

PluginUI

The Wiki Plugin UI allows end users to insert and edit wiki plugins without having to deal with the syntax details. To be supported, the plugin must define the `wikiplugin_wikiplugin_*_info` function (ex: `wikiplugin_annotation_info`). The function is a replacement to the `_help` function previously used. Instead of returning a string with documentation, it returns an array of information about the plug-in.

When the function is available, the plugin help description is also auto-generated.

The info function also provides preferences to be validated. The `prefs` property is an array of preferences that need to have the 'y' value. Failure on any of the preferences will cause the plugin to be disabled. A unique preference must be created for each plugin (in addition to feature checks). The preference must be created in `lib/setup/prefs.php`

The plugin will be manageable, along with all the other plugins, from the administration section. Go to "Plugins" tab under "Text area" administration page (`tiki-admin.php?page=textarea`).

Annotation sample

```
<?php

function wikiplugin_annotation_info() {
    return array(
        'name' => tra('Image Annotation'),
        // link to plugin documentation page on doc.tiki.org (in this case
        http://doc.tiki.org/PluginAnnotation)
        'documentation' => 'PluginAnnotation',
        'description' => tra("Displays an image and enables users to define zones to
highlight information on the image."),
        // list of preferences that need to be enabled to use the plugin
        'prefs' => array('wikiplugin_annotation'),
        'body' => tra('Autogenerated content. Leave blank initially.'),
        'filter' => 'striptags',
        'params' => array(
            'src' => array(
                'required' => true,
                'name' => tra('Location'),
                'description' => tra('Absolute URL to the image or relative
path from tiki root.'),
                'filter' => 'url',
            ),
            'width' => array(
                'required' => true,
                'name' => tra('Width'),
                'description' => tra('Image width.'),
                'filter' => 'digits',
            ),
            'height' => array(
                'required' => true,
                'name' => tra('Height'),
                'description' => tra('Image height.'),
                'filter' => 'digits',
            )
        )
    );
}
```

```

        ),
        'align' => array(
            // if true, this plugin field is mandatory
            'required' => false,
            // field name
            'name' => tra('Alignment'),
            'description' => tra('Image alignment'),
            'filter' => 'alpha',
            // if advanced equals true, this field is hidden by default
            // and only visible inside "Advanced options" section
            'advanced' => true,
            // Field is a select box with the following options instead of
            // a text field
            'options' => array(
                array('text' => tra('Left'), 'value' => 'left'),
                array('text' => tra('Right'), 'value' => 'right'),
                array('text' => tra('Center'), 'value' => 'center'),
            ),
        ),
    ),
);
}

```

Plugin UI development

While the current plugin UI allows defining easily some plugins, some more advanced plugins could use new features.

For example, the FILE plugin is as of Tiki 5 complicated, having 7 parameters: name, desc, page, showdesc, image, fileld and date. If used for Wiki attachments, only the first 5 are used, while if used for files from the galleries, only the last 2 parameters are used. This plugin would ideally have 2 radio buttons to choose between a file from the galleries and an attachment. If it's from the galleries, only the first 5 parameters should show, *name* being required, otherwise only the last 2, *fileld* being required.

Desired features:

- Parameters with radio buttons or checkboxes for boolean fields.
- Dynamic display of parameters depending on the value of other parameters.
- Parameters becoming required or not depending on the value of others parameters.

Below is a proposal of the interface to implement the new functionalities described above. Note that only properties relevant to this discussion were included.

Annotation sample

```

function wikiplugin_file_info()
{
    return array(
        'params' => array(
            /* This is a select box with two options 'Wiki page attachment' and

```

```

'File from file gallery'.
    * Note that other params set values of this one as their parent
meaning that they will only be
    * displayed if this value is selected (they are hidden by default) */
    'type' => array(
        'required' => true,
        'name' => tra('Type'),
        'options' => array(
            array('text' => tra('Wiki page attachment'), 'value'
=> 'attachment'),
            array('text' => tra('File from file gallery'), 'value'
=> 'gallery'),
        ),
    ),
/* Although this field is set as required, since it has a parent it
will only be
    * considered as required if the value from 'type' field is selected.
    * Otherwise the field is not even displayed and not considered as
required.
    * The 'parent' key is an array with two values. The 'name' value is
the
    * name of the parent parameter and 'value' is the value that must be
selected on
    * the parent param for this field to be displayed.*/
    'name' => array(
        'required' => true,
        'name' => tra('Name'),
        'description' => tra('Wiki attachment:') . ' ' . tra("Gives
the name of the attached file to link to"),
        'parent' => array('name' => 'type', 'value' => 'attachment'),
    ),
    'page' => array(
        'required' => false,
        'name' => tra('Page'),
        'parent' => array('name' => 'type', 'value' => 'attachment'),
    ),
    'showdesc' => array(
        'required' => false,
        'name' => tra('Show attachment description'),
        'parent' => array('name' => 'type', 'value' => 'attachment'),
        // New field type bool that automatically generates a checkbox
        'type' => 'bool',
    ),
    'fileId' => array(
        'required' => true,
        'name' => tra('File identifier'),
        'filter' => 'digits',
        'parent' => array('name' => 'type', 'value' => 'gallery'),
    ),
)
);
}

```

Related
[Plugin Alias](#)