

# KARRIEREhandbuch.php5 – Benutzerdokumentation

Markus-Hermann Koch  
[mhk@karrierehandbuch.de](mailto:mhk@karrierehandbuch.de)

11. Februar 2010

# Inhaltsverzeichnis

<b>1 Benutzer</b>	<b>5</b>
1.1 Neuen Benutzer anlegen (Admin) . . . . .	5
1.2 Bestehenden Benutzer tiefgreifend ändern (Admin) . . . . .	7
1.2.1 Benutzer ändern . . . . .	7
1.2.2 Benutzer löschen . . . . .	8
1.3 Ihren eigenen Account bearbeiten . . . . .	8
<b>2 Knoten</b>	<b>11</b>
2.1 Neue Äste anlegen . . . . .	11
2.2 Knoten bearbeiten . . . . .	13
2.2.1 Knoten verschieben . . . . .	13
2.2.2 Knoten entfernen . . . . .	14
2.3 Kids eines Knoten umsortieren . . . . .	14
<b>3 Inhalte bearbeiten</b>	<b>16</b>
3.1 Das Formular „Inhalte bearbeiten“ . . . . .	16
3.1.1 Anmerkungen . . . . .	18
3.2 Formatierung mit Hilfe der Content Converter . . . . .	19
3.2.1 Karrierehandbuch Light mit HTML . . . . .	19
3.2.2 Allgemeingültiges zu den Konvertern . . . . .	19
3.2.3 Der Content Converter STANDARD . . . . .	21
3.2.4 Der Content Converter TABULAR . . . . .	27
3.2.5 Der Content Converter TINYMCE . . . . .	30
3.2.6 Der Content Converter TWOCOL . . . . .	31
<b>4 Jobs – Stellen ausschreiben</b>	<b>34</b>
4.1 Stellenangebote ausschreiben oder bearbeiten . . . . .	35
4.2 Stellenangebote entfernen . . . . .	36
<b>5 Dateien</b>	<b>37</b>
5.1 Download- oder Grafik-Dateien verwalten . . . . .	37
5.1.1 Dateien auf den Server hochladen . . . . .	37
5.1.2 Verzeichnisse Anlegen und Wechseln . . . . .	38
5.2 Dateien und leere Verzeichnisse Löschen . . . . .	38
<b>6 Datenbank</b>	<b>39</b>
6.1 Liste der noch SA-freien Texte . . . . .	39
6.2 Datenbankkonsistenztest . . . . .	39
6.3 Ausgelagerte Skripte . . . . .	40
6.3.1 examineFiles . . . . .	40
6.3.2 nodeInfo.php5 . . . . .	41

<b>7 Aliases</b>	<b>42</b>
7.1 Über Aliase . . . . .	42
7.2 Konventionen . . . . .	43
7.3 Die Aliase . . . . .	43
7.3.1 ae . . . . .	43
7.3.2 bf . . . . .	44
7.3.3 br . . . . .	44
7.3.4 divCenter . . . . .	44
7.3.5 divLine . . . . .	44
7.3.6 em . . . . .	44
7.3.7 larger . . . . .	44
7.3.8 smaller . . . . .	44
7.3.9 underline . . . . .	44
7.3.10 ADD . . . . .	44
7.3.11 ADY . . . . .	44
7.3.12 ALR . . . . .	45
7.3.13 ALLVISITORS . . . . .	45
7.3.14 ALY . . . . .	45
7.3.15 AN . . . . .	45
7.3.16 AN* . . . . .	45
7.3.17 ARR . . . . .	46
7.3.18 ARY . . . . .	46
7.3.19 AUY . . . . .	46
7.3.20 AUTHCLTCOMBO . . . . .	46
7.3.21 AUY . . . . .	47
7.3.22 BBOOK . . . . .	47
7.3.23 BROWSER . . . . .	47
7.3.24 BY . . . . .	47
7.3.25 CHR . . . . .	48
7.3.26 CLT . . . . .	48
7.3.27 COMBO . . . . .	49
7.3.28 CONTACTNOW . . . . .	49
7.3.29 CORPCOL . . . . .	49
7.3.30 D . . . . .	49
7.3.31 DATE . . . . .	50
7.3.32 DD . . . . .	50
7.3.33 DWN . . . . .	50
7.3.34 DWN* . . . . .	51
7.3.35 EML . . . . .	51
7.3.36 FAX . . . . .	51
7.3.37 FILE . . . . .	51
7.3.38 FORM . . . . .	51
7.3.39 FORMe . . . . .	52
7.3.40 GOOGLEMAP . . . . .	52
7.3.41 GOTONEXT . . . . .	53

7.3.42 GOTOTOP . . . . .	53
7.3.43 GTEXT . . . . .	54
7.3.44 HIDDEN . . . . .	54
7.3.45 HOME . . . . .	55
7.3.46 IMG . . . . .	55
7.3.47 JOB . . . . .	56
7.3.48 JOBLIST . . . . .	56
7.3.49 JUS . . . . .	57
7.3.50 KBOOK . . . . .	57
7.3.51 KH_URL . . . . .	57
7.3.52 MPFORM . . . . .	57
7.3.53 NAME . . . . .	57
7.3.54 OBJECT . . . . .	57
7.3.55 PAR . . . . .	58
7.3.56 PDF . . . . .	58
7.3.57 PHP . . . . .	59
7.3.58 REM . . . . .	59
7.3.59 SA . . . . .	59
7.3.60 SATAR . . . . .	59
7.3.61 STATIC . . . . .	60
7.3.62 TEL . . . . .	60
7.3.63 TIME . . . . .	60
7.3.64 TITLE . . . . .	60
7.3.65 TITLELINE . . . . .	61
7.3.66 TITLELINE* . . . . .	61
7.3.67 TOC . . . . .	61
7.3.68 TR . . . . .	62
7.3.69 TRe . . . . .	62
7.3.70 USER . . . . .	62
7.3.71 WRY . . . . .	63
7.3.72 WDY, WLY, WUY . . . . .	63
<b>8 Häufig benötigte Aliase</b>	<b>64</b>

## Abbildungsverzeichnis

1	Das Formular „Neuen Benutzer anlegen“ . . . . .	5
2	Die Hierarchie der Rechte . . . . .	6
3	Das Formular „Benutzer ändern“ . . . . .	8
4	Das Formular „Mein Account“ . . . . .	9
5	Beispiel einer Index-Content-Struktur . . . . .	11
6	Das Formular „Neuer Index-Knoten“ . . . . .	12
7	Das Formular „Knoten bearbeiten“ . . . . .	13
8	Das Formular „Kids eines Knoten umsortieren“ . . . . .	14
9	Ein einfacher Standard-Inhalt . . . . .	22
10	Ein einfacher Standard-Inhalt . . . . .	23
11	Die Checkliste im Browser . . . . .	26
12	Eine einfache Tabelle . . . . .	28
13	Ein Tab-Artikel aus dem Karrierehandbuch . . . . .	30
14	Der TinyMCE-Editor . . . . .	31
15	Eine Div-Folge als zweispaltige Tabelle . . . . .	33
16	Das Formular „Ein automatisch formatiertes Stellenangebot“ .	34
17	Das Formular „Das Menü Jobs“ . . . . .	34
18	Das Formular „Das Hauptformular für die Jobeintragung“ .	35
19	Das Formular „Grafikdateien verwalten“ . . . . .	37
20	Ein Datenbankkonsistenzbericht . . . . .	40
21	Das Tool <b>examineFiles</b> . . . . .	40

# 1 Benutzer

## 1.1 Neuen Benutzer anlegen (Admin)

Um mit dieser Funktion arbeiten zu können, benötigen Sie Admin-Rechte auf mindestens einem Client.

Die Rechte der Benutzer, die Sie hier anlegen, müssen sich auf diejenigen Clients beschränken, auf denen Sie Admin-Rechte besitzen.

The screenshot shows a user creation form titled "Neuen Benutzer anlegen". The left sidebar has a "Benutzer" section selected. The main form contains the following fields:

Anrede	Herr
Vorname	Ullrich
Nachname	Schulz
Sprache	de
E-Mail	hal@web.de
Expires	19.11.2010 um 10:22:42
Subsites	Koch Management Consulting ILLIG Maschinenbau SCHILLER AUTOMATION
Rechte	r:[de]
Login	schulzuh
Passwort	[redacted]
Wiederholung	[redacted]
Herr Schulz ist für...	
Anmerkungen	

At the bottom of the form are two buttons: "Abschicken" and "Zurücksetzen".

Abbildung 1: Das Formular „Neuen Benutzer anlegen“

Die Felder im Einzelnen:

- **Anrede:** Anrede für den neuen Benutzer. Z.B.: Herr oder Frau Dr.
- **Vorname, Nachname:** Name des Benutzers.
- **Sprache:** Bevorzugte Sprache des Benutzers.
- **E-Mail:** Eine gültige E-Mail-Adresse.
- **Expires:** Der Zeitpunkt, zu dem der Account abläuft.
- **Subsites:** Eine Multichoose-Combobox. Angezeigt werden hier diejenigen Clients auf denen Sie Admin-Rechte haben. Wählen Sie diejenigen davon aus, auf denen der neue Benutzer Rechte erhalten soll.

- **Rechte:** Geben Sie hier eine durch Semikoli (;) getrennte Liste von Rechten ein. Diese muss entweder soviele Elemente enthalten wie Subsites ausgewählt wurden, oder aber ein einzelnes, welches dem User dann für die ausgewählten Subsites zugeteilt wird.

Die Reihenfolge der vergebenen Rechte entspricht der Reihenfolge im Subsites-Feld.

In der Beispielgrafik wird Herrn Schulz das Recht **r** bei *Koch Management Consulting* und das Recht **t[de]** bei *SCHILLER AUTOMATION* eingeräumt.

- **Login:** Login-Kennung des Users. Diese muss aus alphanumerischen Zeichen bestehen und darf keine Umlaute enthalten. Per Default hat sie die Form

<Erste 6 Buchstaben des Nachnamens><1. Bs. d. Vornamens>  
 <Letzter Bs d. Nachnamens>

So erhält *Ullrich Schulz* das Login **schulzuh**.

- **Passwort:** Das Login-Passwort für den neuen Benutzer. Es muss mindestens sechs Zeichen haben.
- **Wiederholung:** Erneute Eingabe des Passwortes. Zur Rechtschreibkontrolle.
- **Anmerkungen:** Hier können Sie bei Bedarf Bemerkungen zum neuen Account eintragen. Diese wird nur noch im Formular „Benutzer tiefgehend ändern angezeigt“ und hat ansonsten keine Bedeutung.

Abbildung 2 zeigt die Hierarchie des Rechtesystems der Redaktion.

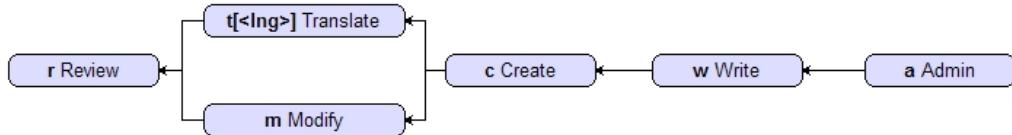


Abbildung 2: Die Hierarchie der Rechte

Es folgt eine Beschreibung der einzelnen Rechte. Höhere Rechte umschließen dabei sämtliche Funktionen aller ihnen untergeordneten Rechte.

- **r:** *Review* ist das kleinste Recht und dazu gedacht, statistische Werte überprüfen zu können. User mit diesem Recht können Datenbankkonsistenztests durchführen. Außerdem sind sie vollwertige User, die zum Beispiel Sprechzeiten haben können.
- **m:** *Modify* erlaubt die Modifikation bestehender Inhalte des rechtegewährenden Mandanten. Die Baumstruktur darf dabei nicht angefasst werden. 'Modifikation' erstreckt sich auch nicht auf die vollständige Entfernung von Inhalten.

