tinyeditor 1.0

the manual

TinyEditor

o Content

1 How to install tinyeditor?

Copy the content of „xoopseditor" folder to the root/class/xoopseditor of your Xoops installation or if folder xoopseditor does not exist, copy the complete folder „xoopseditor" to root/class. A good idea would be to use the xoopseditor package and copy over the files from class/xoopseditor/tinyeditor.

Copy the folder tinyeditor to root/modules, install tinyeditor like any other xoops module and generate the templates for tinyeditor inside the XOOPS templates manager.

The following directories (if available) need special CHMODs

> CHMOD 777
> - tinyeditor / editor / plugins / ibrowser / temp /
> - tinyeditor / editor / plugins / ibrowser / scripts / phpThumb / cache /
> - tinyeditor / editor / plugins / imanager / temp /
> - tinyeditor / editor / plugins / imanager / scripts / phpThumb / cache /
>
> CHMOD 444
> - tinyeditor / editor / plugins / ibrowser / config / config.inc.php
> - tinyeditor / editor / plugins / ibrowser / scripts / phpThumb / phpThumb.config.php
> - tinyeditor / editor / plugins / imanager / config / config.inc.php
> - tinyeditor / editor / plugins / imanager / scripts / phpThumb / phpThumb.config.php
> - tinyeditor / editor / plugins / spellchecker / config.php
> - tinyeditor / include / config.inc.php
> - tinyeditor / include / ftpconfig.php
> - tinyeditor / include / ftpconfigchmod.php
> - tinyeditor / include / ftpconfigplugin.php

2 Setting up tinyeditor

First visit the preferences of tinyeditor. There are some settings to be made

Editor's language: Define your language here. This option can be set to any of the language codes according to iso639-1 standard (http://www.loc.gov/standards/iso639-2/englangn.html). For example: en for English, de for German or fr for French.

Editor's css: Give the path to the CSS-file that should be used for the contents inside the editor area. If left blank, normally the css of the current theme is used. Please **DO** make your own CSS-file for the editor which holds the most important styles for your site. Not all CSS-instructions in your default-theme are needed for formatting your articles.

Force BR elements: This option forces output of BR elements instead of P elements in Internet Explorer.

Force P elements: If this option is enabled Mozilla/Firefox will generate P elements on Enter/Return key and BR elements on Shift+Enter/Return. This option is enabled by default.

Force relative URLs: Enable this option to to convert absolute URLs to relative URLs. There is only one big bug, but tinyeditor is not to blame for this. IF block content edited in WYSIWYG appears on the homepage and is NOT in range of the associated module, relative URLs will fail. E.g.: You display a news block on your homepage, but news is not your start module, information for image/file-URLS isn't valid anymore.

Remove script host: If this option is enabled the host and parts from URLs are removed if they point within the same host as the script. This option is on by default if the „Force relative URLs" option is set to false.

Toolbar location: This option allows you to set the location of the toolbar (top or bottom).

Element path location: Set here the location of the elements path (top, bottom or none). You'll only see the resize-handle for the editor, when the elements path location is visible.

Date format and Time format: Set here the format for the Insert Date/Time plugin according to the following replacement variables:

> %y - year as a decimal number without a century (range 00 to 99)
> %Y - year as a decimal number including the century
> %d - day of the month as a decimal number (range 01 to 31)
> %m - month as a decimal number (range 01 to 12)
> %D - same as %m/%d/%y
> %r - time in a.m. and p.m. notation
> %H - hour as a decimal number using a 24-hour clock (range 00 to 23)
> %I - hour as a decimal number using a 12-hour clock (range 01 to 12)
> %M - minute as a decimal number (range 00-59)
> %S - second as a decimal number (range 00-59)
> %p - either 'am' or 'pm' according to the given time value
> %% - a literal '%' character

Uploads directory: Set the directory for images upload. Note: this directory must be manually created and if on a Unix system, chmoded to 777. Make sure that the upload folder already exists on the server. The upload directory will be used for the plugins ibrowser, imanager and xrmanager.

Enable per-user directories: If enabled, each user manages his own uploads folder preventing disk space cluttering. The folders will be named like user_userid.

