



CORDIS

Content Manual

CORDISCMS2 Version 4.2.0

Contents

Contents.....	2
Tables/Figures	2
1 Introduction	3
2 Editing.....	3
2.1 Textarea based Editing	3
2.2 WYSIWYG Editing.....	4
2.2.1 <i>Kupu</i>	4
2.2.2 <i>FCKeditor</i>	5
2.3 External Editing	6
3 Content	6
3.1 Content Modifications by the WYSIWYG Editors.....	6
3.1.1 <i>WYSIWYG editing and HTML</i>	6
3.1.2 <i>Content Modifications by kupu</i>	6
3.1.3 <i>Content Modifications by FCKeditor</i>	6

Tables/Figures

1 Introduction

This manual describes the aspects of editing HTML documents with the CORDISCMS2.

2 Editing

CORDISCMS2 is supposed to support the editing with the browser.

2.1 Textarea based Editing

HTML documents can be edited in a textarea of the browser on the HTML source level. The WYSIWYG editors have a button in the toolbar to switch them to HTML mode. If the document contains HTML markup errors, only the textarea is available for editing.

Note!

Currently, if a document contains HTML errors, WYSIWYG editing cannot be enabled even if the HTML errors have been removed. This will be changed in the next release.

The following image shows a screenshot of textarea based editing within CORDISCMS2:

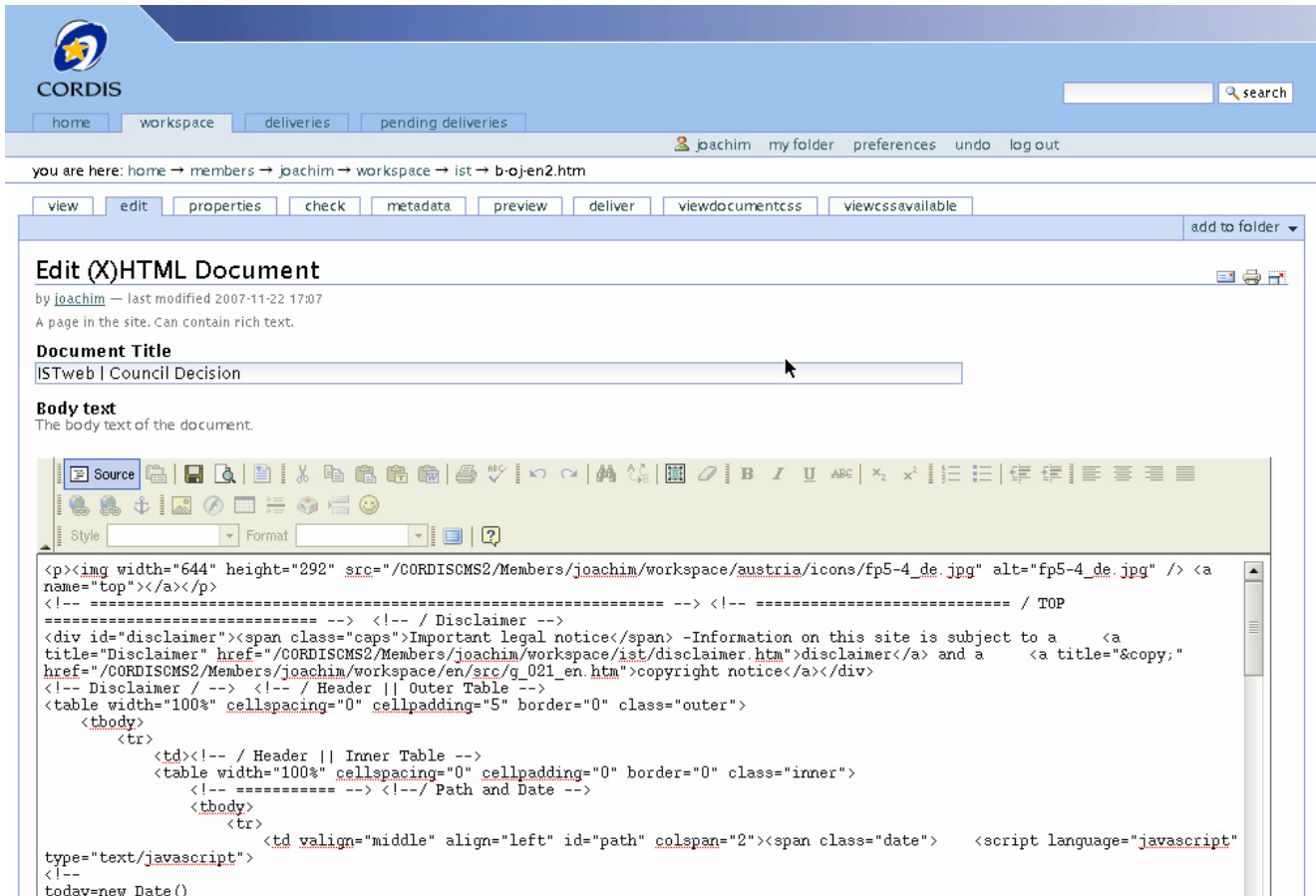


Figure 1: Editing a document in a textarea

Note!