- **t [<lng>]:** *Translate.* Dem Übersetzerrecht ist ein Sprachkürzel nach ISO 639-1 zugeordnet. Ein Übersetzer kann bestehende Inhalte der betreffenden Sprache ändern. Außerdem kann er über die Übersetzermaske neue Inhalte anlegen sofern bereits ein Originaleintrag in einer anderen Sprache existiert der übersetzt werden kann.
- **c:** *Create.* Wer das kreative Recht besitzt kann auf den betreffenden Clients neue Knoten in der Baumstruktur anlegen.
- **w:** *Write.* Ein Benutzer mit Schreibrechten hat vollständige Rechte zum Lesen, Schreiben und Löschen von Inhalten, Knoten und Visitenkarten innerhalb der Karriere-Sub-Site der betreffenden Clients.
- **a:** *Admin.* Admins haben vollständige Rechte auf den betreffenden Clients. Sie können Benutzer anlegen, ändern und entfernen sofern sich die Rechte dieser Benutzer in Clients erschöpfen die von dem betreffenden Admin administriert werden.

## 1.2 Bestehenden Benutzer tiefgreifend ändern (Admin)

Um mit dieser Funktion arbeiten zu können, benötigen Sie Admin-Rechte auf mindestens einem Client.

Die Rechte der Benutzer, die Sie hier bearbeiten, müssen sich auf diejenigen Clients beschränken, auf denen Sie Admin-Rechte besitzen.

### 1.2.1 Benutzer ändern

Das Formular in Abbildung 3 zeigt Ihnen die Daten des zu bearbeitenden Users an und erlaubt deren Modifikation. Das bedeutet auch, dass das Passwort des Users im Quellcode sichtbar wird!

The screenshot shows a user editing interface. On the left is a dark sidebar with navigation links: 'Hinweise', 'Benutzer' (which is bolded), 'Knoten', 'Inhalte', 'Visitenkarten', 'Dateien', 'Datenbank', 'Ausloggen', and 'Hilfe (PDF)'. To the right of the sidebar is a form titled 'Benutzer bearbeiten'. The form contains the following fields:

login	schulzuh
timestamp	2008-11-19 11:33:07
address	Herr
firstname	Ullrich
lastname	Schulz
lng	de
email	hal@web.de
cltsuffix	kh.schiller
auth	r.t[de]
expires	19.11.2010, 11:32:27
comment	Herr Schulz ist für...
passwd1	*****
passwd2	*****

At the bottom of the form are two buttons: 'Abschicken' and 'Zurücksetzen'.

Abbildung 3: Das Formular „Benutzer ändern“

Die Felder entsprechen denen, die bereits im Abschnitt (1.1), „Neuen Benutzer anlegen“ dokumentiert sind.

### 1.2.2 Benutzer löschen

Entfernt den Benutzer komplett aus der Datenbank.

## 1.3 Ihren eigenen Account bearbeiten

Hier können Sie nur Ihren eigenen Account bearbeiten. Insbesondere können Sie nur in diesem Formular persönliche Kontaktdaten und Sprechzeiten für Dritte definieren. Die Kontaktdaten werden während der Sprechzeiten Besuchern der Site angezeigt, wenn diese auf Werbung Ihrer Unternehmen stoßen und die Option 'aktive Ansprechpartner' auswählen.

Hier können Sie Ihre E-Mail-Adresse und Ihr Passwort ändern und Ihren Account um Kontaktdata und Sprechzeiten erweitern.  
Ihre öffentlichen Kontaktdaten können Sie in den Inhalten der Site mit Hilfe des Alias **USER** anzeigen lassen.

Geben Sie hier bitte noch einmal Ihr Passwort ein	*****
Ihre aktuelle E-Mail-Adresse	hal@web.de
gff. Ihr Neues Passwort	
Wiederholung Ihres neuen Passworts	
Firmenauswahl <sup>1</sup>	Koch Management Consulting <b>SCHILLER AUTOMATION</b>
Sprechzeiten <sup>2</sup>	Mo-Do 8.00-13.00 14.00-17.00; Fr 08.00-13.00
öffentliche Telefonnummer	0711 / 123454321
öffentliche E-Mail-Adresse	info@myCom.de
Beliebige URL	http://www.my_com.tv
Text für diese URL	Besuchen Sie uns!
Sprechstunden-Text <sup>3</sup>	Zur Zeit stehe ich Ihnen für Fragen zur Verfügung.
Sperrzeiten-Text <sup>3</sup>	Im Augenblick bin ich nicht erreichbar.

**Abschicken** **Zurücksetzen**

Abbildung 4: Das Formular „Mein Account“

Zu den Feldern des Formulars:

- **Passwortwiederholung:** Um Änderungen an Ihrem Account vornehmen zu können müssen Sie hier noch einmal Ihr Passwort eintragen.
- **Aktuelle E-Mail-Adresse:** Hier steht die E-Mail-Adresse unter der wir Sie erreichen können sollen.
- **Neues Passwort / Wiederholung:** Hier können Sie sich selbst ein neues Zugangspasswort geben. Es benötigt eine Mindestlänge von 6 Zeichen.
- **Firmenauswahl:** *Diese und alle nachfolgenden Felder bezieht sich auf Ihre persönlichen Kontaktdaten, die Sie im Internet veröffentlichen wollen.*  
Auf den Seiten des Karrierehandbuchs können Sie sich als Ansprechpartner für alle oder einige der Unternehmen, auf die sich der Gültigkeitsbereich Ihres Accounts erstreckt, anzeigen lassen.  
Diese Unternehmen werden in der Multichoose-Combobox „Firmenauswahl“ angezeigt und können ausgewählt werden.
- **Sprechzeiten:** In dieses Feld können Sie wochenbezogene Zeiten eintragen, zu denen Sie erreichbar sind. Die genaue Syntax für dieses Feld wird in einer Fußnote des Formulars erklärt.
- **Öffentliche Telefonnummer:** Eine öffentliche Telefonnummer unter der Sie für Dritte erreichbar sind.
- **Öffentliche E-Mail-Adresse:** Eine öffentliche E-Mail-Adresse unter der Sie für Dritte erreichbar sind. Das Kontaktdatenprogramm stellt diese Adresse gegebenenfalls automatisch als Grafik dar. Wenn Sie

das Alias {{USER...;appmail}} verwenden erreichen Sie mit folgendem kombinierten Alias dasselbe:

```
 {{GTEXT'{{USER<login>;appmail}}'}}}
```

Die Kodierung der E-Mail-Adresse als Grafik soll verhindern, dass das Web absuchende Programme diese Adresse auskundschaften. Diese werden meist von Spam-Verbreitern eingesetzt.

- **Beliebiger URL:** Auch ein beliebiger Link kann zu Ihren Kontakt-  
daten zählen.
- **Text für diese URL:** In Ihren automatischen Kontaktdaten wird  
diesem Link auch eine Caption unterlegt, die Sie hier festlegen können.  
Wenn Sie dieses Feld frei lassen, wird die URL sowohl als Linkziel als  
auch als Caption verwendet.
- **Sprechstundentext:** Ein Text, der im Kontaktbereich zusätzlich an-  
gezeigt wird, wenn Sie gerade erreichbar sind.
- **Sperrstundentext:** Im Augenblick noch ohne besondere Funktion.

## 2 Knoten

Struktur und Inhalte der einzelnen Client-Sites sind in der Datenbank getrennt. In den *Index*-Tabellen der einzelnen Clients sind inhaltslose *Knoten* zu einer Baumstruktur verwoben. Jeder dieser Knoten verfügt lediglich über eine „Content id“ (*cid*), mit der auf den eigentlichen Inhalt einer Seite in einer demselben Client zugeordneten *Content*-Tabelle zugegriffen werden kann.

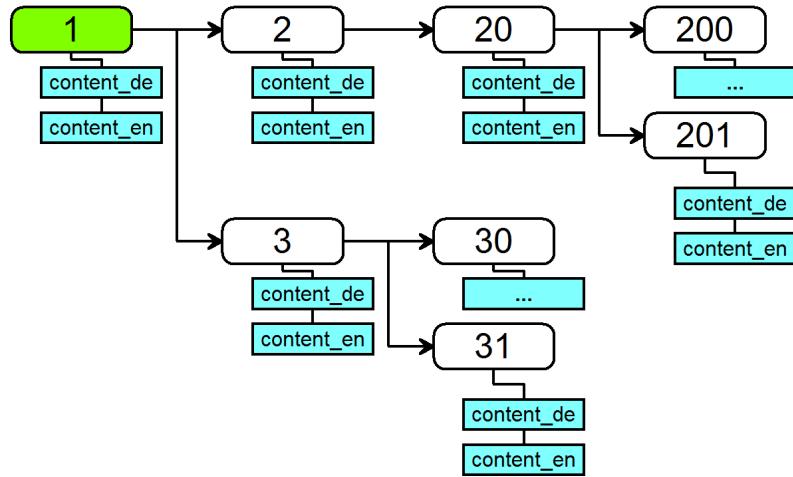


Abbildung 5: Beispiel einer Index-Content-Struktur

Der Nutzen dieses Systems liegt darin, dass ein Client mehrere Content-Tabellen für die selbe Index-Struktur besitzen kann. Beispielsweise eine Content-Tabelle mit Inhalten in deutscher Sprache und eine mit deren Übersetzungen ins Englische<sup>1</sup>.

Auf die Baumstruktur nehmen eine Reihe von Automatismen des CMS Bezug. Eine der Wichtigsten ist die automatische Erstellung von Inhaltsverzeichnissen in den Darstellungsklassen **standard** und **twocol**.

Dieses Kapitel befasst sich mit der Manipulation der *Knoten* der Baumstruktur.

### 2.1 Neue Äste anlegen

Um auf einem Client neue Knoten anlegen zu können, benötigen Sie mindestens das Recht „creative“ (c).

---

<sup>1</sup>Wenn Ihr Unternehmen bei uns eine eigene Sub-Site betreibt, ist Ihnen zunächst nur eine Content-Tabelle zugeordnet. Wir sind jedoch ohne weiteres bereit, Ihnen auf Wunsch weitere Content-Tabellen einzurichten.

| Neuer Index-Knoten |

Wählen Sie einen Artikel für die Bearbeitung aus

1: Willkommen bei NAMEauli			
1: Willkommen bei NAMEauli	100: Der Global Player auf dem Beleuchtungssekt...	110: Die Zukunft gehört dem, der am effiziente...	
		120: Unsere Mitarbeiter sind unsere Bindegliede...	
		130: Headquarters der Automotive Lighting Reutl...	
400: Reutlingen – das Tor zur Schwäbischen Alb			
'title' für den neuen Knoten	Studienarbeiten bei Automotive Lighting		
'id' für den neuen Knoten	140		
'cid' für den neuen Knoten			
'tid' für den neuen Knoten	headlink		
'comment' für den neuen Knoten			
<a href="#">Den grün markierten Artikel bearbeiten</a>			

Abbildung 6: Das Formular „Neuer Index-Knoten“

Nach der Auswahl des zu bearbeitenden Client werden Sie direkt auf den Strukturbrowser weitergeleitet. Der Browser läuft wesentlich flüssiger, wenn Sie Java-Script aktiviert haben, funktioniert aber auch ohne.

Navigieren Sie nun im Browser bis derjenige Knoten, an den Sie den neuen Knoten anhängen wollen, grün unterlegt ist.

Als letzten Schritt füllen Sie das unter dem Browser befindliche, blaue Formular aus und bestätigen mit einem Klick auf *Den grün markierten Artikel bearbeiten*.