Gzip-Compression: The editor will work up to 70% percent faster. The whole javascript-commandset will be saved into cache-folder of XOOPS. One minor setback: If you enable plugins and add buttons to the editor after the cache-file is already written you have to delete the editor's zip-file from the cache-folder. This cache-zip-file will be written anew. Every group will have its own cache-file. tinyeditor and gzip-compression seems to fail on some (namely GDATA Internet Security) antivirus scanners. In this case better turn off gzip-compression instead of browsing unsafe.

Subdirectory: This option has to be set when your XOOPS is working in a subdirectory. Ex.: Instead of http://localhost/ pointing to your homepage, http://localhost/wotever/ is the URL to

your homepage. Fill in the name of your subdirectory starting with a slash. There is no need for an ending slash.

Which library for thumbs creation? Set the graphic library for thumbnails creation. Use GD if your PHP install has safe_mode=On. safe_mode information will be displayed if you visit the administration index of tinyeditor.

Path to the library Set here the server path to Image magick or NetPBM library. If you use GD, leave the field as is.

Thumbnails width (px): Here you can define the max. width for thumbnails in xrmanager.

Thumbnails height (px): Here you can define the max. height for thumbnails in xrmanager.

Prefix for thumbnail files: Optional setting. You don't really need to change this. Do NOT change anything when your server is running safe_mode = on. The dot before the file name means, that the file will be invisible when you browse the directories.

Directory to store thumbnail files: Optional setting. You don't really need to change this. When safe_mode is on the thumbnail will be saved into the same directory as the image. The dot before the directory name makes it invisible when you browse the directories.

Default thumbnail The default thumbnail if the thumbnails can not be created, either due to error or bad image file. The image has to reside somewhere in the tinyeditor structure.

Image Editor temporary filename prefix Optional setting. You don't really need to change this. The dot before the filename makes it invisble.

Thumbnails columns Enter the desired number of columns for thumbnails display.

Organize upload space by: Select a criteria to prevent uploads space messing.

Directories for HTML- and CSS-files Type in where the HTML- and CSS-files are located you would like to edit with tinyeditor (templates-plugin must be installed). E.g.: modules/content/html - you can type in more than one directory, but they must be separated by a blank. No smarty-support in this version. For saving back the files the target directory needs to be writable.

Use FTP? You can use PHP-FTP commands instead of the normal file operation commands from FTP. If your PHP is configured with safe_mode = on you cannot use all the features of the xrmanager plugin. Another nice side effect is that you are not bound to upload limits anymore.

Override group toolbars? Sounds tricky, but is in fact very simple. If users are in different

groups, without this setting only the first group id will be taken for toolbars. With this you can set a priority. Or as a nice side effect you can set a bunch of groups to use a toolbar of one specific group. You have to write like this: 2|3|4|5|**2** which means that the groups with the group ids from 2 to 5 will use the toolbar of group 2. Always the last value will determine the final group!! Be careful that a group id appears only once in this list. Separate with a blank to make more combinations.

Unpack ZIPs uploaded with xrmanager? This will unpack ZIPs in the selected directory. Subdirectories inside the ZIP will be preserved. **WARNING:** There is at the moment no **RELIABLE** method to detect what is inside the ZIP, so select carefully who is allowed to upload ZIP (this can be done inside the mimetypes management of tinyeditor).

Now you finished the preferences and you can on to toolsets! There you can define the toolbars for every group individually.

3 Setting up the toolbars

First select the group you want to make the toolbars for. Please note, that only for the first THREE groups toolbars are predefined. For other groups the toolbar have to be submitted first, the revisit this group to create the toolbars! See also above for overriding group ids.

Toolbar rows: You have three rows for your tools. Just click the icons to place the tools into the textarea. The order of the textarea is the order where the tools finally appear in the editor. Do not worry about the blanks that are inside the textarea, these blanks will be trimmed nicely on submission.

Max. image width: This is the maximum size for the width of images. This only applies for the plugins iBrowser and iManager. If you don't want to use these plugins just leave the value there.

Max. image height: This is the maximum size for the height of images. This only applies for the plugins iBrowser and iManager. If you don't want to use these plugins just leave the value there.

Disk quota: Here you can define how much disk space a user can use for images. This only applies for the plugins iBrowser, iManager and xrmanager. If you don't want to use these plugins just leave the value there. If you are now think that you want to have such restriction, just wait until we visited the permissions-page for tinyeditor.

