

# Word Cleaner 4.6 | Help File | Revised Nov 09

---

Word Cleaner is the perfect tool for web designers and people who have to maintain websites. Word Cleaner enables you to batch convert Word files (and .rtf & .txt files) to HTML/XHTML/TXT files and clean out all the unnecessary tags, reducing the size of your files and thus improving download times. Word Cleaner will save you **hours** of boring repetitive work.

Remember to visit our website for the latest program updates, change log and support advice:  
<http://www.ConvertWordToHtml.com>

*To jump to a section just click on the section name below in the table of contents...*

## Table of Contents

Introduction.....	2
Guide to the Main Screen.....	3
Guide to the Settings Screen .....	5
General Options .....	5
Add folders .....	5
Preview browser paths.....	5
Default HTML Editor .....	5
MS Word Button Addin .....	5
Frequently Asked Questions.....	6
The formatting is not right or content is missing?.....	6
How do I get proper list formatting for lists in Word HTML?.....	6
I want to replace <b> with <strong>, or <i> with <em>.....	6
I can't see the Word Cleaner button in Word?.....	6
How to disable auto renaming of files.....	6
How to convert Word Files to HTML files.....	7
1: Converting one file at a time .....	7
2: Convert an entire directory .....	7
3: Convert a file from within Word.....	7
4: Right click on a file.....	8
How to clean up HTML pages .....	8
Right click support .....	8
How to Create Templates .....	9
Easy template editor .....	9
Advanced Template editor screen .....	10
Guide To The Templates That Come With Word Cleaner.....	12
Editing The Templates In An Text Editor.....	13
Backing up/transferring templates.....	13
Template Commands.....	14
Updating old templates .....	26
Regular Expressions Support .....	27
Command Line Support.....	27
How to get support.....	29
Feedback and Suggestions.....	29

## Introduction

### Word Cleaner will allow you to:

- Automatically convert Word files(and .rtf & .txt files) to HTML/XHTML/TXT files. It will even convert all tables, images and other content of Word files.
- Automatically clean up existing HTML files.
- Create/edit conversion templates to allow you to perform powerful tasks such as find and replace/delete code/text.
- Use the command line feature to automate tasks

Microsoft Word files are the most common way that clients and users will give you information to put on the web. Word does have a 'save as web page' feature but the problem is that the HTML files that are created contain a lot of unnecessary code that simply increases the size of your files. In the past web designers and webmasters have had to clean out the code by hand - even tools such as the 'clean up word html' command in Dreamweaver are not very effective.

**It's important to note that the clean HTML feature can be used to clean up any HTML page and not just pages created by Word.**

We are web designers ourselves, who spent a lot of time converting our clients' Word files and cleaning them up for use in their websites. We searched the web for a tool to automate the process but we couldn't find one, so we decided to build a tool ourselves!

The result is Word Cleaner, a utility designed by web designers to meet the needs of web designers.

### How to use the program

Word Cleaner is a very simple program to use, but it is also an extremely powerful program that you can customise to meet your own needs. At a basic level Word Cleaner has 4 main features:

1. **Convert Word documents (and .rtf & .txt files) to HTML/XHTML/TXT files**
2. **Clean up existing HTML Pages**
3. **Create your own conversion templates**
4. **Powerful find and replace function**
5. **Command line feature to automate tasks**

## Guide to the Main Screen

When you launch Word Cleaner you will see that we try to keep the interface simple. The areas below correspond to the screen grab at the bottom of this page.

### Area 1 - Toolbar:

- **Add files button:** click this button to browse for a file to convert. Note in the settings screen you can choose the files types you want to be selected.
- **Add folder button:** click this button to select a folder and convert/clean all the files in the folder. In settings screen you can turn off/on the option to scan sub folders.
- **Convert all button:** clicking this will start the conversion process and convert all files in the list.
- **Convert selected button:** clicking this will only convert the files you have selected. Remember you can hold down the control/shift key to select multiple files.
- **Template Drop Down List:** this is where you select the template that you want to use for the conversion. Remember you can also select a template sequence.
- **Edit Templates Button:** this is where you can add/edit and delete the templates.
- **Edit file in editor button:** clicking on this button will open the selected converted file in your text editor. You can set the default text editor in the settings menu.
- **Open file folder button:** selecting a file in the convert list and clicking on this button will open the folder that contains the file in windows explorer.
- **View in browser button:** clicking on this button will open the selected converted file in your browser. If you click on the small down arrow to the left of the button you can select to preview the file in IE, Firefox or Opera (you need to have these browsers installed on your system). You can set the default browsers in the settings screen.
- **Select all files button:** this will select all the files in the list, you can then delete them, or reset there file status.
- **Reset file status button:** click this button to reset the status of the converted file. For example say you converted a file but you need to reconvert it. If you highlight the file then click this button, it will reset the file status you can convert it again.
- **Remove selected button:** selecting files in the 'Convert these files' area and then clicking on this button will remove the files from the list. Note it will not delete the actual files.
- **Settings button:** clicking this will open up the Word Cleaner Settings screen
- **Help button:** clicking this will open the help file
- **About:** tells you which build of Word Cleaner you have
- **Website:** links you to our website

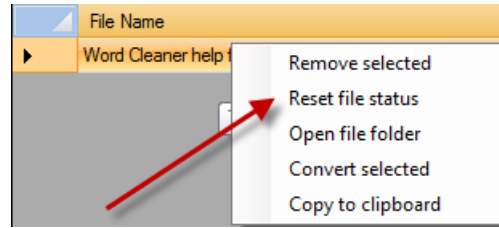
### Area 2 - Files Area:

- **Convert these files area:** this is where you can see what files you are converting and the status of the conversion.
- **Filename column:** this is the list of files that you have selected for conversion
- **Converted file name column:** word cleaner will automatically create a html file name based on your word file name. If your word file had spaces in the name word cleaner will remove them and use the - character instead, as spaces in html file names can cause problems. So 'about us.doc' will become

'about-us.html'. You can turn this option off in the settings screen.

**Note:** You can also change the default converted file name by double clicking on the converted file name, and entering your own file name.

- **Status column:** this shows the status of the file, ready=not converted yet, converted=file has been converted already
- **Right click support:** you can right click on any file in the list and show a range of common commands as shown in the screen grab to the right:

