

Documentation

Project Description

The [EZ Wild Apricot Web Designer](#) (aka WATM) is a JavaScript library that is written using the [jQuery](#) framework included with every [WildApricot](#) website. EZ Wild Apricot Web Designer will help any administrator with some knowledge of the [Document Object Model \(DOM\)](#) manage and replace nearly every piece of hard-coded and configurable text in Wild Apricot.

You can make a variety of single page and whole site changes such as:

- search and replace of a particular text string
- modify field labels on forms, button labels
- change hard-coded errors and warnings
- hide text on any user interface items

EZ Wild Apricot Web Designer module can be used to make Wild Apricot sites bi-lingual by dynamically replacing strings with a translated string. A translator will use a comma separated value (CSV)-formatted configuration file that is easily editable, making it easy to maintain hundreds of modifications in a human-readable file. A standard file with over 600 English-to-French Wild Apricot labels and strings is included.

Browser Based CSS Inspector

After successful installation, go to the public view of your Wild Apricot website, and add `?dev` into the URL. This will switch EZ Wild Apricot Web Designer into a "development" mode. You can then click on any element on the page to inspect its CSS naming properties.

There are 3 potential values that will show up when you click on any element on a page:

- the [Element ID](#)
- the CSS Path
- the [Element Class\(es\)](#).

Element ID

An Element ID is unique per web page and identifies a particular object on a web page. IDs always have a `#` at the beginning of the name and appear in the optional `id` attribute of an element. The `id` attribute is used to point to a specific style declaration in a style sheet. An example ID is `#mytable`. Using the Element ID is the most specific and usually the most accurate way to target a particular element on a page. Unfortunately not all elements in a WildApricot page have an `id` but many do.

CSS Path

The CSS Path also can define a particular element using a "chain" of HTML elements. A CSS Path describes an element in relation to other elements that contain the specific element so it can be composed of a combination of several tags. A CSS path is precise definition of an element that may be nested within multiple elements. Note that a CSS Path can change if the order of elements on a page are added or removed. As a result a CSS Path can "mutate" due to the configuration of a Wild Apricot database for example. If an Element ID is not available a CSS path can be used instead.

Element Class

An Element class describes one or more CSS classes applied to the element. A class can be used by multiple HTML elements, so as a result it is the least specific of all targets. Use the element class when you wish to apply your changes to all elements that share this class.

NOTE: The effects of using an Element class can have unintended consequences. When applying changes to an element class be aware that the changes can be applied wherever the class is applied. A change to an Element class change can cascade into multiple places on a page or even across the website. Using an Element ID or CSS Path is always unique to a specific web page.

####

Demo: [Visit the NewPath Wild Apricot Sandbox in dev mode](#). Click on any element to inspect the Element ID, Element Class(es) or CSS Path.

Using Dev Tools to inspect

[Using Copy Selector and Copy Styles to quickly find a CSS Selector](#)

[Learning more about about Cascading Style Sheets \(CSS\)](#).

[How to use Chrome Developer Tools](#)

[How to use Chrome Developer Tools to find CSS classes or IDs](#)

Change History

0.1 - Initial Release 11/19/18

0.2 - Added support for many more system flags, added French translation into CSV configuration file, changed testing button toggle label 11/27/18

0.3 - Added support for adding a style, added several more system labels, added support for limiting the language toggle display "flicker" 12/5/18

0.4 - Added support for effective primary menu design changes, added support to override any CSS class or ID 12/13/18

0.5 - Added support for mouseover, mouseout (hover) and :before and :after pseudo-elements 1/15/19

0.6 - Added support for SCSS-style variables 1/24/19

0.7 - Now managed in GitHub, fixed issues related to Internet Explorer 11 support in the CSV parser and in the WATM library 2/23/19

0.71 - The Wild Apricot Text Manager config file is not cached while in testing mode 2/6/20

0.8 - WATM will now automatically enable when it goes into public view in Wild Apricot, new `attribute` function has been added 6/15/20