Die Felder des Formulars:

- **title:** Mit dem Knoten wird auch ein Rocheintrag in der Content-Tabelle der Defaultsprache des Clients angelegt. Dieser Eintrag enthält neben seiner Id zunächst nur die Überschrift, die Sie hier eingeben können.
- **id:** Jeder Knoten benötigt eine im aktiven Client eindeutige Id aus der Menge  $\{2, \dots, 2.147.483.647\}$ . Die Vergabe dieser Id ist für den Knoten endgültig.
- **cid:** Für die **Content Id** gelten dieselben Regeln wie für die **id**. Wenn Sie das Feld einfach frei lassen, wird für die **cid** dieselbe Nummer wie für die **id** verwendet. Bislang gibt es noch kein Feature, dass eine Abweichung von diesem Default sinnvoll macht.
- **tid:** Hier kann für den Knoten ein zusätzlicher Tipp definiert werden, der dann am Ende des Artikels angezeigt wird. Bislang erklärt sind

- **headlink:** *Mit Klick auf die Überschrift öffnen Sie das Inhaltsverzeichnis*
- **headcm:** *Mit Klick auf die Überschrift gelangen Sie zur Karrieremesse*

Default ist **NULL**. In diesem Fall wird dem Knoten kein Tipp zugeordnet.

- **comment:** Ein Kommentar, der in die Datenbank eingetragen wird, sonst aber ohne Funktion ist.

Der neue Knoten wird sofort angelegt und sein Inhalt kann unter *Inhalte→Inhalte bearbeiten* für die Hauptsprache des Clients erstellt und gegebenenfalls unter *Inhalte→Inhalte übersetzen* übersetzt werden.

Bei erfolgreicher Anlegung des Knotens wird keine eigene Bestätigungsmeldung ausgegeben. Stattdessen wird der Strukturbrowser aktualisiert und das Knotenerstellungsformular erneut angezeigt.

## 2.2 Knoten bearbeiten

*Knoten bearbeiten* ist ein mächtiger Handlungsstrang, der das Verschieben oder Entfernen ganzer Äste der Baumstruktur ermöglicht.

1. Willkommen bei NAMEmanz			
1: Willkommen bei NAMEmanz	100: Systemlösungen in der Automation	101: Manz - der Technologieanbieter	
		102: Internationale Kunden erfordern internatio...	
		103: step by step - one step ahead	
		104: Innovationen bei Manz	

200: Reutlingen - das Tor zur Schwäbischen Alb	
Den ausgewählten Knoten verschieben	<input checked="" type="radio"/>
Den ausgewählten Knoten entfernen	<input type="radio"/>
Knoten ggf. mit Unterknoten entfernen	<input type="checkbox"/>

Den grün markierten Artikel bearbeiten
--

Abbildung 7: Das Formular „Knoten bearbeiten“

### 2.2.1 Knoten verschieben

In vier kurzen Schritten können Sie ganze Äste Ihrer Indexstruktur verschieben oder miteinander vertauschen:

1. Navigieren Sie im Strukturbrowser bis die Wurzel des zu verschiebenden Astes erreicht ist.

2. Aktivieren Sie den Radio Button „Den ausgewählten Knoten verschieben“ und Klicken auf „Den grün markierten Artikel bearbeiten“. Das Programm führt Sie nun auf eine Seite mit einem weiteren Strukturbrowser und dem Vermerk „Wählen Sie eine Aktion für den ersten Knoten“
3. Navigieren Sie zum Zielknoten für Ihre Aktion.
4. Wählen Sie je nach Bedarf „Vertauschen“ oder „Anhängen“ aus und bestätigen Sie abermals mit „Den grün markierten Artikel bearbeiten“. Das Programm überprüft dann, ob Ihr Vorhaben der Logik des Baums entspricht und führt es gegebenenfalls aus. Andernfalls erhalten Sie eine Fehlermeldung.

### 2.2.2 Knoten entfernen

Gegebenenfalls können Sie Äste auch komplett entfernen. Beachten Sie dabei, dass das Karriere CMS keinen „Mülleimer“ enthält. Gelöschte Inhalte gehen verloren.

Um Knoten zu entfernen gehen Sie wie folgt vor:

1. Navigieren Sie im Strukturbrowser bis der zu entfernende Knoten grün markiert ist.
2. Wählen Sie nun im blauen Formular darunter den Radio-Button „Den ausgewählten Knoten entfernen“ und, falls erforderlich, die Checkbox „Knoten ggf. mit Unterknoten entfernen“. Bestätigen Sie mit einem Klick auf „Den grün markierten Artikel bearbeiten“ und die ausgewählten Knoten werden vollständig gelöscht.

### 2.3 Kids eines Knoten umsortieren

Jeder Knoten in der Baumstruktur eines Clients kann weitere Knoten haben, die von ihm ausgehen. Die Ids dieser Tochterknoten werden im Karriere-CMS als *Kids* („Kid Ids“) bezeichnet.

The screenshot shows a user interface for reordering child nodes. At the top, there is a yellow button labeled 'Kids umsortieren'. Below it is a table with two columns. The left column contains the text 'Ordnen Sie die Liste nach Bedarf neu.' and the right column contains a list of integers from 1 to 51. At the bottom of the table are two buttons: 'Abschicken' and 'Zurücksetzen'.

Ordnen Sie die Liste nach Bedarf neu.	1;2;3;4;5;6;7;8;9;10;11;12;13;14;21;22;23;24;25;26;27;28;30;50;51
	<input type="button" value="Abschicken"/> <input type="button" value="Zurücksetzen"/>

Abbildung 8: Das Formular „Kids eines Knoten umsortieren“

Eine solche Tochterknotenliste hat auch eine Reihenfolge, die bei der automatischen Erstellung von Inhaltsverzeichnissen beibehalten wird.

Gelegentlich kommt der Wunsch auf, diese Reihenfolge zu ändern. Das ist die Aufgabe des Handlungsstranges „Kids eines Knoten umsortieren“. Der einfache Prozess ist in zwei Schritte gegliedert:

1. Nach der Wahl des Zielclients wird der Strukturbrowser angezeigt. Navigieren Sie hier bis der Knoten, dessen Kids Sie umsortieren wollen, grün hervorgehoben ist und bestätigen Sie mit einem Klick auf „Den grün markierten Artikel bearbeiten“.
2. Das Programm führt Sie nun auf ein Miniformular wie es in Abbildung 8 dargestellt ist. In der Textbox steht die durch Semikoli getrennte Liste der Tochterknoten des ausgewählten Wurzelknotens. Sortieren Sie diese nach belieben um und bestätigen Sie die Aktion mit einem Klick auf „Abschicken“.  
Das Programm prüft dann, ob der neue String der Logik des Baums entspricht und führt die Umsortierung sofort aus. Ansonsten wird eine Fehlermeldung ausgegeben.

## 3 Inhalte bearbeiten

### 3.1 Das Formular „Inhalte bearbeiten“

Im Formular „Inhalte bearbeiten“ können alle strukturunabhängigen Aspekte eines Inhalts bearbeitet werden.

Die einzelnen Felder der Reihe nach:

- **Title:** Dieses Feld ist für die Überschrift des Artikels bestimmt.  
Neben den *Content Convertern* (siehe 3.2) werten vor allem die automatische Inhaltsverzeichniserzeugung und einige Aliase diesen Eintrag aus.  
Wenn Sie keinen Content Converter verwenden wird der Inhalt dieses Feldes im fertigen Artikel nicht angezeigt.  
`title` unterstützt sowohl HTML als auch das Aliaskonzept.
- **Content:** Der eigentliche Inhalt des Artikels.  
`Content` unterstützt sowohl HTML als auch das Aliaskonzept.
- **CConv:** Im Feld *Content Converter* können Sie eine der implementierten Formatierungsklassen auswählen und so den Inhalt vor der Darstellung im Browser vorformatieren lassen.  
Wenn Sie dieses Feld einfach leer lassen wird der Content des Artikels unverändert ins Hauptfenster eingetragen – lediglich Aliase werden interpretiert.  
Eine ausführliche Dokumentation zum Thema Content Converter finden Sie unter 3.2.
- **returnUrl:** Die Content Converter erzeugen aus dem Eintrag unter `title` eine Überschrift die gleichzeitig als Link fungieren kann.  
Grundsätzlich können Sie hier unter drei Optionen wählen:
  1. Sie lassen das Feld leer. In diesem Fall werden die Content Converter die Überschrift in einen Link auf das naheliegendste, in der Hierarchie oberhalb des Artikel liegende Inhaltsverzeichnis umwandeln. Als Unteranker wird die id des Artikels verwendet. Letzteres führt dazu, dass in automatisch erzeugten Inhaltsverzeichnissen direkt an die Stelle gesprungen wird die auf den aufrufenden Artikel verweist.
  2. **NONE.** Wenn Sie `NONE` eintragen stellen die Content Converter die Überschrift dar ohne einen Link daraus zu machen.
  3. Eine beliebige URL auf die der Überschriftlink dann verweist.

`returnUrl` unterstützt das Aliaskonzept. Die Hauptanwendung hierfür bietet das Alias `{{CLT..}}` mit dem Sie eine URL auf einen Artikel innerhalb der Karriere-Site ohne umschließendes HTML-Tag erzeugen

lassen können. `{{{CLT}}}` ist unter 7.3.26 ausführlich dokumentiert. Dennoch ein kurzes, typisches Anwendungsbeispiel im Feld `returnUrl`: `{{{CLT};42;;1}}` erzeugt eine URL auf den Artikel mit der Index Id 42 innerhalb der aktiven Sub-Site und in der aktiven Sprache. Sind der Mandant `muster` und die Sprache `de` aktiv, wäre als Return Url nun folgendes eingestellt:

```
http://www.karrierehandbuch.de/  
karrierehandbuch.php5?con=42&lng=de&clt=muster
```

- **secRetCid:** Alle Content Converter die aus dem Eintrag unter `title` eine Überschrift erzeugen bieten die Option auf eine *Secondary* und bei Bedarf auch eine *Ternary Return Cid* für die Überschrift. Dabei wird links und ggf. auch rechts von der Überschrift ein Symbol eingeblendet welches gleichzeitig als Link auf den Artikel mit der in `secRetCid` angegebenen Content Id fungiert. Sie haben hier bislang drei Optionen:
  1. Sie lassen das Feld leer.
  2. Sie geben die *Content Id* eines Artikels innerhalb Ihrer Subsite an. Dann wird links neben der Überschrift ein Symbol (per Default ein nach links weisender Pfeil) erzeugt, welches auf diesen Zielartikel verweist.
  3. Sie fügen dieser Content Id noch ein ; und eine weitere Content Id hinzu. Dann wird per Default rechts neben der Überschrift noch ein weiterer, nach rechts weisender Pfeil angezeigt, der auf diese zusätzliche Zielseite verweist. Das ist z.B. für Fotostrecken nützlich.
- **toc-Status:** Grundsätzlich fasst Karriere-CMS Artikel mit leerem Content-Feld als automatisch zu erzeugendes Inhaltsverzeichnis auf. Anstelle des leeren Inhaltes wird ein im Stil des eingestellten Content Converters<sup>2</sup> dargestelltes Verzeichnis der dem Artikel hierarchisch untergeordneten Artikel erzeugt.  
Die automatische Erzeugung von ReturnUrls nutzt dieses Verfahren ebenfalls: Wenn Sie das Feld `returnUrl` leer lassen verlinken durch Content Converter dargestellten Überschriften auf den ersten hierarchisch übergeordneten leeren Artikel.  
Gelegentlich ist es allerdings erforderlich, ein Inhaltsverzeichnis von Hand zu erstellen und dieses hat dann einen nichtleeren Content der vom CMS nicht als Inhaltsverzeichnis erkannt wird. Mit dem Eintrag einer 1 im Feld `toc-Status` können Sie das beheben.
- **pdf-Meta:** Bislang wird für einfach gestrickte und mit dem Content Converter `standard` dargestellte ein auf dem Open-Source-Projekt FPDF

---

<sup>2</sup>sofern dieser Inhaltsverzeichnisse unterstützt – bislang nur für *standard* und *twocol*

aufbauender pdf-Erzeuger angeboten.

