

# rte\_htmlarea in TYPO3 4.2.1

---

**Problem:** Installation und Grundkonfiguration des RTE hat sich in TYPO3 4.2 etwas geändert, so dass vorhandene Konfigurationen nicht mehr problemlos übernommen werden können.

## Grundlagen

**Installation:** Der rte\_htmlarea muss nicht mehr eigens installiert werden, da er seit Vs. 4.2 zum Core gehört und mit TYPO3 installiert und aktiviert wird. Im TER ist nur noch eine alte Version für frühere TYPO3-Versionen zu finden.

**Manual:** das Manual ist jetzt eine eigene Extension und so im TER zu finden: rtehtmlarea\_api\_manual (Link: [http://typo3.org/documentation/document-library/extension-manuals/rtehtmlarea\\_api\\_manual/1.7.4/view/](http://typo3.org/documentation/document-library/extension-manuals/rtehtmlarea_api_manual/1.7.4/view/))

## Demo-Installation

ich habe für Testzwecke eine Demo-Installation erstellt, die jeder benutzen kann:

<http://p102292.typo3server.info/typo3>

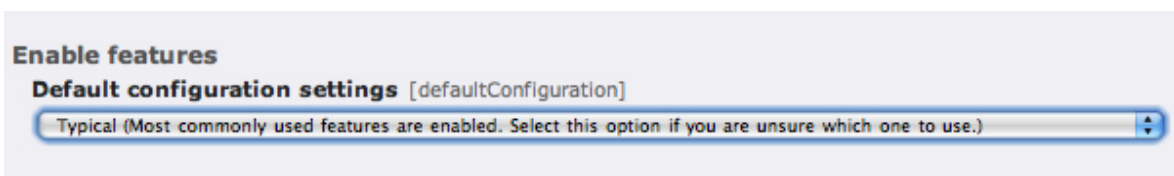
Login: test

Passwort: test

Ich stelle dort verschiedene Grundkonfigurationen vor; jeder ist eingeladen, sie zu testen und mir ggf. Fehler mitzuteilen. Bitte die vorhandene Konfiguration nicht zerstören, sondern für Tests eigene Unterseiten mit eigener Konfiguration anlegen.

## Grundkonfiguration

Der RTE verändert erheblich sein Verhalten abhängig von seiner Grundkonfiguration, die über den Extensionsmanager erfolgt.



Ganz entscheidend ist der Unterschied zwischen den Default configuration settings »Minimal«, »Typical« und »Demo«. Minimal ist eine Minimalkonfiguration, die nur grundlegende Werkzeuge erlaubt, während Demo alle Features des RTE freischaltet. Empfohlen wird »Typical«, eine Einstellung, die nur Werkzeuge freischaltet, die auch modernes HTML produzieren. Nicht enthalten sind hier z. B. alle Werkzeuge, die mit Hilfe des <font>-Tags arbeiten.

Ich gehe im Weiteren von der Einstellung »Typical« aus, die auch die Grundeinstellung bildet, mit der der RTE ausgeliefert wird.

**Enable additional inline elements** [enableInlineElements]

If set, the potential use of additional inline elements will be enabled in the default configuration of the RTE.



Mit dieser Funktion werden Bearbeitungsmöglichkeiten freigeschalten, die HTML-Inline-Tags verwenden, wie zum Beispiel `<code>`, `<cite>` und so weiter. Diese Option ist standardmäßig aus (somit stehen standardmäßig nur `<b>`, `<i>`, `<sup>` und `<sup>` als Inline-Tags zur Verfügung.

Ich möchte jedoch mehr einsetzen, deshalb schalte ich diese Option frei

**Enable links accessibility icons** [enableAccessibilityIcons]

If set, accessibility icons will be added in front of links.



Die Accessibility-Icons vor Links sind jetzt standardmäßig nicht aktiv, sie erscheinen also nicht mehr automatisch. Da ich diese Icons lieber über CSS regle, bleibt die Option aus.