0.81 - fixed code typo in wildapricot-textmanager.js due to reformatting 6/29/20

0.82 - added display of WATM version on console log 9/8/20

0.9 - added CSS class/id inspector invoked with `?dev` in URL, added more error/information logging, strip leading/trailing space in "function" field of configuration file 11/20/20

0.91 - Fixed `replace_delay` function that works for form dropdowns, added "-n" suffix to the `replace-delay` function. For example, `replace-delay-3` will delay for 3 seconds 11/23/20

0.93 - support for switching languages using embedded WildApricot "widgets" in 3rd party content management system, added a standard French translation file 2/15/22

0.94 - added support for entering/exiting Inspect mode, easy copy/paste of elements in the dev mode panel, more details in install script

0.95 - added support for Inspector Hover mode which shows the outlines of each element on a page when using the Inspector

Installation

Script Setup

1. In Admin View of Web Apricot, browse to Settings -> Site -> Global JavaScript. More details on inserting JavaScript into Wild Apricot are available in [Wild Apricot Documentation](#).
2. Copy and paste the following lines to the Global JavaScript.

NOTE: this code snippet assumes you have uploaded all files into the folder `/resources/Theme/WildApricotTextManager`

```
<!-- jQuery-CSV Source: https://github.com/evanplaice/jquery-csv -->
<script src="/resources/Theme/WildApricotTextManager/jquery.csv-0.8.9-mod.js">
</script>
<script src="/resources/Theme/WildApricotTextManager/wildapricot-textmanager.js">
</script>
<script>

    var csvFile = "/resources/Theme/WildApricotTextManager/wildapricot-textmanager-
config.csv"; // location of EZ Wild Apricot Web Designer configuration file

    var showInspectorLink = true; // this will show the Inspector mode link in the
footer of the website, set to false to turn off

    var showInspectorHover = true; // this will show the Inspector Hover mode, set to
false to turn off
    var inspectorHoverColor= "red"; // this will change the the Inspector Hover mode
outline, set to any CSS color or hex value

    var inspectorKeyword = "?dev"; // this is the URL keyword that enables Inspector
mode
    var inspectorContainerId = "el-details"; //this is the Element ID that contains the
Inspector
    var inspectorlocation = "bottom"; //this is the location of the Inspector, can be
set to "top" as well

    var textManagerMultilingualMode = false; // alternative language button toggle, set
```

```

to true to turn on multi-lingual mode
var primaryLanguageButtonName = "English"; // default language button name
var alternativeLanguageButtonName = "Français"; // alternative language button name
var languageButtonHtmlID = "languageButton"; // the Element ID of the language
button
var alternativeLanguageClassName = ".french"; // the CSS class name for all
alternative language content gadgets
var primaryLanguageClassName = ".english"; // the CSS class name for all default
language content gadgets

$(window).bind("load", function() {$('#textmanager_overlay').css('display',
'none'); }); // Fail-safe to remove white overlay
</script>

```

3. To deactivate the multilingual mode edit this JavaScript variable declaration as follows:

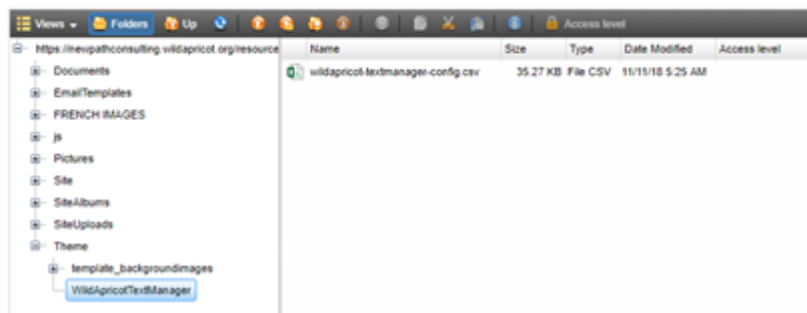
```
var textManagerMultilingualMode = false;
```

Files Setup

1. In Admin view of Wild Apricot, browse to [Website -> Files](#).
2. Create a new folder named `WildApricotTextManager` under the Theme folder.

