

# Custom fields in Joomla

## ~~3.6~~ 3.7

**Allon Moritz**

JAB 21 May 2016



**DIGITALPEAK**

# About Me

Allon Moritz

@digitpeak / @laoneo

Founder Digital Peak GmbH

Doing Joomla extensions since 2007

Member Architecture Team Joomla X

Developer by passion



# Agenda

- Information
- Concept
- End user perspective
- Developer perspective
- Demo
- How to integrate
- Hidden features
- Final notes
- Questions



# Information

- DPFields  
<https://joomla.digital-peak.com/products/dpfields>
- Com\_fields Repository  
<https://github.com/joomla-projects/custom-fields>
- We need Testers/Coders/Code Stylers and people who can write documentation



# Concept

It's not a CCK



# Concept

It's not a standalone component



<https://joomla.digital-peak.com>

**DIGITAL**PEAK

# Concept

What is it then?



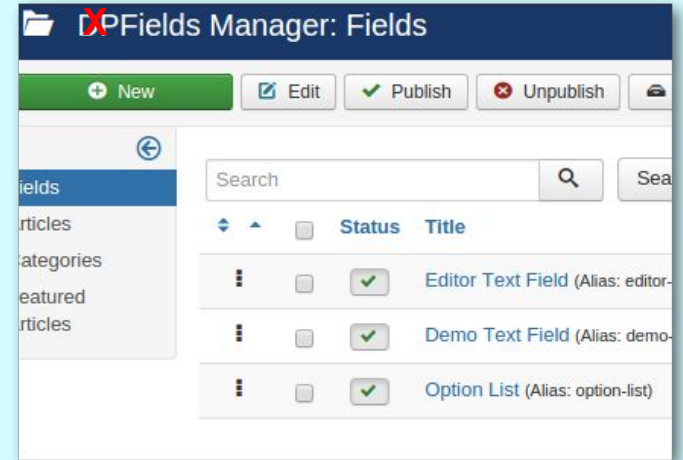
<https://joomla.digital-peak.com>

**DIGITAL**PEAK

# Concept

It

- is a component and a system plugin
- integrates in the background through Joomla events
- extends JForm
- uses JFormFields
- is MVC coded
- works like com\_categories
- uses JLayouts





