WP Smart Sort Premium

User Manual
V 2.0

Contents

Introduction

WP Smart Sort Premium is a WordPress plugin that provides options to users of your blog to customize the display of postings in a variety of ways. WPSS Premium hugely enhances WordPress's capability as a CMS and allows regular blog readers the freedom to view exactly what they want out of your blog.

(This plugin has been built around core WordPress hooks, filters and actions and offers maximum compatibility with other plugins.)

Features

- Sort/Order your posts by any field you can think of. Includes sorting by any field in the WP posts table e.g. Title, Date, Comment Count Etc as well as any custom fields found.
- Filter/Refine your view of posts. Users can fill out form of options and display only what they want to see, great for CMS implementation of WordPress.
- Advanced Search form. Allow you users to comprehensively search through posts and choose to limit to categories, tags custom fields etc.
- Fully compatible with WP Super Cache so database load is drastically reduced.
- Uses WP Rewrite to create "pretty permalinks".
- Custom Index table means your index database only includes what you need and doesn't bloat your WP installation.
- Index table automatically updates when posts are added/deleted.
- Custom Fields on steroids! Assign your custom fields a data type and even allow array values.
- Custom implementation of form fields so that it fits perfectly with your blog's look and feel.

Installation

You should have already received a zip file containing all the WPSS Premium files when you purchased your copy. If you did not or if you ever loose these files please login to your client services account and download a fresh copy.

Extract the "wp-smart-sort" folder and upload it into the "plugins" directory of your WordPress installation (On WP 2.7 You can simply navigate to the "Add new" tab of the plugins menu and upload the zip file you received directly). Activate the plugin, you will now see a new menu "WP Smart Sort" you will find a licensing tab for the plugin under this menu.

Licensing

You have two options.

1. Upload the license.lic file you received to the wp-smart-sort-premium folder.
2. Open the license.lic file in a text editor like notepad and copy and paste the license code into the box provided on the plugin license page and "Save License".

For setup and settings refer to that section in this manual.

Core Concepts

WPSS Premium functions by creating an additional table in your MYSQL database, the "Index" table. This table's structure is flexible and includes an index of data for each post in your blog. This index table is the core of the plugin and allows for rapid searching, sorting and filtering of posts. In order for a field to be available in the Sort/Search/Filter options pages in the admin that field first needs to be included in the Index table.

Example:
If you wanted to allow users to sort by the date a post was last modified you would need to "Add" the 'post_modified' field to the Index table first. Once this is done, this plugin automatically creates a new column in the index table containing 'post_modified" dates. You now can view this option in the Sort/Search/Filter options pages.

Setup

Index Table

The index table options pages allows you to choose which fields to include for use in the Sort/Search/Filter options pages of the plugin. You will see 4 major sections on this page: posts table, meta table, tags, categories.

Posts Table

This section pulls all the fields that are included in WordPress's MYSQL post table. You will see the option to enter in the display text, this is text that the user will see on the blog front end to identify this field.

It is a good idea to change this text to something easily understandable and intuitive to the user.

Meta Table

This section shows all the fields that are included in the custom fields of WordPress posts. Besides the usually display text option there are 3 other options for custom fields.

1. **Field Type**

- This feature allows you to specify what kind of data is found in the custom field.

  By specifying this you allow MYSQL to better handle the information in this field. For example if you have a field of ratings of 1-10 and the field is left as LONG TEXT and you sort the posts by field MYSQL will sort in alphabetical order: 1, 10, 2, 3, 4 …

  For this reason you can choose to specify if a field contains numbers that the type should be "BIGINT".

  It should be noted that if you specify a field type and a post has custom field and the value is not of that type that MYSQL may delete or modify that field to 'fit' it to that data type.

2. **Field is Array**

- This option allows you to specify that a single custom field includes a list/array of information.

  E.g. You may create a custom field with key: "Brands" and values: "Nike, Reebock, Addidas"

3. **Array Separator**

- This box allows you to type the characters that you use to separate the information in the array usually a person would use a comma.

Tags

Include your blog post tags in the index table.

Categories

Include your blog post categories in the index table.

Sort

Displayed on this page is a list of available field, from the index table, which you can select to include in the users list of sort options. You will note that Category, Tag and Custom Fields that are arrays are not included in this list as it would make little sense to sort by an array of values unless the 1st value was the dominate value which this plugin does not provide for.

Default Sort Direction: This option sets the default sort direction for your blog.