The screenshot shows the textarea mode of FCKeditor. If kupu is selected as the preferred editor or if the document was uploaded as a pure text object, the textarea may look slightly different.

2.2 WYSIWYG Editing

The CMS2 supports two WYSIWYG editors for editing HTML documents, [kupu](#), which is the standard WYSIWYG editor of [Plone](#), and [FCKeditor](#), which is an open source javascript based WYSIWYG editor that is used in a number of different applications and provides a number of additional useful features.

Each of the editors has its strengths and weaknesses.

The user can set his preferred WYSIWYG editor in his user preferences. The selected editor will be used for WYSIWYG editing. It's currently not possible to make the editor selection on a per document basis but such a feature will be added in one of the next releases.

2.2.1 Kupu

[kupu](#) is the WYSIWYG editor that's included with [Plone](#). It deeply integrated into [Plone](#) and can be customized by an administrator to a certain degree. It's relatively small on features and toolbar buttons but sufficient for most simple editing cases.

The following screenshot shows kupu in action.

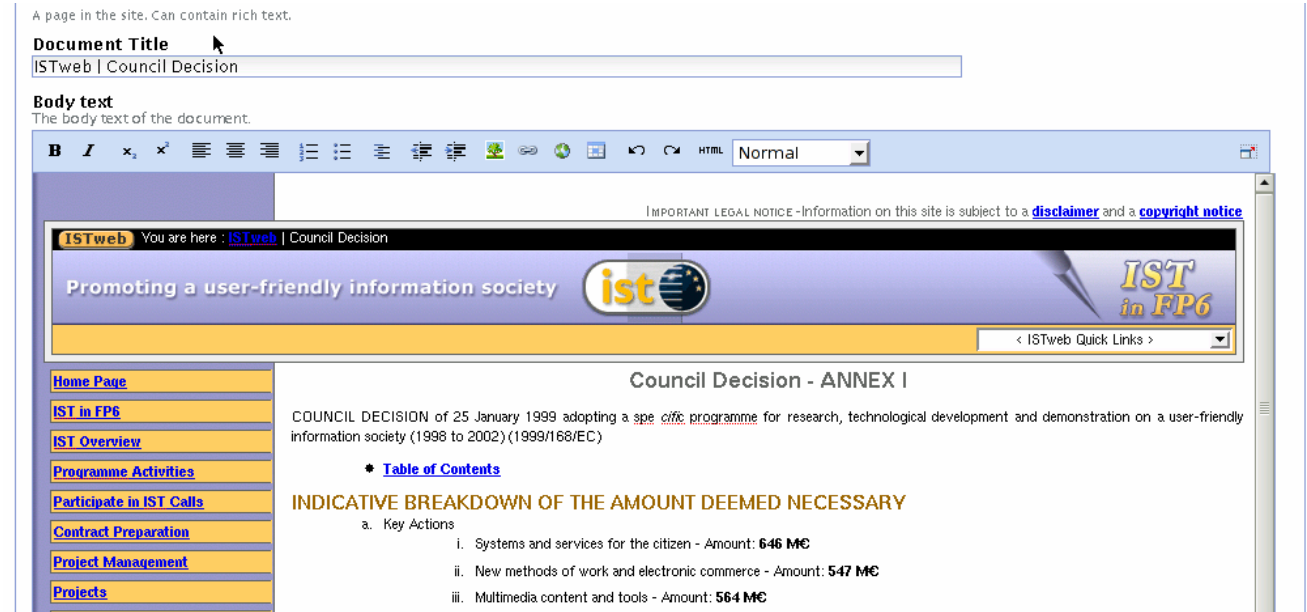


Figure 2: Editing a document with kupu

2.2.2 FCKeditor

FCKeditor is a WYSIWYG editor with a lot of features and a toolbar that looks similar to Word's toolbar. It has been integrated into a number of different products like Adobe ColdFusion or Oracle Application Express. In addition to HTML it even supports PHP or CFM markup, but this has not yet been tested in conjunctions with CORDISCMS2.