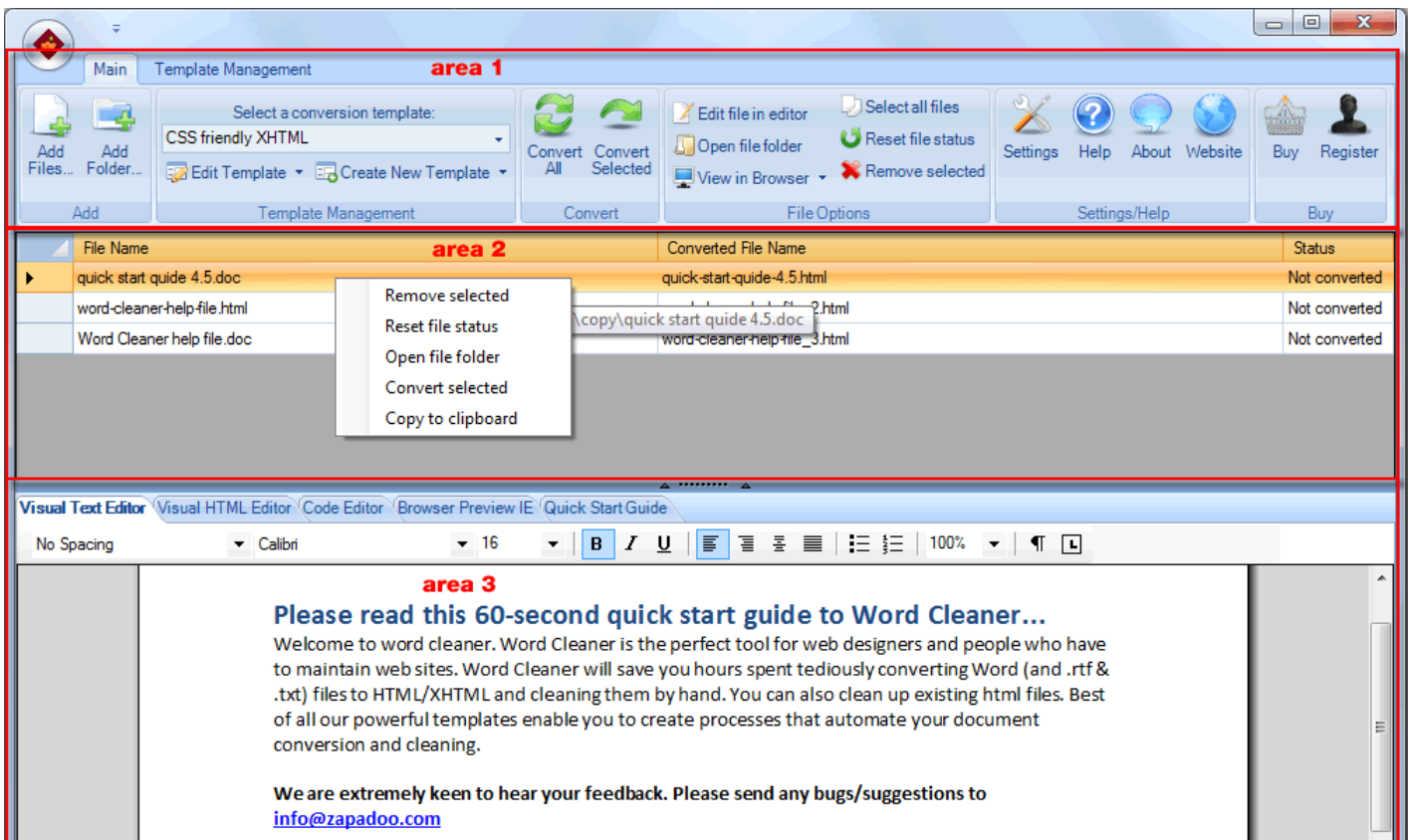


**Area 3 - Preview/Editors Area:**

This area allows you to view and edit the files you are converting.

**Important:** You need to double click on a file in the file list to show it in the preview area.

- **Visual text Editor:** In this area you can preview and edit the text file you have selected. **Note:** Word Cleaner **does not** need MS Word to be installed to work.
- **Visual HTML Editor:** In this area you can preview and edit the converted HTML file you have selected.
- **Code Editor:** In this area you can preview and edit the converted HTML file code you have selected. **Note:** right clicking in this area will show options for saving, copying etc.
- **Browser Preview:** This will show you how the converted file will look in Internet Explorer. **Note:** for technical reasons the built in browser preview only works with IE, not Firefox or Opera. For previewing files in Firefox or Opera use the drop down option in the 'View in browser' button to the right of the file list.



## Guide to the Settings Screen

Clicking on the settings button in the main screen will open the settings page, here you can choose all the options for Word Cleaner.

*Most of the settings are self explanatory but here is a guide to the main settings:*

### General Options

**Enable word button:** Ticking this will make Word Cleaner insert a button into Microsoft Word. When you click on this button it will launch the Word Cleaner program and add the document to the file list.

**File name web friendly format:** One of the cool features of Word Cleaner is the ability to automatically rename output files to make them web friendly. For example if your Word file is called Contact Us.doc Word Cleaner will rename the converted file to contact-us.html To make them web friendly we change spaces to dash's and convert uppercase letters to lowercase. If you do not want this feature you can disable it by un-ticking the box.

**Auto file name renaming if converted file already exists:** if you tick this then if the converted file name already exists in the folder word cleaner will not overwrite it. For example if your word file is called aboutus.doc, and aboutus.html exists in the same folder then word cleaner will rename the new converted file to aboutus-2.html. Turning this option off will overwrite the existing html file so be careful.

**Backup html file before cleaning:** Ticking this will make Word Cleaner create a backup of any html file before it cleans it. Be careful switching this off.

### Add folders

**Add folder file types:** you can choose what type of files will be visible when you are adding files to the file list.

**Scan subfolders:** if you tick this option, when you add a folder to the file list, it will also include the contents of any sub folders within the folder.

### Preview browser paths

you can preview files in up to 3 browsers. By default we link to IE, Firefox & Opera, but you can change these. Note these browsers do need to be installed on your system to be used by word cleaner.

### Default HTML Editor

**Use internal editor:** word cleaner has a built in editor but you can specify a different editor if you like.

### MS Word Button Addin

The button feature has a separate installer; you can find the installer by clicking on the settings button on the main screen, click on ms word button addin, then click install:



### Very important! Before you install the Button

Please note you will need to close Word and Outlook before you run the installer. If you have problems installing it, please reboot your computer and do not open any programs before running the installer again. The installer requires Word to be closed, but sometimes it can run in the background as programs like Outlook will link into Word. For advanced users you can also option task manager and kill any references to winword.exe before installing.

## Frequently Asked Questions

Here we outline the common questions users have:

### The formatting is not right or content is missing?

Please try converting both with the internal conversion engine and using Word to convert. This option is in the easy template editor main screen.

### How do I get proper list formatting for lists in Word HTML?

If you convert the Word file with Word Cleaner and you tick the use internal conversion engine option then we do use the proper list tags.

If you are cleaning an existing word html file then you will have problems. The issue is Word uses CSS to control lists, it does not use the standard ul li tags. You have 3 options:

1. Re convert the Word file with Word Cleaner
2. Keep the CSS styles in your word html file
3. Re create the lists manually

### I want to replace <b> with <strong>, or <i> with <em>

All you need to do is tick the option in the easy template editor main screen.

### I can't see the Word Cleaner button in Word?

[Please click here for instructions on activating the button...](#)

### How to disable auto renaming of files

One of the cool features of Word Cleaner is the ability to automatically rename output files to make them web friendly.

For example if your Word file is called **Contact Us.doc** Word Cleaner will rename the converted file to **contact-us.html**

To make them web friendly we change spaces to dash's and convert uppercase letters to lowercase.

**If you do not want this feature you can disable it in the settings screen.**

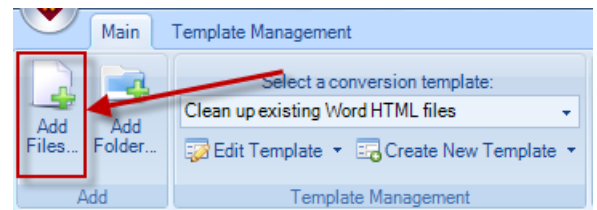
Still have a question? Please email: [info@ConvertWordToHtml.com](mailto:info@ConvertWordToHtml.com)

## How to convert Word Files to HTML files

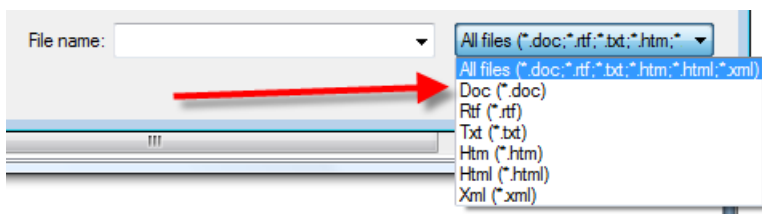
*There are four ways of doing this:*

### 1: Converting one file at a time

Open up the Word Cleaner program. Click on 'add...' and from the dialog box that opens select the Word file that you wish to convert and click on OK. The file will be added to the list of files to be converted. If you wish you can click on 'add...' again to select another file to be added to the conversion list. Once you have selected all the files you wish to convert, you will now need to select the template you want to use for the conversion.