Title for Drop Down Menu: This option sets the title that will be used when displaying the Drop Down Menu.

Filter

The Filter options page provides several options for displaying filter fields to your users. All the fields in the Index table are available for inclusion into your filter form.

Once you select the checkbox of a field you want a drop down menu will become available to select the type of field you want the user to utilize to make their selections.

Form Field Types:

- Checkboxes
  - Display options as a list of checkboxes, user can select as many as they like.
- Drop Down
  - Drop down select box of options, only one can be selected.
- Range – User Input
  - This will display an output of 2 text boxes for the user to type in minimum and maximum values.
- Range – Drop Down
  - This will display and output of 2 drop down select boxes.

Data Source:

Can choose to let WPSS Premium automatically generate a list of options for your chosen filter field or you can select from 2 manual options.

- Semi Manual
  - WPSS Premium will automatically generate a list of available options and you will be able to select with options you want included.

- Fully Manual
  - You are able to Type a display title and the hidden value for each option you wish to be included in the filter options.

Search

Search form is very similar to the filter form except it's scope lies across all the blog posts on the blog where as the filter's scope lies only within the current view of category, tag, date, archive etc.

Theme Implementation

Basic Implementation

Open your template file for editing, WordPress has a built in editor under the design tab. You will need to edit all the template files that you wish the form to display on. Eg. Category.php, index.php, archive.php etc.

Most commonly you would find the section in the template where it says:

```
<?php if (have_posts()) : ?>
```

Straight after this you would need to include one or all of the following.

For the Sort Drop down:

```
<?php if (function_exists('placesort')) { placesort(); } ?>
```

For the Filter Form:

```
<?php if (function_exists('placefilter')) { placefilter(); } ?>
```

For the Search Form:

```
<?php if (function_exists('placesearch')) { placesearch(); } ?>
```

For the Post's Per Page Form:

```
<?php if (function_exists('placelimit')) { placelimit(); } ?>
```

Plugin API

There are several functions available that you can call to customize you implementation of WPSS Premium in the plugin API.

Sort Form

Function: ss_get_placesort()

Returns: array of sort fields

- title
- asc_links
- desc_links
- display text

Example:

```
<?php
$sort_fields = ssp_get_placesort();

    if ($sort_fields['display'] == "CSS Dropdown Menu") {
        $f_html = '';
        foreach ($sort_fields['fields'] as $field) {
            $f_html .= '<li><span class="dir">'.$field->field_title.'</span>
                                <ul>
                                    <li><a href="'.$field-
>asclink.'">Ascending</a></li>
                                    <li><a href="'.$field-
>desclink.'">Descending</a></li>
                                </ul>
                            </li>';
        }

        $sorthtml = '<div>
                        <ul class="dropdown dropdown-horizontal">
                            <li><span
class="dir">'.$sort_fields['title'].'</span>
                                <ul>
                                    '.$f_html.'
                                </ul>
                            </li>
                        </ul>
                    </div>';
    }
    else {
        $f_html = '';
        foreach ($sort_fields['fields'] as $field) {
            $f_html .= '<option value="'.$field->asclink.'"
onClick="window.location = \''.$field->asclink.'\'; return false;" >'.$field-
>field_title. " " . __('ASC') .'</option>';
            $f_html .= '<option value="'.$field->desclink.'"
onClick="window.location = \''.$field->desclink.'\'; return false;" >'.$field-
>field_title. " " . __('DESC') .'</option>';
        }

        $sorthtml = '<select>'.$f_html.'</select>';
    }

    echo "$sorthtml";
?>
```

Filter Form

Function: ss_get_placefilter()

Returns: array of filter fields.

- title
- formname
- formaction
- formmethod
- hiddenfields
- clearfilterurl
- display
- fields [array]
  - (more info)