**Enable the RTE in Opera 9.5+** [enableInOpera9]

If set, the RTE will be enabled when the browser is Opera 9.5+. This feature is still experimental!



Standardmäßig ist die Unterstützung für Opera nicht eingeschaltet, da ich es hier testen möchte, aktiviere ich diese Option. Opera-Anwender bitte testen :-)

**Enable Mozilla/Firefox extension** [enableMozillaExtension]

If set, enables the triggering of installation of a Mozilla/Firefox extension to allow the RTE to access the clipboard.



Mit dieser Option wird die Copy-Paste-Funktion auf der rechten Maustaste in Firefox freigeschalten. Diese Option ist standardmäßig aus – ich würde sie auch aus lassen, da das dazu erforderliche Plugin noch nicht in Firefox 3 funktioniert. Ausserdem funktioniert Copy/Paste natürlich über die Tasten Str-C und Str-V .

Wenn diese Option aktiviert wird, so sollte auch ein Pfad angegeben werden. unter dem die Extension zu finden ist:

[http://releases.mozilla.org/pub/mozilla.org/addons/852/allowclipboard\\_helper-0.5.5-fx+mz+zm.xpi](http://releases.mozilla.org/pub/mozilla.org/addons/852/allowclipboard_helper-0.5.5-fx+mz+zm.xpi)

**Others****Url of AllowClipboard Helper** [mozAllowClipboardURL]

Full absolute Url of the AllowClipboard Helper extension for Mozilla and Firefox.

[http://releases.mozilla.org/pub/mozilla.org/addons/852/allowclipboard\\_helper-0.5.5-fx](http://releases.mozilla.org/pub/mozilla.org/addons/852/allowclipboard_helper-0.5.5-fx)

## Bilder im RTE

Standardmäßig ist im RTE das Platzieren und der direkte Bildupload nicht aktiv. Um das Platzieren von Bildern im RTE zu erlauben, muss in der Grundkonfiguration diese Option aktiviert werden:

### Enable images in the RTE [enableImages]

If set, the use of images will be enabled in the default configuration of the RTE.



Damit wird der Umgang mit Bildern prinzipiell möglich, allerdings nicht der Bildupload direkt im RTE.

Zusätzlich ist daran zu denken, dass im PageTSConfig die maximale Bildbreite und Höhe angegeben werden muss:

```
buttons.image.options.plain.maxWidth = 600
```

```
buttons.image.options.plain.maxHeight = 800
```

# bzw.

```
buttons.image.options.magic.maxWidth = 300
```

```
buttons.image.options.magic.maxHeight = 400
```

Um den Bildupload direkt aus dem RTE zu ermöglichen, muss zusätzlich ein UserTSConfig hinzugefügt werden, entweder für jeden User oder besser in den User-Gruppen:

```
# Bildupload im RTE erlauben
```

```
options.uploadFieldsInTopOfEB = 1
```

```
# Ordner erstellen erlauben
```

```
options.createFoldersInEB = 1
```

The screenshot shows a configuration interface with three main tabs: 'Allgemein', 'Zugriffsrechte', and 'Freigaben und Arbeitsumgebungen'. The 'Options' sub-tab is selected. Under 'Auf Domäne beschränken:', there is a checkbox and an empty text input field. Under 'IP-Einschränkung für Benutzer deaktivieren', there is a checkbox. The 'TSconfig:' section contains a text area with the following configuration:

```
# Bildupload im RTE erlauben
options.uploadFieldsInTopOfEB = 1
# Ordner erstellen erlauben
options.createFoldersInEB = 1
```

# Eigene Klassen im RTE verwenden

---

Die Verwendung eigener Klassen im RTE ist nicht schwierig, allerdings darf man keine Einstellung vergessen, sonst funktioniert es nicht. Ausserdem muss man mit einigen Tricks den Browser-Cache überlisten, sonst wird man davon in die Irre geführt bzw. man sieht geänderte Klassen nicht oder mit erheblicher Verspätung im RTE.