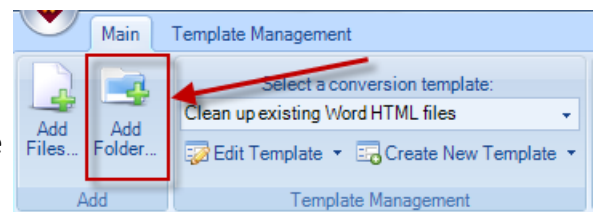


🔧 Remember you can show only specific file types by selecting from the drop down file type list. See screengrab below:



### 2: Convert an entire directory

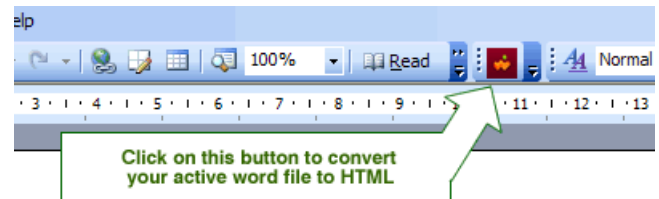
This option allows you to select an entire directory and Word Cleaner will convert all the files within it. Just open the program and click on the 'add folder' button on the left and then browse to the folder you want to choose. Once you have selected the folder you can then select the mode and template options as outlined in 'Converting one file at a time'.



You can select text files only, html files only or both.

### 3: Convert a file from within Word

Word Cleaner inserts a button into Microsoft Word - if you don't see the button go to the view menu: toolbars: and tick the option for Word Cleaner. You can hide this button by unselecting Word Cleaner from the toolbars menu and you can also drag this button around and position it where you please. When you click on this button it will launch the Word Cleaner program and convert the document you have open in Word into HTML.

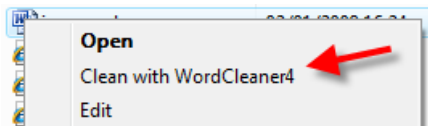


### Can't see the button in Word?

[Please click here for instructions on activating the button...](#)

### 4: Right click on a file

In Windows Explorer or in any folder you can right click on a Word or HTML file and select the 'Clean with Word Cleaner' option from the menu that appears. The file you right click on will be opened in Word Cleaner and all you have to do is to select the conversion options.



## How to clean up HTML pages

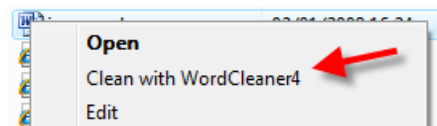
You can choose to clean up existing HTML files.

This works the same way as converting files to HTML - you can choose to clean up individual files, folders or to create a batch conversion. Add the HTML files the same way you add the Word files, by clicking on the 'add' or 'add folder' buttons.

It's important to note that the clean HTML feature can be used to clean up any HTML page and not just pages created by Word.

## Right click support

In Windows Explorer or any folder you can right click on a Word or HTML file and select the Clean with Word Cleaner4' option from the menu that appears.

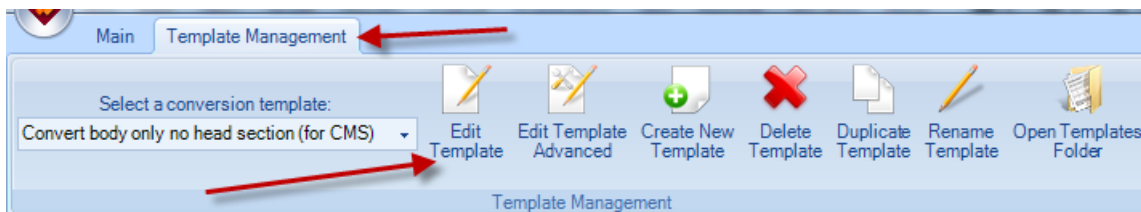
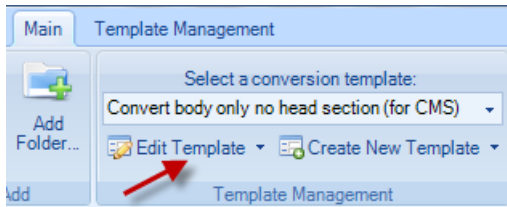


The file you right click on will be opened in Word Cleaner and all you have to do is to select the conversion options.

# How to Create Templates

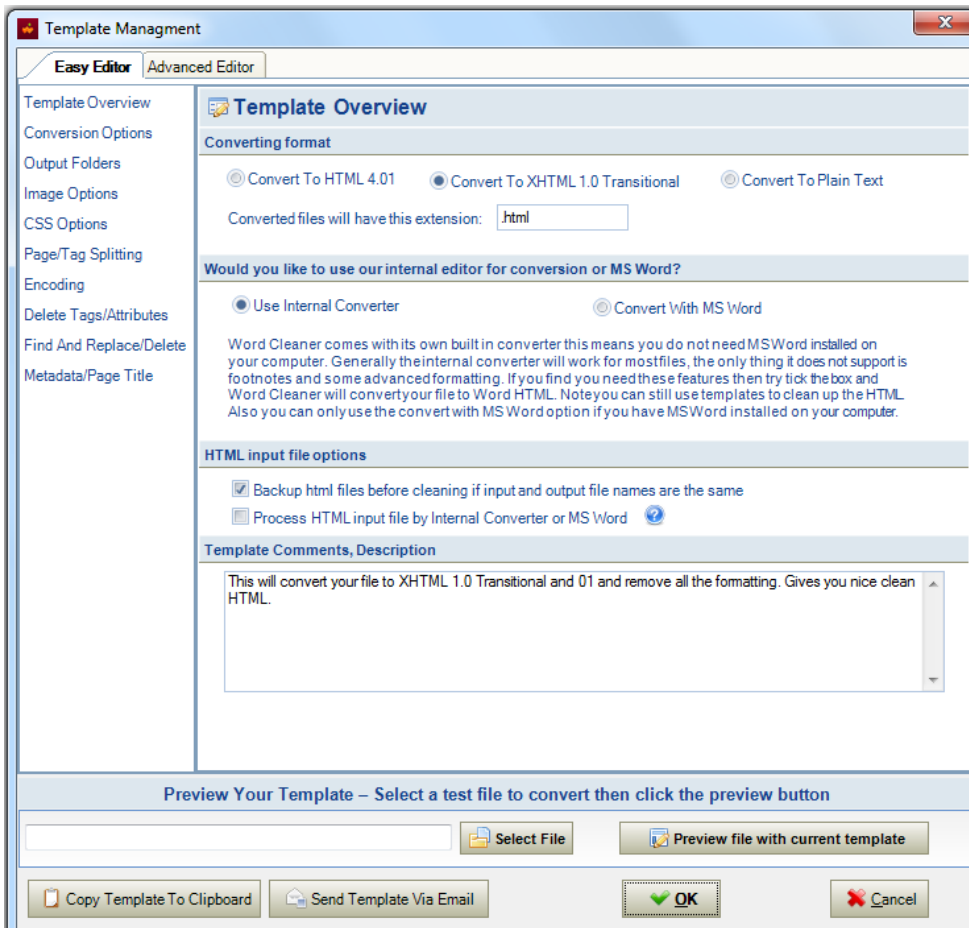
## Easy template editor

The core of Word Cleaner is the powerful template system. Here you can edit the built in templates, duplicate templates, and create your own templates. To enter this section just click on the Templates tab on the main screen, or the edit templates button on the main tab. Remember to first select the template you want to edit/duplicate/delete by selecting a template from the drop down list.