Example:

```
<?php
$filter_fields = ssp_get_placefilter();

    if ($filter_fields->display == "CSS Dropdown Menu") {
        $filter_html = "<form name='$filter_fields->formname'
action='$filter_fields->formaction' method='$filter_fields->formmethod'>
                        $filter_fields->hiddenfields
                            <div style='float:left;'>
                                <ul class='dropdown dropdown-horizontal'>
                                    <li><span class='dir'>$filter_fields-
>title</span>
                                        <ul>";

        foreach ($filter_fields->fields as $field) {
            $filter_input = '';
            switch ($field->type) {
               case 'dropdown':
                 $filter_input = "<select name='$field->id' id='$field->id'>";
                 $filter_input .= "<option value=''></option>";
                 foreach ($field->values as $txt => $value) {
                    $filter_input .= "<option value='$value'>$txt</option>";
                 }
                 $filter_input .="</select>";
                 break;
               case 'rangeuser':
                 $filter_input = "Between:<br>";
                 $filter_input .= "<input style='width:90%;' type='text'
name='$field->id' id='$field->id' />";
                 $filter_input .= " and ";
                 $filter_input .= "<input style='width:90%;' type='text'
name='$field->id2' id='$field->id2' />";
                 break;
               case 'rangedropdown':
                 $filter_input = "Between:<br>";
                 $filter_input .= "<select name='$field->id' id='$field->id'>";
                 $filter_input .= "<option value=''></option>";
                 foreach ($field->values as $txt => $value) {
                    $filter_input .= "<option value='$value'>$txt</option>";
                 }
                 $filter_input .="</select>";
```

```
                $filter_input .= "<br> and <br>";
                $filter_input .= "<select name='$field->id2' id='$field-
>id2'>";
                $filter_input .= "<option value=''></option>";
                foreach ($field->values as $txt => $value) {
                   $filter_input .= "<option value='$value'>$txt</option>";
                }
                $filter_input .="</select>";
                break;
              case 'checkbox':
                foreach ($field->values as $txt => $value) {
                   $filter_input .= "<input type='checkbox' name='$field->id'
id='$field->id' value='$value' />
                                    <label for='$field->id'>$txt</label><br>";
                }
                break;
            }

            $filter_html .= '<li><span class="dir">'.$field->title.'</span>
                            <ul>
                                <li><span>'.$filter_input.'</span></li>
                            </ul>
                        </li>';
        }

        $filter_html .= '              </ul>
                                </li>
                              </ul>
                             <input class="dropmenupushbutton" type="submit"
value="Go" /><input class="dropmenupushbutton" type="submit" value="Clear
filters" onClick="window.location = \''.$filter_fields->clearfilterurl.'\';
return false;" /></div></form>';

    }
    else {
        $filter_html = "<form name='$filter_fields->formname'
action='$filter_fields->formaction' method='$filter_fields->formmethod'>
                        $filter_fields->hiddenfields
                        <div><span>$filter_fields->title</span>";

        foreach ($filter_fields->fields as $field) {
            $filter_input = '';
            switch ($field->type) {
               case 'dropdown':
                 $filter_input = "<select name='$field->id' id='$field->id'>";
                 $filter_input .= "<option value=''></option>";
                 foreach ($field->values as $txt => $value) {
                    $filter_input .= "<option value='$value'>$txt</option>";
                 }
                 $filter_input .="</select>";
                 break;
               case 'rangeuser':
                 $filter_input = "Between:";
                 $filter_input .= "<input style='width:90%;' type='text'
name='$field->id' id='$field->id' />";
                 $filter_input .= " and ";
                 $filter_input .= "<input style='width:90%;' type='text'
name='$field->id2' id='$field->id2' />";
                 break;
               case 'rangedropdown':
                 $filter_input = "Between:";
                 $filter_input .= "<select name='$field->id' id='$field->id'>";
                 $filter_input .= "<option value=''></option>";
```

```
                foreach ($field->values as $txt => $value) {
                   $filter_input .= "<option value='$value'>$txt</option>";
                }
                $filter_input .="</select>";
                $filter_input .= " and ";
                $filter_input .= "<select name='$field->id2' id='$field-
>id2'>";
                $filter_input .= "<option value=''></option>";
                foreach ($field->values as $txt => $value) {
                   $filter_input .= "<option value='$value'>$txt</option>";
                }
                $filter_input .="</select>";
                break;
              case 'checkbox':
                foreach ($field->values as $txt => $value) {
                   $filter_input .= "<input type='checkbox' name='$field->id'
id='$field->id' value='$value' />
                                     <label for='$field->id'>$txt</label>";
                }
                break;
            }

            $filter_html .= '<div><span style="font-weight:bold;">'.$field-
>title.':</span>
                           <span>'.$filter_input.'</span></div>';
        }

        $filter_html .= '<input class="" type="submit" value="'. __("Go").'"
/><input class="" type="submit" value="'. __("Clear Filters").'"
onClick="window.location = \''.$filter_fields->clearfilterurl.'\'; return
false;" /></div></form>';

    }

    echo "$filter_html";
?>
```

Search Form

Function: ss_get_placesearch()

Returns: array of search fields.

