# CustomList4SobiPro

Display module for listing custom queries on SobiPro

v1.1.0 July - 2011

Contents	
CustomList4SobiPro	3
Installing CustomList4SobiPro	4
Configuring CustomList4SobiPro	5
Setting up a Simple Rule	7
Setting up an Advanced Rule	8
Setting up an ItemID	9
Sample Configurations	10
Configuring language files	11

### CustomList4SobiPro

CustomList4SobiPro is a display module built specifically for SobiPro and is the end-all of all SobiPro listings modules. Unlike modules built for a specific criterion, Custom-List 4SobiPro allows users to select any field or a combination of fields in a SobiPro listing to automatically generate a customised list. This can vary from latest items display, featured items display to a dynamic related items listing.

CustomList4SobiPro can work under two modes:

In Simple Rules mode, the module allows you to use any of the field combinations in an item to be used to define a filter criteria to display the items by. This is the most recommended mode and allows you to build a list such as latest items, featured items etc.

In Advanced Rules mode (meant for developers), you can define the entire function that CustomList4SobiPro uses to display the results. In simple terms, this is like Jumi inside a module. Developers can access global variables, session variables, perform SQL queries etc. to increase the flexibility of the extension.

Other features of CustomList4SobiPro include

- Custom templating through template files (pre-built with three template files). Display any field from the items.
- Custom javascript effects (pre-built with three java script effects).
- Sort by any field.
- Filter by Category or by entire Section.
- Ability to set up an ItemId for the links.

# Installing CustomList4SobiPro

LatestItems4SobiPro is installed as any other Joomla Module. Unzip the package before installation to extract the user manual and the installation package of the plugin.

# Configuring CustomList4SobiPro

Configuring LatestItems4SobiPro consists of configuring the following parameters of the module:

#### Module class suffix

CSS class suffix to be used by the module. Can be left empty.

#### Select latest from

Specifies whether the latest items should be selected from an entire Section or a single Category. In either case, the module automatically traverses to sub-category level to find all items.

#### Section

If 'Select latest from' is set to Section, the latest items from this section (and its categories and subcategories) are selected to be displayed. If 'Select latest from' is set to Category, then this setting is ignored.

### Category

If 'Select latest from' is set to Category, the latest items from this category (and its subcategories) are selected to be displayed. If 'Select latest from' is set to Section, then this setting is ignored.

### Items to display

Number of latest items to display.

# Order By

Defines the field which should be used to order items by.

# Order By Priority

Whether the items should be ordered in Ascending or Descending order of the 'Order By' field.

#### ItemId for Links

Allows the user to specify an ItemID for the items that are displayed by the module. If not specified, the links to the items will not have an ItemId.

# Rule Type

Defines the rule mode in which the system operates. If set to 'Simple' users can define a simple rule which invloves logical comparisons of the fields of the items. If set to 'Advanced' mode, users can define a full php function to define a filter criteria for the custom list.

#### Rule

Defines the rule to be used to display the items. To display all items, set to 1==1 or specify a field based rule as explained in Setting Up a Simple Rule.

# Display Type

If set to Static, renders the output based on the template file specified. If set to Dynamic, renders the output based on the template file specified and then animates based on the javascript effect file.

# Template File

Template file to be used for rendering. The files are stored at /modules/mod\_customlist4sobipro/tmpl/templates folder with the extension .tpl.

The extension is pre-built with three templates:

vcard.tpl:

Displays the fields field\_name, field\_description and field\_image.

• image\_only.tpl:

For each item, displays the image specified in the field\_image field.

• simple\_list.tpl:

Items are displayed as a simple list using the values specieid in field name.

# Javascript File

Javascript file to be used for effects. The files are stored at /modules/mod\_customlist4sobipro/tmpl/js folder. The module comes with two pre-built javascript effects:

Slideshow:

The displayed item is faded out and then the new item is faded in. The time duration between the slides can be configured by specifying {\interval':duration} in Javascript Params. The \interval' refers to milliseconds.

E:q:

{'interval':5000} refers to 5 second interval.

Scroller:

Items are displayed as a vertical or a horizontal scroller. The direction of scrolling and the scroll interval can be controlled via Javascript Params.

{'interval':5000} - Set the scrolling interval to 5 seconds. Vertical scrolling will be applied by default.

{'interval':10000,'dir':'horizontal'} - Set scrolling interval to 10 seconds, set scrolling direction to horizontal.

# Javascript Params

Javascript parameters to be passed to the javascript files. Parameters are specified in JSON format.

# Setting up a Template File

CustomList4SobiPro allows users to set up their own template files similar to vcards to control the display of the items better.

Each template file is stored in /modules/mod\_customlist-4sobipro/tmpl/templates folder with the extension .tpl. Each template file defines the layout of each item that is displayed.

The module starts with an entire list of filtered and sorted items and applies the .tpl file to each of the items. The entire layout of the .tpl file can be defined using html and php.

As a good practice, it is recommended to start and end each template with an enclosing div tag. Data for current item can be accessed via the \$item object.

