affix gloss attached to a target word in your target text, this means that **STAMP** did not find the affix in the target lexicon.

A good method of tracking down what is going wrong in a particular situation is to put just one word into your source text file and try and get that word to translate correctly.

The following table shows the modules in sequence and the files they create. It may be helpful in tracking down problems.

Module	File Name Property	Default Name	In Temp Folder
Extract Source Text	AnalyzedTextOutputFile	source_text.aper	No
Extract Bilingual Lexicon	BilingualDictOutputFile	bilingual.dix	No
<i>Transfer Module</i>	TargetTransferResultsFile	target_text.aper	No
Catalog Target Prefixes	TargetPrefixGlossListFile	target_pfx_glosses.txt	No
Convert Text to STAMP Format	TargetOutputANAFFile	myText.ana	Yes
Extract Target Lexicon	TargetOutputSynthesisFile	myText.syn	Yes
Extract Target Lexicon	TargetProject	<TargetProject>_pf.dic	Yes
Extract Target Lexicon	TargetProject	<TargetProject>_sf.dic	Yes

Module	File Name Property	Default Name	In Temp Folder
Extract Target Lexicon	TargetProject	<TargetProject>_rt.dic	Yes
Extract Target Lexicon	TargetProject	<TargetProject>_stamp.dec	Yes