# Concept

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

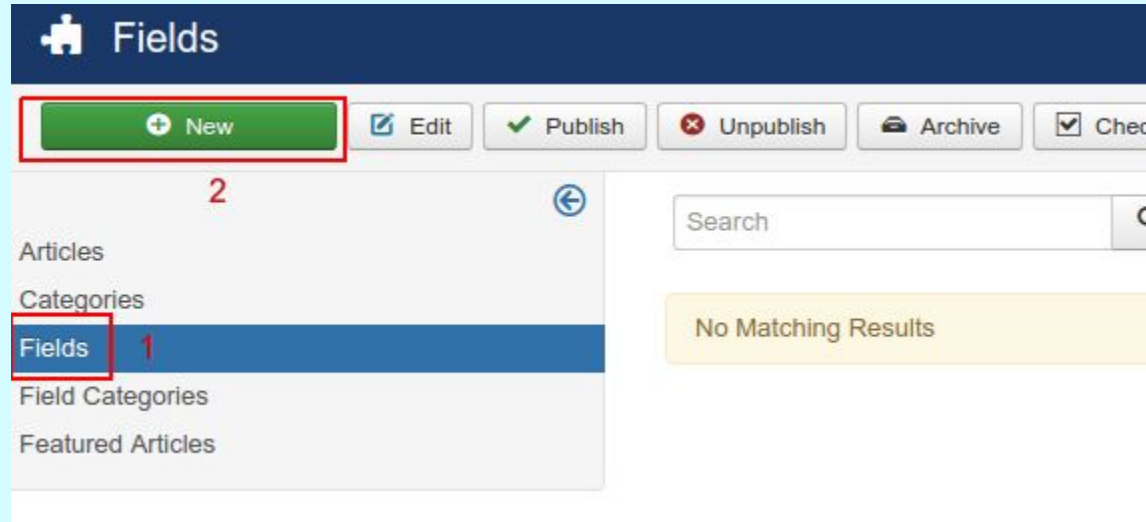
```
<field name="title" type="text" label="JGLOBAL_TITLE"
description="JFIELD_TITLE_DESC"
class="input-xxlarge input-large-text"
size="40"
required="true" />

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



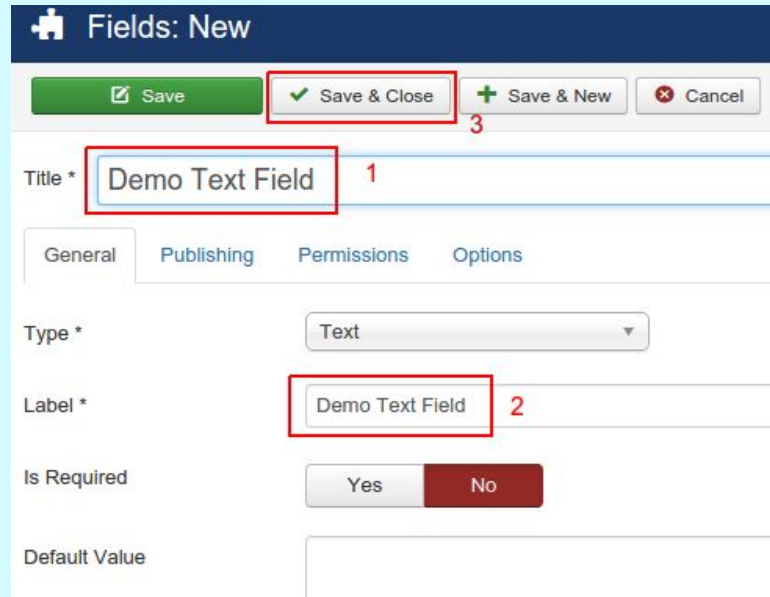
# End user perspective

## Menu item sidebar



# End user perspective

## Creating a field



The screenshot shows the Joomla! 'Fields: New' form. The title 'Demo Text Field' is entered in the 'Title \*' field, which is highlighted with a red box and a red '1'. The 'Save & Close' button is highlighted with a red box and a red '3'. The 'Label \*' field contains 'Demo Text Field', highlighted with a red box and a red '2'. The 'Type' dropdown is set to 'Text'. The 'Is Required' section has 'No' selected. The 'Default Value' field is empty. The form has tabs for 'General', 'Publishing', 'Permissions', and 'Options'.



# End user perspective

## Edit article (item of your component)

The screenshot shows the Joomla! 'Articles: New' form. At the top, there is a dark blue header with a pencil icon and the text 'Articles: New'. Below the header, there is a row of buttons: 'Save' (green), 'Save & Close' (highlighted with a red box and labeled '3'), 'Save & New' (green with a plus icon), and 'Cancel' (grey with a red X icon). The 'Title \*' field contains the text 'Demo Article'. Below the title, there are four tabs: 'Content', 'Publishing', 'Images and links', and 'Options' (labeled '1'). The 'Fields' tab is highlighted with a red box. Below the tabs, there is a 'Demo Text Field' label and a text input field containing 'Demo field value' (labeled '2').



# End user perspective

## Front end view



# Developer perspective

## Menu item sidebar

Sidebar entries are added in your helper class through JHtmlSidebar

```
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');
```

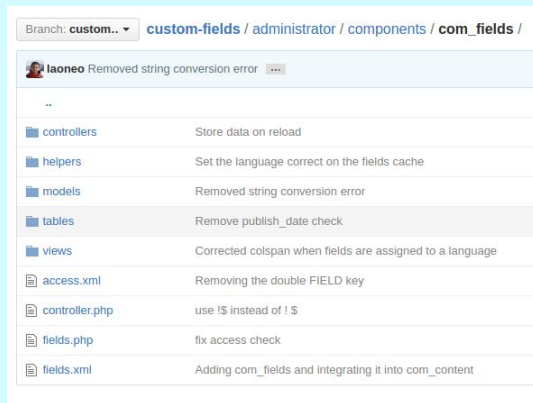
*administrator/components/com\_content/helpers/content.php*



# Developer perspective

## Creating a field

Editing a field is done in com\_fields like any other ordinary J3 component



The screenshot shows the Joomla! administrator interface for the 'com\_fields' component. At the top, the breadcrumb trail is 'Branch: custom... / custom-fields / administrator / components / com\_fields /'. Below this, a commit log entry is shown for user 'laoneo' with the message 'Removed string conversion error'. The main part of the screenshot is a table listing the changes made in this commit.

File	Description
controllers	Store data on reload
helpers	Set the language correct on the fields cache
models	Removed string conversion error
tables	Remove publish_date check
views	Corrected colspan when fields are assigned to a language
access.xml	Removing the double FIELD key
controller.php	use !\$ instead of ! \$
fields.php	fix access check
fields.xml	Adding com_fields and integrating it into com_content

*administrator/components/com\_fields*



# Developer perspective

## Edit article (item of your component)

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