Variable/Object Property	Meaning
\$item->id	ID of the SobiPro item.
\$item->validSince	Since when the item has start publishing.
\$item->validUntil	When the item stops pub- lishing
\$item- >updatedTime	When the item was last updated.
\$item->author	Owner ID of the item.
\$item- >authorname	Name of the Owner.
\$item->fieldname	Value of the fieldname. E.g.: \$item->field_de- scription is the value stored in description field.

### For Image Fields

\$item->imagefieldname returns an array of four URLs of the image:

- thumbnail
- icon
- original
- image

### These can be accessed in the following manner:

```
$img = $item->imagefieldname;
echo $img["thumb"]; //Thumbnail URL
echo $img["icon"]; //Icon URL
echo $img["image"]; //Image URL
echo $img["orignal"]; //Original Image URL
```

#### For URL Fields

\$item->urlfieldname returns an array with the index `url'. This can be accessed in the following manner:

```
$url = $item->urlfieldname;
echo $url["url"]; //URL
```

### For select, checkbox, radio button fields

Information of select, check box, radio button fields are stored in SobiPro in a manner similar to an array. Therefore each field needs to be called with array[0] to get the actual value.

#### E.g.:

```
$selected_option = $item->field_yesno;
echo $selected_option[0]; //will print the se-
lected option
```

\$selected\_options = \$item->fieldmultiselectoption;
echo implode(",",\$selected\_options); //will print
all the selected options comma separated.

# Setting up a Custom Template

In addition to the three pre-built templates, Custom List 4 SobiPro allows you to set up your own template to customize the display. The templates are applicable to Static view as well as Dynamic view.

Setting up a template is a simple matter of creating a .tpl file with the placeholders for the fields you want to display. The extension will apply the specified template to each and every entry that is being displayed inside the module. An example template is show below:

```
<div id="<?php echo $item->id; ?>" >
   <div class="left" style="width:50%; float:left">
        <div class="title" style="clear:both;">
             <?php echo $item->field name; ?>
        </div>
        <div class="sh des" style="clear:both;">
             <?php echo $item->field shortdes; ?>
        </div>
   </div>
   <divclass="right"style="width:50%; float:left">
        <div class="image">
             <ima
                     src="<?php echo</pre>
                                            $item-
>image["thumb"]; ?>" />
        </div>
   </div>
</div>
```

The above template will divide each entry in to two portions, left and right and will display the fields field\_name and field\_shortdes in the left portion. The right portion will display the image of the entry.

Data assigned to each entry can be accessed via the \$item object in the following manner:

Object Property	Description
\$item->id	ID assigned to the entry by SobiPro.
\$item->author	User ID of the author of the entry.
\$item->authorname	User <b>name</b> of the author of the entry.
\$item->validSince	Date from when the item is active/pub-lished.
\$item->validUntil	Date when the item is de-activated/unpub-lished.
\$item->updatedTime	Date the item was last updated.
\$item->fieldname	Value stored in the field fieldname. Can access any field using this method.

It is recommended to always wrap each item with a div tag. Clients can further use the variable id (not id) to access the ID of the module that displays this item. This would be useful in setting up individual stylings for each module.

All templates should be stored in modules/mod\_custom-list4sobipro/tmpl/templates directory.

# Setting up a Custom Javascript Effect

Custom List 4 Sobi Pro is pre-built with two javascript effects and allows you to extend the effects or add new javascript effects via .js files. Users can select the required javascript file in each module to activate the desired display effects.

The extension will call the javascript function with the name <code>customlist4sobipro\_filename(id, data, jsparams)in</code> the .js file with the following parameters in the same order:

- id: ID of the module. Used primarily to identify between multiple modules.
- data: All entries which should be displayed in this position with the template applied and passed as a JSON array.
- jsparams: Content of the JS Params set up in the module.

E.g. If a user creates a slideshow.js file. Then the user should create a customlist4sobipro\_slideshow(id, data, js-params) function in the slideshow.js file.

# Setting up a Simple Rule

Setting up a simple rule in CustomList4SobiPro allows users to perform any logical operator on any field of the item or any combination of the fields.

E.g.

- 1. [field\_featured] == "yes"
- 2. ([field\_featured]=="yes") AND ([field\_license]=="non-commercial")

In the two sample rules above, it is assumed that the field\_featured defines if an item is a featured item or not and field\_license defines if an item is a commercial one or not. The module configured with the first rule lists all the featured items. The second rule lists all the featured and non-commercial items.

# Specifying field names

Field names should be specified within square brackets "[]" and the field name should refer to the field alias. You can find the field alias by going to Edit Field option in SobiPro.



# **Logical Operators**

Rule	Meaning
AND	Logical AND
OR	Logical OR
NOT	Logical NOT

# **Comparison Operators**

Rule	Meaning
IS_EQUAL_TO(a,b)	Equal to
IS_GREATER_THAN_OR_ EQUAL_TO(a,b)	a is greater than or equal to b
IS_LESS_THAN_OR_ EQUAL_TO(a,b)	a is less than or equal to b
IS_ GREATER_THAN(a,b)	a is greater than b
IS_LESS_THAN(a,b)	a is less than b

# Setting up an Advanced Rule

An advanced rule provides users with more flexibility in filtering items that should be displayed. Effectively, an Advanced Rule allows users to define the filtering function which is used to filter in the items. Users who are familiar with MVC framework may also assign additional properties to the items from within this procedure so that they can overide the display template to display the additionally assigned items.

