

---+ Package TWiki::Form::FieldDefinition

Base class of all field definition classes.

Type-specific classes are derived from this class to define specific per-type behaviours. This class also provides default behaviours for when a specific type cannot be loaded.

## **ClassMethod new ( % . . . )**

Construct a new FieldDefinition. Parameters are passed in a hash. See Form.pm for how it is called. Subclasses should pass @\_ on to this class.

## **ObjectMethod finish ( )**

Break circular references.

## **isEditable() -> \$boolean**

Is the field type editable? Labels aren't, for example. Subclasses may need to redefine this.

## **isMultiValued() -> \$boolean**

Is the field type multi-valued (i.e. does it store multiple values)? Subclasses may need to redefine this.

## **isTextMergeable() -> \$boolean**

Is this field type mergeable using a conventional text merge?

## **isMandatory() -> \$boolean**

Is this field mandatory (required)?

## **renderForEdit( \$web, \$topic, \$value ) -> (\$col0html, \$col1html)**

\$web - the web containing the topic being edited \$topic - the topic being edited Render the field for editing. Returns two chunks of HTML; the \$col0html is appended to the HTML for the first column in the form table, and the \$col1html is used as the content of the second column.

## **cssClasses(@classes ) -> \$classes**

Construct a list of the CSS classes for the form field. Adds additional class specifiers related to the attributes of the field e.g mandatory. Pass it a list of the other classnames you want on the field.

## **getDefaultvalue() -> \$value**

Try and get a sensible default value for the field from the values stored in the form definition. The result should be a value string.

Some subclasses may not support the definition of defaults in the form definition. In that case this method should return `undef`.

## **renderHidden(\$meta) -> \$html**

Render the form in `$meta` as a set of hidden fields.

## **populateMetaDataFromQuery( \$query, \$meta, \$old ) -> (\$bValid, \$bPresent)**

Given a CGI `$query`, a `$meta` object, and an array of `$old` field entries, then populate the `$meta` with a row for this field definition, taking the content from the query if it's there, otherwise from `$old` or failing that, from the default defined for the type. Refuses to update mandatory fields that have an empty value.

Return `$bValid` true if the value in `$meta` was updated (either from the query or from a default in the form).  
Return `$bPresent` true if a value was present in the query (even it was `undef`)

## **ObjectMethod renderForDisplay ( \$format , \$attrs ) -> \$html**

Render the field for display, under the control of `$attrs`.

The following vars in `$format` are expanded: `$title` - title of the form field `$value` - expanded to the **protected** value of the form field

The value is protected by `TWiki::Render::protectFormFieldValue`.

---

This topic: [TWiki > TWikiFormFieldDefinitionDotPm](#)

Topic revision: [r1](#) - 2008-01-22 - [TWikiContributor](#)



Copyright © 1999-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)

**Note:** Please contribute updates to this topic on [TWiki.org](#) at `TWiki:TWiki.TWikiFormFieldDefinitionDotPm`.