**Tip:** Remember you can edit or duplicate the templates that come with Word Cleaner.

In the easy editor screen all you need to do is click on an option on the right and select the options you need:



## Advanced Template editor screen

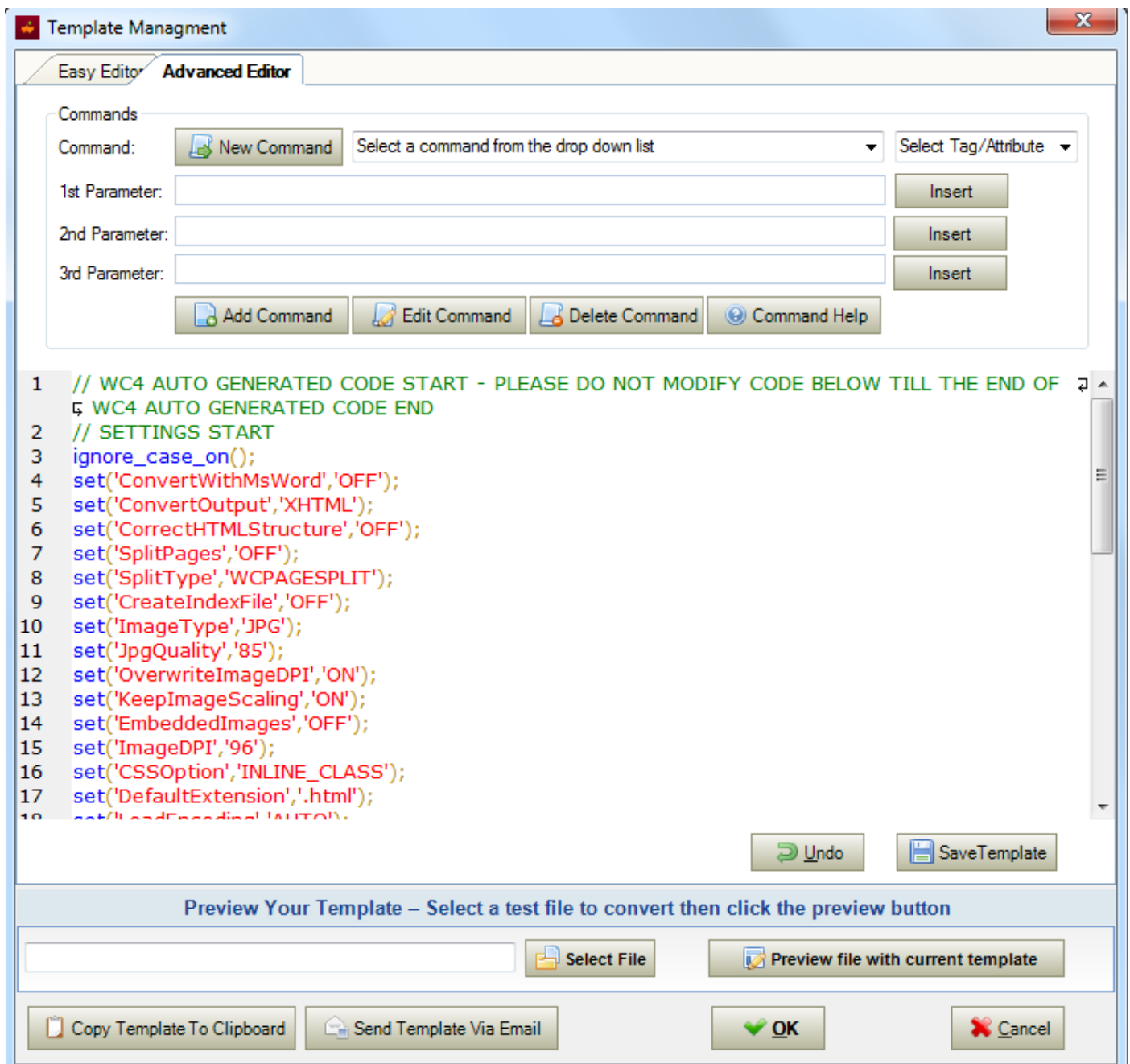
For advanced users we have the advanced template editor. You access this by clicking the tab at the top.

The first thing to be aware of is creating templates is not as complicated as it first looks. We have created a custom template system that should cover the needs of most users.

At a basic level the template system should be considered like a find and replace system. You can look for a specific code then edit or delete it.

 **We support Regular Expressions (RegEx) in commands, [see this page for more information](#).**

*On the screen grab below you can see the template editor screen, it has several main parts:*



**Command area:**

this is where you create the command you want to add to the template.

1. The first step is to select a command from the drop down list. For example say we want to delete the span tag for a document. We select the delete\_tag command.
2. The second step is too add the parameters, for example if we enter the span tag. You can enter the tag manually or select it from the drop down tag/attribute list.
3. The third step is too add the command to the template by clicking the add command button.

**Edit/delete a command:** to editor or delete a command you should first click on it in the editor area then click on the edit or delete command button.

**Template preview area:**

you can test out your template by select a test file. This enables you to easily tweak and test your template until you are happy with it.

**Editor area:**

here you can directly edit the template commands. Pressing control and space will show you a list of commands.

**In the screengrab example below there are two commands:**

`delete_tag('span');` - this will delete the span tag from your file.

`delete_all_tag_attributes('body');` - this will clean the p tag, for example convert `<body class="BODYSTYLE">` to `<body>`

2nd Parameter:

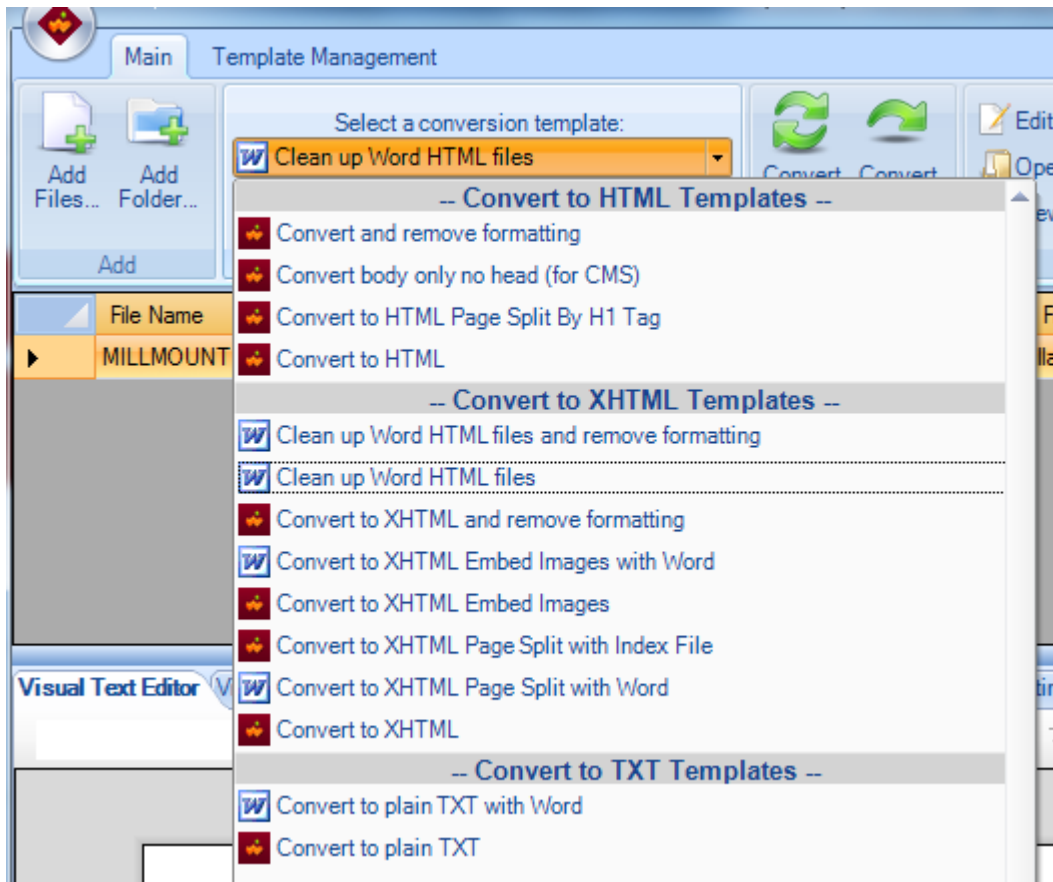
3rd Parameter:

```
21 set('BodyOnly','OFF');
22 set('ProcessHTMLInputFile','OFF');
23 // SETTINGS END
24 // PLAINTEXTCONVERSION START
25 delete_attribute('class');
26 delete_attribute('style');
27 delete_all_tag_attributes('body');
28 delete_tag('span');
29 delete_tag('base');
30 delete_tag_with_content('style');
31 delete_all_tag_attributes('li');
32 replace_regexp_text('>(&nbsp;){1,}<','><');
33 replace_regexp_text('<p>&nbsp;','<p>');
34 replace_regexp_text('{2,}','');
35 delete_all_empty_tags();
36 // PLAINTEXTCONVERSION END
37 // COMMENTS START
38 // This will convert your file to XHTML 1.0 Transitional and 01 and remove all the formatting. Gives you nice clean HTML
```