Valid Elements, Extended Valid Elements and Invalid Elements: With these three fields you are able to define which html-tags are allowed to use inside the editor. The following example is taken from the original tinyMCE-manual which you will find http://www.tinymce.com. There is a complete explanation on how to set up valid elements, extended elements and invalid elements. Valid elements: This example string tells the editor to remove all elements that are not a „a, strong, div or br" element, convert b elements to strong elements, default target to „_blank" and keep the href, target and align attributes of the elements.

```
a[href|target=_blank],strong/b,div[align],br
```

Put your definition into their textareas and save it to the database. If you don't want to invest more time on this, just leave the fields as they are – a basic instruction set will then be used. If you experience that some html-tags are empty in your html-code, come back here and fill in the missing elements. For your convenience there is a pulldown to insert the standard instruction set and a reduced instruction set for XOOPS DHTML editor look-a-like.

Once you defined the toolbars for all your groups, go to permissions!

4 Set the permissions

The small icons will show you which permission is meant for which module. If there is no icon it is a module wide permission.

Allowed to upload files: Here you can define which group is allowed to upload images.

Allowed to delete files: Here you can define which group is allowed to delete images.

WYSIWYG: Here you can define which group is allowed to use the WYSIWYG-feature. If unchecked the users will only get the standard DHTML-textarea. **This setting does not apply for the webmaster-group, because they will always have WYSIWYG!!**

Allowed to create folders: Here you can define which group is allowed to create directories.

Use quota: Here you can define which group has only limited disk space for image upload (how much disk space was defined in toolsets). **Please note that the webmaster-group is NOT automatically included.** If you want the webmaster-group to have also limited disk space you have to check there as well.

Override user-dirs for images: If the user has only limited disk space he can only navigate within his directories. With this permission you are able to override this so this group is allowed to see **ALL** upload-directories. This permission makes only sense if the use quota permission is used.

Change image attributes: Define which group is allowed to change image attributes.

Rename images: Here you can define which group is allowed to rename the images. **Please note that this might lead to unexpected results if images are renamed when already used in written articles.**

Allowed to delete non-empty folders: Here you can define which group is allowed to delete complete directory structures although being not empty.

Rename directories: Here you can define which groups are allowed to rename directories. Please note that this might lead to unexpected results if directory names are already used.

Copy directories: Here you can define which groups are allowed to copy directories. Content in the directories will also be copied.

Move directories: Here you can define which groups are allowed to move directories. Content in the directories will also be moved.

Copy file: Here you can define which groups are allowed to copy single files.

Move file: Here you can define which groups are allowed to move single files.

Edit file: Here you can define which groups are allowed to edit files. At the moment only images can be edited.

Allow to use xrmanager: Here you can define which groups are allowed to use the xrmanager. This is a special permission because this is bundled with the php ftp feature.

Use FTP Debug? This permission is only for groups which have admin right for tinyeditor to detect possible problems. Be WARNED that paths, username and password will be displayed. So be sure that in the webmaster group are only trusted people when FTP Debug is on.

One last hint: If system PHP commands like copy are disabled on your server you will not see all of the above permissions. You can get them back if you use FTP with the module (if you trust my progamming abilities).

Now you are ready to work with tinyeditor WHEN the modules you want to use tinyeditor with are ready for tinyeditor. So here are some instructions on how to implement the editor:

5 Make your modules tinyeditor-ready

If you're using an „old" version of Xoops (< 2.2) you can still use tinyeditor with your modules.

In this case only a few changes to some of your module's files are required. Following is a sample of integration with an old News module:

In your favourite text editor open the file

```
XOOPS_ROOT_PATH/modules/news/include/storyform.inc.php
```

about line 35 add the following line

```
include_once XOOPS_ROOT_PATH .
    „/class/xoopseditor/tinyeditor/formtinyeditortextarea.php";
```

Make sure this is the last file to be included.

