Custom fields in Joomla 3.7 for developers

Allon Moritz
JDD 17 September 2016
About Me

Allon Moritz

@digitpeak / @laoneo

Founder Digital Peak GmbH

Doing Joomla extensions since 2007
Agenda

● Information
● Concept
● End user perspective
● Developer perspective
● Demo
● How to integrate
● Hidden features
● Fields plugin group
● Questions
Information

- DPFIELDS is the base of com_fields
  https://joomla.digital-peak.com/products/dpfields
- Com_fields development repository (CLOSED)
  https://github.com/joomla-projects/custom-fields
- Final version can be found in PR 11833 in the 3.7 branch
Concept

- It is a component and a system plugin
- It integrates in the background through Joomla events
- It extends JForm
- It uses JFormFields
- It is MVC coded
- It works like com_categories, it is not a standalone component
- It uses JLayouts
Concept

It fills the gap between the XML declaration in administrator/components/com_content/models/forms/article.xml and JForm.

```
<field name="title" type="text" label="JGLOBAL_TITLE"
      description="JFIELD_TITLE_DESC"
      class="input-xxlarge input-large-text"
      size="40"
      required="true" />

<field name="alias" type="text" label="JFIELD_ALIAS_LABEL"
      description="JFIELD_ALIAS_DESC"
      hint="JFIELD_ALIAS_PLACEHOLDER"
      size="40" />
```
End user perspective

Menu item sidebar
End user perspective

Creating a field

[Image of a Joomladay Germany 2016 slide showing a section on creating a field in Joomla, with highlighted text indicating the title of the field as "Demo Text Field"]

https://joomla.digital-peak.com
End user perspective

Edit article (item of your component)
End user perspective

Front end view

Joomla Custom Fields

Home
Demo Article

Details
- Written by Super User
- Category: Uncategorised
- Published: 10 March 2016
- Hits: 1

Demo Text Field: Demo field value

Demo description of the article.

You are here: Home > Uncategorised > Demo Article

https://joomla.digital-peak.com
Developer perspective

Menu item sidebar

Sidebar entries are added in your helper class through JHtmlSidebar

```php
JHtmlSidebar::addEntry(
    JText::_('JGLOBAL_FIELDS'),
    'index.php?option=com_fields&context=com_content.article',
    $vName == 'fields.article'
);
JHtmlSidebar::addEntry(
    JText::_('JGLOBAL_FIELD_CATEGORIES'),
    'index.php?option=com_categories&extension=com_content.article.fields',
    $vName == 'categories.article'
);
```

`administrator/components/com_content/helpers/content.php`
Developer perspective

Creating a field

Editing a field is done in com_fields like any other ordinary J3 component

administrator/components/com_fields

https://joomla.digital-peak.com
Developer perspective

Edit article (item of your component)

When an item is loaded, through the onContentPrepareForm event the fields are added to the form

```php
$node = $parent->appendChild(new DOMElement('field'));
$node->setAttribute('name', $field->alias);
$node->setAttribute('type', $field->type);
$node->setAttribute('default', $field->default_value);
$node->setAttribute('label', $field->label);
$node->setAttribute('description', $field->description);
$node->setAttribute('class', $field->class);
$node->setAttribute('required', $field->required ? 'true' : 'false');
$node->setAttribute('readonly', $field->params->get('readonly', 0) ? 'true' : 'false');
```

// Loading the XML fields string into the form
$form->load($xml->saveXML());

plugins/system/fields/fields.php
Developer perspective

Edit article (item of your component) 2

When an item is saved, through the onContentAfterSave event the fields are stored

```php
// Setting the value for the field and the item
$model->setFieldValue($field->id, $context, $id, $fields[$field->alias]);
```

`plugins/system/fields/fields.php`
Developer perspective

Front end view

When an item is rendered, through the Plugin/Events/Content events the fields are displayed on the front

```php
public function onContentAfterTitle($context, $item, $params, $limitstart = 0) {
    return $this->display($context, $item, $params, 1);
}
```

```php
public function onContentBeforeDisplay($context, $item, $params, $limitstart = 0) {
    return $this->display($context, $item, $params, 2);
}
```

```php
public function onContentAfterDisplay($context, $item, $params, $limitstart = 0) {
    return $this->display($context, $item, $params, 3);
}
```

