



From The Blog

[Homepage](#) \ [Blog](#) \ [Other](#) \ [Adding Custom Attributes To WordPress Menus](#)

Calendar

Page ▾

Navigation Label	Title Attribute
<input type="text" value="Calendar"/>	<input type="text"/>
Subtitle	
<input type="text" value="My Sub-title here !"/>	
Original: Calendar	
Remove Cancel	

[Add a custom field to WordPress menus](#)

Newsletter

Join the newsletter to receive news updates in your inbox.

[SUBSCRIBE](#)

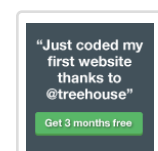
Adding Custom Attributes To WordPress Menus

Published on: 21st November 2012 **Author:** [Remi](#)

Under: [Other](#) **Comments:** [30](#)

They are so many tutorials on the web to explain “how to add a custom field to WordPress menus” but most of them treat about how to use the default CSS field of the default Description field. None of them gives a real method to add a real new field to menus. So, today we will create a simple plugin that will add a “subtitle” field to any WordPress menu.

Advertisements



👍 Share This Post



Menu

+

Menu Name
Menu
☐ Automatically add new top-level pages

Delete Menu

Calendar

Page ▼

Navigation Label

Title Attribute

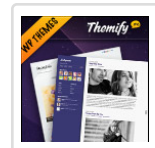
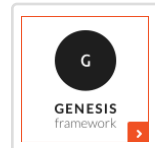
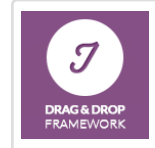
Calendar

Subtitle

My Sub-title here !

Original: [Calendar](#)

Remove | Cancel



Step 1: The Plugin Creation

Let's go, as we saw in one of my previous post, [creating a plugin](#) isn't complex. You just need to create a new folder under wp-content/plugins and called it "sweet-custom-menu", and to create inside of it a file called "sweet-custom-menu.php". Then open this file and add this code:

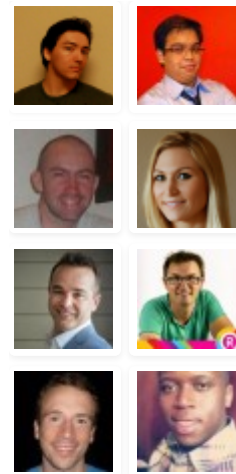
```

1 <?php
2 /*
3 Plugin Name: Sweet Custom Menu
4 Plugin URL: http://remicorson.com/sweet-
  custom-menu
5 Description: A little plugin to add
  attributes to WordPress menus
6 Version: 0.1
7 Author: Remi Corson
8 Author URI: http://remicorson.com
9 Contributors: corsonr
10 Text Domain: rc_scm
11 Domain Path: languages
12 */

```

This code is all we need to define a new plugin. Now we are

Contributors



 **Share This Post**

Here is how to create the class:

```
1 class rc_sweet_custom_menu {  
2  
3     /*-----  
4     * Constructor  
5     *-----  
6  
7     /**  
8     * Initializes the plugin by setting  
9     * localization, filters, and administration  
10    * functions.  
11    */  
12    function __construct() {  
13    } // end constructor  
14  
15    /* All functions will be placed here */  
16 }  
17  
18 // instantiate plugin's class  
19 $GLOBALS['sweet_custom_menu'] = new  
    rc_sweet_custom_menu();
```

The last line instantiates the class and add the whole class in a global variable.

Step 2: Adding Custom Fields Filter

Now that we have our class, we are going to create custom functions. The first function to add is the one that will register the “sub-title” custom field we want to be included as an advanced menu attribute. To do so, here the function to create, place this code instead of `/* All functions will be placed here */`:

```
1 /**
```

Categories

- ▶ [WordPress Tips](#)
- ▶ [WordPress Tutorials](#)
- ▶ [WordPress News](#)
- ▶ [WordPress Hosting](#)
- ▶ [WordPress Plugins](#)
- ▶ [WordPress Product Reviews](#)
- ▶ [WordPress Snippets](#)
- ▶ [WordPress Security](#)
- ▶ [WordPress SEO](#)
- ▶ [WordPress Post Series](#)

Connect on
Facebook

Recent Tweets

👍 Share This Post



```

5  * @access      public
6  * @since       1.0
7  * @return      void
8  */
9  function rc_scm_add_custom_nav_fields(
10 $menu_item ) {
11     $menu_item->subtitle = get_post_meta(
12 $menu_item->ID, '_menu_item_subtitle', true
13 );
14     return $menu_item;
15 }