Then move on and replace the lines which render the textareas with

```
$sform->addElement(new XoopsFormTinyeditorTextArea
    (array('caption'=> _NW_THESCOOP, 'name'=>'hometext', 'value'=>$hometext,
    'width'=>'100%', 'height'=>'400px'),true));
```

for the Scoop text and

```
$sform->addElement(new XoopsFormTinyeditorTextArea
    (array('caption'=> _AM_EXTEXT, 'name'=>'bodytext', 'value'=>$bodytext,
    'width'=>'100%', 'height'=>'400px', 'xEditor'=>'1'),false));
```

for the Extended text. That's all!

Here is a short description of how to make „new" modules work with tinyeditor that were designed for the xoopseditor framework.

We take the new news-module as example. Koivi was the first editor used for the xoopseditor framework. So lets search for occurrances of koivi to mimic this to make tinyeditor work.

Open xoops_version.php and look for this array (may vary):

```
array(_MI_NEWS_FORM_DHTML=>'dhtml', _MI_NEWS_FORM_COMPACT=>'textarea',
    _MI_NEWS_FORM_SPAW=>'spaw', _MI_NEWS_FORM_HTMLAREA=>'htmlarea',
    _MI_NEWS_FORM_KOIVI=>'koivi', _MI_NEWS_FORM_FCK=>'fck');
```

and change this to:

```
array(_MI_NEWS_FORM_DHTML=>'dhtml', _MI_NEWS_FORM_COMPACT=>'textarea',
    _MI_NEWS_FORM_SPAW=>'spaw', _MI_NEWS_FORM_HTMLAREA=>'htmlarea',
    _MI_NEWS_FORM_KOIVI=>'koivi', _MI_NEWS_FORM_FCK=>'fck',
    'tinyeditor'=>'tinyeditor');
```

Now open include/functions.php and look for:

```
case „koivi":
if(!$x22) {
if (is_readable(XOOPS_ROOT_PATH.„/class/wysiwyg/formwysiwygtextarea.php"))
{
include_once(XOOPS_ROOT_PATH . „/class/wysiwyg/formwysiwygtextarea.php");
$editor = new XoopsFormWysiwygTextArea($caption, $name, $value, '100%',
'400px', '');
}
} else {
$editor = new XoopsFormEditor($caption, „koivi", $editor_configs);
}
break;
```

After this insert these lines:

```
case „tinyeditor":
if ( is_readable(XOOPS_ROOT_PATH . „/class/xoopseditor/tinyeditor/
    formtinyeditortextarea.php")) {
include_once(XOOPS_ROOT_PATH . „/class/xoopseditor/tinyeditor/
    formtinyeditortextarea.php");
$editor = new XoopsFormTinyeditorTextArea(array('caption'=>$caption,
    'name'=>$name, 'value'=>$value, 'width'=>'100%', 'height'=>'400px',
    'xEditor'=>'1'));
}
break;
```

Update the news-module and you are done (hopefully). Now you have a good example on how to implement tinyeditor in other modules.

Making tinyeditor the default editor for XOOPS: If you're running Xoops 2.0.15 or higher you can edit the file /class/xoopsform/formdhtmltexteditor.php and change the line

```
var $htmlEditor = array();
```
to
```
var $htmlEditor = array(,XoopsFormTinyeditorTextArea', ,/class/xoopseditor/
tinyeditor/formtinyeditortextarea.php');
```

NEW: A new parameter has been added in the current version of tinyeditor (from 1.0 BETA onwards): xEditor. You can use this new parameter to imitate the XOOPS DHTML editor. When the value is set to 0 a reduced instruction and command set will be used. This could be useful if you want to turn other dhtml textareas into WYSIWYG.

6 Plugins

All available plugins (not basic commands) will explained roughly here.

Advanced horizontal rule: This plugin allows you to define some more values like width, height and shadow for the horizontal rule.

Advanced Image: This is an advanced version of the standard button image. You can set some extra options here. Please note that with this plugin uploads are not possible. To make use of this plugin the standard button needs to be placed too.

Advanced link: This is an advanced version of the standard button link. You can set many extra options here. As an extra I included the possibility of adding links from XOOPS modules. Currently links for the modules mylinks, newbb, news and wfdownloads are possible. You'll find the extra code for adding XOOPS links inside the plugin advlink in folder plugins. Maybe YOU want to enhance these php scripts for better use?

Emotions: This plugin queries the XOOPS database for smilies and inserts them into the editor.