## Stylesheet-Datei einbinden

Erster Schritt ist das Anlegen einer Stylesheet-Datei mit den gewünschten Klassen. Diese Klassen können in der einfachen Schreibweise mit einem Punkt geschrieben werden, zum Beispiel so:

```
.align-center {text-align:center}
```

oder aber jeweils an einen HTML-Tag gebunden, so zum Beispiel für ein Blockformat:

```
p.align-center, div.align-center {text-align:center}
```

oder für ein Inline-Format:

```
span.rot {color:#ff0000}
```

Die erste Lösung ist einfacher, erfordert aber eine Konfigurationsänderung im RTE:

```
RTE.default.showTagFreeClasses=1
```

Diese CSS-Datei muss in den RTE eingebunden werden mit folgender Anweisung (Pfad anpassen!):

```
RTE.default.contentCSS=fileadmin/tmpl/rte.css
```

Das heisst im Klartext, dass der RTE selbst eine Stylesheetdatei laden muss, es reicht nicht, die Datei nur für das Frontend einzubinden.

**Achtung:** Bevor Sie die Konfiguration der Klassen im TSConfig vornehmen, muss diese Datei existieren, die richtigen Klassen enthalten und korrekt geladen sein.

**Tipp:** Öffnen Sie ein Browserfenster und laden Sie dort die Datei direkt ins Browserfenster, so dass sie als Text erscheint. Nach jeder Änderung laden Sie diese Seite neu

im Browser mit gedrückter Umschalt-Taste (Mac) oder Str-Taste (Windows), damit der Browser-Cache gelöscht wird.

**Tipp:** Aufgrund eines Fehlers im Safari sollten Sie für die Klassenbezeichnungen immer Kleinschreibung verwenden, nie die Camel-Notation!

## Klassennamen definieren

Danach müssen (sollten) die Klassen im PageTSConfig des RTE genauer definiert werden. Unbedingt erforderlich ist das nicht, aber wenn es weggelassen wird, so erscheinen die Klassen mit den Selektornamen aus der CSS-Datei, was unschön aussieht.

```
RTE.classes {
  csc-frame-rulerbefore {
    name = Linie davor
    value =
  }
  dunkelgrau {
    name = dunkelgrauer Text
    value = color:#666666
  }
}
```

dunkelgrau: Klassenselektor (= Name in der CSS-Datei)  
name: Name im Menü  
value: Aussehen im Menü (nicht im Frontend!)

## Klassen in den Menüs aktivieren

Für jedes Pulldownmanü bzw. jede Klassenauswahl im RTE müssen die Klassen einzeln zugewiesen bzw. aktiviert werden. Zum Beispiel für die Zeichenstile:

```
RTE.default.classesCharacter = serifen, starkebetonung,
betonung, hellgrau, dunkelgrau
```

**Achtung:** Dieser Schritt ist notwendig, sonst erscheinen die Klassen nicht in den Menüs!

## Klassen im Parser freischalten

Damit die Klassen nicht nur im RTE zur Verfügung stehen, sondern auch beim Sichern in die Datenbank geschrieben werden, müssen sie alle noch aktiviert werden:

```
RTE.default.proc.allowedClasses (  
  align-left,align-center,align-right,  
  csc-frame-rulerbefore,csc-frame-rulerafter,  
  csc-frame-indent,csc-frame-frame1,csc-frame-frame2,  
  kleinertext, grossertext,  
)
```

Hier müssen alle Klassen aufgeführt werden, die verwendet werden sollen.

Beispiel einer CSS-Datei:

<http://p102292.typo3server.info/fileadmin/tmpl/rte.css>

Beispiel eines TSConfig-Setups:

<http://p102292.typo3server.info/fileadmin/tmpl/eigene-klassen-tsconfig.txt>

**Hinweis:** Diese Konfiguration am besten in das PageTS-Config der Rootseite einfügen (Seiteneigenschaften – Optionen – TSconfig).