```

Then, we need to tell WordPress to take our function into account, so place the code below in the `__construct()` function:

```

1  // add custom menu fields to menu
2  add_filter( 'wp_setup_nav_menu_item', array(
3  $this, 'rc_scm_add_custom_nav_fields' ) );

```



WPExplorer

@WPExplorer

Create a professional Real Estate website using the great [#WordPress](#) Theme wpexplorer.com/real-estate/

Show Summary



WPExplorer

@WPExplorer

Whenever you customize a [#WordPress](#) theme you should be using a child theme wpexplorer.com/wordpress/

wpexplorer.com/wordpress/

Show Summary



WPExplorer

@WPExplorer

The Best CDN for [#WordPress](#) - A Review by [@MaxCDN](#) by [@SouravKundu92](#)

wpexplorer.com/maxcdn/

Step 3: Saving Custom Fields

Even if the custom “sub-title” field doesn’t really exist we must create a function that will save its value on menu edition. Menu elements are in two ways custom post types, so we can use the custom post type APIs and `post_meta` method. So, to save custom menu field value, add this function under `rc_scm_add_custom_nav_fields()`:

```

1  /**
2   * Save menu custom fields
3   *
4   * @access      public
5   * @since       1.0
6   * @return      void
7   */
8  function rc_scm_update_custom_nav_fields(
9  $menu_id, $menu_item_db_id, $args ) {

```

 **Share This Post**

```
12     'item-subtitle' ])) ) {  
13         $subtitle_value = $_REQUEST['menu-  
14         item-subtitle'][$menu_item_db_id];  
15         update_post_meta( $menu_item_db_id,  
16         '_menu_item_subtitle', $subtitle_value );  
17     }  
18 }
```

In this function, we are checking if the custom field value is sent from the form we are about to create and then we simply store its value. Now we have to add the function to the appropriate hook. To do so, add these lines to `__construct()`:

```
1 // save menu custom fields  
2 add_action( 'wp_update_nav_menu_item', array(  
3     $this, 'rc_scm_update_custom_nav_fields' ),  
4     10, 3 );
```

Step 4: The Form

If you followed this tutorial step by step, you'll probably guess that we haven't created the form that must contain our sub-title field. This part is a bit more complex than the previous ones. This time we have to deal with [Walker](#). I really encourage you to read the codex about the walker class, this will really help you to understand what it is and what it does. By the way there are many great tutorials everywhere on the internet to give you more details about this class, so please check them out! In most part of the time, walkers are used to modify the HTML output of a menu. Here, we are working on the admin menu output form. Simply add this function to your main class:

```
1 /**  
2  * Define new Walker edit
```

 **Share This Post**

```
6  * @return void
7  */
8  function rc_scm_edit_walker($walker,$menu_id)
9  {
10     return 'Walker_Nav_Menu_Edit_Custom';
11 }
12 }
```

and then this to the constructor:

```
1  // edit menu walker
2  add_filter( 'wp_edit_nav_menu_walker', array(
    $this, 'rc_scm_edit_walker'), 10, 2 );
```

What this does is replacing the default admin edit menu form by a custom one. Now that the filter, is added, copy these two lines at the bottom of the sweet-custom-menu.php file, outside the rc_sweet_custom_menu class:

```
1  include_once( 'edit_custom_walker.php' );
2  include_once( 'custom_walker.php' );
```

We are about to include two files. The first one

“edit_custom_walker.php” is the one that will modify the default form to edit the menu. It’s in this file that we are going to add the sub-title field.

The second one is the welker used on the website frontend, this is the file that will modify the menu output to your visitors.

As the “edit_custom_walker.php” is a bit long, i am not going to paste the whole code. You can download the file [here](#). The only code i added to it is from line 174 to line 185. Here is the added code:

 **Share This Post**

```
3 echo $item_id; ?>">
4 <?php _e( 'Subtitle' ); ?><br />
4 <input type="text" id="edit-menu-item-
  subtitle-<?php echo $item_id; ?>"
  class="widefat code edit-menu-item-custom"
  name="menu-item-subtitle[<?php echo $item_id;
  ?>]" value="<?php echo esc_attr(
  $item->subtitle ); ?>" />
5 </label>
6 </p>
```

If you want to add some more fields to the menu, simply duplicate these lines and copy & paste them. Once this step is done, the menu should be working in the administration. You should now be able to see the new “sub-title” field when adding a new element to a menu. If it’s not the case, make you followed the tutorial step by step. It’s now time to output the subtitle value on the frontend.