Preview Your Template – Select a test file to convert then click the preview button

## Guide To The Templates That Come With Word Cleaner

Below is a quick guide to the templates that come with Word Cleaner.



As you can see from the screen grab above

### **Convert to HTML**

This will convert your file to HTML 4.01 and keep all the formatting, e.g. fonts background colours etc.

### **Convert to XHTML**

This will convert your file to XHTML 1.0 Transitional and keep all the formatting, e.g. fonts background colours etc.

### **Convert to HTML and remove formatting**

This will convert your file to HTML 4.01 and remove all the formatting. Giving you nice clean HTML.

### **Convert to XHTML and remove formatting**

This will convert your file to XHTML 1.0 Transitional and 01 and remove all the formatting. Giving you nice clean HTML.

### **Clean up Word HTML files**

If you have existing Word HTML files Word Cleaner will clean them up for you. It will remove all unneeded tags. I will still keep your layout and formatting but your files will be smaller and cleaner.

### **Clean up Word HTML file and remove formatting**

If you have existing Word HTML files Word Cleaner will clean them up for you. It will remove all unneeded tags as well as removing all the formatting from the text etc. Giving you nice clean HTML.

### **Convert body only no head section (for CMS)**

This will convert your file to HTML 4.01 and keep all the formatting, e.g. fonts background colours etc. It will remove the head section leaving you with the body code. This is useful if you want to paste HTML into a content management system (CMS) or a template.

## **Editing The Templates In An Text Editor**

We recommend you use the built in editor on Word Cleaner to edit templates but you can edit the templates in any text editor of your choice, for example notepad etc.

We store the templates in .wc4 file format. This format is basically a text file so you can open the template files in notepad or any other editor.

The templates are located in this folder: **C:\Program Files\Zapadoo\WordCleaner4\Templates**


## **Backing up/transferring templates**

Each template has its own file, this makes it very easy to back up the templates folder, or share your templates with a colleague.

The templates are located in this folder: **C:\Program Files\Zapadoo\WordCleaner4\Templates**

You can copy and paste them, just like any normal file. When you open Word Cleaner, the program will automatically detect any new templates and add them to the drop down templates list.

We store the templates in .wc4. This format is basically a text file so you can open the template files in notepad or any other editor.

 If you have created a template that you think might benefit other users then please email send to: [info@ConvertWordToHtml.com](mailto:info@ConvertWordToHtml.com) and we will consider bundling it with the next version.

## Template Commands

Below is a guide to the template commands and examples of how they can be used.

We know templates can appear complex, so if you have any questions or need any help just email us at [info@ConvertWordToHtml.com](mailto:info@ConvertWordToHtml.com)

### List of commands:

- `ignore_case_on();`
- `ignore_case_off();`
- `replace_text(Text to replace in HTML file,New text);`
- `replace_inner_text(Text to replace only inside HTML tags(inner text),New text);`
- `replace_regexp_text(Enter regular expression to match in html file, Text used to replace any regexp matches);`
- `replace_tag_attribute(HTML tag to replace,HTML Attribute,New HTML Attribute);`
- `replace_tag_attribute_no_quotes(HTML tag to replace,HTML Attribute,New HTML Attribute);`
- `replace_tag_attribute_value(HTML tag to replace,HTML Attribute value to replace,New value);`
- `replace_tag_attribute_value_no_quotes(HTML tag to replace,HTML Attribute value to replace,New value);`
- `replace_inner_text_for_tag(HTML tag in which inner text will be replaced,Text to replace,New text);`
- `delete_tag(HTML tag to delete in whole file);`
- `delete_all_empty_tags(Removes all tags that are empty);`
- `delete_text(Text to delete in HTML file);`
- `delete_inner_text(Text to delete only inside HTML tags[inner text]);`
- `delete_regexp_text(Enter regular expression to delete in html file);`
- `delete_inner_text_for_tag(HTML tag in which inner text will be replaced,Text to delete);`
- `delete_tag_attribute(HTML tag in which attribute will be deleted,Attribute name);`
- `delete_all_tag_attributes(HTML tag in which all attributes will be deleted);`
- `add_tag_attribute(HTML tag to which attribute will be added,New attribute with value);`
- `add_html_after_opening_tag(HTML opening tag after which HTML will be added,Text/HTML string);`
- `add_html_after_closing_tag(HTML closing tag after which HTML will be added,Text/HTML string);`
- `add_html_before_opening_tag(HTML opening tag before which HTML will be added,Text/HTML string);`
- `add_html_before_closing_tag(HTML closing tag before which HTML will be added,Text/HTML string);`

### New commands:

- `delete_all_empty_tags();`

Removes all tags that are empty (Empty tag: `<tag>any white space inside</tag>`) for example:  
`<p></p>` or `<p> </p>` - will be removed from html

- `delete_attribute('Attribute to delete in HTML file for every tag');`

Remove defined html tag attribute from all html tags in source html.

Example:

```
delete_attribute('style');
```

Input html:

```
<p class="test" style="width:100px">Test</p>  
<span class="test2" style="height:100px">Test2</span>
```

Output html:

```
<p class="test">Test</p>  
<span class="test2">Test2</span>
```

- `delete_empty_tag('HTML tag to delete in whole file');`

Removes defined tag that is empty (Empty tag: `<tag>any white space inside</tag>`) for example a command to remove all empty 'p' tags will be like this:

**`delete_attribute('p');`**

Will result that all empty tags will be removed from html file

Input html:

```
<p> </p>  
<p>TEST</p>
```

Output:

```
<p>TEST</p>
```

**`delete_tag_with_content('HTML tag to delete in whole file');`**

Removes defined tag with it's inner content (inner text or html).

Example:

**`delete_tag_with_content('body');`**

Will remove body tag with all internal (inner) content it has.

**`find_and_replace_regexp_text('Enter regular expression to find in html file','Enter regular expression to match and replace in text strings that were found', 'Regular expression or text used to replace any regexp matches');`**,

This powerful command can be used to do some regular expression find/replace/remove html processing for a given text that will be found by the first parameter of that command.