Damit das funktioniert müssen Sie unter **cconv** den Converter **standard** auswählen. Dann können Sie eine §§-Liste von im Acrobat-Reader mit Hilfe von <Strg>+<D> abrufbaren PDF-Metadaten eintragen.

Die Syntax ist wie folgt:

<Name des Verfassers>[§§<Thema>§§<Stichworte>] Aus der Tatsache, dass Sie das Feld nicht leer lassen oder NULL eintragen schließt das CMS, dass ein PDF-Erzeugungsbutton erwünscht ist. Wenn Sie das Feld **pdf-Meta** leer lassen wird die PDF-Erzeugung des Artikels nicht angeboten.

- **additional:** Hier können Sie eine ', '-Liste von ebenfalls auf dieser Seite anzuzeigenden Artikeln eingeben. Die Syntax eines einzelnen Eintrags ist dabei von der Form einer ', '-Liste und wie folgt:

[**clt**,]**cid**[,**lng**] [,**type**]

- **clt:** Clientsuffix des Zusatzartikels.  
Per Default der aktive Client.
- **cid:** Content id des Zusatzartikels.
- **lng:** Sprachkürzel des Zusatzartikels.  
Per Default die aktive Sprache.
- **type:** *ToDo. Dieses Feature ist noch nicht funktionsfähig. Im Augenblick funktioniert additional nur für den Type pdf und den cconv 'standard'.*

Typ der Erweiterung. Mögliche Einträge (die auch durch Leerzeichen getrennt hintereinander gestaffelt werden können) sind

- \* **all:** Fügt die Erweiterung für alle unterstützten Medien hinzu.
- \* **regular:** Fügt die Erweiterung der HTML-Variante hinzu.
- \* **pdf:** Fügt die Erweiterung der automatisch generierten PDF-Variante hinzu.

Default ist 'pdf'.

- **comment:** Hier können Sie persönliche Anmerkungen eintragen. Das Karriere-CMS notiert hier außerdem User-Login und Zeitpunkt der letzten Änderungen am Artikel.

### 3.1.1 Anmerkungen

Die von Ihnen getätigten Einträge kommen (nach einer SQL-Injections erschwerenden Kodierung) direkt in die SQL-Datenbank sofern die zugehörigen Felder irgendwelche Zeichen enthalten. Leere Felder werden ignoriert. Wenn Sie einen vorher befüllten Eintrag einmal leeren wollen können Sie das bewirken indem Sie das Wort **NULL** eintragen.

## 3.2 Formatierung mit Hilfe der Content Converter

Ursprünglich wurde die PHP-Neufassung des Karrierehandbuchs entwickelt, um die Vielzahl der meist sieben Zeilen langen Artikel auf gut wartbare Art und Weise verwalten zu können. Dabei sollte ein einzelner Datenbankeintrag so wenig HTML wie möglich enthalten.

Da die meisten dieser Artikel aus einfachen, einheitlich formatierten Fließtext bestehen lag der Gedanke nahe, sie auch als Fließtext in die Datenbank aufzunehmen und ihre HTML-Formatierung zu automatisieren. Diese Aufgabe übernehmen die diversen Content Converter.

Die Content Converter koexistieren friedlich mit den Aliasen. Unabhängig davon ob und falls ja welchen Converter Sie verwenden stehen Ihnen sämtlich Aliase zur Verfügung und umgekehrt.

### 3.2.1 Karrierehandbuch Light mit HTML

Die Content Converter wie auch die Aliase haben das Karrierehandbuch wie es sich heute präsentiert überhaupt erst möglich gemacht. Ich empfehle Ihnen zumindest einen Blick auf den Converter Standard im Abschnitt 3.2.3 und auf den Abschnitt 8 mit den alltäglichsten Aliasen zu werfen.

Grundsätzlich können Sie auf diese Hilfsmittel aber auch komplett oder teilweise verzichten!

Wenn Sie im Formular *Inhalte bearbeiten* CConv auf NULL setzen oder gar nicht erst befüllen wird der Eintrag des Feldes **Content** als plain HTML genauso ausgegeben wie Sie ihn eingeben. Lediglich Aliase werden vor der Darstellung ggf. interpretiert.

Aber auch bei Verwendung von Content Convertern können Sie HTML einsetzen, müssen dabei aber die Regeln beachten nach denen der Converter den Content nachbearbeitet. Da die Converter zeilenbasiert arbeiten können Sie jedoch die weitaus meisten Probleme dieser Art vermeiden, indem Sie einzelne HTML-Sequenzen nicht über mehrere Eingabezeilen ausdehnen. Das führt mitunter zu recht langen Zeilen, hält den Code aber auch über die Konvertierung hinaus fehlerfrei.

Sehr komfortabel können Sie Inhalte mit dem integrierten TinyMCE-Editor erstellen. Falls Sie sich dazu entschließen, sollten Sie den Content Converter **tinymce** verwenden<sup>3</sup>.

### 3.2.2 Allgemeingültiges zu den Konvertern

#### Optional: Die Parameterzeile

Jeder Content kann in der ersten Zeile mit der Zeichenfolge PAR%% begonnen werden. Damit wird dem Parser mitgeteilt, dass die erste Zeile keinen Inhalt

---

<sup>3</sup>siehe Abschnitt 3.2.5

sondern Parameter für diesen enthält. Unterstützt werden bislang folgende Parameter:

- **GOTONEXT**: Beendet den Artikel mit einem GOTONEXT-Pfeil (siehe auch unter dem gleichnamigen Alias) auf den 'nachfolgenden' Artikel. Als 'nachfolgend' zählt in absteigender Priorität
  - Der in den (optionalen) Parametern angegebene Zielartikel: Es ist möglich einen String mit Parametern für das Alias GOTONEXT zu übergeben. Z.B.: GOTONEXT('n; ;15').
  - Der erste Tochterartikel des aktiven Artikels.
  - Der in der Kid-Liste des Parentknoten des aktuellen Artikels dem aktuellen Artikel nachfolgende Knoten.
  - Der in der Kid-Liste des Parentknotens des Parentknotens des aktuellen Artikels dem aktuellen Artikel nachfolgende Knoten.
  - ...
  - Falls kein nachfolgender Knoten gefunden wird hat das Parameterzeilenkommando keinen Effekt.
- **NOTITLE**: unterdrückt die Darstellung der Überschrift.
- **NOGOTOTOP**: unterdrückt ggf. die Darstellung eines am Ende aufgrund der Artikellänge automatisch eingefügten Pfeil-Links „Zum Seitenanfang“.
- **SRS('<text>')**: verwendet den durch <text> erzeugten HTML-Code als *Second Return Symbol*. In <text> eingesetzte Aliase werden interpretiert.
- **TRS('<text>')**: verwendet den durch <text> erzeugten HTML-Code als *Third Return Symbol*. In <text> eingesetzte Aliase werden interpretiert.
- **CSS('<fname>')**: Diese Option ist bislang stark auf die Bedürfnisse der Stamm-Site ausgerichtet. Sie bindet ein zusätzliches Stylesheet ein. <fname> ist der Dateiname. Das CSS selbst muss im Pfad der Style-Sheets des Karrierehandbuchs abgelegt sein. Diese Option kann auch mehrfach verwendet werden.
- **STYLE('<css-Code>')**: Der in <css-Code> eingegebene Text wird im Header der erzeugten HTML-Seite eingebunden.

### Ein paar Worte für die technisch interessierten

Die Programmierstrategie hinter den Konvertern ist objektorientiert. Am Anfang steht die Klasse `htDisplay` die eine Vielzahl der immer wieder auftau chenden allgemeinen Aufgaben zu lösen vermag. Dazu gehören beispielsweise

das sachgemäße Einbinden von Style Sheets, die Interpretation der Aliase, die Erzeugung von Überschriften im Karriere-Stil, die Verwaltung von Visitenkarten, oder auch der Import von Inhalten aus der Datenbank.

Etwas spezieller ist die Formatierung dieser Inhalte. `htDisplay` verwendet drei Methoden, um Inhalte und Visitenkarten darzustellen oder (nur eine Dummy-Methode) Inhaltsverzeichnisse zu erstellen. *Darstellen* heißt dabei soviel wie „Interpretiere ggf. Aliase, stelle das Ergebnis bereit und gut ist“. Dieses Verhalten erhalten Sie, wenn Sie im Formular *Inhalte bearbeiten* das Feld `CConv` leer lassen.

Wenn Sie in `CConv` nun allerdings ein beliebiges Wort eintragen, z.B. `standard`, wird Karriere-CMS nicht mit `htDisplay` arbeiten, sondern (sofern implementiert) versuchen die Klasse `standardDisp` zu benutzen. `standardDisp` ist von `htDisplay` abgeleitet und erbt alle Methoden dieser Urklasse. Lediglich die drei oben erwähnten Darstellungsmethoden werden überschrieben und bei Bedarf um weitere Methoden ergänzt.

Die bisherigen Bedürfnisse des Karrierehandbuchs werden durch die drei bislang existierenden Ableitungen `standardDisp`, `twocolDisp` und `tabularDisp` abgedeckt und demzufolge können als `CConv` bislang lediglich die drei zugehörigen Stile gewählt werden. Weitere Ableitungen sind in Zukunft jedoch denkbar.

### 3.2.3 Der Content Converter STANDARD

Um diesen Stil zu verwenden wählen Sie `CConv` als `standard`.

Der Standardkonverter ist das Arbeitstier des Karrierehandbuchs. Per Default verwendet er das Style Sheet

`~/Styles/KarriereCSS/standardCon.css`

Außerdem erzeugt er die Standardinhaltsverzeichnisse: Um ein Inhaltsverzeichnis des an einen Knoten angehängten Artikelbaumes zu erzeugen wählen Sie als Content Converter `standard` und lassen das Feld `Content` einfach leer oder befüllen es mit dem Wort `NULl`.

In seiner Hauptaufgabe ist der Standardkonverter für einfache Artikel und Listen entworfen worden. Er formatiert die Überschrift im Design des Karrierehandbuchs und legt sie wie in Abschnitt 3.1 unter `returnUrl` beschrieben als Link an.

Wenn Sie den `Content` zeilenweise wie in einem Texteditor eingeben setzt er jede einzelne Zeile in eine Umgebung

```
<div class='contentline'> ... </div>
```

Dieses div ist in `standardCon.css` wie folgt erklärt:

```
div.contentline {
    margin-top : 8px;
    margin-left: 25px;
}
```

Ein Beispiel für einen sehr einfachen Inhalt ist folgender Artikel



Abbildung 9: Ein einfacher Standard-Inhalt

Der zugehörige Formulareintrag hat die Gestalt  
Bemerkungen:

- Das Fazit wird mit Hilfe des Bold-Face-Alias fett gesetzt: {{bfFazit:}}
- Dass eine Visitenkarte der Firma SEITZ angezeigt wird hat mit dem Inhalt des Artikels nichts zu tun. Visitenkarten werden bislang nur von Standard-Inhalten unterstützt, werden aber nicht über die Content-Tabellen verteilt. Eine ausführliche Dokumentation zu diesem Thema finden Sie im Abschnitt refTODO.

### Checklisten

Viele Inhalte des Stamm-Karrierehandbuchs werden in Listenform präsentiert. Das betrifft insbesondere den Hauptpunkt *Checklisten*.