Step 5: Custom Field Output

If everything is working fine in the administration, you probably now want to display the subtitle on the frontend. Open `custom_walker.php` and add this code to it:

```
1 <?php
2 /**
3  * Custom Walker
4  *
5  * @access      public
6  * @since       1.0
7  * @return      void
8  */
9 class rc_scm_walker extends Walker_Nav_Menu
10 {
11     function start_el(&$output, $item,
12     $depth, $args)
13     {
14         global $wp_query;
```

 Share This Post


```

16         $class_names = $value = '';
17
18         $classes = empty( $item->classes )
19 ? array() : (array) $item->classes;
20
21         $class_names = join( ' ',
22 apply_filters( 'nav_menu_css_class',
23 array_filter( $classes ), $item ) );
24         $class_names = ' class="'.
25 esc_attr( $class_names ) . '"';
26
27         $output .= $indent . '<li
28 id="menu-item-'. $item->ID . '" . $value .
29 $class_names . '>';
30
31         $attributes = ! empty(
32 $item->attr_title ) ? ' title="'. esc_attr(
33 $item->attr_title ) . '" : '';
34
35         $attributes .= ! empty(
36 $item->target ) ? ' target="'. esc_attr(
37 $item->target ) . '" : '';
38
39         $attributes .= ! empty( $item->xfn
40 ) ? ' rel="'. esc_attr(
41 $item->xfn ) . '" : '';
42
43         $attributes .= ! empty( $item->url
44 ) ? ' href="'. esc_attr(
45 $item->url ) . '" : '';
46
47         $prepend = '<strong>';
48         $append = '</strong>';
49         $description = ! empty(
50 $item->description ) ? '<span>'. esc_attr(
51 $item->description ) . '</span>' : '';
52
53         if( $depth != 0 )
54         {
55             $description = $append =
56 $prepend = "";
57         }
58
59         $item_output = $args->before;
60         $item_output .= '<a'. $attributes
61 . '>';
62
63         $item_output .=
64 $args->link_before . $prepend . apply_filters(
65 'the_title', $item->title, $item->ID
66 ) . $append;
67
68         $item_output .=
69 $description . $args->link_after;
70         $item_output .= '
71 . $item->subtitle . '</a>';
72         $item_output .= $args->after;
73
74         $output .= apply_filters(

```


 **Share This Post**

```
48 | }
```

This code is pretty standard. The only important thing here is these line:

```
1 | $item_output .= ' '.$item->subtitle.'';
```

Step 6: Call The Walker!

In the last step you will need to open the header.php file in your theme folder. It's where most part of the time the menu is called. If not, then contact the theme creator to know where it's called.

Find the `wp_nav_menu()` function.

Then simply add the walker parameter:

```
1 | 'walker' => 'rc_scm_walker'
```

That's it ! I hope you enjoyed adding a custom field to a WordPress menu. You can now modify a bit this code to add other fields such as checkboxes, select dropdown, textareas etc... Feel free to add a comment in the comments section. If you found a bug, if you improve it etc...

Sweet Custom Menu Plugin


👍 Share This Post



Enjoy!

[Get The Plugin →](#)

About the Author

BIO	TWITTER	GOOGLE+
<div></div> <div>Remi<p>Remi is an expert WordPress developer that coded many great free and premium themes and plugins. His experience on WordPress allows him to produce great stuff and to propose advanced tutorials. (remicorson.com)</p></div>		

Latest Articles Posted Under: Other

- ▶ [WordPress 3.9+ TinyMCE 4 Tweaks: Adding Styles, Buttons, Fonts, Drop-downs & Pop-Ups](#)
- ▶ [10 Premium WordPress Add-ons You Must Have](#)
- ▶ [9 Women in WordPress You Should Be Following](#)
- ▶ [Dear Diary: A Letter from WordCamp San Francisco](#)

 [Share This Post](#)



Gregg says:

[REPLY TO THIS MESSAGE](#)



Remi, This is awesome! I had no idea you could do this. Thanks for the great tutorial.