```
$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

```
// 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

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

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

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

*plugins/system/fields/fields.php*



# Time for another picture



# 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>

**DIGITAL**PEAK

# How to integrate

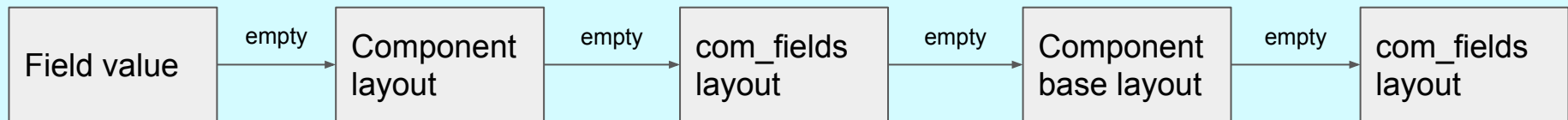
- It's all about the context
  - Form name is the context
  - Context save event
  - Context render event
- Load the params fieldsets, all of them
- Fields are added to the item as fields array
- Basic search needs a join, look at the plg\_search\_content plugin. Smart search is automatically supported.
- New field types in administrator/components/com\_foo/models/types
  - class FieldTypeFoo extends FieldTypeBase
  - forms/foo.xml



# Hidden features

## Rendering: Prepare the value

The value of a field is prepared through the following steps



`$field->params->get('output')`

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

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

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

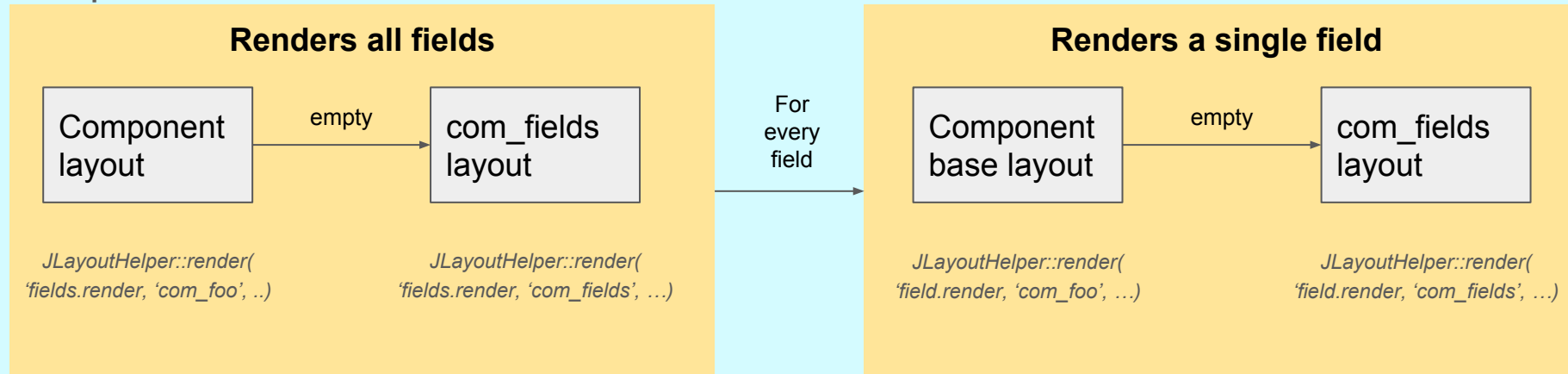
`JLayoutHelper::render(  
'field.prepare.base', 'com_fields', ...)`



# Hidden features

## Rendering: Prepare the output

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





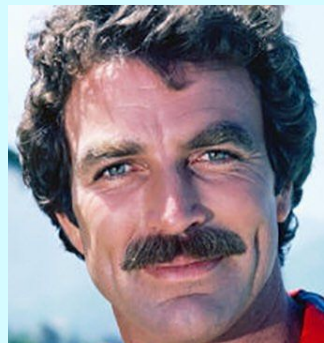
# Hidden features

## Rendering: Prepare trough the content event

Com\_fields allows us to use the fields in the description trough onContentPrepare by using the simple mustache syntax from <http://mustache.github.io>.