CustomItems4SobiPro employs a function evaluate\_custom() to define the user's filter criteria. In effect, the user defines the body of the function when defining a rule in the Advanced Mode.

(array) function custom\_evaluate(\$params, \$items)

**\$params:** A JParameter object that stores the module parameters. get('paramname') method can be used on this object to retrieve the value of the parameter.

**\$items:** An array of objects consisting of all items that belongs to the specified Section/Category and are current (i.e. the item has a valid since date that is either equal to or less than today and has a valid until date that is either equal to or greater than today).

Each object in the array consists of following properties:

**id:** ID of the item as stored in sobipro.

**title:** Title of the item based on what is defined to be the 'Title' field.

**image:** A URL to the representative image of the item based on what s defined to be the 'Image' field.

**description:** Description of the item based on what is defined to be the 'Description' field.

author: User ID of the user who submitted the item.

**data:** An array consisting of field alias as the key and data stored for the particular field. (i.e. array(['field\_name']=>"name of the item", ['field\_des']=>"description of the item"))

Users may use any of the above to define their own filtering algorithm. As the module allows to use any valid PHP code within the rule, users can easily extend this functionality. Please refer Sample Codes section for a wide range of samples of such filtering algorithms.

#### return:

Users **should** return the items array in the format specified above through their code. To compenstate for any possible problems in the user's code, the function returns a NULL by default. Therefore it is necessary to manually return the filtered array back to the main program through a simple return clause.

### Extending further

It is possible to further extend the functionality through the user rule by assigning additional data to each of the objects in \$items array or to the \$data array of an object. If this functionality is used, then the users should manually override the template to format the output to display additional details.

Please note that due to a limitation, template override works only in the static display mode.

# Setting up an ItemID

Every single module that is set up in Joomla needs to be assigned with a menu item that it needs to show up on. This mechanism causes problems with auto/dynamically generated links as by definition, they do not have an assigned menu item. It is possible to over come this problem by some third party extensions (such as Modules2Pages).

LatestItems4SobiPro overcomes this by allowing users to specify an itemid for the dynamically generated links. Though not necessary, it is advised that this itemid be the same as the itemid for existing SobiPro menus.

# Sample Configurations

#### Latest Items Module

CustomList4SobiPro can be configured as a Latest Items module by easily setting the following configurations:

• Number of Items: Number of items to display.

Order by field: validSinceSort Direction: Descending

Rule Type: Simple

• Rule: 1==1

#### Featured Items Module

To configure CustomList4SobiPro , each item in SobiPro needs to be added with a field to identify if a particular item is a featured item or not. In this example it is assumed that the field 'field\_featured' is a field which has two options 'Yes' and 'No'

• Number of Items: Number of items to display.

Order by field: validSinceSort Direction: Descending

• Rule Type: Simple

• Rule: [field\_featured]=="yes"

# Listings by Author

CustomList4SobiPro can be configured to display the SobiPro items by each author. As this could be a long list, it is suggested to configure the module as a slideshow.

Number of Items: Leave empty.Type of display: Rotate Items

• Order by field: Author

• Sort Direction: Ascending

• Rule Type: Simple

• Rule: 1==1

# Listings by Logged in User

One common feature of listings directories is the ability to view one's own listings when they are logged on. CustomList4SobiPro can be easily configured for this purpose:

• Number of Items: Leave empty

Order by field: validSinceSort Direction: Descending

Rule Type: SimpleRule: [author]==\$uid

#### Slideshow Module

Any of the above functionality can be displayed as a standard list or as a slideshow module (e.g. LatestItems Slideshow). Simply set the 'Type of Display' to 'Rotate Items'

# Configuring language files

Geolok is built with multi-language support for administration view. The module is shipped with the default support for en-GB (English - UK). Users can create their own language files to support their native languages.

Language files for Geolok module is uploaded to 'language' folder of the main site under the sub-folder for the relevant language (e.g. 'language/en-GB'). The file name of the language file for the module is en-GB.plg\_geolok.ini.

# kBanners4VM

kBanners4VM is the ultimate banner management solution for Virtuemart that allows users to set up HTML and image banners for your Virtuemart eshop.

kBanners4VM allows you to set up banner groups and assign them to Virtuemart categories allowing you to target your advertiestments to the correct audience.

# Modules2Pages

Modules2Pages is the ultimate module management solution for Joomla that allows users to assign any module to any page.

Ideal for websites that use multipage components like Virtuemart, JomSocial, Commmunity Builder and SOBI2.

# kBanners4Joomla

kBanners4Joomla is the end-all of all banner components for Joomla. Providing the same flexible interface of kBanners4VM and kBanners4SOBI, kBanners4Joomla allows users to target their banners based on conditional URL parameters.