Remi says:

[REPLY TO THIS MESSAGE](#)

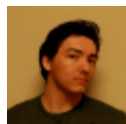


Thanks Gregg, this tutorial is just a simple example, but you could do much more using this method. I'm currently working on an awesome plugin related to WordPress menus that should be really handfult!

AUTHOR

AJ Clarke | WPExplorer says:

[REPLY TO THIS MESSAGE](#)



Can't wait to see what you're working on!

 **Share This Post**



Remi says: [REPLY TO THIS MESSAGE](#)



All i
can
say
is
that i
am
doing
it in
parntership
with
the
codecanyon
top
seller!!!

Mark says:

[REPLY TO THIS MESSAGE](#)



Sounds good but missing a
showcase. And got a question:
Could I add in this way some
more targets (_top) in menu link?

👍 Share This Post



Hi Mark, i sent the plugin to the WordPress official repo, so you can downolad and use it easily. And to answer your question, yes you can use this method to add links target (even id wordpress gives you by default a checkbox to open a link in a new window)

AUTHOR

Kev says:

REPLY TO THIS MESSAGE

 **Share This Post**

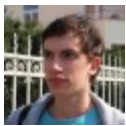
I tried this using a twenty-twelve child theme and get this error when I add the walker parameter to headers.php:

Fatal error: Using \$this when not in object context in .../wp-includes/class-wp-walker.php on line 185

I'm pretty sure I followed the instructions that came with the plugin correctly.

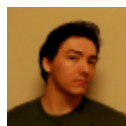
jumbowei says:

REPLY TO THIS MESSAGE



I recently ran into the same problem and could fix it. You have to use `wp_nav_menu(array('menu' => 'main', 'walker' => new rc_scm_walker));`

AJ Clarke | WPExplorer says: REPLY TO THIS MESSAGE



Thanks!

👍 Share This Post



Hi,Remi

I got these notice after using
debug bar plugin : Trying to get
property of non-object online : 39
41 42 44 please help please.

Best regards,
Decneo

Remi says:

REPLY TO THIS MESSAGE



Where did you place this
code ?

AUTHOR

Decnoe says:

REPLY TO THIS MESSAGE



Hi, Remi

I've placed this
code on my
theme_functions.php

 **Share This Post****Steven says:****REPLY TO THIS MESSAGE**

This might be asking a lot, but this is the closest I've come across in my research so far.

I'm looking to add a `data-reveal-id=""` field to my menu items (specifically one of them) to trigger a PopPop! (<http://wordpress.org/extend/plugins/poppop/>) modal window.

I added the additional section to the `edit_custom_walker.php` file like you recommended:

```
<label for="edit-menu-item-subtitle-">
```

```
<input type="text" id="edit-menu-item-subtitle-"  
class="widefat code edit-menu-item-custom" name="menu-item-subtitle[]" value="data-reveal-id "; ?>" />
```

And the attribute item appears in

 **Share This Post**

ensure that it actually outputs the field's contents into the front-end of my site:

```
$item_output .= '  
'.$item->subtitle.'";
```

Also, currently, whatever I type in the field reverts to a value of 0 when I save and replaces the subtitle value. Could be related to the above unmodified section, but I thought I'd post that too, in case it was indicative of me missing something else along the way.

Thanks in advance if you get the opportunity to answer this.

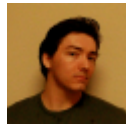
- Steven

Steven says:

REPLY TO THIS MESSAGE



Never mind, I just used a 'jQuery var newHTML =' function to find and replace the menu item in question.

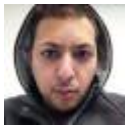
 **Share This Post**

Sorry no one
ever got back to
you for the
initial comment!
Glad you got it
working though
;)

ADMIN

Rami Girgis says:

REPLY TO THIS MESSAGE



I think it's easier to edit the
custom_walker.php file and
remove \$item_output .= '
'.\$item->subtitle."; (make
sure to leave the close in)
and then add a new line for
\$attributes above that (e.g.
copy and paste \$attributes
.= ! empty(\$item->xfn) ? '
rel="' . esc_attr(\$item->xfn)
' : "'; and then change xfn
(two instances) to subtitle,
and rel to data-reveal-id.
Then in
edit_custom_walker.php
you can change the label
"Subtitle" to "Data Reveal"

👍 Share This Post



in the form in the wp admin menu section

Alex says:

REPLY TO THIS MESSAGE



The plugin is not working. Like already said above it gives an error: Fatal error: Using \$this when not in object context in /wp-includes/class-wp-walker.php on line 185.