Files

You are currently using 1 MB (0%),
100 MB limit for each file upload.



3. Upload [jquery.csv-0.8.9-mod.js](#) to the new folder.
4. Upload [wildapricot-textmanager.js](#) to the new folder.
5. Upload [wildapricot-textmanager-config.csv](#) to the new folder.

The configuration file must be saved as a CSV file format in UTF-8 format. The file must be named `wildapricot-textmanager-config.csv`.

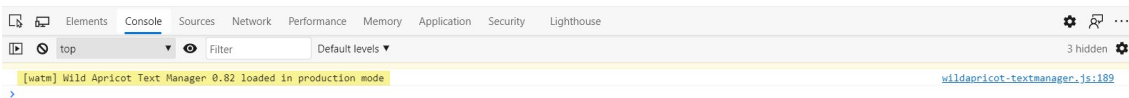
Testing Installation

To test whether EZ Wild Apricot Web Designer has been installed properly:

1. Click Public View to browse the Wild Apricot website.

2. Click the Show Inspector link in the footer of the website. You should see the "Inspector". You can also add a `/?dev` to the URL to turn on the Inspector manually.
3. Use the inspector to select any highlighted element by hovering over any element on the web page.
4. Click Exit Inspector button to leave Inspector mode.

If you do not see the Inspector or the Show Inspector link, check Dev Tools and see if EZ Wild Apricot Web Designer loaded. You should see the version number message in the Console. If you do not see this message, or you see errors EZ Wild Apricot Web Designer may not have been properly installed.



Overlay Setup (Optional)

Normally, when the EZ Wild Apricot Web Designer is enabled, the original text is displayed on the page load and then replaced within one second. This "flicker" can be hidden with the HTML and CSS below. EZ Wild Apricot Web Designer will automatically remove the overlay once the script has been completed, reducing the flicker.

Open the each Wild Apricot Page Template in use, add a [Custom HTML](#) gadget anywhere in the template. Edit the code of the Custom HTML gadget and include this HTML snippet:

```
<div id="textmanager_overlay"></div>
```

You must also add these lines to the Wild Apricot global CSS in the [Website -> CSS](#) menu:

```
#textmanager_overlay {
  position: fixed;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  background-color: white;
  z-index: 2;
}
```

Multilingual Setup (Optional)

Standard WildApricot translation file

EZ Wild Apricot Web Designer can be also be used to enable 2 languages for WildApricot websites. EZ Wild Apricot Web Designer does *not* require you to create any additional site pages or page templates. Translations can be applied for any Wild Apricot module without the need to create multiple versions of a page, event, donation form etc. A French translation file is included as sample CSV configuration file, named `FR-`

wildapricot-textmanager-config.csv . Remember to rename this file to wildapricot-textmanager-config.csv before using it. This file can be edited in Microsoft Excel or another CSV editor. When saving the file, make sure to save with the UTF-8 encoding format to preserve any accented or multi-byte characters.

Moving the language switcher button

To move the Language button from the default location:

1. In Admin View, browse to each page template in use.
2. Click Edit, then Gadgets, then drag and drop Custom HTML gadget to where you would like to place the button.
3. Click Edit code and delete the default "Insert your HTML code here".
4. Add a single space into the gadget. It will show empty but there will be a "placeholder" space.
5. Click Settings button for the new gadget. On the left panel, show Advanced and then change the HTML ID to languageButton .
6. Save the template.
7. Repeat this for each page template.
8. (Optional) Customize the look of the button by applying custom CSS to the ID #languageButton in the configuration file on the Custom CSS section.

Add the translated content

The below is an example of adding translated content to a Wild Apricot "Contact" page.

1. Start with your Contact page. In Admin Mode, browse to Website -> Site pages -> Contact and click Edit. You may have a different page on your site, and you can start with any page.
2. Click on Gadgets and drag a new Content gadget below the existing Contact us heading. If your second language is French, then type in Contactez nous into this new space. Adjust font sizes and padding to match the original content.