**Hinweis:** Diese Konfiguration erlaubt die Verwendung eigener Klassen, es handelt sich noch nicht um eine vollständige bzw optimale Konfiguration des RTE. Allerdings ist sie zusammen mit der allgemeinen Grundeinstellung des RTE in TYPO3 4.2.1 bereits recht gut verwendbar.

Sie finden eine entsprechende Konfiguration in meiner Testseite (siehe erste Seite) in der Seite »Minimalkonfiguration). Sie können sich einloggen, die Konfiguration anschauen und die Verwendung des RTE mit dieser Konfiguration testen.

# Daten aus der Zwischenablage

Das größte Problem bei einem Rich Text Editor sind immer die Daten, die über die Zwischenablage eingefügt werden. Je nach Herkunft und grundsätzlichem Verhalten des Browsers entstehen durch die erforderliche Konvertierung eine Unmenge an zusätzlichen Tags und Attributen, die nicht erwünscht sind:

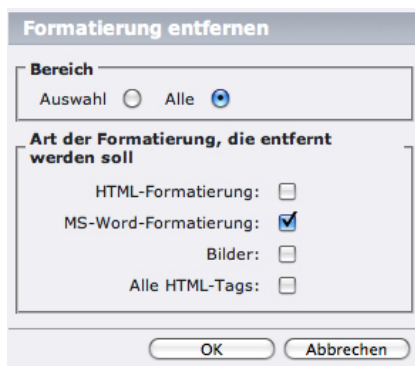
- Wird ein Text im Browser kopiert, so werden von Safari, Internet Explorer und Firefox unter Windows die Tags und Klassen der Originalseite mit kopiert. Diese sind aber im neuen Kontext zumindest überflüssig, wenn nicht schädlich.

- In Safari kommt hinzu, dass aus den Stilvorlagen der Originalseite, aus der kopiert wird, direkte Formate erstellt werden und an jeden Tag angehängt werden – es entstehen Konstruktionen wie `<h1 style= „„>`, die nur schwer manuell im RTE beseitigt werden können.

- Beim Kopieren aus Word gelangen viele überflüssige Klassen, style-Attribute und Anker in die Datei, die sich ebenfalls schädlich auswirken können.

Die beste Methode, dies zu umgehen ist folgende Vorgehensweise durch die Redakteure:

1. Originaltext kopieren
2. In RTE einfügen
3. Quellcode bereinigen lassen:



4. Speichern.

Dadurch wird der Quellcode zweimal bereinigt: einmal durch Entfernen der Word-Formatierungen (bezieht sich nicht nur auf reine Word-Formatierungen sondern bezieht auch andere mit ein) und anschließend beim Speichern wirken die konfigurierbaren Einstellungen des RTE-Parsers.

Das gilt nicht für Texte, die aus Word übernommen werden; auch bei Kopien aus Webseiten oder OpenOffice sollte genauso vorgegangen werden.

Die Browser verhalten sich beim Kopieren über die Zwischenablage sehr unterschiedlich:

- Internet Explorer (Windows)

Es werden praktisch alle Wordformatierungen übernommen – der größte Teil davon ist störend und muss auf dem beschriebenen Weg entfernt werden.

- Firefox (Windows)

Es werden viele Wordformate übernommen, die auch zum Teil entfernt werden müssen.

- Firefox (Macintosh)

Es werden keinerlei Formate übernommen und die Absätze des Originals landen als Zeilenschaltungen im RTE. In diesem Fall muss der Quellcode nicht bereinigt werden, es ist aber zu empfehlen, unmittelbar nach dem Einfügen zu Speichern, da dadurch aufeinanderfolgende Zeilenschaltungen wieder zu Absätzen werden.

- Safari (Macintosh)

Auch hier werden praktisch alle Wordformate übernommen (und nicht nur aus Word, sondern aus allen Programmen wird die RTF-Formatierung übernommen). Gerade beim Safari muss der Quellcode nach dem Einfügen unbedingt bereinigt werden.