```
{{#fields alias=foo}}  
  <b>{{label}}</b>: <p>{{title}}</p>  
{{/fields}}
```

*We have an editor button from DPFields, if needed.*



# Hidden features

## Adding own fields

Every component can add it's own fields in the folder  
administrator/components/com\_foo/models

- fields/foo.php class *JFormFieldFoo* extends *JFormField*
- types/foo.php class *FieldsTypeFoo* extends *FieldsTypeBase*
- types/forms/foo.xml



# Hidden features

## Adding own fields

The screenshot displays a code editor with three panels. The left panel shows the project structure of 'DPCalendar' in a file explorer. The middle panel shows the 'dpcalendar.php' file, which contains PHP code for a custom field type. The right panel shows the 'dpcalendar.xml' file, which contains the XML metadata for the field type.

**PHP Explorer:**

- DPCalendar [DPCalendar master]
  - com\_dpcalendar
    - admin
      - cli
      - controllers
      - helpers
      - language
      - libraries
      - models
        - fields
          - dpcalendar.php
        - forms
      - types
        - forms
          - dpcalendar.xml

**dpcalendar.php:**

```

1 <?php
2 /**
3  * @package      DPCalendar
4  * @author        Digital Peak http://www.digital-peak.com
5  * @copyright      Copyright (C) 2012 - 2015 Digital Peak. All
6  * @license        http://www.gnu.org/licenses/gpl.html GNU/GPL
7  */
8 defined('_JEXEC') or die();
9
10 JLoader::import('components.com_fields.models.types.base',
11
12 class DPFieldsTypeDpcalendar extends FieldsTypeBase
13 {
14
15     public function prepareValueForDisplay ($value, $field)
16     {
17         if (! is_array($value))
18         {
19             $value = array(
20                 $value
21             );
22         }
23
24     JLoader::import('components.com_dpcalendar.helpers.
  
```

**dpcalendar.xml:**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <form>
3     <fields name="fieldparams" label="COM_DPFI
4         <fieldset name="fieldparams">
5             <field name="multiple" type="radio"
6                 label="COM_DPFI
7                 <option value="1">JYES</option>
8                 <option value="0">JNO</option>
9             </field>
10         </fieldset>
11     </fields>
12 </form>
13
  
```

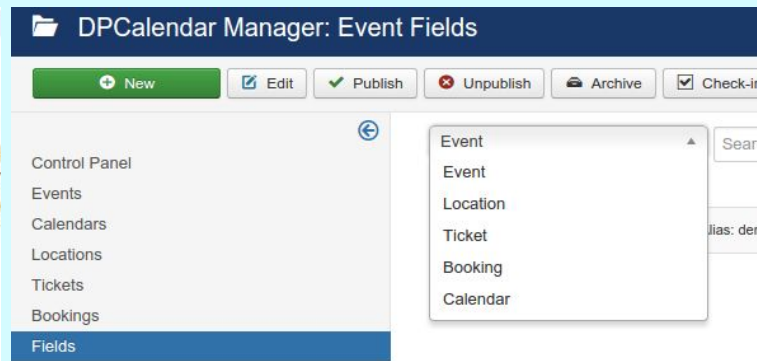


# Hidden features

## Multiple contexts

If your component has multiple contexts, for example DPCalendar has events, locations, tickets..., add a filter form file to *com\_foo/models/forms/filter\_fields.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<form>
  <fields name="custom">
    <fieldset name="custom">
      <field name="section" type="section" default="com_dpcalendar.event">
        <option value="com_dpcalendar.event">COM_DPCALENDAR_FIELDS_SECTIO
        <option value="com_dpcalendar.location">COM_DPCALENDAR_FIELDS_SEC
        <option value="com_dpcalendar.ticket">COM_DPCALENDAR_FIELDS_SECTI
        <option value="com_dpcalendar.booking">COM_DPCALENDAR_FIELDS_SECT
      </field>
    </fieldset>
  </fields>
</form>
```



# Hidden features

## Public helper classes and API

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

```
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.

```
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()->authorise('edit.value', 'com_foo.bar.field.' . (int) $field->id);
```



## Final notes

- Photos are taken from <https://unsplash.com>
- Hello world demo contains a DPFields branch, can be used for com\_fields  
<https://github.com/Digital-Peak/Joomla-3.2-Hello-World-Component>
- Joomla X will have custom fields on steroids
- Help testing and coding on the custom fields project <https://github.com/joomla-projects/custom-fields>



# Questions?





Thank you and enjoy the day :-)