Flash: Inserts Flash movies. Enter the URL, height and width of the movie and insert it.

Insert date: Inserts the current date formatted as defined in preferences.

Insert time: Insert the current time formatted as defined in preferences.

Preview: With this button you'll get a preview of the current content in a popup.

Print: With this button you can print the current content of the editor.

Search: Search throught the content of the editor for specified words.

Replace: Search throught the content of the editor for specified words and replace them.

Table: With this plugin you'll have all features to design tables. Once activated you get 11 buttons for designing or editing tables.

Zoom: With this plugin you can zoom the content of the editor up to 250% (IE only).

Spellchecker: With this plugin you can spellcheck the content of the editor (IE only). You need IESPELL (http://www.iespell.com/download.php) to be installed.

XOOPS-Quote: Highlight text in your editor and make it formatted like a XOOPS-Quote.

Block properties: With this plugin you can set some extra options for block-elements like headings and paragraphs. You have to click inside a block-element to edit its properties.

XOOPS-Code: Highlight text in your editor and make it formatted like a XOOPS-Code.

ML-Content: This plugin opens where you paste your text and select the language markup. This plugin is to be used with the multilingual hacks of XOOPS.

Paste Word: Copy your text from a Word-document to insert it into the editor. The Word-code will be cleaned. There are much more options for this plugin, have a look at the origdocs. There is a small difference between the behaviour of this plugin in IE and Gecko-browsers. Working with IE just copy the text from Word and click the button in the editor. In Gecko-browsers you have to insert the text to the popup, then click Insert to return it to the editor.

Paste Text: This plugin returns simple text to the editor. Existing formatting will be removed. You can preserve linebreaks if you want.

Select All: With this plugin you can select the complete content of the editor.

LTR / RTL: With these plugin you can change the writing direction of a paragraph.

Full Screen: This plugin makes the editor occupy the whole screen. Clicking again on the icon reverts the editor to normal size.

iManager: This is a full-featured image manager. You can upload files, edit them, add some special effects etc.pp. Although almost perfect, there some things that might cause problems with this plugin: a) In some cases there will be problems with the GD-library. This happens when there is no GD installed on the server or GD has not the right version number. These problems the plugin can not avoid. If you have problems with GD take the image manager plugin which is the XOOPS image manager wrapped as a plugin.

iBrowser: Nearly the same as iManager, but with less features. Same problems as the iManager-plugin.

Char counter: Counts the words and characters inside the editor area.

