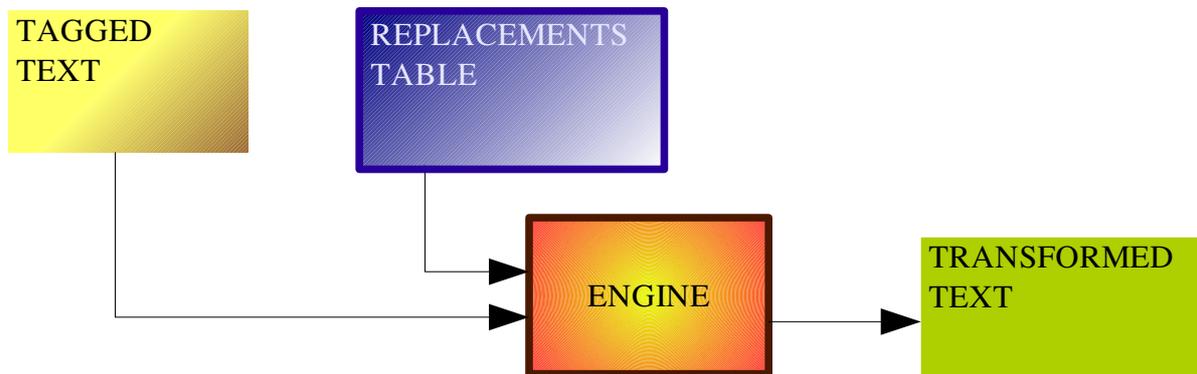


TDTxE: Tagged Document Transformer Engine

Abstract

This Project was born from a specific need to convert tagged text html4+css (with style code) to html 3.2., due to an incompatibility issue with an application that has to handle.

This need is seen as an opportunity to build a more generic application that can transform any tagged document from a replacement tables, that the users themselves can develop. The functional structure would be as follows:



The two main components are:

-**Engine process:** that from a text input applies a series of replacements.

-**Replacements Table:** It contains the list of all the substitutions that we want to apply with any additional information is needed. These boards should be able to be easily manipulated by the user to build different conversion tables and be able to choose at every moment which is to be used with a particular text.

We also need 2 operation modes:

-**Command Line:** get as parameters the incoming and outgoing documents, and the Replacement Table. Designed to enable batch operations or calls from other processes.

-**Graphic User Interface:** With a edit box to type/paste text and another for the processed text output, once it is chosen the table of substitutions to apply. To use it in cases where it wants to make such a "fast transformation" from fragments of a document, resident text in another application and ultimately, where there is not proper to convert a file.

With the appropriate tables and the same engine can also be used to clean code, suppress comments lines form code text...

Engine process

As usual in open source projects, we start with a **first basic version** that may evolve, we expect, to include new options, performance improvement, and so on. Also depending on what the community itself wants to make and to request.

The initial release consists of an iterative process: for each entry in the Replacements Table (RT from now), is processed with the document entered. When it reaches the end of the RT and once tried his latest entry, will complete the process and we will have the file converted. Therefore, it has to bear in mind the following (for the first version):

- Substitutions are textual, do not apply patterns of strings.
- Only operate in console mode (command line). The program will receive 3 arguments: path to the original file, path to the RT file and path to the output file.
- The RT entries are applied in sequential order, so that you can get different results with different ordinations.

In subsequent versions, the features we want to mainstream are:

- **Graphic User Interface**: With a edit box to type/paste text, an option to load a RT, and a box for text output.
- Options to **manage the RT** (GUI).
- Possibility of using **strings patterns**.
- Distinct process methods for each rule (for each entry in the TS)

Replacements Table

This component is the "*raison d'etre*" of this project and that really make the difference from other similar projects. Each table will contain general information (such as the description) and a collection of substitution rules. The attributes of the general data are:

Description	Descriptive title of the table. It will show when loaded via a drop-down.
Comments	To include information such as author, creation date, etc.
Version	Of the schema used. In anticipation that in the future is likely to alter the structure of this table, here indicate what version contains.
Begin string	To add to the begin.
End string	To add to the end.
Condition	...to add the strings above: 1-Always. 2-Only if there has been any replacement.

The attributes of the entries are as follows:

Initial Source Tag	String for searching into the incoming text. If not appears, nothing is done (skips to the next entry in the table).
Final Source Tag	Once found the initial string label, now seeking for this final tag.
Initial Destination Tag	Is the tag that will replace the Initial Source Tag.
Final Destination Tag	Is the tag that will replace the Final Source Tag.
Remove?	To remove all the content between the Initial and Final Source Tags.
Type Process	In the initial version there is only one, but in the future we will to define different ways to make the search process and substitution.
Comments	Observations to document this entry (is ignored by the engine process).