Die Darstellungsklasse `standard` unterstützt die Erstellung einfacher Listen im LATEX-Stil. Mit einer Zeile der Form

`CL<Parameterzeichen>`

wird eine Checkliste begonnen und mit einer Zeile der Form

`CLe`

<b>Inhalt bearbeiten</b>	
Title	Ihr Fundament tragt Sie auch durch Krisen
Content	<p>Sein Bau einer Brücke zweifelt wohl niemand daran, dass das Fundament das Wichtigste ist.</p> <p>Aber auch eine Karriere verlangt ein festes Fundament, das auch Stürme überstehen kann.</p> <p>Manche Menschen spült ihr Glück im wahrsten Wort Sinn ohne eigenes Zutun nach oben.</p> <p>Wenn diese Phase einmal zu Ende geht, reißt sie die Flut genauso schnell wieder nach unten.</p> <p>Nur wer seine Karriere auf einen festen Fundament startet, wird auch dauerhaft bestehen.</p> <p>Ihr Fundament sollte ein solides Wissen in wenigstens einem betrieblichen Fachgebiet sein.</p> <p>((btFootnote:)) Auch Unternehmerkinder erben mit ihren Genen selten alle erforderlichen Fähigkeiten !</p>
CConv	standard
ReturnUrl	
SecRetId	
toc_Status	
pdf-Meta	
Modified 18.03.2008, 15:10:14 by *****	

Abbildung 10: Ein einfacher Standard-Inhalt

wird sie wieder beendet.

Zwischen diesen beiden Schaltzeilen kann normaler Fließtext stehen, dessen Zeilen aber nicht in Divs der Klasse `contentline` sondern in Divs der Klasse `checklistline` gestellt werden.

Die Ausnahme bilden Zeilen die mit der Zeichenfolge `CLI%%` beginnen. Diese werden in ein Div der Klasse `checklistitem` gesetzt. Außerdem wird der Zeile per Default eine bei 1 beginnende fortlaufende Nummer vorangestellt.

In `standardCon.css` werden beide Klassen definiert:

```
div.checklistitem { /* Checklistenzeile: Fett. */
    margin-top : 8px;
    margin-left: 25px;
    font-weight: bolder;
}

div.checklistline { /* Checklistenzeile: Eingerueckt. */
    margin-top  : 8px;
    margin-left : 30px;
    padding-left: 3ex;
}
```

## Checklisten formatieren

- Dem Marker `CL` können bei Bedarf die Kleinbuchstaben `m` und `i` angehängt werden.

`m` steht für *mute list*. Per Default sorgt die Anwesenheit einer Checkliste im Artikel dafür, dass die Überschrift um das Präfix *Checkliste* erweitert wird. Ist das unerwünscht müssen alle Checklisten im Artikel mit der Option `m` begonnen werden.

`i` steht für *indented*. Ist diese Option ausgewählt wird für die normalen Zeilen in der Checkliste anstelle der Klasse `checklistline` die Klasse `extraIndentLine` verwendet, für die `margin-left: 40px` gesetzt ist. Falls diese Möglichkeit nicht ausreicht, empfehle ich Ihnen die Verwendung des im Abschnitt 3.2.2 beschriebenen Artikelparameters `STYLE('...')`.

Eine „stumme“, weiter eingerückte Checkliste beginnen Sie also mit der Zeile

`CLmi`

- Wie oben beschrieben werden die mit `CLI%%` beginnenden Zeilen fortlaufend nummeriert und als Item-Zeilen dargestellt. Diese Nummerierung können Sie beeinflussen indem Sie wie folgt einen Parameter übergeben:

`CLI%%<Parameter>%<Titel des Items>`

Dieser Parameter kann sein:

- Eine ganze Zahl. In diesem Fall wird der Counter für die fortlaufende Nummer auf diesen Wert gesetzt und die Zeile dann ausgegeben.
- Eine beliebige sonstige Zeichenfolge außer dem Symbol „>“. Diese Zeichenfolge wird dann als Listenmarker verwendet. Der Item-Counter wird in diesem Fall nicht erhöht.
- Das Symbol „>“ wird durch das Alias *wedge right yellow*, `{}{WRY}`, ersetzt. Dies ist ein Zugeständnis an die Tatsache, dass fast alle nicht durchnummerierten Listen innerhalb des Stamm-Karrierehandbuchs ihre Items mit einem gelben, nach rechts weisenden Keil hervorheben.

### Bereiche die nicht der Klasse `contentline` angehören sollen

Eine `contentline` hält nach oben 8, nach links 25 Pixel Abstand. Das ist nicht immer erwünscht. Daher gibt es Möglichkeiten die Klasse einzelner Zeilen oder auch ganzer Textbereiche zu verändern.

- Eine einzelne Zeile können Sie einer beliebigen Klasse zuordnen indem Sie ihr die Zeichenfolge  
`STD<Klassenbezeichner>%%`  
voranstellen. Soll zum Beispiel eine bestimmte Zeile der Klasse `center`

angehören so können Sie diese Zeile beginnen mit  
STDcenter%%

- Wenn Sie einen ganzen Block von Zeilen einer anderen Klasse angehören lassen wollen können Sie auch folgende Konstruktion verwenden:

BEGIN\_<Klassenbezeichner>

...

END

- Eine Sonderrolle nimmt die Umgebung

BEGIN\_NOSTD

...

END

ein. Ihre Inhalte werden überhaupt nicht in Divs gesetzt sondern nach Interpretation der Aliase direkt ausgegeben. Prinzipiell können Sie in dieser Umgebung ganz normales HTML einsetzen.

In `standardCon.css` sind bereits eine Reihe von alternativen Klassen vorgesehen:

```
div.center {  
    margin-top : 8px;  
    text-align : center;  
}  
div.right {  
    margin-top : 8px;  
    margin-right: 25px;  
    text-align : right;  
}  
div.packed {  
    margin-left: 25px;  
}  
div.packedcenter {  
    text-align : center;  
}  
div.packedright {  
    margin-right: 25px;  
    text-align : right;  
}
```

### Beispiel einer Checkliste aus dem Stamm-Karrierehandbuch

Die unten abgebildete Checkliste wird durch einen Datenbankeintrag erzeugt für den als `title` die Worte `thing big`, als Content Converter `standard`, als `SecRetCid` die cid 70610 gesetzt und für den pdf-Metadaten angegeben sind.

Der Content selbst lautet<sup>4</sup>

```
CL
CLI%%Sie wollen Ihren Erfolg
und das mit allen Phasen Ihres Herzens.
CLI%%Sie sind bereit, etwas zu riskieren,
selbst wenn Sie hierbei scheitern sollten.

...
CLI%%Sie sind in der Lage,
sich und andere zu organisieren.
CLI%%Sie verfuegen ueber die Geduld
die noetig ist, um auch grosse Ziele zu erreichen.
STDcenter%{{bfUnd Sie verlieren Ihr Ziel niemals aus den Augen!}}
CLe
```

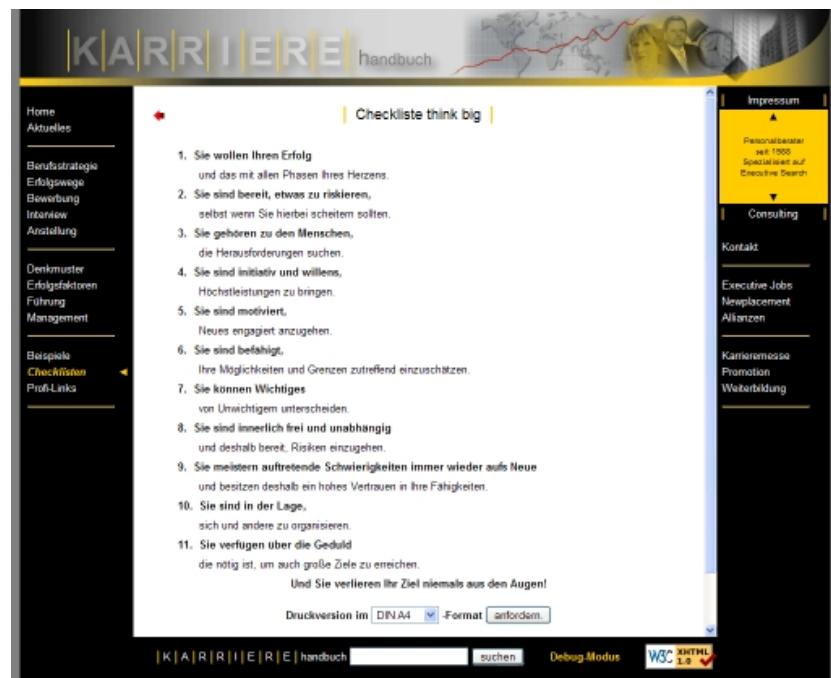


Abbildung 11: Die Checkliste im Browser

<sup>4</sup>deutsche Sonderzeichen wurden hier ersetzt, da nicht vom Font unterstützt.

### 3.2.4 Der Content Converter TABULAR

Es ist durchaus wahrscheinlich, dass STANDARD und TABULAR eines Tages zu einem einzelnen Konverter verschmolzen werden.

TABULAR bietet bereits heute alle Funktionen des Standardkonverters bis auf die Darstellung von Visitenkarten und das Angebot einer PDF-Erzeugungsleiste.

#### Tabellenumgebung

Eine Tabelle steht in einer Umgebung der Form

```
BEGIN_TABULAR
...
END_TABULAR
```

Sollte der gesamte Artikel kein BEGIN\_TABULAR enthalten gehen das Programm davon aus, dass der Artikel lediglich aus einer großen, zweispaltigen, default-formatierten Tabelle besteht.

Text der nicht in einer Tabellenumgebung steht wird wie ein STANDARD-Content interpretiert.

#### Defaulttabellen

Die Defaulttabelle geht über die Seitenbreite, hat zwei gleichbreite Spalten und keinen sichtbaren Rahmen.

Der Quellcode der Tabelle wird zeilenweise ausgewertet wobei die linke und rechte Spalte abwechselnd gefüllt werden.

Sonderregel: Bei *zweispaltigen Tabellen* kann eine linke Spalte mit dem Wort COMBINE beginnen. In diesem Fall erstreckt sich die linke Spalte in dieser Zeile über die gesamte Tabellenbreite und ihr Inhalt wird zentriert. Die daraufliegende Zeile im Quellcode bezieht sich direkt auf die linke Spalte der nachfolgenden Tabellenzeile.

Ein Beispiel: Die Zeilen

```
BEGIN_TABULAR
COMBINEDauerbrenner
H. Groenemeyer
4630 Bochum
Tim Curry
Sweet Transvestite
The Doors
Crystal Ship
END_TABULAR
```

Erzeugen die Tabelle in Abbildung 12.

H. Groenemeyer	Dauerbrenner
Tim Curry	4630 Bochum
The Doors	Sweet Transvestite Crystal Ship

Abbildung 12: Eine einfache Tabelle

## Anangepasste Tabellen

Eine Tabellenumgebung kann auch mit einer Zeile der Form

```
BEGIN_TABULAR%%<Parameterliste>
```

begonnen werden. Diese <Parameterliste> kann sich aus einer Reihe von Anweisungen zusammensetzen und recht lang ausfallen. Hier sind die Möglichkeiten.

