Auto Text Converter

By SaelSoft

Users Manual

For Version 0.8100 Beta

Prepared by David Saelman

March 14, 2016

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Key Features 1

1.3 Requirements 1

2. Overview 2

3. Operation 4

3.1 Running the Application 4

3.2 Running Modes 5

3.3 Review, Reverting or Committing Changes 5

3.4 Exporting the list of Search Extractions or the List of Before-After Text Replacements 5

4. Preferences 6

4.1 Colorization 6

5. Making Custom Searches and Replacements 6

5.1 Header block 6

5.2 Search sections 6

5.3 XML Example 7

5.4 Plugin Assemblies 8

5.5 Plugin Validation Function Prototype 8

5.6 Plugin Format Function Prototype 9

6. Using the Examples 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Auto Text Converter is a powerful search-and-replace tool for Microsoft Word Documents. It utilizes regular expressions, a feature missing in standard MS Office installations, as the heart of its search-and-replace engine. With this application one can accomplish such tasks as capturing all of the URLs in a document, converting words of British spelling into their American equivalents and vice-versa. The search and replace are only limited by one’s imagination.

## Key Features

This document is intended for developers

* Automated or manual controlled searching and or text replacements.
* Exporting the list of searched entries to a text or EXCEL file.
* The inclusion of several plugins which are used to extend the ability to perform sophisticated text replacements such as converting a date format, e.g., 3/17/2016 into March 17, 2016.
* The ability of writing your own plugins and adding them to the application.
* The ability of reviewing all the changes that were made during a search-and-replace session and choosing to either revert or commit that change.
* One can instantly navigate to any search result by simply double-clicking the entry in the application list view.
* And many more options.

## Requirements

Before running the application please read, understand, and agree to the following:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE SaelSoft or David Saelman BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following must be present on your PC in order to run the application:

* Operating System: WIN XP SP3, VISTA, Windows 8, 8.1, 10
* .NET 4.0
* Microsoft WORD 2007, 2010, 2013, 2016

# Overview

Below is a picture of the main application GUI.

The tool bar shows the following function buttons (from left to right)

Open file

Save file

Close file

Operations COMBO box

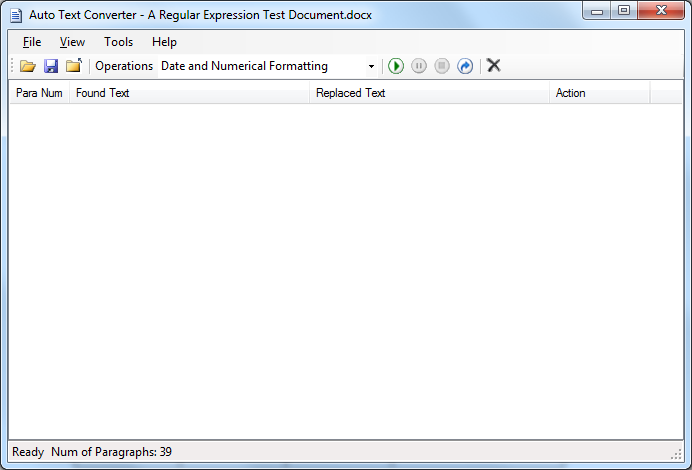
Auto process button

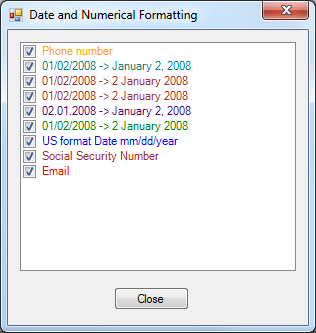
Pause button

Stop button

Step button

Clear the Listview button





# Operation

## Running the Application

After starting the application, the user can choose one of several search profile XML files from the Operations combobox. By going to the View-> After opening the file with the application, the user can press the <process> button and the application will search paragraph by paragraph for a match. The processing can be paused or resumed by pressing the <pause> button. Pressing the <stop> button will stop the processing.

What is required to run the application:

To run the application .NET 4.0 runtime must be installed. The application was developed using Visual Studio 2015 Community edition which is a free download from the Microsoft website.

## Running Modes

One can process the file in one of two mode:

(1) Process mode, which will process the entire file automatically. This mode is very effective on large documents. One can run in process mode, leave the computer and return when the job is finished. A record of all of the changes will appear in the main GUI List View.

(2) Step mode, which will stop at each change. The user then needs to hit the step button again and the program will proceed until the next change. Step mode comes in handy when the user wishes to review the changes in real time as the application makes them. The user can change modes simply by pressing the <Pause> button and pressing the mode of their choice.