3. Show [Advanced settings section](#) on the left settings panel and add french to the CSS class text box.
4. Click on the original English Contact us heading and add english to the CSS class text box.

* ADVANCED

Use CSS to fine-tune appearance and behavior.
[Learn more](#)

HTML ID

CSS class

Inline style

5. Repeat for other content blocks you wish to localize and Save page. The result should look similar to below:

Contact us

Contactez nous

This is an example of a page that can be used to display contact information for your organization.

You could add a social profile gadget or subscription form gadget to provide additional ways for visitors to contact you.

The more clear your contact information is, the more more likely people will contact you with their questions.

Place your address here:

456 Street name, City, State, Country, Zip/postal code
+1 (000) XXX-XX-XX
http://examplewebsite.com
email@wildapricot.com

You can include here an interactive Google map with your organization address instead of this text block.

Ceci est un exemple de page pouvant être utilisée pour afficher les informations de contact de votre organisation.

Vous pouvez ajouter un gadget de profil social ou un gadget de formulaire d'abonnement pour fournir aux visiteurs d'autres moyens de vous contacter.

Plus vos coordonnées sont claires, plus il est probable que des personnes vous contacteront pour vous poser des questions.

Placez votre adresse ici:

456 Nom de la rue, ville, état, pays, code postal
+1 (000) XXX-XX-XX
http://examplewebsite.com
email@wildapricot.com

Vous pouvez inclure ici une carte Google interactive avec l'adresse de votre organisation au lieu de ce bloc de texte.

Note that if EZ Wild Apricot Web Designer is enabled, one of the two languages will be hidden based on the state of the language, even in the Edit mode. Please make sure `textManagerProductionMode = false` is set in Global JavaScript and then click the button to disable EZ Wild Apricot Web Designer to see both sets of languages at once.

Support for embedded WildApricot widgets on 3rd party Content Management Systems

If you are using the WildApricot widget code using iframes, use the `?secondLanguage` argument at the end of widget URL to trigger translations to be loaded.

1. Alter the WildApricot widget code to support the `?secondLanguage` widget.
2. Embed the widget on the translated page on the 3rd party CMS. This works with systems like Drupal, Squarespace and WordPress.

Example code:

For ENGLISH pages use the default language widget:

```
<iframe width='750px' height='400px' frameborder='no'
src='https://mysite.wildapricot.org/widget/join-us'
onload='tryToEnableWACookies("https://mysite.wildapricot.org");' ></iframe>
<script type="text/javascript" language="javascript"
src="https://mysite.wildapricot.org/Common/EnableCookies.js" ></script>
```

For FRENCH pages add the ?secondLanguage widget:

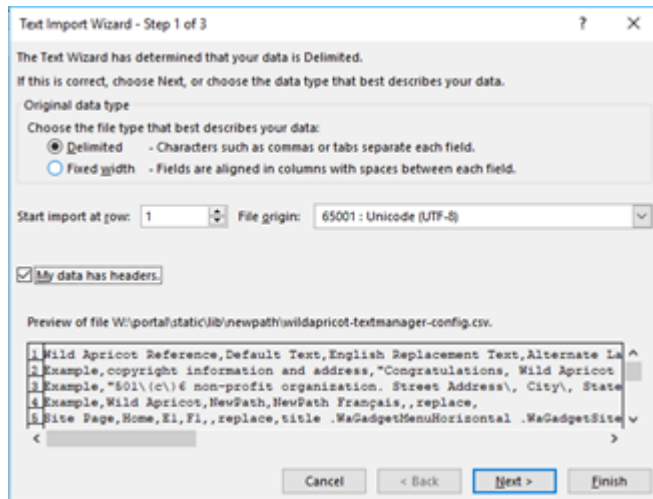
```
<iframe width='750px' height='400px' frameborder='no'
src='https://mysite.wildapricot.org/widget/join-us/?secondLanguage'
onload='tryToEnableWACookies("https://mysite.wildapricot.org");' ></iframe>
<script type="text/javascript" language="javascript"
src="https://mysite.wildapricot.org/Common/EnableCookies.js" ></script>
```

