



# scribite! 2.1

## Administratoren-Dokumentation

scribite! ist ein Modul, welches die WYSIWYG-Editoren Xinha, TinyMCE, FCKeditor, openWYSIWYG oder nicEdit in Textarea-Felder integriert, um einen optisch ansprechenden Text gestalten zu können.

scribite! in der Version 2.1 ist ausschließlich unter Zikula 1.0 und höher lauffähig.

Diese Dokumentation ist für Administratoren einer Internetseite bestimmt. Modulautoren können in der Developer-Dokumentation nachlesen, wie scribite! direkt in Module integriert werden kann.

Autor: Sven Schomacker (hilope)

URL: <http://trac.zikula.org/projects/scribite>

Veröffentlicht unter der GPL (pndocs/license.txt)

## ***1. Leistungsbeschreibung***

scribite! fügt in folgenden Modulen den Textarea-Feldern WYSIWYG-Funktionalitäten hinzu:

1. About
2. Admin\_Messages
3. Book
4. ContentExpress
5. cotype
6. crpCalendar
7. element
8. eventia
9. FAQ
- 10.htmlpages
- 11.Mailer
- 12.mediashare
- 13.News
- 14.PagEd
- 15.Pages
- 16.pagesetter
- 17.PhotoGallery
- 18.pncommerce
- 19.pnForum
- 20.pnhelp
- 21.pnMessages
- 22.pnWebLog
- 23.Profile
- 24.PostCalendar
- 25.Reviews
- 26.ShoppingCart
- 27.tFAQ

## ***2. scribite! Aktualisierung***

Wenn scribite! in einer Vorgängerversion installiert war, kann der normale Updatevorgang für Zikula-Module durchgeführt werden, nachdem alle Dateien im Ordner modules/scribite mit den neuen Dateien überschrieben wurden.

Seit Version 1.2 liegen die Editordateien/Javaskripte im Unterordner /javascript/scribite\_editors und nicht mehr in /modules/scribite/pnincludes.

Im Administrationsmenü wird eine Warnung angezeigt, da die Popups aufgrund einer .htaccess-Einschränkung für den modules-Ordner nicht mehr laufen.

### ***3. scribite! Installation***

1. Alle Dateien müssen in den Ordner **modules/scribite** hochgeladen werden.
2. Die gewünschten Editoren müssen sich in den Ordner **javascript/scribite\_editors** hochgeladen werden.
3. Darauf achten, dass die Datei **config/plugins/function.scribite.php** existiert. Falls nicht kann die Datei aus dem Ordner **modules/scribite/pntemplates/plugins** in den oben genannten Ordner kopiert werden.
4. In der Moduladministration die Module neu generieren, das neu angezeigte Modul scribite! Initialisieren und danach aktivieren.
5. In der scribite!-Administration die gewünschten Einstellungen vornehmen.
6. Das scribite!-Plugin in die Templates einfügen, wobei darauf geachtet werden muss, dass scribite! NACH dem Additionalheader und möglichst als letzter Eintrag vor `</head>` eingetragen wird.

#### **1. Xanthia-Themes**

In allen Theme-Templates (master.htm, admin.htm etc.) muss das Plugin mit `<!--[scribite]-->` eingefügt werden.

Beispiel:

```
<!--[scribite]-->
</head>
```

#### **2. AutoTheme-Themes**

Hier muss die **theme.html** angepasst werden:

Beispiel:

```
<?php $scribite = pnModFunc('scribite', 'user', 'editorheader');
echo $scribite ?>
</head>
```

#### **3. (alte) Zikula/Postnuke-Themes**

Anpassen der **theme.php** wie angegeben.

Beispiel:

```
$scribite = pnModFunc('scribite', 'user', 'editorheader');
echo ($scribite);
```

7. Einige Module laufen nicht standardmäßig mit scribite! und bedürfen kleinerer Anpassungen:
  1. **PagEd, Pagesetter (v6.x), Postcalendar (v4.02 CVS), Book (v1.12), pnhelp (v.1.0.0.0)**  
Die erforderlichen Dateien liegen unter **pndocs/ext\_modules** bereit und müssen in den jeweiligen Modulordner kopiert werden (Backup der alten Dateien anlegen!)
  2. **pncommerce**  
In der Datei **pntemplates/pncommerce\_admin\_itemedit.html** nach folgendem Eintrag suchen:  

```
<textarea name='ItemDescription' rows='10' cols='40' wrap><!--  
[ $Item.Desc ]-->
```

  
Und wie folgt bearbeiten:  

```
<textarea id='ItemDescription' name='ItemDescription' rows='10'  
cols='40' wrap><!--[ $Item.Desc ]-->
```
  3. **CoType 1.0**  
Die erforderlichen Dateien liegen unter **pndocs/ext\_modules** bereit und müssen in den jeweiligen Modulordner kopiert werden (Backup der alten Dateien anlegen!)
8. Leeren der Cache-Ordner sowie der kompilierten Templates.
  - pnTemp/Xanthia\_compiled
  - pnTemp/Xanthia\_cache
  - pnTemp/pnRender\_compiled
  - pnTemp/pnRender\_cache
9. Fertig.