Do you have a working version?

Alex says:

REPLY TO THIS MESSAGE



Just to let other know, the description at <http://wordpress.org/extend/plugins/sweet-custom-menu/installation/> is wrong, you should define the walker in wp_nav_menu like this: 'walker' => new rc_scm_walker

 **Share This Post**

Hi Alex, thanks for mentioning the error. You're totally true. I have to update the description.

AUTHOR

julian says:

[REPLY TO THIS MESSAGE](#)



awesome! now I can have everything I want on my menu

Fernando says:

[REPLY TO THIS MESSAGE](#)



I'm writting 'walker' => 'rc_scm_walker' and I get the error: Using \$this when not in object context in /wp-includes/class-wp-walker.php on line 185. What I need to write or to change to make functional this plugin?

Rami Girgis says:

[REPLY TO THIS MESSAGE](#)

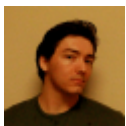
 **Share This Post**

make sure you are using
the same type of single
quotes that are already
being used in the other
values/keys in the array,
and then change 'walker'
=> 'rc_scm_walker' to
'walker' => new
rc_scm_walker

Ciprian Tepes (@cippo_ro) says:[REPLY TO THIS MESSAGE](#)

Thank you very much for your
post! It really helped me out to
build a color picker for each
menu item.

Cheers!

AJ Clarke | WPExplorer says:[REPLY TO THIS MESSAGE](#)

Nice! Would love to see
what you came up with, is
this for a theme/plugin
available somewhere?

👍 Share This Post



'walker' => new rc_scm_walker"
is causing a parse error.. Can you
post the code intact inside the
wp_nav_menu() ?

Rami Girgis says:

[REPLY TO THIS MESSAGE](#)



Native Imaging: you are using
the wrong type of quotes (not
your fault, the code in the
tutorials is using the wrong type).
Make sure you manually type it in
instead of copying and pasting.
Or copy and paste an existing
key/value from the array and just
change the values to walker and
new rc_scm_walker.

AJ Clarke | WPExplorer says:

[REPLY TO THIS MESSAGE](#)



The issue is that when we
updated to 3.6 recently it
appears our syntax
highlighter has stopped
working. I'll get that fixed

👍 Share This Post



ADMIN

ozenyxozzy says:

REPLY TO THIS MESSAGE



Try:

```
1 $walker = new  
  rc_scm_walker;
```

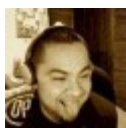
and add that as

```
1 'walker' =>  
  $walker,
```

This sholt fix it

onilinkcr says:

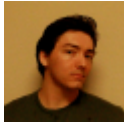
REPLY TO THIS MESSAGE



This may be a dumb question but
can you call TWO Walkers or not?
I mean one would be for your
tutorial and the other I need for a
description of the menu item.

Possible?

👍 Share This Post



You should combine them
into one walker.

ADMIN

Leave a Reply

Enter your comment here...

← PREVIOUS


5 WORDPRESS TWEAKS FOR THE HOLIDAY
SEASON

NEXT →

SITE RE-DESIGN, NEW FREEBIES & GIVEAWAY
(CLOSED)

Notice how fast this site loads? It's because
WPEngine.

[LEARN MORE](#)

 Share This Post



Recent Coupons

© 2014 WP Explorer All Rights Reserved. A SimpleWorkz Company.

WordPress

Twitter

Facebook

Google+

Privacy Policy | Site Terms & Disclosures | Hosted by WPEngine

25% Off EngineThemes

15% Off MotoPress

50% Off Hostgator VPS Hosting

2 Months Free Hosting

60% Off SiteGround Hosting

Themify Buy 1 Get 1 Free

Recent Themes

Fashionista Responsive Masonry Blog WordPress Theme

Explore Directory Child WordPress Theme

Ambiance Pro Genesis WordPress Theme

Event Parallax WordPress Theme

Magtastico Masonry WordPress Theme

Kraken Business WordPress Theme

Recent From Blog

RSS

Newsletter

Best Real Estate WordPress Themes

MaxCDN Review: The Best CDN for WordPress?

WordPress Tips: The Anatomy of A WordPress Theme

10 WordPress Resources You Need to Bookmark Right Now

WordPress 3.9+ TinyMCE 4 Tweaks: Adding Styles, Buttons, Fonts, Drop-downs & Pop-Ups