- **border('<Parameter>')**: Schreibt `border='<Parameter>'` in das Tabelleneröffnungs-Tag. Damit können sichtbare Rahmen gesetzt werden.
- **div('<Klassenname>')**: Die gesamte Tabelle wird zusätzlich in eine Umgebung
 

```
<div class='<Klassenname>'> ... </div>
```

 gesetzt.
- **multicol('<n>')**: <n> steht für eine natürliche Zahl die die Anzahl der Spalten in der Tabelle angibt.
- **table-style('<css-Code>')**: Bewirkt, dass das eröffnungstag `<table>` der Tabelle um den Ausdruck
 

```
style='<css-Code>'
```

 erweitert wird. Auf dieselbe Weise funktionieren die Kommandos `tr-style('...')` für die Defaultzeile, sowie `td0-style('...')`, `td1-style('...')`, `td2-style('...')`, ... für die einzelnen Spalten.
- **trClasses('<n>')**: Für  $n > 0$ . So viele unterschiedliche Zeilenklassen wechseln sich ab. Die Klassen heißen `tr<1fdNr>` beginnend bei 0. Default für n ist 2 mit dem Wechsel `tr0 - tr1 - tr0 - ...`  
 Sonderfall: Statt einer natürlichen Zahl kann auch `eo` übergeben werden. In diesem Fall wird zwischen `class='even'` und `class='odd'` gewechselt. Diese beiden Klassen werden in `common.css` wie folgt erklärt:

```
tr.even, tr.even0 { /* Eher dunkler Farbton */
  background-color: #CAE1FF; /* #87CEFA; */
}
```

```
tr.odd, tr.odd0 { /* Eher heller Farbton */
```

```
background-color: #BFEFFF; /* #AFEEEE; */
}
```

Das bedeutet, dass Sie mit `trClasses('eo')` eine Tabelle erzeugen können, deren Hintergrundfarbe sich von Zeile zu Zeile abwechselt. Vgl. hierzu auch das Zeilenkommando **NEXT-TR-CLASS**

### Steuerzeilen

- Eine Zeile

`NEXT_TR_CLASS%<Klassenname>`

sorgt dafür, dass die nächste Tabellenzeile die Klasse `<Klassenname>` bekommt. Auf die Klassen aller anderen Zeilen hat das keinen Einfluss. Insbesondere kann anstelle eines Klassennamens eine natürliche Zahl übergeben werden. Dann wird der reguläre Klassenname der betroffenen Zeile um diese Zahl erweitert. Diese Option wurde mit Blick auf den weiter oben beschriebenen Wechsel `even ↔ odd` eingeführt. In `common.css` sind weitere Farben definiert:

```
tr.even1 { background-color: #eecbad; } /* Dunkler Orange-Ton */
tr.odd1 { background-color: #ffdab9; } /* Heller Orange-Ton */
tr.even2 { background-color: #9acc46; } /* Dunkler Gruen -Ton */
tr.odd2 { background-color: #bcee68; } /* Heller Gruen -Ton */
```

Ferner können Sie mit Hilfe des unter 3.2.2 beschriebenen Parameters `STYLE('...')` Ihre Zeilenklassen auch beliebig gestalten.

- Eine Zeile

`NEXT_TR_STYLE%<css-Anweisungen>`

Verwendet für das nachfolgende `<tr>`-Tag die übergebenen Styleanweisungen.

Zu dem Zeitpunkt, zu dem diese Dokumentation geschrieben wird, lassen sich `NEXT_TR_STYLE` und `NEXT_TR_CLASS` nicht gemeinsam verwenden.

Beispiel: Der folgende Code<sup>5</sup> erzeugt den Artikel in Abbildung 13.

```
1 PAR%%NOGOTOTOP NOTITLE
  STYLE('span.space {letter-spacing: 0.5ex; font-size: larger}
        table a.clt { text-decoration: none; }')
2 STDcenter%{{IMGallianzen.gif}}
```

---

<sup>5</sup>deutsche Sonderzeichen wurden hier ersetzt, da nicht vom Font unterstützt.

```

3 BEGIN_TABULAR%%trClasses('eo')
4   table-style('font-weight: bolder;') td0-style('width:24%')
5   {{WRY}} Baden-Wuerttemberg
6   {{WRY}} {{CLT;2601#mami}}Management + Mitarbeiter{{ae}}
7   {{WRY}} Bayern
8   {{WRY}} {{CLT;2601#arin}}Arbeitsratgeber - Informationen aus
      dem Arbeitsleben{{ae}}
9   {{WRY}} Nordrhein-Westfalen
10  {{WRY}} {{CLT;2601#stpa}}Dr. Strunk & Partner{{ae}}
    {{WRY}} {{CLT;2601#wkc}}Weiser, Kuck & Comp.{{ae}}
11   
12   
13 COMBINE{{WRY}}{D}<span class='space'>{{CLT;2601;;toc}}</span>
14 NEXT_TR_CLASS%%1
15 COMBINE{{WRY}} {{CLT;260101;;toc}}
16 END_TABULAR

```

Dabei ist das richtige Setzen der Zeilenumbrüche entscheidend! Zur Verdeutlichung sind obigem Code-Beispiel Zeilennummern beigefügt.



Abbildung 13: Ein Tab-Artikel aus dem Karrierehandbuch

### 3.2.5 Der Content Converter TINYMCE

TinyMCE ist ein freies WYSIWYG<sup>6</sup>-Programm von MOXIECODE SYSTEMS<sup>7</sup>, welches JavaScript verwendet, um ein normales Formulartextfeld in einen Rich Text Editor umzuwandeln. Das Arbeiten mit dem Editor fühlt sich an, wie Arbeiten unter Microsoft Word. Die Umsetzung Ihres Dokuments in HTML übernimmt das Programm für Sie.

<sup>6</sup>What you see is what you get

<sup>7</sup>siehe [HTTP://TINYMCE.MOXIECODE.COM/](http://TINYMCE.MOXIECODE.COM/)

Unter dem Inhaltsfeld im Hauptformular „Inhalte bearbeiten“ finden Sie einen Link, mit dessen Hilfe Sie das Textfeld *Content* in einen TinyMCE-Editor umwandeln können. Das Problem dabei ist, dass Inhalte im Karrierehandbuch in der Regel nicht im HTML-Format abgelegt werden und der Output von TinyMCE in der Folge nur sehr bedingt mit den meisten Content Convertern kompatibel ist.

Für diesen Fall ist der Content Converter **tinymce** gedacht: **tinymce** sorgt lediglich dafür, dass ein mit dem TinyMCE-Editor erstellter Inhalt in eine geeignete Umgebung eingepasst und mit einer Überschrift versehen wird.

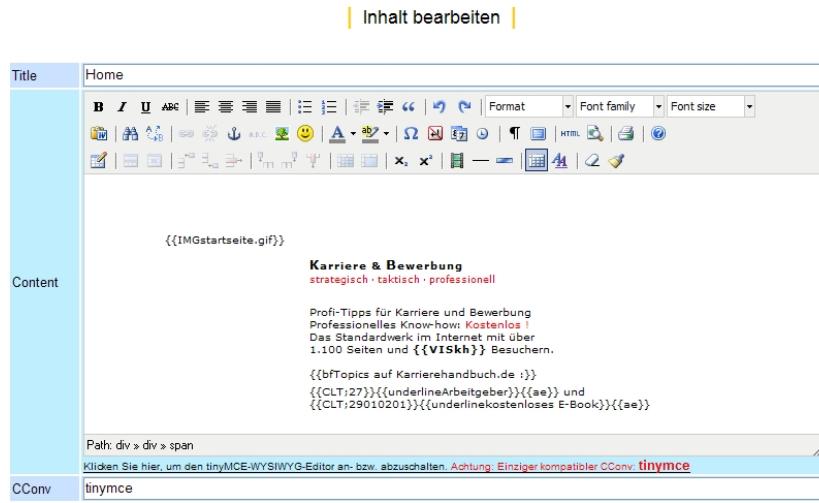


Abbildung 14: Der TinyMCE-Editor

### 3.2.6 Der Content Converter **twocol**

Hauptsächlich für die Profilinks wurde der Content Converter **twocol** entworfen. Er bietet sowohl einen eigenen Stil für Inhaltsverzeichnisse als auch einen Stil für mit Hilfe von CSS erzeugte, zweispaltige Artikel an.

#### Inhaltsverzeichnisse mit **twocol**

Die Methode **twoCol->tocTpl()** ist scharf auf die Bedürfnisse der Profilink-Listen zugeschnitten. Der Algorithmus geht nur einen Knoten in die Tiefe und fügt ihn dem entstehenden Verzeichnis im Stile eines Checklist-Items hinzu. Außerdem wird der Content des Knotens wie folgt analysiert:

- Es werden alle Einträge **{{AN}}** gesucht. Ein solches, parameterloses Anker-Alias erzeugt lediglich einen leeren Link, welcher als **name** eine bei 1 beginnende, fortlaufende Nummer trägt.
- Aus dieser Information, sowie dem Textinhalt der zugehörigen Zeile wird ein Unterpunkt des Inhaltsverzeichnisses angelegt der auf die entsprechende Stelle im Ziel-Content verweist.

### Artikel im TWOCOL-Stil

Wie bei der Klasse `tabular` wird der Artikel wie ein Artikel im `standard`-Stil interpretiert bis der Parser auf eine Umgebung

```
BEGIN_FLOAT
...
END_FLOAT
```

stößt – sollte keine solche Umgebung vorkommen, wird der gesamte Artikel als `FLOAT` aufgefasst.

Die Gestalt eines `FLOAT`-Bereichs ist die einer zweispaltige Tabelle. Seinem Wesen nach ist er aber ein Stapel von Divs die abwechselnd der Klasse `colleft` und der Klasse `colright` angehören. In `twocolDisp.css` sind sie wie folgt erklärt:

```
div.colleft {
    margin-top: 8px;
    clear: left;
    float: left;
    min-width: 5%;
}

div.colright {
    margin-top: 8px;
}
```

Die `float`-Eigenschaft erlaubt das Vorkommen mehrerer Blockelemente in einer Zeile.

### Die CHECK-Line

Die Letzte Zeile des Artikels kann mit dem Wort `CHECK` begonnen werden. In diesem Fall wird der Rest der Zeile in ein Div der Klasse `checkline` gesetzt:

```
div.checkline {
    margin-top: 20px;
    font-size: smaller;
}
```

Damit kann man den Artikel z.B. mit einem klein gesetzten Datum unterschreiben und so den letzten Zeitpunkt angeben, an dem alle Links der betreffenden Profi-Link-Liste überprüft wurden.

### Ein Beispiel für eine Profilinkliste

Da der ganze Artikel eine ununterbrochene `twocol`-Liste ist, muss keine Umgebung `FLOAT` gesetzt werden.

```

1 {{WRY}}
2 {{AN}} {{bfAuswahlverfahren}}
3
4 {{D}}{{AN*http://www.arbeitskreis-ac.de/}}Arbeitskreis
Assessment Center{{ae}} - Forum fuer Personalauswahl und
Entwicklung
5
6 {{D}}{{AN*http://www.personalpsychologie.de/}}S &
F{{ae}} - die Personalpsychologie GmbH des Prof. Schuler
7
8 {{D}}{{AN*http://www.psychologie.uni-wuerzburg.de/fips/
skripten/neu/grund/tuf/tuf_lewand-skript_hm_2004.pdf}}}
Testverfahren{{ae}} - eine Info (PDF) der Uni Wuerzburg
9 {{WRY}}
10 {{AN}} {{bfEntgeltfindung}}
11 {{AN;gesp}}{{CLT;613}}{{ALR}}{{ae}}
12 {{D}}{{AN*http://www.lohnspiegel.de/}}Gehaltsspiegel{{ae}}
- zum Mitmachen und Mitprofitieren - unterstuetzt von ver.di
13
14 {{D}}{{AN*http://www.boeckler.de/}}Hans-Boeckler-Stiftung{{ae}}
- mit Tarifloehnen und -gehaeltern sowie einem Gehaltsspiegel
15
16 {{D}}{{AN*http://www.sueddeutsche.de/jobkarriere/erfolggeld/
special/126/44082/}}Sueddeutsche{{ae}} - Gehaltsangaben in
der Sueddeutschen Zeitung

```

Auch hier ist das richtige Setzen der Zeilenumbrüche entscheidend. Zur Ver deutlichung wurden der Quellcode um Zeilennummern ergänzt. Grundsätzlich beziehen sich die ungeraden Zeilen auf die schmale, höchstens ein Symbol enthaltende linke und die geraden Zeilen auf die rechte „Spalte“.