For example following command will search input html for a body tag and it would process its manually entered emails (as text) and it would convert it to html versions of it.

```
find_and_replace_regexp_text('<body.*?</body>','([\w-]+)@([\w-]+\.)+\w+','<a href="mailto:$0">$0</a>');
```

Input html:

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
  <title>EMAIL@TEST.COM</title>
</head>
<body>
  <p>
    EMAIL@TEST.COM
  </p>
</body>
</html>
```

Output html:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>EMAIL@TEST.COM</title>
  </head>
  <body>
    <p>
      <a href="mailto:EMAIL@TEST.COM">EMAIL@TEST.COM</a>
    </p>
  </body>
</html>
```

That command only processed body (blue part) of input html and it changed its EMAIL@TEST.COM to html version. If that command would be executed for the whole input html it would also change email that is entered in <title>EMAIL@TEST.COM</title> which would cause html syntax issue.

Example 2:

```
find_and_replace_regex_text('<p[^\<>]*?>.*?</p>', '([\w-]+)@([\w-]+\.)+\w+', '<a href="mailto:$0">$0</a>');
```

Input html:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>EMAIL@TEST.COM</title>
  </head>
  <body>
    <p>
      EMAIL@TEST.COM
    </p>
    <span>
      EMAIL_SPAN@TEST.COM
    </span>
    <p>
      EMAIL2@TEST2.COM
    </p>
  </body>
</html>
```

Output html:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>EMAIL@TEST.COM</title>
  </head>
  <body>
    <p>
      <a href="mailto:EMAIL@TEST.COM">EMAIL@TEST.COM</a>
```

```
</p>
<span>
  EMAIL_SPAN@TEST.COM
</span>
<p>
  <a href="mailto:EMAIL2@TEST2.COM">EMAIL2@TEST2.COM</a>
</p>
</body>
</html>
```

That command was looking for any <p>...</p> tag occurrence in the input html and it changed its EMAIL@TEST.COM to html version. If that command would be executed for the whole input html it would also change email that is entered in <title>EMAIL@TEST.COM</title> which would cause html syntax issue. Please note that it did not process <span> tag email.

### **ignore\_case\_on();**

Turns on case ignoring when matching(searching for) a text to replace.

### **ignore\_case\_off();**

Turns off case ignoring when matching(searching for) a text to replace.

### **replace\_text(Text to replace in HTML file,New text);**

Simple command to find and replace any text in html file.

**Example** - finds all 'foo1' strings in html file and replaces it with 'foo2':

*Command:*

```
replace_text('foo1','foo2');
```

Input html:

```
<p>foo1 example</p>
<p>foo1foo1foo1</p>
<foo1>foo1</foo1>
```

Output:

```
<p>foo2 example</p>
<p>foo2foo2foo2</p>
<foo2>foo2</foo2>
```

### **replace\_inner\_text(Text to replace only inside HTML tags(inner text),New text);**

Simple command to find and replace in inner text in html file.

**Example** – finds all 'foo1' strings in inner text in html file and replaces it with 'foo2':

*Command:*

```
replace_inner_text('foo1','foo2');
```

Input html:

```
<p>foo1 example</p>
<p>foo1foo1foo1</p>
<foo1>foo1</foo1>
```

Output:

```
<p>foo2 example</p>
<p>foo2foo2foo2</p>
<foo1>foo2</foo1>
```

**replace\_regexp\_text(Enter regular expression to match in html file, Text used to replace any regexp matches);**

Command to find and replace text in html file. First parameter can contain regular expressions and second one can contain backreferences if there any in first parameter. (For regexp backreferences please look in regular expression tutorial).

**Example 1**– finds all <p>...</p> tags in html file and replaces it with 'foo' string:

*Command:*

```
replace_regexp_text('<p>.*?</p>', 'foo');
```

Input html:

```
<p>foo1 example</p>
<p>foo1foo1foo1</p>
<foo1>foo1</foo1>
```

Output:

```
foo
foo
<foo1>foo1</foo1>
```

**Example 2**– finds all <p>...</p> tags in html file and leaves only inner text of <p> tag:

*Command:*

```
replace_regexp_text('<p>(.*?)</p>', '$1');
```

Input html:

```
<p>foo1 example</p>
<p>foo1<p>foo2</p><b>bold</b></p>
<foo1>foo1</foo1>
```

Output:

```
foo1 example
foo1<p>foo2<b>bold</b></p> !- this html line is broken because <p></p> tag was embedded in other <p></p> tag
and example 2 regexp is mathing <p> and first occurrence of </p> closing tag
<foo1>foo1</foo1>
```

**Example 3**– deletes all <p>...</p> tags in html file:

Command:

```
replace_regexp_text('<p[^>/*>.+?</p>','');
```

Input html:

```
<p>foo1 example</p>
<p>foo1<p>foo2</p><b>bold</b></p>
<foo1>foo1</foo1>
```

Output:

!- first line was removed correctly but line below was not all removed because <p> tags were embedded

```
<b>bold</b></p>
<foo1>foo1</foo1>
```

**replace\_tag\_attribute(HTML tag to replace,HTML Attribute,New HTML Attribute);**

Command replaces defined tag attribute with new one in html file.

**Example** – finds all <p> tags with attribute 'id="..."' in in html file and replaces it with **other\_attribute="ABC"**:

Command:

```
replace_tag_attribute('p','id','other_attribute="ABC"');
```

Input html:

```
<p id="123" bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id="XYZ" >foo1 example</p>
<p>foo1 id=100</p>
```

Output:

```
<p other_attribute="ABC" bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" other_attribute="ABC" >foo1 example</p>
<p>foo1 id=100</p>
```

**replace\_tag\_attribute\_no\_quotes(HTML tag to replace,HTML Attribute,New HTML Attribute);**

Command replaces defined tag attribute with new one in html file. No quotes means that attribute value has no "" (quotes).

**Example** – finds all <p> tags with attribute 'id=...' in in html file and replaces it with **other\_attribute="ABC"**:

Command:

```
replace_tag_attribute_no_quotes('p','id','other_attribute="ABC"');
```

Input html:

```
<p id=123 bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id="XYZ" >foo1 example</p>
<p>foo1 id=100</p>
```

Output:

```
<p other_attribute="ABC" bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id="XYZ">foo1 example</p>
<p>foo1 other_attribute="ABC"</p>
```

**replace\_tag\_attribute\_value(HTML tag to replace,HTML Attribute value to replace,New value);**

Command replaces defined tag attribute value with new value in html file.

**Example** – finds all <p> tags with attribute 'id="..."' in in html file and replaces it with **ABC**:

*Command:*

**replace\_tag\_attribute\_value('p','id','ABC');**

Input html:

```
<p id="123" bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id='XYZ' >foo1 example</p>
<p>foo1 id=100</p>
```

Output:

```
<p id="ABC" bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id='ABC' >foo1 example</p>
<p>foo1 id=100</p>
```

**replace\_tag\_attribute\_value\_no\_quotes(HTML tag to replace,HTML Attribute value to replace,New value);**

Command replaces defined tag attribute value with new value in html file. No quotes means that attribute value has no "" (quotes).

**Example** – finds all <p> tags with attribute 'id=...' in in html file and replaces it with **ABC**:

*Command:*

**replace\_tag\_attribute\_value\_no\_quotes('p','id','ABC');**

Input html:

```
<p id=123 bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id="XYZ" >foo1 example</p>
<p>foo1 id=100</p>
```

Output:

```
<p id=ABC bgcolor="#ff00ff">foo1 example</p>
<p bgcolor="#ff00ff" id="XYZ" >foo1 example</p>
<p>foo1 id=ABC</p>
```

**replace\_inner\_text\_for\_tag(HTML tag in which inner text will be replaced,Text to replace,New text);**

Command replaces defined tag inner text first occurrence only of parameter string to replace with new string in html file.