- title
- formname
- formaction
- formmethod
- hiddenfields
- mainsearchboxhtml
- mainsearchboxid
- submitid
- display
- fields [array]
  - (more info)

Example:

```
<?php
$search_fields = ssp_get_placesearch();

    $search_html = "<form name='$search_fields->formname'
action='$search_fields->formaction' method='$search_fields->formmethod'>";
    // Make this optional based on user choice.
    $search_html .= "<label class='hidden' for='$search_fields-
>mainsearchboxid'>".__('Search for:')."</label>
                        <div>
                            $search_fields->mainsearchboxhtml
                        </div>
                        <ul>";
    //make this optional based on if advanced search is being used.
    $toggle_js      =
"document.getElementById('ssp_advanced_search').style.display = 'block';

document.getElementById('ssp_advanced_toggle').style.display = 'none';
                        return false;";
    $search_html   .= '<div id="ssp_advanced_toggle" style="'.$hide_toggle.'">
                        <a href="" onClick="'.$toggle_js.'" >Show Detailed
Search</a>
                        </div>';

    $hide_adv       =  "display:none;";
    $search_html   .= '<div id="ssp_advanced_search" style="'.$hide_adv.'">';

    $toggle_js      =
"document.getElementById('ssp_advanced_search').style.display = 'none';

document.getElementById('ssp_advanced_toggle').style.display = 'block';
                        return false;";
    $search_html   .= '<div><a href="" onClick="'.$toggle_js.'" >Hide Detailed
Search</a></div>';

    foreach ($search_fields->fields as $field) {
        $search_input = '';
```

```
        switch ($field->type) {
           case 'dropdown':
             $search_input = "<select name='$field->id' id='$field->id'>";
             $search_input .= "<option value=''></option>";
             foreach ($field->values as $txt => $value) {
                $search_input .= "<option value='$value'>$txt</option>";
             }
             $search_input .="</select>";
             break;
           case 'rangeuser':
             $search_input = "Between:<br>";
             $search_input .= "<input type='text' name='$field->id' id='$field-
>id' />";
             $search_input .= " and ";
             $search_input .= "<input type='text' name='$field->id2' id='id2'
/>";
             break;
           case 'rangedropdown':
             $search_input = "Between:<br>";
             $search_input .= "<select name='$field->id' id='$field->id'>";
             $search_input .= "<option value=''></option>";
             foreach ($field->values as $txt => $value) {
                $search_input .= "<option value='$value'>$txt</option>";
             }
             $search_input .="</select>";
             $search_input .= "<br> and <br>";
             $search_input .= "<select name='$field->id2' id='$field->id2'>";
             $search_input .= "<option value=''></option>";
             foreach ($field->values as $txt => $value) {
                $search_input .= "<option value='$value'>$txt</option>";
             }
             $search_input .="</select>";
             break;
           case 'checkbox':
             foreach ($field->values as $txt => $value) {
                $search_input .= "<input type='checkbox' name='$field->id'
id='$field->id' value='$value' />
                                  <label for='$field->id'>$txt</label><br>";
             }
             break;
           case 'textbox':
             $search_input = "<label for='$field->id'>Search:</label>";
             $search_input .= "<input type='text' name='$field->id' id='$field-
>id' />";
             break;
        }

        $search_html .= '<li><a href="">'.$field->title.'</a><br />
                    '.$search_input.'</li>';
    }
    //make this optional based on if advanced search is being used.
    $search_html .= '</div>';
    $search_html .= "</ul>
                      $search_fields->hiddenfields
                     <input type='submit' id='$final_search_fields->submitid'
value='Search' />
                     </form>";

    echo $search_html;
?>
```

Posts per Page

Function: ss_get_placesearch()

Example:

```
<?php
$limit_values = array(5,10,25,50,100,200);

    $thelimitvar = ssp_get_placelimit();

    $html = "<select id='sshow' name='show'>";
    foreach ($limit_values as $value) {
        $selected = "";
        if ($thelimitvar->current == $value) {
            $selected = "selected='selected'";
        }
        $html .= "<option value='$value' onClick='window.location=".'"'.
$thelimitvar->url . "$value/" . '"'. "' $selected>$value</option>";
    }
    $html .= "</select>";

    echo $html;
?>
```

Support

Visit http://dyasonhat.com/support/