## 4. Neue Module hinzufügen

Falls sich das gewünschte Modul nicht in der Liste findet, kann es zu scribite! hinzugefügt werden.

Dafür werden folgende Informationen benötigt:

- Modulname (nicht Aliasname, sondern Verzeichnisname im modules-Ordner)
- Modulfunktionen, unter denen scribite! aktiviert wird  
Falls es sich um ein altes Modul ohne func-Parameter handelt oder alle Modulfunktionen benutzt werden sollen, kann der Parameter „all“ benutzt werden.
- Textarea-ID's, die vom Editor umgewandelt werden sollen  
Sofern alle Textarea-Bereich umgewandelt werden sollen, kann der Parameter „all“ benutzt werden.

## 5. Modul bearbeiten

Sofern sich Modulfunktionen ändern oder geändert wurden, können über den Editieren-Schalter die Modulfunktionen und die Textarea-ID's des betroffenen Moduls bearbeitet werden.

Hier gelten die gleichen Grundsätze wie bei „Neue Module hinzufügen“ („all“-Parameter). Für PagEd kann der Parameter „PagEd“ im ID's-Feld benutzt werden, damit nicht alle Felder umgewandelt werden.

## 6. Module löschen

Hier kann ein Modul komplett aus der scribite!-Verwaltung gelöscht werden.

## 7. Templates

Das Standardtemplate ist **scribite\_EDITOR\_editorheader.htm** und bietet die meisten Optionen. Für Modifikationen sollte diese Datei in den Ordner /themes/YOURTHEME/templates/modules/scribite/ kopiert und dort bearbeitet werden.

Wenn eine Datei **scribite\_EDITOR\_MODULENAME.htm** existiert, wird diese anstelle des Standardtemplates benutzt. So kann für verschiedene Module eine unterschiedliche Editorkonfiguration erstellt werden.

Modulautoren haben zusätzlich die Möglichkeit, scribite! direkt über die API in ihren Modulen aufzurufen (siehe Extradokumentation), wobei ein spezielles Template als Parameter angegeben werden kann.

## 8. Xinha - erweiterte Konfigurationen



Die Konfiguration des Editors kann in der Datei **pntemplates/scribite\_xinha\_editorheader.htm** vorgenommen werden. (Siehe auch Punkt 7. Templates)

Die Datei **pnconfig/xinha/editor.css** kann angepasst werden, um das Aussehen des Editors an das Theme anzupassen (zur Zeit nur unter IE lauffähig). Anpassungen der reduzierten und Mini-Menüleiste können in den Dateien **scribite\_xinha\_toolbar\_reduced.htm**, bzw. **scribite\_xinha\_toolbar\_mini.htm** vorgenommen werden.

Für weitere Informationen sei die Lektüre des Xinha-Wiki unter <http://xinha.webfactional.com/> empfohlen.

Die Pluginbeschreibungen können unter <http://xinha.webfactional.com/wiki/Plugins> eingesehen werden.

Empfohlen ist auf jeden Fall die Aktivierung des GetHtml-Plugins.

Das Plugin „ExtendedFileManager“ ist leider zur Zeit noch nicht über die Administration konfigurierbar.

Bei Bedarf muss die Datei `/javascript/scribite_editors/xinha/plugins/ExtendedFileManager/config.inc.php` angepasst werden.

Bsp.:

```
...
$IMConfig['images_dir'] = $_SERVER['DOCUMENT_ROOT'].'images/xinha';
...
$IMConfig['images_url'] = $_SERVER['SITE_ROOT'].'images/xinha';
...
```

## 9. TinyMCE - erweiterte Konfigurationen



Die Konfiguration des Editors kann in der Datei **pntemplates/scribite\_tinymce\_editorheader.htm** vorgenommen werden. (Siehe auch Punkt 7. Templates)

Für weitere Informationen sei die Lektüre des TinyMCE-Wiki unter <http://tinymce.moxiecode.com/> empfohlen.

Die Pluginbeschreibungen können unter <http://wiki.moxiecode.com/index.php/TinyMCE:Plugins> eingesehen werden.