Insert Code: Opens a popup where you can paste source-code. Select the programming language. Once done your source code appears nicely formatted inside the editor using the geshi-syntax-highlighter. If you want to use this plugin you have to add some css-classes to the css-file of the editor.

```css
pre {
border-top: 1px solid #ddd;
border-left: 5px solid #ddd;
border-right: 1px solid #ddd;
border-bottom: 1px solid #ddd;
background-color: #eee;
padding: 2px;
margin: 0;
color: #000066;
}

.css .de1, .css .de2, .html4strict .de1, .html4strict .de2, .javascript
.de1, .javascript .de2, .mysql .de1, .mysql .de2, .php .de1, .php .de2,
.smarty .de1, .smarty .de2, .sql .de1, .sql .de2, .xml .de1, .xml .de2
{font-family: 'Courier New', Courier, monospace; font-weight:
normal;color: #000020;}
.css .imp, .html4strict .imp, .javascript .imp, .mysql .imp, .php .imp,
.smarty .imp, .sql .imp, .xml .imp
{font-weight: bold; color: red;}
.css .kw1, .html4strict .kw1, .javascript .kw1, .mysql .kw1, .php .kw1,
.smarty .kw1, .sql .kw1, .xml .kw1
{color: #b1b100;}
.css .kw2, .html4strict .kw2, .javascript .kw2, .mysql .kw2, .php .kw2,
.smarty .kw2, .sql .kw2, .xml .kw2
{color: #000000; font-weight: bold;}
.css .kw3, .html4strict .kw3, .javascript .kw3, .mysql .kw3, .php .kw3,
.smarty .kw3, .sql .kw3, .xml .kw3
{color: #000066;}
.css .kw4, .html4strict .kw4, .javascript .kw4, .mysql .kw4, .php .kw4,
.smarty .kw4, .sql .kw4, .xml .kw4
{color: #f63333;}
.css .co1, .css .co2, .css .coMULTI, .html4strict .co1, .html4strict .co2,
.html4strict .coMULTI, .javascript .co1, .javascript .co2, .javascript
.coMULTI, .mysql .co1, .mysql .co2, .mysql .coMULTI, .php .co1, .php .co2,
.php .coMULTI, .smarty .co1, .smarty .co2, .smarty .coMULTI, .sql .co1,
.sql .co2, .sql .coMULTI, .xml .co1, .xml .co2, .xml .coMULTI
{color: #808080; font-style: italic;}
.css .es0, .html4strict .es0, .javascript .es0, .mysql .es0, .php .es0,
.smarty .es0, .sql .es0, .xml .es0
{color: #000099; font-weight: bold;}
.css .br0, .html4strict .br0, .javascript .br0, .mysql .br0, .php .br0,
.smarty .br0, .sql .br0, .xml .br0
{color: #66cc66;}
.css .st0, .html4strict .st0, .javascript .st0, .mysql .st0, .php .st0,
.smarty .st0, .sql .st0, .xml .st0
```

```
{color: #ff0000;}
.css .nu0, .html4strict .nu0, .javascript .nu0, .mysql .nu0, .php .nu0,
.smarty .nu0, .sql .nu0, .xml .nu0
{color: #cc66cc;}
.php .me1 {color: #006600;}
.php .me2 {color: #006600;}
.php .re0 {color: #0000ff;}

pre ol{margin-left: 0;padding-left: 30px;}

pre .head {font-family: Verdana, Arial, sans-serif;color: #333333;font-
weight: bold;background-color: #f0f0ff;border-bottom: 1px solid
#d0d0d0;padding: 2px;font-size: 12px;}
pre li, pre li {font-family: `Courier New`, Courier, monospace; color:
black; font-weight: normal; font-style: normal;font: normal normal 95%
`Courier New`, Courier, monospace; color: #003030;}
pre li.li2 {font-weight: bold;font-weight: bold; color: #006060;}
pre .foot {display: none;}
pre a:link {color: #000060;}
pre a:hover {background-color: #f0f000;}
```

Insert DIV: Makes a DIV from your selected text. This plugin comes with tons of features and options.

Custom Bullet: Use this plugin to make your own definition lists.

Non editable: Once activated you can select the style Non editable from the style selector to protect elements against editing.

XOOPS Image Manager: This is the well-known XOOPS image manager wrapped as a plugin. Meant for those who have problems with iBrowser or iManager.

Context Menu: Gives you some commands to be used in the editor when right-clicking in the editor area or on elements. Please note that although this plugin appears inside the textarea in toolsets, there is no icon in the toolbar of the editor.

xrmanager: This is a plugin which plugs into the plugins advlink, advimage and flash. Once activated you are able to browse to files or links on your server. If you browse images you can edit them.

Keyboard: This is a Unicode Keyboard for various languages.

Equation: Returns some defined equations (square root, divide etc.) to the editor.

Layer: Turns marked text into div-layers. Less options than insertdiv.

Style: Add some more options to your your marked text.

Templates: With this plugin you upload/download/edit HTML- or CSS-files. The directories for upload and editing have to be defined in the module's preferences. **Please note**

that you can NOT edit files containing smarty-commands (this will possible in a later version - hopefully).

Spellchecker: This plugin is browser-independent. You can run this plugin in different modes: pspell or Google-services.

Nonbreaking: This plugin adds a -character.

Pagebreak: This plugin adds the tag [pagebreak] to the content, which is used by some modules to split content into different pages. When in WYSIWYG-mode a horizontal line will be displayed.

Media: This plugin handles embedded media such as QuickTime, Flash, ShockWave, RealPlayer and Windows Media Player.

¶ Visual chars: This plugin adds the possibility to see invisible characters like .

Xhtmlxtras: This plugin adds support for some XHTML elements these include cite, ins, del, abbr abd acronym.

Youtube: With this plugin you can add Youtube-movies. Just enter the id-code of the movie. This plugin might collide with the flash plugin.

Devkit: Use this plugin if you want to debug or develop tinyMCE plugins. Once activated a small iframe will be written to the top to every page. Click and an information window with various tabs will pop down.