EZ Wild Apricot Web Designer Configuration File

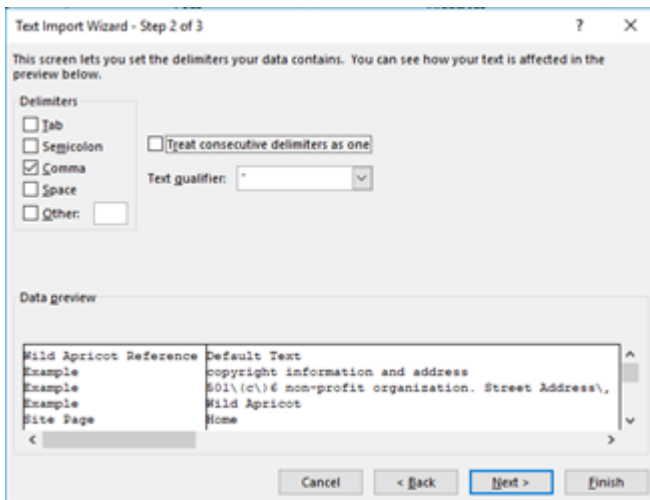
To maintain all the EZ Wild Apricot Web Designer text changes, translations and styling changes, you must edit and maintain the CSV configuration file `wildapricot-textmanager-config.csv` using Excel or another commas separated file (CSV) editor.

Instructions for editing the file in Excel

1. Open the `wildapricot-textmanager-config.csv` file Choose Delimited, check My data has headers and click Next



2. Check Comma for the Delimiters option, Click Finish



Saving the configuration file

When saving the file, use the CSV UTF-8 (Comma delimited) (.csv). **Other formats will break the configuration file!*

The configuration file may be cached in the web browser between changes while in production mode. If you don't see your configuration file changes reflected on the website, you will need to [force-refresh your browser](#) to reload the latest configuration file.

Fields in the configuration file

Note: Inserting or moving columns will break the configuration file!

Wild Apricot Reference : The field can be used to group configuration lines together in the configuration file. It is for reference use only.

Default Text : The text that will be searched for replacement in a Wild Apricot gadget or label. This is a case-sensitive and space sensitive field.

English Replacement Text : If you wish to replace the text with another value, place the new text in this field. This is an optional field.

Alternative Language Text : If you wish to translate text with another language, place the translated language in this field. This field is an optional field.

Notes : Notes can be added here for reference.

Function : This field is used to manage text. If a function is used the **Default Text** , **English Replacement Text** and **Alternative Language Text** fields will be used. It is an optional field.

Query : This is the targeted CSS ID, CSS Path or CSS Style to target for changes. It is a required field.

Style : Contains the CSS style properties and value to apply to the **Query** element

Note: You can leave the **Function** field empty if you wish to apply CSS using the **Query** and **Style** fields.

Text Management Functions for Function field:

The functions below change the text with the contents of `English Replacement Text` or the `Alternative Language Text` column. Any text in the `Default Text` column is ignored.

- `text` - sets text of the element selected by the `Query` column.
- `hide` - hides the element selected by the `Query` column.
- `button` - sets value of the button selected by the `Query` column.
- `placeholder` - sets placeholder attribute. Only used for search boxes.
- `delay` - page will pause one second before replacing text. This can be used on text elements that are written with JavaScript after page load (eg the member directory gadget).

The following functions expect to use `Default Text` column as the search criteria for replacement.