**Example** – finds all <p ...> tags with 'foo' inside tag and replaces it with **XXX**:

*Command:*

**replace\_inner\_text\_for\_tag('p','foo','XXX');**

Input html:

```
<p id="123">foo1 example</p>
<p bgcolor="#ff00ff">xxx foo examplefoo</p>
<p>foo2</p>
```

Output:

```
<p id="123">XXX1 example</p>
<p bgcolor="#ff00ff">xxx XXX examplefoo</p> !- blue underlined foo was not changed because this command matches first occurrence of foo only
<p>XXX2</p>
```

**delete\_tag(HTML tag to delete in whole file);**

Command deletes defined tag html but leaves it's contents in html file.

**Example** – delete all <p ...></p> tags:

*Command:*

**delete\_tag('p');**

Input html:

```
<p id=123>foo1 example</p>
<p bgcolor="#ffeccc">foo1<p>foo2</p><b>bold</b></p>
```

Output:

```
foo1 example
foo1foo2<b>bold</b>
```

**delete\_tag\_with\_content(HTML tag to delete in whole file);**

Command deletes defined tag html but leaves it's contents in html file.

**Example** – delete all <p ...>\*</p> tags:

*Command:*

**delete\_tag\_with\_content('p');**

Input html:

```
<p id=123>foo1 example</p>
<p bgcolor="#ffeccc">foo1<p>foo2</p><b>bold</b></p>
<b>foo</b>
```

Output:

!- first line was removed correctly but line below was not all removed because <p> tags were embedded

```
<b>bold</b></p>
```

```
<b>foo</b>
```

### **delete\_text(Text to delete in HTML file);**

Command deletes defined text in html file.

**Example** – delete all 'foo' strings:

*Command:*

### **delete\_tag('foo');**

Input html:

```
<p id=foo>foo1 example</p>
```

```
<p bgcolor="#ffecc">foo1<p>foo2</p><b>bold</b></p>
```

```
<foo1>foo1</foo1>
```

Output:

```
<p id=>1 example</p>
```

```
<p bgcolor="#ffecc">1<p>2</p><b>bold</b></p>
```

```
<1>1</1>
```

### **delete\_inner\_text(Text to delete only inside HTML tags[inner text]);**

Command deletes first occurrence of defined text in inner text of html tag in html file.

**Example** – delete all 'foo' strings in inner html:

*Command:*

### **delete\_tag('foo');**

Input html:

```
<p id=foo>foo1 example</p>
```

```
<p bgcolor="#ffecc">foo1<p>foofoo2</p><b>bold</b></p>
```

```
<foo1>foo1</foo1>
```

Output:

```
<p id=foo>1 example</p>
```

```
<p bgcolor="#ffecc">1<p>foofoo2</p><b>bold</b></p> !- foofoo was not removed because it was second and third occurrence
```

```
<foo1>1</foo1>
```

### **delete\_regexp\_text(Enter regular expression to delete in html file);**

Command to delete text in html file. Parameter can contain regular expression.

**Example 1**– deletes all <p>...</p> tags in html file:

*Command:*

**delete\_regexp\_text('<p>.\*?</p>');**

Input html:

```
<p id=foo>foo1 example</p>
<p bgcolor="#ffecc">foo1<p>foofoofoo2</p><b>bold</b></p>
<p>foo1foo1foo1</p>
<foo1>foo1</foo1>
```

Output:

```
<p id=foo>foo1 example</p>
<p bgcolor="#ffecc">foo1<b>bold</b></p>
<foo1>foo1</foo1>
```

**delete\_inner\_text\_for\_tag(HTML tag in which inner text will be replaced,Text to delete);**

Command deletes second parameter string in defined tag inner text in html file. It deletes first occurrence of second parameter only.

**Example** – finds all <p ...> tags with 'foo' inside tag and deletes it's fist occurrence:

*Command:*

**delete\_inner\_text\_for\_tag('p','foo');**

Input html:

```
<p id="123">foo1 example</p>
<p bgcolor="#ff00ff">xxx foo examplefoo</p>
<p>foo2</p>
```

Output:

```
<p id="123">1 example</p>
<p bgcolor="#ff00ff">xxx examplefoo</p> !- blue underlined foo was not deleted because this command matches first occurrence of foo only
<p>2</p>
```

**delete\_tag\_attribute(HTML tag in which attribute will be deleted,Attribute name);**

Command deletes defined tag attribute in html file.

**Example** – finds all <p> tags with attribute 'id="..."' in in html file and deletes it's first occurrence:

*Command:*

**replace\_tag\_attribute('p','id','other\_attribute="ABC"');**

Input html:

```
<p id="123" bgcolor="#ff00ff">foo1 example</p>
<p id="123" bgcolor="#ff00ff" id="123">foo1 example</p>
<p>foo1 id="100"</p>
```

Output:

```
<p bgcolor="#ff00ff">foo1 example</p>
```

```
<p bgcolor="#ff00ff" id="123">foo1 example</p> !- blue underlined text was not deleted because this command matches first occurrence of foo only
```

```
<p>foo1 id="100"</p>
```

### **delete\_all\_tag\_attributes(HTML tag in which all attributes will be deleted);**

Command deletes defined tag attribute in html file.

**Example** – finds all <p> tags with attribute 'id="..."' in in html file and deletes it's first occurrence:

*Command:*

```
replace_tag_attribute('p','id','other_attribute="ABC"');
```

Input html:

```
<p id="123" bgcolor="#ff00ff">foo1 example</p>
```

```
<p id="123" bgcolor="#ff00ff" id="123">foo1 example</p>
```

```
<p>foo1 id="100"</p>
```

Output:

```
<p>foo1 example</p>
```

```
<p>foo1 example</p>
```

```
<p>foo1 id="100"</p>
```

### **add\_tag\_attribute(HTML tag to which attribute will be added,New attribute with value);**

Command adds defined attribute to a tag in html file.

**Example** – adds to all <p> tags attribute 'id="123"' in html file:

*Command:*

```
add_tag_attribute('p','id="123"');
```

Input html:

```
<p id=123 >foo1 example</p>
```

```
<p bgcolor="#ffeccc">foo1<p>foo2</p><b>bold</b></p>
```

```
<b>foo</b>
```

Output:

```
<p id=123 id="123">foo1 example</p>
```

```
<p bgcolor="#ffeccc" id="123">foo1<p>foo2</p><b>bold</b></p>
```

```
<b>foo</b>
```

**add\_html\_after\_opening\_tag(HTML opening tag after which HTML will be added,Text/HTML string);**

Command adds defined attribute after opening tag in html file.

**Example** – adds after all <p> tags '**<b>A Bold Text Example</b>**' in html file:

*Command:*

**add\_html\_after\_opening\_tag('p','<b>A Bold Text Example</b>');**

Input html:

```
<p id=123 >foo1 example</p>
<p bgcolor="#ffecc">foo1<p>foo2</p><b>bold</b></p>
<b>foo</b>
```

Output:

```
<p id=123><b>A Bold Text Example</b>foo1 example</p>
<p bgcolor="#ffecc"><b>A Bold Text Example</b>foo1<p><b>A Bold Text Example</b>foo2</p><b>bold</b></p>
<b>foo</b>
```

**add\_html\_after\_closing\_tag(HTML closing tag after which HTML will be added,Text/HTML string);**

Command adds defined attribute after closing tag in html file.