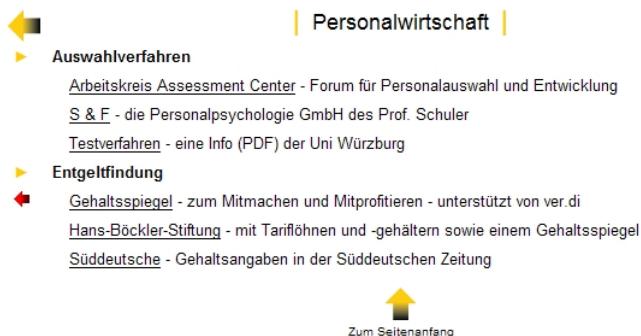
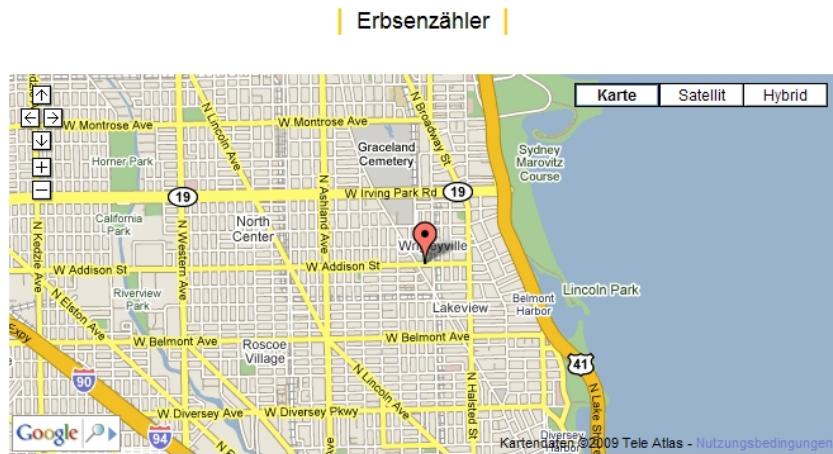


Abbildung 15: Eine Div-Folge als zweispaltige Tabelle

## 4 Jobs – Stellen ausschreiben

Auf Karrierehandbuch.de können Sie auch freie Positionen in Ihrem Unternehmen online ausschreiben.



Für unsere hochmoderne Buchhaltung suchen wir einen amtlichen Erbsenzähler,  
der mindestens bis 1000 zu zählen vermag...

<b>Job-Typ</b>	Niederqualifizierte Tätigkeit
<b>Deadline</b>	01.07.2009
<b>Gehalt</b>	12 1000€
<b>Unternehmen</b>	Graduell Inc.
<b>Url</b>	<a href="http://www.karrierehandbuch.de">http://www.karrierehandbuch.de</a>
<b>Anschrift</b>	1060 W Addison St Chicago IL 60613 United States

Abbildung 16: Das Formular „Ein automatisch formatiertes Stellenangebot“

Dazu loggen Sie sich im Redaktionssystem ein und wählen auf der linken Seite das Register *Jobs* aus.

Jobs sind auf den Seiten des Karrierehandbuchs sprachabhängig. Das bedeutet, dass Sie für jede Sprache eine eigene Job-Tabelle führen können. Diese wird bei Erzeugung des ersten Stellenangebotes in der ausgewählten Sprache automatisch angelegt.



Abbildung 17: Das Formular „Das Menü Jobs“

## 4.1 Stellenangebote ausschreiben oder bearbeiten

Neu erstellte Jobangebote erhalten automatisch eine **id**. Außerdem wird der Einstellzeitpunkt **inserttime** erfasst. Anhand des Einstellzeitpunkts werden später die Stellenangebote in automatisch erzeugten {{JOBLIST..}}-Listen sortiert dargestellt: Die neuesten Angebote zuerst, die ältesten zuletzt.

Bezeichnung	Financial Analyst (m/w)
Job-Typ	Analyst
Beschreibung	Für unser (em Risk Controlling) suchen wir...
Deadline**	01-05-2009 00:00:00
Gehalt	45000
Währung	€
Unternehmen*	Finanz-Bank
Rechtsform*	AG
Anschrift*	Nordbahnhof 2 70144 Stuttgart
PLZ	70144
Url*	<a href="http://www.bank-o-mat.de/">http://www.bank-o-mat.de/</a>
E-Mail*	info@fb.de
Telefon*	
FAX*	NONE
Modified 25.03.2009, 10:23:41 by kochmn	
Kommentar	

Abbildung 18: Das Formular „Das Hauptformular für die Jobeintragung“

Um einen neuen Job anzulegen benötigen Sie das Recht 'c'. Für die Bearbeitung eines bereits bestehenden Angebots genügt 'm'.

Der Dialog für die Jobauswahl zeigt neben der Bezeichnung des Angebots auch die **id** an.

Die Felder im Einzelnen:

- **Bezeichnung:** Name der Position. Gewissermaßen die Überschrift der Stellenausschreibung. Das Feld akzeptiert Aliase.
- **Job-Typ:** Für wen ist die Stelle gedacht? Z.B. „Elektroingenieur (m/w)“.
- **Beschreibung:** Ausführliche Beschreibung der Stelle. Das Feld akzeptiert Aliase.
- **Deadline:** Datum, bis zu dem die Stelle besetzt sein muss. Das Alias {{JOBLIST..}} stellt nur Stellen dar, deren Deadline noch nicht erreicht ist. Default ist ein 90-tägiger Zeitraum.
- **Gehalt:** Jährliche Vergütung.
- **Waehrung:** Währung der Vergütung. Z.B. €.
- **Unternehmen:** Namen des die Stelle ausschreibenden Unternehmens.

- **Rechtsform:** Rechtsform des Unternehmens.
- **Anschrift:** Postalische Adresse der Stelle.
- **PLZ:** Nocheinmal die Postleitzahl der Adresse der Stelle.
- **Url, E-Mail, Telefon, FAX:** Selbsterklärend.
- **Kommentar:** Admin-Kommentar, der ansonsten keine Bedeutung hat.

Die im Formular mit einem \* gekennzeichneten Felder können auch komplett frei gelassen werden. Das Programm holt sich diese Informationen später aus den Daten des Unternehmens, dem die Job-Tabelle zugeordnet ist. Falls Sie das nicht wollen, können Sie explizit das Wort **NONE** in die betreffenden Felder eintragen.

## 4.2 Stellenangebote entfernen

Diese Handlung erfordert das Recht 'c'. Sie ermöglicht das restlose Entfernen von Jobangeboten aus der Jobtabelle und lässt sich nicht rückgängig machen.

## 5 Dateien

Im Verzeichnis *Dateien* gibt es bislang nur zwei Handlungsstränge: *Grafik-Dateien verwalten* und *Download-Dateien verwalten*. Beide funktionieren identisch und erfordern das Recht **w** (**write**) auf dem bearbeiteten Client.

Ein Client im Karrierehandbuch hat per Default eine Gesamt-Quota von 30 MB für Grafiken und Dateien, die er auf unserem Server hinterlegen kann. Insbesondere die Aliase `{{IMG..}}` und `{{FILE..}}` greifen auf diese besonderen Verzeichnisse zu.

### 5.1 Download- oder Grafik-Dateien verwalten

Die Grafik zeigt das Formular zur Verwaltung von Grafikdateien. Das Formular für die Verwaltung von Downloads ist identisch dazu.

Abbildung 19: Das Formular „Grafikdateien verwalten“

Das Textfeld zeigt zunächst das Stammverzeichnis der Grafikdateien des ausgewählten Clients.

Dieses Formular erfüllt mehrere Aufgaben auf einmal:

#### 5.1.1 Dateien auf den Server hochladen

1. Tragen Sie in der Textbox „Wählen Sie die Zielfile(en) aus“ die hochzuladenden Dateien samt Pfad auf Ihrem Rechner ein.
2. Geben Sie nach Wunsch das „optionale Zielunterverzeichnis“ an. Per Default wird die Datei im Stammverzeichnis des Clients abgelegt. Falls Sie ein noch nicht existierendes Unterverzeichnis angeben, wird es angelegt.

Die Verzeichnis-Syntax betreffend ist der Parser einigermaßen gutmütig. Vorzugsweise orientiert er sich aber an der Verzeichnissyntax wie sie auf

Unix-Systemen üblich ist:

Für das Stammverzeichnis des Clients schreiben Sie einfach ./ und hängen die durch Slashes / getrennten Unterverzeichnisse an.

*Beispiel:* Angenommen Sie wollen Ihre Datei in das Verzeichnis Sa/New/ eintragen. Dann geben Sie als Pfad an: ./Sa/New/

*Hinweis:* Wie die meisten Unix-Server hat auch der Server auf dem das Karrierehandbuch gehostet ist ein case-sensitives Dateisystem. Das heißt, dass das System Groß- und Kleinschreibung unterscheidet und einen Unterschied macht, ob Sie ein Verzeichnis beispielsweise `directory` oder `Directory` nennen!

3. Bestätigen Sie die Anfrage mit einem Klick auf „Abschicken“. Das Zielverzeichnis wird dann gegebenenfalls angelegt und die Dateien werden hochgespielt. Auerdem wechselt das Formular automatisch in das Zielverzeichnis.

### 5.1.2 Verzeichnisse Anlegen und Wechseln

Wenn Sie einen Pfad angeben, ohne eine hochzuladende Datei zu spezifizieren wird einfach nur das Verzeichnis gewechselt. Falls es noch nicht existiert, wird es auch angelegt.

## 5.2 Dateien und leere Verzeichnisse Löschen

Um Dateien und leere Verzeichnisse zu löschen, müssen Sie zunächst die Checkbox „Ausgewählte Dateien vom Server löschen“ aktivieren.

Das den Verzeichnisinhalt anzeigennde Fenster ist eine Kombobox in der Sie Dateien und leere Verzeichnisse mit der Maus auswählen können.

Das Programm versucht diese nach Klick auf „Abschicken“ zu löschen.

## 6 Datenbank

Das Inhaltsverzeichnis *Datenbank* enthält bislang zwei Funktionen und eine Linkliste auf außerhalb des Redaktionssystems gelagerte Skripte.

### 6.1 Liste der noch SA-freien Texte

Das Einblenden von Visitenkarten unter den Fachtexten ist eine wichtige Dienstleistung des Karrierehandbuchs, die praktisch jedem unserer Kunden zu Gute kommt.

Dabei wird unterschieden zwischen *fest vergebenen* und *freien* Fachtexten.

Ein bestimmter Fachtext gilt als fest vergebenen, wenn es eine Visitenkarte gibt, die diesen Fachtext als Ziel hat. D.h.: Immer wenn dieser Fachtext geöffnet wird, wird diese Visitenkarte dazu angezeigt.

Grundsätzlich lässt das CMS zu, dass mehrere Visitenkarten ein und denselben Fachtext als Ziel haben. Dessen ungeachtet hat jeder Client ein Recht auf „seine“ Fachtexte, unter denen exklusiv seine Visitenkarten eingeblendet werden.

Dieses Tool liefert eine Liste derjenigen Fachtexte zurück, die noch frei sind.

### 6.2 Datenbankkonsistenztest

Dieses Tool analysiert die Datenbank auf logische Inkonsistenzen.

Die Hauptfunktionen sind:

1. Überprüfung der logischen Konsistenz der Indextabellen: Existieren die Knoten, auf die verwiesen wird?
2. Stimmen die Gegenverlinkungen? D.h.: Wenn ein Knoten A auf einen Knoten B als „kid“ verweist, verweist dann auch Knoten B auf A als „parent“? Und umgekehrt?
3. Existiert für jeden Knoten zumindest ein Content in der Defaultsprache des Clients? Und umgekehrt?
4. Sind in den Inhalten `title` gesetzt? Sind gegebenenfalls die angegebenen Content Converter-Klassen definiert? Existieren die `secretcid`-Inhalte?

Die Ausgabe erfolgt tabellarisch.

```
Testergebnis für (clt=='kh',lng=='de, en, ru')

global
Zeit 2008/11/26 ,09:47:24
Gegenstand Untersucht werden Tabellen der Kunden kh in den Sprachen de, en, ru.
kh
Existenz Tabelle sa_en_kh, sa_ru_kh nicht gefunden.

Keine Fehler in diesem Teil der Datenbank gefunden. (Criticals: 0 / Errors: 0 / Warnings: 0)

Schlusswertung
Der Konsistenzcheck ergab keine Fehler!
```

Abbildung 20: Ein Datenbankkonsistenzbericht

## 6.3 Ausgelagerte Skripte

### 6.3.1 examineFiles

Dieses Tool befasst sich mit der Korrektheit der auf den Seiten eines Client auftauchenden {{CLT..}}, {{DWN..}}, {{FILE..}} und {{IMG..}}-Aliase.