- `replace` - Searches for `Default Text` column and replaces this sub-string in any element. If `Query` column is blank, the entire page is searched.
- `replace_element` - Searches text in `Default Text` column and replaces the text of the entire element. If `Query` column is blank, the entire page is searched.
- `replace_delay` - Replaces string after one second delay. Add a `_n` suffix after `replace_delay` to increase delay beyond the default 1 second. `replace_delay_3` will delay 3 seconds.
- `attribute` - Replaces string contained in any HTML `attribute` tag

There is a subtle difference between the `text`, `replace` / `replace_delay` and `replace_element` functions.

Function `text` ignores the `Default Text` value and simply replaces the text inside the element that matches the `Query`. It is *not* a search/replace function.

Example:

If the ID is `#myelement` and you wish to replace the content with "Wild Apricot is awesome!" create the following configuration row:

- Function is `text`
- Query is `#myelement` and
- English Replacement Text is `Wild Apricot is awesome!`

Functions `replace`, `replace_delay` and `replace_element` do a search & replace based on the `Default Text` field.

Example:

If the ID is `#myelement` and you wish to find the text `Personify` anywhere in an element and replace it with `Wild Apricot` create the following configuration row:

- Function is `replace`
- Query is `#myelement` and
- Default Text is `Personify`. Make sure case is correct as the search is case-sensitive.
- English Replacement Text is `Wild Apricot`

If the original text is `Personify is awesome!` the configuration above will replace the string with `Wild Apricot is awesome!`.

Example:

If the ID is #myelement and you wish to find the text Personify anywhere inside an element and replace *the whole element* with Wild Apricot is great! create the following configuration row:

- Function is replace_element
- Query is #myelement and
- Default Text is Personify . Make sure case is correct as the search is case-sensitive.
- English Replacement Text is Wild Apricot is great!

If the original text is Personify is awesome! the configuration above will replace the element with Wild Apricot is awesome! . It simply searches for the the Default Text and when found uses the English Replacement Text .

The replace_delay function waits a preset number of seconds before making a replacement to allow any other Javascript to execute before making element changes. This is particularly useful on any content that is rendered with Javascript, such as the membership directory gadget. If you are having trouble with EZ Web Designer rendering changes, try the replace_delay function.

In the above examples, the English Replacement Text can be replaced with Alternative Language Text if multilingual mode is turned on.

Utility Functions

- inactive - disables the configuration row. This function can be used to save a configuration, but not activate it for use.

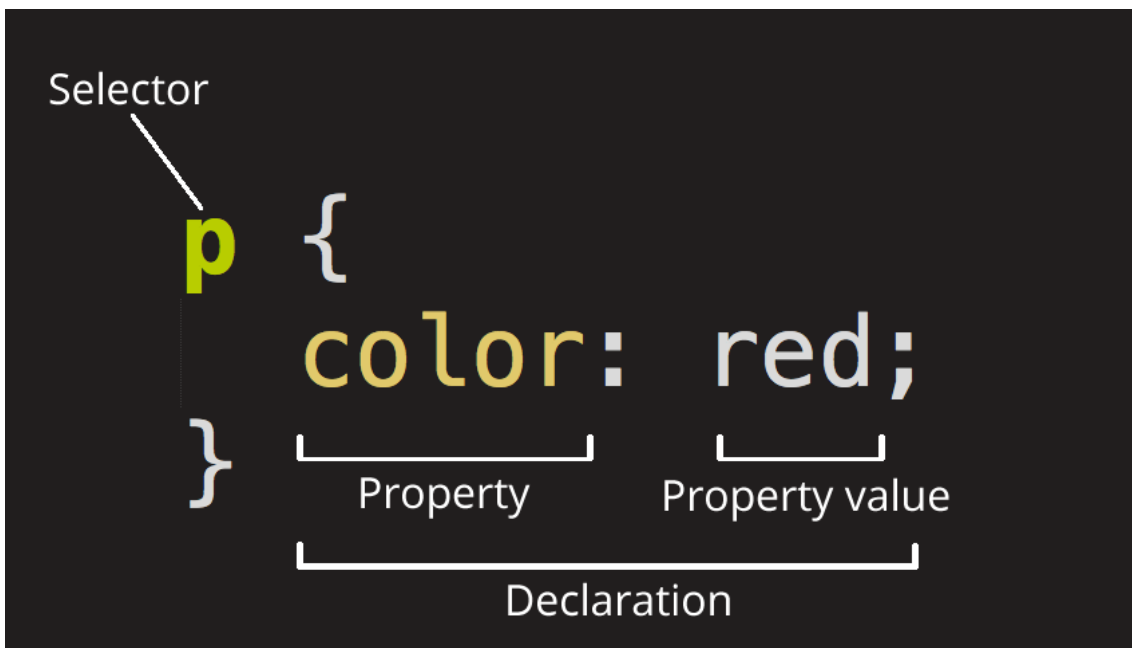
CSS-only functions:

- mouseover - Sets CSS inside an event handler when hovering over an element.
- mouseout - Sets CSS inside an event handler when no longer hovering over element.

Query : The CSS selector to target for the change. A selector can include HTML elements, classes and IDs as well as a CSS path (eg div.class > td > #myID). [Using Copy Selector and Copy Styles to quickly find a CSS Selector](#)

Style : The CSS ruleset (properties & values) to apply to selector in the Query column. Optional.

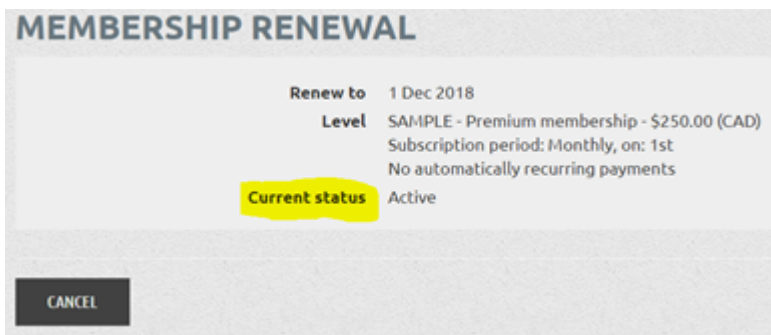
Each property and value should be enclosed with a double quote (") (eg "color":"red";)



Example of a Text Change

Change the label “Current status” on Membership Renewal system gadget:

Here is what the standard Membership Renewal system gadget looks like:



- ☐ Open wildapricot-textmanager-config.csv in your text editor or Excel.
- ☐ Search the file for “Current status”. Verify **Wild Apricot Reference** column matches the name of the gadget we’re looking at.
- ☐ Type in new text `My New Text!` in **English Replacement Text** column.

116	WaGadgetContactProfile	Social network login				text
117	WaGadgetMembershipRenewal	Renew to				text
118	WaGadgetMembershipRenewal	Level				text
119	WaGadgetMembershipRenewal	Current status	My New Text!			text
120	WaGadgetMembershipRenewal	Cancel				button
121	WaGadgetMembershipRenewal	Update and next				button
122	WaGadgetMembershipRenewal	Your profile				text

- ☐ (Optional) Type translated language text in **Alternative Language Text**.
- ☐ Save as CSV UTF-8 (Comma delimited) (*.csv)

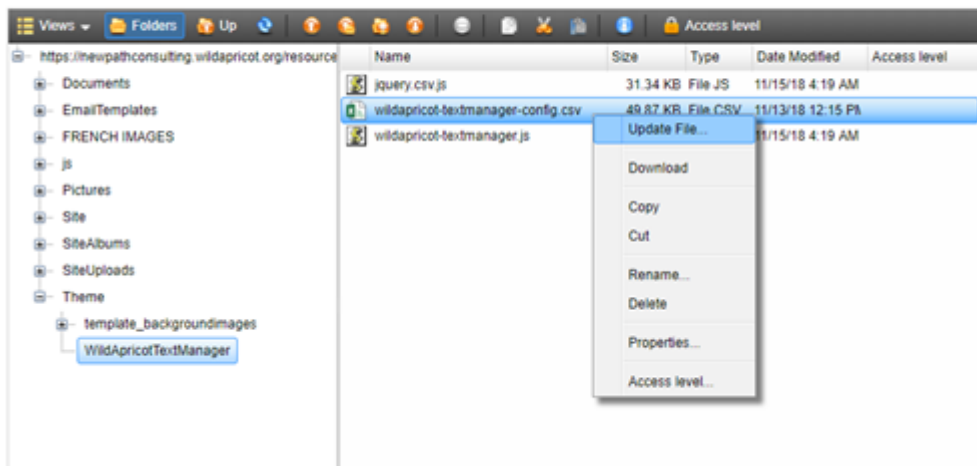
File name:

Save as type:

- ☐ Upload/Update .csv file - Note: [WebDAV is available to update](#) the WATM configuration file to speed up the process of updating the file between changes.

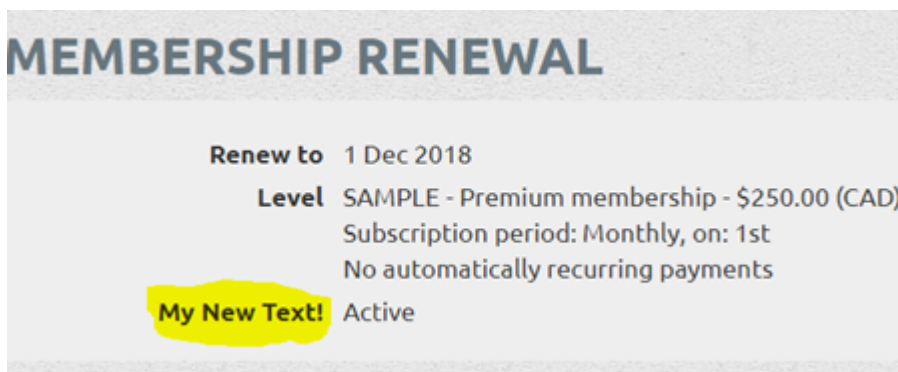
Files

You are currently using 1 MB (0%).
100 MB limit for each file upload.



- ☐ Load the membership renewal page and use the toggle buttons to test.

Here is what the new Membership Renewal system gadget should look like:



System Requirements

EZ Wild Apricot Web Designer must be used with a free or paid WildApricot account. EZ Wild Apricot Web Designer is supported on the latest versions of Chrome, Safari, Firefox and Edge. Older browsers like Internet Explorer on Windows are supported "best effort," without formal testing or 100% compatibility.

Further Examples

Hide an element:

```
Default Text: Type the 6 characters you see in the picture
Function: hide
Query: .captchaGuiding
```

Replace title of page on all menus:

```
Default Text: Directory
English Replacement Text: Members Directory
Function: replace
Query: title .WaGadgetMenuHorizontal .WaGadgetMenuVertical .WaGadgetSiteMap
.WaGadgetNavigationLinks .WaGadgetBreadcrumbs
```

Replace title of page on only main menu:

```
Default Text: Directory
English Replacement Text: Members Directory
Function: replace
Query: .WaGadgetMenuHorizontal
```

Replace text of any element of any page:

```
Default Text: 501(c)(3) non-profit organization
English Replacement Text: My Organization Name
Function: replace_element
(Note: Use backslash to escape special characters such as commas, parentheses, brackets.)
```

Set CSS of the H2 CSS selector:

```
Query: h2
Style: { "color" : "red", "font-style" : " italic" }
```

Set CSS of :before or :after (Note the use of standard CSS formatting):

```
Query: .WaGadgetCustomMenu.menuStyleNone .orientationVertical li:before
Style: { content: "|"; color: #5e5e5e }
```

Set CSS :hover of a button (Note, :hover is not needed in Query):

```
Query: .functionalButton
Style: { "color": "white", " background": "#a80532" }
```

Replace text "Purchase Organ Music" in the HTML title attribute

```
Default text: Purchase Organ Music
English Replacement Text: test
Function: attribute
Query: title
```