The following screenshot shows FCKeditor inside CMS2:

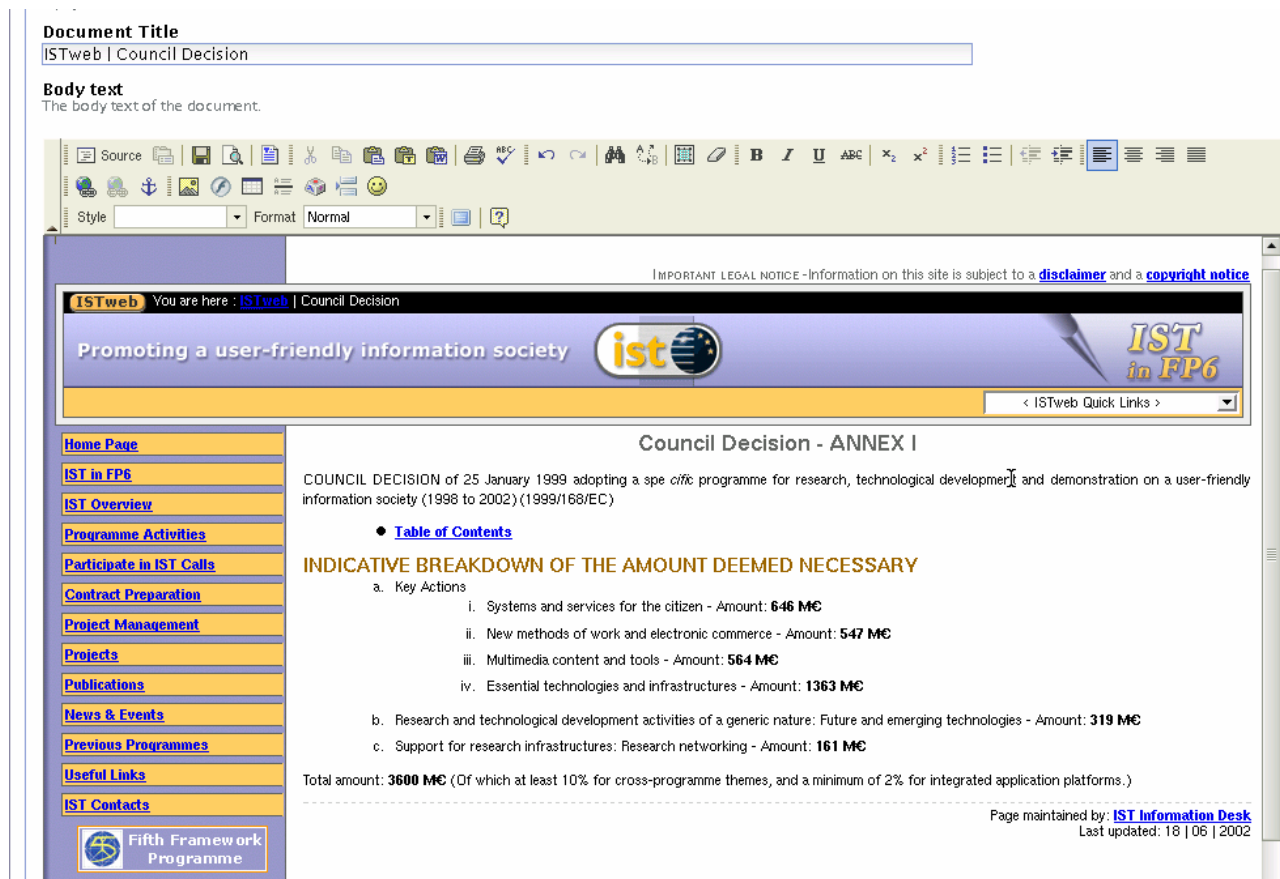


Figure 3: editing a document with FCKeditor

2.3 External Editing

Web pages can also be used with special HTML editors like Dreamweaver or GoLive. Editing with external editors will be described in one of the next releases.

3 Content

This chapter contains information concerning the content in general, e.g. what a user has to consider when working with HTML documents in the context of CORDISCMS2.

3.1 Content Modifications by the WYSIWYG Editors

As described earlier, CORDISCMS2 supports two WYSIWYG editors which work inside the browsers. As such, interactive elements for a web page may interfere with the editing component.

This chapter is work in progress and will be enhanced as experience with both editors evolves.

3.1.1 WYSIWYG editing and HTML

WYSIWYG editors in [Plone](#) are implemented in Javascript and use a so called RichTextEdit control of the browser. As such, the user sees an HTML page that he can edit. However mouse movements and keyboard presses are intercept and sent to a controlling instance, typically a parent frame of the WYSIWYG frame which can check if the input is valid or update buttons and dropdown boxes of the controlling toolbar.

In order to work correctly, interactive elements from the page being edited must be deactivated, e.g. the typical event handler attributes like onclick, onkeydown, onkeyup etc.

3.1.2 Content Modifications by kupu

Kupu simply removes all all interactive element attributes, so they are lost.

3.1.3 Content Modifications by FCKeditor

FCKeditor keeps the HTML in its original form in a textarea that is not shown normally. When editing in WYSIWYG mode, the HTML is copied to the WYSIWYG frame of the editor and dangerous HTML attributes are prefixed by a unique string which is removed again, when the HTML is copied back to the (normally hidden) textarea or when the document is saved.

By this method, no attributes are being lost.