**Example** – adds after all </p> tags '**<b>A Bold Text Example</b>**' in html file:

*Command:*

**add\_html\_after\_closing\_tag('p','<b>A Bold Text Example</b>');**

Input html:

```
<p id=123 >foo1 example</p>
<p bgcolor="#ffecc">foo1<p>foo2</p><b>bold</b></p>
<b>foo</b>
```

Output:

```
<p id=123>foo1 example</p><b>A Bold Text Example</b>
<p bgcolor="#ffecc">foo1<p>foo2</p><b>A Bold Text Example</b><b>bold</b></p><b>A Bold Text Example</b>
<b>foo</b>
```

**add\_html\_before\_opening\_tag(HTML opening tag before which HTML will be added,Text/HTML string);**

Command adds defined attribute before opening tag in html file.

**Example** – adds before all <p> tags '**<b>A Bold Text Example</b>**' in html file:

*Command:*

### `add_html_before_opening_tag('p','<b>A Bold Text Example</b>');`

Input html:

```
<p id=123 >foo1 example</p>
<p bgcolor="#ffeec">foo1<p>foo2</p><b>bold</b></p>
<b>foo</b>
```

Output:

```
<b>A Bold Text Example</b><p id=123>foo1 example</p>
<b>A Bold Text Example</b><p bgcolor="#ffeec">foo1<b>A Bold Text Example</b><p>foo2</p><b>bold</b></p>
<b>foo</b>
```

### `add_html_before_closing_tag(HTML closing tag before which HTML will be added,Text/HTML string);`

Command adds defined attribute before closing tag in html file.

**Example** – adds before all `</p>` tags `'<b>A Bold Text Example</b>'` in html file:

*Command:*

### `add_html_before_closing_tag('p','<b>A Bold Text Example</b>');`

Input html:

```
<p id=123 >foo1 example</p>
<p bgcolor="#ffeec">foo1<p>foo2</p><b>bold</b></p>
<b>foo</b>
```

Output:

```
<p id=123>foo1 example<b>A Bold Text Example</b></p>
<p bgcolor="#ffeec">foo1<p>foo2<b>A Bold Text Example</b></p><b>bold</b><b>A Bold Text Example</b></p>
<b>foo</b>
```

## Updating old templates

In Word Cleaner version 4.5 all settings are now at the template level, not the global level. What this means is these settings are now at the template level:

- **General options:** convert to xhtml, plain text conversion, convert urls & emails to links, clean MS word html.
- **Images:** what image formats to convert to
- **CSS:** you can have internal, linked or no CSS.
- **Page splitting:** split a converted file into several html files
- **Output folders:** select where you want to place converted files and images
- **Delete tags/attributes:** remove code to suit your needs.
- **Find and replace:** find specific code and change it to suit.
- **Encoding:** select the input and output encoding formats - UTF8, ANSI, ASCII

- **Metadata:** insert and control meta tags such as title, author, subject, keywords.
- **Comments/description:** write a description of the template to remind you of changes.

We have made templates easier to use, most options now can be selected by ticking options.

Updating templates is very easy.

All you need to do is edit the old template, set the options above, then click save. That's all :-)

## Regular Expressions Support

The Word Cleaner template system support regular expression commands (regex).

regular expressions are very powerful, but they can be complex. For beginners we recommend just using the standard template commands.

**Note:** its important to note that we only support .net regular expressions.

Generally .net expressions are the same as standard ones but we recommend you look at some of these third party sites for guidance on how to format commands.

[http://msdn2.microsoft.com/en-us/library/hs600312\(VS.71\).aspx](http://msdn2.microsoft.com/en-us/library/hs600312(VS.71).aspx)

<http://www.regular-expressions.info/dotnet.html>

<http://regexlib.com/CheatSheet.aspx>

<http://www.google.com/search?hl=en&q=.net+regular+expressions&meta=>

## Command Line Support

With Word Cleaner you can automatic your conversions with command line support. Please note that this feature is more for experienced programmers, it is not recommended for novice users.

In the folder: **C:\Program Files\Zapadoo\Word Cleaner4\** you will see a file called **WordCleaner4.exe**, this is the command line version.

In command mode typing:

**WordCleaner4.exe /h**

will give you the list of options.

**Command line usage:**

WordCleaner4.exe [/q] [/r] [/p] [pstag tag\_string] [/w] [/t template] [nl] /f file | folder [/o folder] [/of output\_file\_name.html] [/x] [/kis] [/css NO|FILE|INLI

NE(default)) [/dpi 96(default)] [/i JPG 85(default)|PNG|BMP|GIF|WMF] [/e .html(default)] [/b] [/a] [/fn] [/h]

Parameter List (Please see WordCleaner Settings dialog for more options):

/q - Optional, Quiet mode, no output information  
/t template\_name - Optional Template name  
/ns - Optional, Do not load settings from template source.  
/r - Optional, Include Sub-Folders  
/w - Optional, Convert with MS Office Word  
/p - Optional, Files will be converted and split by pages to separate html files (that option does not work for html input files and MS Word)  
/pstag - Optional, Files will be converted and split by tag string to separate html files  
/pidx - Optional, Files with split mode on will generate html index file  
/f source file or folder\_name - File name or folder to convert (works with /r option)  
/o output directory - Optional Output directory - by default input file directory will be used  
/of output file name - Optional Output file name - by default input file name will be used  
/x - Optional, Convert to XHTML  
/kis - Optional, Keep Image Scaling  
/dpi 96 - Optional, Enable Overwrite Image DPI with default 96 DPI value, can be from 1 up to 600 DPI  
/i JPG 85|PNG|BMP|GIF|WMF - Optional, Image type, default JPG with 85 quality (1-100), other image types do not have quality parameter  
/e .html - Optional, Output file extension, default .html  
/b - Optional, Backup file before converting  
/a - Optional, Auto rename file if file exists  
/fn - Optional, Web friendly file name format  
/h - Optional, shows this help information

### Examples:

Example 1 - convert Example.doc with example template to current folder, no messages will be shown:

```
WordCleaner4.exe /q /t example /f Example.doc
```

Example 2 - convert all supported files (checked in Settings) from Docs folder with template called 'example\_template' to 'c:\' folder:

```
WordCleaner4.exe /r /t example_template Docs /o c:\
```

Example 3 - convert Example.doc with example template to current folder, output image type PNG, XHTML:

```
WordCleaner4.exe /x /i PNG /t example /f Example.doc
```

Example 4 - convert Example.doc with example template to 'c:\' folder with 'test.txt' file name:

```
WordCleaner4.exe /o c:\ /of test.txt /t example /f Example.doc
```

## Questions:

### Can I stop the GUI showing?

No for technical reasons the GUI needs to show during the conversion process. If you use the /q command, then the GUI will only show for the duration of the conversion, it will close automatically after the conversion is done.

## How to get support

Feel free to send us a support request at any time. You can visit our support area at:

<http://www.ConvertWordToHtml.com> or email: [info@ConvertWordToHtml.com](mailto:info@ConvertWordToHtml.com)

**Remember to visit our website for the latest program updates and advice:**

<http://www.ConvertWordToHtml.com>

## Feedback and Suggestions

We welcome any feedback or suggestions you have about the Word Cleaner program. We will keep the program up to date with new releases, so do let us know your ideas for future versions. Unlike other software companies we actively respond to customer feedback, bugs are fixed within a few days of users informing us about them and if we like your feature suggestions then we usually implement these within a few weeks. Please send all comments to: [info@ConvertWordToHtml.com](mailto:info@ConvertWordToHtml.com)

## Thank you

We really appreciate the time you have taken to try our software, we really hope you like it.

© Freshideas.ie Ltd