Falls TinyMCPUK genutzt werden soll, sollten folgende Punkte überprüft werden (danke an Erik Spaan für die Umsetzung):

- Einen Unterordner zum Hochladen der Bilder erstellen (Standard: /images/tinymcpuk)
- In diesem Ordner Unterordner für die verschiedenen Dateitypen erstellen (Image, Media, File und Flash).
- CHMOD 777 auf alle oben genannten Ordner anwenden.

Falls ein anderer Ordner als der oben genannte images/tinymcpuk zum Hochladen benutzt werden soll, müssen folgende Dateien auf den neuen Pfad angepasst werden:

- /javascript/scribite\_editors/tiny\_mce/imagemanager/config.inc.php:  
\$IMConfig['base\_dir'] = \$\_SERVER['DOCUMENT\_ROOT'] . '/images/tinymcpuk/Image';  
\$IMConfig['base\_url'] = 'http://' . \$\_SERVER['SERVER\_NAME'] .  
'/images/tinymcpuk/Image';
- /javascript/scribite\_editors/tiny\_mce/filemanager/connectors/php/config.php  
\$fckphp\_config['UserFilePath'] = "/images/tinymcpuk" ;



## ***10.FCKeditor - erweiterte Konfigurationen***



Die Konfiguration des Editors kann in der Datei **pntemplates/scribite\_fckeditor\_editorheader.htm** vorgenommen werden. (Siehe auch Punkt 7. Templates)

Für weitere Informationen sei die Lektüre des FCKeditor-Wiki unter <http://www.fckeditor.net/> empfohlen.

## ***11.OpenWYSIWYG - erweiterte Konfigurationen***



Die Konfiguration des Editors kann in der Datei **pntemplates/scribite\_openwysiwyg\_editorheader.htm** vorgenommen werden. (Siehe auch Punkt 7. Templates)

Für weitere Informationen sei die Lektüre des OpenWYSIWYG-Dokumentation unter <http://www.openwebware.com/products/openwysiwyg/docs/> empfohlen.

Der integrierte Imagemanager (nur im vollen Modus) benutzt den Ordner /images, damit allgemeine Bilder der Zikula-Installation genutzt werden können.

Falls ein Hochladen ermöglicht werden soll müssen folgende Punkte durchgeführt werden:

- Erstellen eines Ordners images/openwysiwyg und Anwenden von CHMOD 777 auf diesen Ordner
- Anpassen der Datei (nur, wenn ein anderer Ordner als images/ openwysiwyg benutzt werden soll)

/javascript/scribite\_editors/openwysiwyg/addons/imagelibrary/config.inc.php

```
$imagebasedir = '../..../images/openwysiwyg';
```

```
$imagebaseurl = 'images/openwysiwyg';
```

## ***12. NicEdit - erweiterte Konfigurationen***



Die Konfiguration des Editors kann in der Datei **pntemplates/scribite\_nicedit\_editorheader.htm** vorgenommen werden. (Siehe auch Punkt 7. Templates)

Für weitere Informationen <http://www.nicedit.com/> .

## ***13.Danksagungen***

- InteractiveTools.Com - they created htmlArea (<http://interactivetools.com/> ), xinha is based on htmlarea
- all programmers who made xinha possible (<http://xinha.webfictional.com/>) especially gogo
- Moxiecode Systems AB - the creators of tiny\_mce (<http://tinymce.moxiecode.com> )
- the programmers at <http://p4a.sourceforge.net/tinymcepuk> for their TinyMCPUK
- FCKeditor (<http://www.fckeditor.net> )
- openWYSIWYG (<http://www.openwebware.com>)
- nicEdit by Brian Kirchoff (<http://www.nicedit.com> )
- Sébastien Gruhier for Prototype Window Class (<http://prototype-window.xilinus.com>)
- the programmers of Snoopy Class
- Andreas Krapohl, who made the base module (<http://krapohl.info/>)
- Frank Schummertz, Steffen Voss, Carsten Volmer und Axel Guckelsberger for their help and inspiration (<http://pn-cms.net>)
- many thanks to all other pn users for their feature request, bug reports and hints making scribite! better
- translations by Daniel Neugebauer (por), Wilfried Santer (ita), tuerkyilmaz (tur), Arjen Tebbenhof (nld), Blaz Carli (slv), Dinmana (bos/hrv/srp), AnyKiller (spa), Yokav (fra), Jussuf (pol)
- everyone spending time in supporting and developing Zikula/Postnuke, xinha, TinyMCE, FCKeditor, openWYSIWYG and all the other fine scripts
- and special thanks to my wonderful girls Steffi and Charlotte ;)

## ***14. Changelog***

**See changelog.txt**