## Review, Reverting or Committing Changes

After a file has been processed with the Auto Text Converter, one can quickly go back and check any particular change by simply double clicking the appropriate line in the ListView. Double-clicking will cause the application to navigate to and highlight the exact spot in the document where the change has been made in the target document. Any change can quickly be reverted back to the original text simply by right clicking the <revert> option in the menu. Similarly, a particular change can be committed by right clicking the <commit> button.

An alternative method of committing the changes in a document quickly, is to use the <Commit All> menu option available from the Tools option.

## Exporting the list of Search Extractions or the List of Before-After Text Replacements

After running an operation involving text formatting or replacements, the results displayed in the application List View can be exported to either a text file or to CSV file suited for EXCEL. The Export option is accessible from the File->Export menu option.

# Preferences

The preference dialog, available from the Tools Menu, gives options on how the application should behave.

## Colorization

By default, the application changes the color of any text that has been searched or replaced according to the color field. In addition, when text is reverted to its original format, it becomes pink. If the user wishes to disable the feature, the Colorization checkbox needs to be “unchecked”.

# Making Custom Searches and Replacements

The user can create his own XML profiles for custom searches and replacements. In this version, the user must manually create the XML file. Existing files can be used as a template or starting point. Below is a description of the XML format :

## Header block

Gives general information about the XML file including version, owner and a description. The only required field is the ProfileVersion which should be set to 1.0

## Search sections

Gives general information about the XML file including version, owner and a description.

### RegEx

The regular expression. Note: one should first test the regular expression using the handy regular expression tester available from the Tools Menu.

### Identifier

This field can be used for a private description of the section.

### FindColor

The color the application will use for the text that will be searched for or replaced. The list of colors which can be used can be accessed from the Tools Color Guide menu. In the future, one will be able to use the auto keyword if they do not wish to change the color of the text.

### Action

If the action is defined as “find”, the engine will only perform a search but no replacement will be made. If the Action = “replace”, the text will be replaced.

### Plugin

If the section supports a plugin, Plugin Description =”yes”, otherwise <Plugin></Plugin>

#### PluginName

The name of the Plugin Assembly

#### PluginValidationFunction (Optional)

The name of the Plugin function to validate the input string. For example, a numerical sting can be checked if it is a valid social security number or credit card number.

#### PluginFormatFunction

The Plugin function name that formats the input.

#### Reserved

Currently not used.

### Reserved1

Currently not used

### Reserved2

Currently not used

### Description

The description used to identify what the regular expression will look for. The description appears in the RegExListBox available from the View menu.

## XML Example

Gives general information about the XML file including version, owner and a description.

<Searches>

<Header>

<Product>Auto Text Converter by SaelSoft</Product>

<ProfileOwner>David Saelman</ProfileOwner>

<CreationDate>March 10, 2016</CreationDate>

<ProfileVersion>1.0</ProfileVersion>

<ProfileDescription>Converts dates and performs other numerical formatting</ProfileDescription>

<ProfileHistory />

</Header>

<Search RegEx= "(0[1-9]|1[012])[-\u2013/.](0[1-9]|[12][0-9]|3[01])[-\u2013/.]((19|20)\d\d)">

<Identifier>URL</Identifier>

<FindColor>Teal</FindColor>

<Action>replace</Action>

<Plugin Description="yes">

<PlugInName>NumericDateValidator.dll</PlugInName>

<PlugInValidationFunction>ValidateUSDate</PlugInValidationFunction>

<PlugInFormatFunction>GetDateMatch\_1</PlugInFormatFunction>

<Reserved>0</Reserved>

</Plugin>

<Resrved1/>

<Resrved2/>

<Description Text= "01/02/2008 -> January 2, 2008"/>

</Search>

</Searches>

## Plugin Assemblies

Plugin assemblies can be called via the XML search profile to perform complex text formatting or validation. The application package comes with a few sample plugin assemblies. The user can write write his own custom assemblies. Once they are compiled, the user must move the assembly .DLL files into the application PlugIn folder. The appropriate functions must be entered into the desired section of the Profile XML. There is no limit to the number of validation or formatting functions which can reside in a particular assembly. The pluggin functions must be written according to the following prototypes:

## Plugin Validation Function Prototype

public bool MyValidationFunction(Match matchResult)

Function input – RegEx Match (contains one or more matches from the particular paragraph.

Function output – return true if validation passes, otherwise false.

## Plugin Format Function Prototype

public string MyFormatingFunction(Match matchResult)

Function input – RegEx Match (contains one or more matches from the particular paragraph.

Function output – returns reformatted string which will replace the input string(s).

# Using the Examples

The Auto Text Converter packages comes with several sample documents that you can try along with the sample XML profiles. Below is an example