# Erweiterte Einstellungen

---

Meine Minimal-Konfiguration erlaubt bereits die Standardwerkzeuge des RTE und eigene Klassen, allerdings möchte man normalerweise den RTE genauer an die Bedürfnisse anpassen, z. B. Werkzeuge anzeigen oder verstecken und so weiter. Ausserdem möchte man eventuelle den Quellcode noch stärker bereinigen lassen.

Sie finden eine Beispielkonfiguration in der Testinstallation in der Seite »Erweiterte Konfiguration« bzw. unter diesem Link:

<http://p102292.typo3server.info/fileadmin/tmpl/erweiterte-konf.txt>

## HTML-Tags für Inline-Formatierungen

Wenn in den Grundeinstellungen die Option »Enable additional Inline elements« aktiviert wurde, müssen Sie diese Tags wie zum Beispiel `<code>` oder `<cite>` im RTE erlauben und zusätzlich sicherstellen, dass der Parser, der für die Darstellung im Frontend durchlaufen wird, diese ebenfalls durchlässt.

1. Im RTE erlauben Sie HTML-Tags durch diese Einstellung:

```
RTE.default.proc.allowTags (  
    a,abbr, ... etc.  
)
```

Achten Sie darauf, hier nur die Tags anzugeben, die Sie im RTE auch wollen. Eine Zusammenstellung der erlaubten Tags in meiner Konfiguration finden Sie hier:

<http://p102292.typo3server.info/index.php?id=6>

Damit werden die Tags korrekt in die Datenbank geschrieben.

2. Damit die Tags auch korrekt aus der Datenbank in das Frontend gelangen, muss noch im Haupt-TypoScript-Setup eine Änderung vorgenommen werden. Dort werden nämlich alle Inhaltsobjekte nochmals geparkt (hier der Übergang von der Datenbank ins Frontend) und nochmals unerwünschte Tags entfernt. Sie müssen hier folgende Zeilen einfügen:

```
lib.parseFunc.allowTags = a,abbr,acronym,address,b,bdo,  
blockquote,br,cite,code,del,div,em,h1,h2,h3,h4,h5,h6,hr,i,  
img,li,ol,p,span,strike,strong,sub,sup,u,ul
```

```
lib.parseFunc_RTE.allowTags = a,abbr,acronym,address,b,  
bdo,blockquote,br,cite,code,del,div,em,h1,h2,h3,h4,h5,h6,  
hr,i,img,li,ol,p,span,strike,strong,sub,sup,u,ul
```

**Achtung:** Bedenken Sie, dass insbesondere `lib.parseFunc.allowTags` alle Einträge parst – wenn Sie also den RTE restriktiver fassen wollen und eher Tags verbieten als zusätzliche erlauben, so sollten Sie diese Anpassung nicht vornehmen, da sonst auch Tags aus anderen Inhaltselementen deaktiviert werden.

## Werkzeuge konfigurieren

Mit `RTE.default.showButtons` und `RTE.default.hideButtons` können Sie vorgeben, welche Werkzeuge im RTE gezeigt werden. Zusätzlich können diese Werkzeuge noch detailliert konfiguriert werden. Beispiele finden Sie in meinem (kommentierten) `TSCConfig`:

<http://p102292.typo3server.info/fileadmin/tmpl/erweiterte-konf.txt>

und natürlich im Manual.

## Parser konfigurieren

Über `RTE.default.proc.entryHTMLparser_db` kann das Parsen beim Übergang vom RTE in die Datenbank genau konfiguriert werden. Beispiele finden Sie wieder in meiner Konfiguration

<http://p102292.typo3server.info/fileadmin/tmpl/erweiterte-konf.txt>

und natürlich im Manual.

# Disclaimer, Test und Rückmeldungen

---

Ich »erforsche« den RTE gerade – meine Angaben sind also alle vorläufig. Es ist »work in progress« und heisst im Klartest: es kann alles noch ganz anders werden :-)

Falls jemand Fehler findet oder sonstige Anregungen hat, bitte ich um Rückmeldungen unter <mailto:info@lisardode>

Peter Linzenkirchner