`plugins/system/fields/fields.php`
Developer perspective recap

- Plugin events to integrate into other components
- Com_fields component to manage the fields
- Layouts to prepare the value and render the field
Demo

https://joomla.digital-peak.com
How to integrate

- It’s all about the context
  - Form name is the context
  - Context save event
  - Context render event
- Load the params fieldsets, all of them
- Fields are added to the item as fields array
- Basic search needs a join, look at the plg_search_content plugin. Smart search is automatically supported.
How to integrate: My own fields

- Field needs to implement JFormDomfieldinterface and to extend JFormField
- Field parameters (eg. thumbnail width) as form in ..../parameters/foo.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<form>
  <fieldset name="fieldparams">
    <field
      name="multiple"
      type="radio"
      class="btn-group btn-group-yesno"
      default="g"
      label="JLIB_FORM_FIELD_PARAM_LIST_MULTI_LABEL"
      description="JLIB_FORM_FIELD_PARAM_LIST_MULTI_DESC"
   >
      <option value="1">YES</option>
      <option value="0">NO</option>
    </field>
  </fieldset>
</form>
```

- If the output of the field for the front needs some special functionality, create a layout in components/com.foo/layouts/field/prepare/foo.php
Hidden features

Rendering: Prepare the value

The value of a field is prepared through the following steps:

1. **Component layout** → **com_fields layout** (empty)
2. **com_fields layout** → **Component base layout** (empty)
3. **Component base layout** → **com_fields layout** (empty)

```
JLayoutHelper::render(
    'field.prepare.foo', 'com_fields', ...)
```

```
JLayoutHelper::render(
    'field.prepare.base', 'com_fields', ...)
```

```
JLayoutHelper::render(
    'field.prepare.foo', 'com_foo', ...)
```

```
JLayoutHelper::render(
    'field.prepare.base', 'com_foo', ...)
```
Hidden features

Rendering: Prepare the output

The output for the display events is prepared through a couple of steps:

- **Renders all fields**
  - Component layout
  - com_fields layout
  - `JLayoutHelper::render('fields.render', 'com_fields', ..)`

- **Renders a single field**
  - Component layout
  - Empty
  - For every field
  - `JLayoutHelper::render('field.render', 'com_fields', ..)`

Details
- Written by Super User
- Category: Uncategorised
- Published: 13 September 2016
- Hits: 1

Demo Text Field: Demo Text Value
Demo List Field: Demo List Value 2
Hidden features

Multiple contexts

If your component has multiple contexts, for example Contacts has contacts and E-Mail form fields, add a filter form file to administrator/com_foo/models/forms/filter_fields.xml.

```xml
<?xml version="1.0" encoding="utf-8"?>
<form>
  <fields name="custom">
    <fieldset name="custom">
      <field name="section" type="section" default="com_contact">
        <option value="com_contact.contact">COM_CONTACT</option>
        <option value="com_contact.mail">COM_CONTACT_MAIL</option>
      </field>
    </fieldset>
  </fields>
</form>
```
Hidden features

Public helper classes and API

The class FieldsHelper has some public API functions to work with fields.

```php
JLoader::register('FieldsHelper', JPATH_ADMINISTRATOR . '/components/com_fields/helpers/fields.php');
$fields = FieldsHelper::getFields('com_foo.bar', $item, true);
```

The field model itself allows to get and store the value of a field.

```php
JModelLegacy::addIncludePath(JPATH_ADMINISTRATOR . '/components/com_fields/models', 'FieldsModel');
$fieldModel = JModelLegacy::getInstance('Field', 'FieldsModel', array('ignore_request' => true));
$fieldModel->setValue($fieldId, 'com_foo.bar', $item->id, 'demo value');
```
Hidden features

ACL support

Every field has an access level

Every field has a new permission edit.value

JFactory::getUser()-&gt;authorise('edit.value', 'com_foo.bar.field.' . (int) $field-&gt;id);
Fields plugin group

- New plugin group fields
- Every plugin must have a folder fields with the JFormFields
- Layouts should be placed in the layouts folder field/prepare
- Actually there are no plugin events, configuration is done by convention
Final notes

- Photos are taken from https://unsplash.com
- The way to a CCK is not far anymore
- Hello world demo contains a DPFields branch, can be used for com_fields https://github.com/Digital-Peak/Joomla-3.2-Hello-World-Component
Questions?

Slides on https://joomla-digital-peak.com/jd16de
Thank you and enjoy the day :-)