7 FTP functionality

Right here some extra information about what changed since the last versions (beta, RC1). In my opinion tinyeditor was working quite good, but not with server setting safe_mode = on. This setting prevented a lot of actions like creating folders in other folders. So I had to add ftp functionality to the module.

If you feel uneasy using ftp for all groups just don't use. But if you want to use it, make a ftp access that only points to your upload directory and NOT to the root. I TRIED to make this safe as possible, but one never knows...

If ftp is not working as expected turn on ftp-Debug in permissions. This permission is only for the group which has admin rights to tinyeditor and shows every ftp transaction including usernames and passwords so take care that only trusted people are in the webmaster group when this debug mode is turned on.

As an extra menu in the administration you will have now FTP CHMOD. With this you can give CHMODs to every single file or directory that is within the range of your ftp access. Every change of **CHMOD to 777** will be inserted into the database so that you can hold track of your changes.

To make FTP work you have to edit a few files manually: include/ftpconfig.php for general ftp inside the plugins, include/ftpconfigchmod.php for chmodding files and directories and include/ftpconfigplugin.php for plugin- and language-file upload. You have to give 3 to 4 informations: a) the name of your ftp-server. b) an optional path if your ftp-access starts somewhere below your XOOPS installation. If there is no such optional path, leave a forwarding slash (/) there. c) the name of the ftp-user. d) the password for ftp-access. Save the files back and **write-protect them immediately!!**

A good advice before deciding and using FTP or not: To avoid „severe" problems, make your decision first about using FTP or not. In my opinion is not a good idea to use the editor without FTP in the first place and then after a while switching to FTP. If doing so you will get problems e.g. deleting files because the permissions on files and folders are different before and after use of FTP. Small example: Let's say you decided first to use tinyeditor without FTP and uploaded files, copied them, made directories and so on. Then you decided to use FTP, but this means that the files and directories which are already there have a different owner, so deleting will be denied.

8 Plugin Upload

This part of the administration can be used to upload new plugins or language files. If your server has safe_mode = on you have to activate FTP inside the preferences. If you are not using FTP you have to set tinyeditor/admin/images and tinyeditor/editor/plugins to CHMOD 777 to get this working.

Just browse the files on your local harddisk and upload them. Be sure that the file have names like plugin_nameoftheplugin.zip or de_langfile_nameoftheplugin.zip (de stands for the abbreviation of the language).

The zip-files will be uploaded, passing a 4-stages check, unzipped and copied to the correct directory. If this fails for any reason you have to manually upload the unzipped files. As I told in a previous section ZIPs cannot not be fully checked, but I tried my best.

9 Credits

I would like to thank all the people listed here in a non-specific order.

http://tinymce.moxiecode.com/ Moxiecode's crew for building such a great and useful tool

http://www.zhuo.org/htmlarea/docs/index.html Wei Zhuo for the advanced image manager/ editor

Phppp for fixing tinyeditor's backward compatibility with Xoops < 2.2 and many more hints

ralf57 for writing tinyeditor.

paketep for giving code and ideas.

irmtfan for bugging debuggin and ideas.

http://www.frxoops.org for being my greatest „fans" and providing me a lot of bugs and problems and great documentation and translation.

silver74 for giving inspiration to group overriding.

satanas for giving me webspace to upload imanager.

searche for donating me temporarily a domain for testing and developing debaser (sorry for misusing this place).

terrion from ultranetdomains.com for donating me webspace and being always helpful.

http://www.phpconcept.net for the great pclzip-script.

Ryan Demmer from which I borrowed code for xrmanager and his templates plugin which I modified a bit and the charcount plugin.

TOMO from Mambo for writing his excellent FTP class.

Andrew Tetlaw for his blockproperties plugin.

Nephesh Studios for their custombullet plugin.

CDL for the equation plugin.

mmjaeger for ibrowser and imanager plugins.

Maxime Lardenois for insertcode plugin.

kristofik for insertdiv plugin.

Misc. for keyboard plugin. See keyboard.html for detailed information.

Adrian Hope-Bailie for youtube-plugin.

McDonald for making my job and giving support for tinyeditor.

And all the others who contributed their plugins and help.