An dieser Stelle sei angemerkt, dass das {{DWN..}}-Alias einem nicht mehr gepflegten Konzept angehört.

```
Dateiverlinkungstest

Mit diesem Admin-Tool werden die (falls vorhanden) Download-Tabellen sowie die Aliase DWN, FILE und IMG in den Content-Tabellen eines gegebenen Client darauf untersucht, ob die angegebenen Zielfile existieren.
Außerdem wird überprüft, ob die CLT-Aliase auf existierende Knoten verweisen, und ob es zu jeder in den Download-Tabellen bereitgestellten Datei ein diese Tatsache nutzendes DWN-Alias gibt.

Untersuche Client tesat
Überprüfung der Download-Tabellen
Überprüfung der Content-Tabellen
Keine Inkonsistenzen in content_de_tesat gefunden.
Überprüfung auf hängende Verlinkung
```

Abbildung 21: Das Tool **examineFiles**

**examineFiles** überprüft, ob

1. alle Knoten, auf die mit CLT-Aliassen verwiesen wird, tatsächlich existieren.
2. alle Downloads, auf die mit DWN-Aliassen verwiesen wird, sowohl in der zugehörigen Download-Tabelle, als auch auf dem Server an der dort angegebenen Stelle existieren.
3. Alle in den Download-Tabellen bereitgestellten Dateien an wenigstens einer Stelle in den Content-Tabellen aufgerufen werden. Achtung: Die Sa-Tabellen werden nicht untersucht!

4. Die in {{FILE..}}- und {{IMG..}}-Aliasen angegebenen Dateien tatsächlich existieren.

### 6.3.2 nodeInfo.php5

Ein winziges Tool, welches im Augenblick nur die Unterknoten eines gegebenen Knoten anzeigt und deren Töchter auszählt. Das Werkzeug kann über die GET-Parameter `id` (für die Content Id) und `clt` (für das Client Suffix) gesteuert werden.

## 7 Aliases

### 7.1 Über Aliase

Ein *Alias* ist eine Zeichenfolge die beim Kompilieren der Seite durch eine andere Zeichenfolge ersetzt wird.

Die Idee mit Aliasen zu arbeiten stammt für mich aus der Zeit an der Uni Stuttgart als JOACHIM WIPPER, JOACHIM GAUKEL und ich unter Anwendung des L<sup>A</sup>T<sub>E</sub>X-Paketes *psfrag* das Paket *mtp* programmierten. Mit Hilfe von *mtp* wurde es möglich MATLAB-Grafiken so in L<sup>A</sup>T<sub>E</sub>X-Dokumente einzubinden, dass sämtliche Beschriftungen der Grafik mit dem von L<sup>A</sup>T<sub>E</sub>X verwendeten Font in der im Dokument eingestellten Skalierung neu gesetzt wurden.

Joachim Gaukel entwickelte damals eine ganze Reihe von zusätzlichen Tools für das elegante Erstellen von Grafiken unter MATLAB und brachte dabei auch die Idee ein, Aliase als Parameter für die diversen Funktionen zu verwenden. So war es beispielsweise möglich

```
>> mtparrow(.,'red','dotted');
```

in MATLAB einzugeben, um einen roten, gepunkteten Pfeil zu erhalten. Der Aliasgedanke und seine Umsetzung trugen einiges zur Benutzerfreundlichkeit von *mtp* bei.

Als ich Anfang 2008 das Karrierehandbuch neu entwickelte und dabei das Ziel verfolgte auch innerhalb der Datenbank Struktur und Inhalt zu trennen stand ich rasch vor der Notwendigkeit, relativ dynamische Strukturinformationen in die an sich recht statischen Inhalte aufzunehmen. So schrieb ich die Tool-Methode *replaceAliases*, deren Aufgabe es war, jeden Text vor der Darstellung im Browser nach Zeichenfolgen der Form

```
%%PHPkaesebrot%%
```

abzusuchen und dann (im Beispiel) durch den Wert der zur Laufzeit im Speicher befindlichen Variablen *\$kaesebrot* zu ersetzen. Dies ermöglichte beispielsweise die Aufnahme von wartungsfreien Links in die Artikel des Karrierehandbuchs welche sich automatisch anpassten falls der Artikel auf den sie verwiesen einmal innerhalb der Strukturtabelle umziehen sollte.

Außerdem war der Grundstein gelegt beliebige Zeichenfolgen unter Verwendung des Arbeitsspeichers, der Festplatte und oder der Datenbank zu ersetzen.

Heute ist *replaceAliases* eine der zentralen Methoden der für die Darstellung aller Inhalte verantwortlichen Klassenfamilie *htDisplay* und unterstützt über 60 verschiedene Aliase die bei Bedarf auch ineinander verschachtelt werden können. Die grundsätzliche Syntax hat sich marginal gewandelt:

```
 {{<Name des Aliases><Durch ',' getrennte Parameterliste>}}
```

Die Aliase können ganz regulär in Artikeln eingegeben werden. Den Satz „Wir sind die **lustigen** Holzhackerbuam !“ könnte man aliasgestützt wie folgt darstellen:

```
 {{underline{{emWir}}} sind}} die {{bf lustigen}} Holzhackerbuam !
```

## 7.2 Konventionen

1. Die grundsätzliche Syntax lautet

```
 {{<aliasName><',';-getrennte Parameterliste>}}
```

Hierbei kann nur das letzte Argument das Zeichen ; als Teil seines Inhaltes enthalten.

2. Aliase sind *Case Sensitive*. Unmodifiziert aus LATEX oder HTML übernommene Aliase sind klein, alle anderen groß geschrieben.
3. Manche ältere Aliase erfordern ein weiteres Alias als Endtag. Dieses hat denselben Namen wie das eröffnende Alias gefolgt von einem kleinen 'e'. Z.B. {{FORM..}} und {{FORMe}}
4. Jedes Alias ist in der SQL-Tabelle `karriere.aliases` registriert. Dort liegt auch für jedes Alias eine Kurzdokumentation. Außerdem ist dort für die einfacheren Aliase Code hinterlegt. Das Alias {{smaller..}} beispielsweise findet dort die Zeichenfolge

```
<span class='smaller'>%</span>
```

und ersetzt dann nur noch % durch seinen ersten (und in diesem Fall einzigen) Parameterstring.

5. Manche Aliase legen HTML-Umgebungen an. Als `class` wird in der Regel der Name des Aliases angegeben.
6. Eine Sonderrolle nimmt das Alias {{CHR..}} ein. Es ist redundant unter `staticTool::hackSecureEncode(..)` implementiert und wird in dieser Form von der Datenbankverwaltungsklasse Kdb verwendet. Da Kdb-Objekte innerhalb der htDisplay-Objekte instanziert werden, muss Kdb selbst auf die Funktionalität von `replaceAliases(..)` verzichten. Da {{CHR..}} bei der Anti-SQL-Injection-Verschlüsselung zum Einsatz kommt ist es wünschenswert, dass Kdb dieses spezielle Alias interpretieren kann.

## 7.3 Die Aliase

### 7.3.1 ae

{{ae}} wird ersetzt durch </a>

### 7.3.2 bf

„Bold Face.“ {{bf}} wird ersetzt durch <span class='bf'>%</span>

### 7.3.3 br

„Line Break.“ {{br}} wird ersetzt durch <br />

### 7.3.4 divCenter

{{divCenter}} wird ersetzt durch <div class='centerline'>%</div>

### 7.3.5 divLine

{{divLine}} wird ersetzt durch <div class='line'>%</div>

### 7.3.6 em

„Emphasis.“ {{em}} wird ersetzt durch <span class='em'>%</span>

### 7.3.7 larger

{{larger}} wird ersetzt durch <span class='larger'>%</span>

### 7.3.8 smaller

{{smaller}} wird ersetzt durch <span class='smaller'>%</span>

### 7.3.9 underline

{{underline}} wird ersetzt durch <span class='underline'>%</span>

### 7.3.10 ADD

\verb|{{ADD[<cltSuffix>[;div]]}}|

wird ersetzt durch die Adresse des Mandanten mit dem angegebenen Client Suffix. Wird als zweiter Parameter die Zeichenfolge div angegeben, so wird die Adresse zusätzlich in <div class='ADD'>..</div> verpackt.

Siehe auch: EML, FAX, HOME, NAME, TEL

### 7.3.11 ADY

{{AUY}} ('Arrow Down Yellow') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
      {{PHPi_images_arrows}}{{PHPi_gif_arrows;dn_yellow}}
      alt='go to next' />
```

Dies bindet die Grafik eines gelben, nach unten weisenden Pfeils ein.

### 7.3.12 ALR

`{{ALR}}` ('Arrow Left Red') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;left_red}},
    alt='siehe auch' />
```

Dadurch wird die Grafik eines roten, nach links weisenden Pfeils eingebunden.

### 7.3.13 ALLVISITORS

`{{ALLVISITORS}}` wird ersetzt durch die Summe aller Besucherzähler in der Client-Tabelle.

### 7.3.14 ALY

`{{ALY}}` ('Arrow Left Yellow') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;left_yellow}},
    alt='see also' />
```

Dadurch wird die Grafik eines gelben, nach links weisenden Pfeils eingebunden.

### 7.3.15 AN

Opening Tag für einen Anker.

`{{AN[<href>; [<name>]]}}`

- `<href>`: Ziel-Url für den Anker.
- `<name>`: Name für den Anker.

Diese Version des Aliases ist für interne Links gedacht. Der Anker führt nicht dazu, das ein neues Fenster geöffnet wird.

### 7.3.16 AN\*

Variation von `{{AN}}`. Diese Version des Aliases ist für externe Links gedacht und soll eines Tages dazu führen, das bei Aufruf ein neues Fenster geöffnet wird. Dafür kenne ich im Augenblick leider noch keine W3C-XHTML1.0/STRICT standardkompatible Möglichkeit.

### 7.3.17 ARR

`{{ARR}}` ('Arrow Right Red') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;right_red}},
    alt='see also' />
```

Dadurch wird die Grafik eines roten, nach rechts weisenden Pfeils eingebunden.

### 7.3.18 ARY

`{{ARY}}` ('Arrow Right Yellow') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;right_yellow}},
    alt='see also' />
```

Dadurch wird die Grafik eines gelben, nach rechts weisenden Pfeils eingebunden.

### 7.3.19 AUY

`{{AUY}}` ('Arrow Up Yellow') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;up_yellow}},
    alt='go to top' />
```

Dadurch wird die Grafik eines gelben, nach oben weisenden Pfeils eingebunden.

### 7.3.20 AUTHCLTCOMBO

Alias für eine Combobox die in Abhängigkeit des gesetzten Redaktions-Cookies diejenigen Clients anbietet, auf die der User unter den geforderten Rechten Zugriff hat.

`{{AUTHCLTCOMBO[<reqRights>[;<name>[;<multi>[;<size>]]]]}}`

- `<reqRights>`: Welches Rechtesymbol wird benötigt? Default ist 'a'
- `<name>`: Name für die Combobox. Default ist 'authcltcombo'
- `<multi>`: Boolean. Wird an `{{COMBO..}}` weitergeleitet.
- `<size>`: Wird an `{{COMBO..}}` weitergeleitet.

Auf der technischen Seite erzeugt dieses Alias einen Hash `$authcltcombo` der auf das Array `$GLOBALS` gepusht wird. Danach wird das Alias ersetzt durch

`{{COMBOauthcltcombo;<gefilterte Argumente>}}`

### 7.3.21 AUY

`{{AUY}}` ('Arrow Up Yellow') wird ersetzt durch

```
<img class='thumb' src='{{PHPi_protocol}}{{PHPi_html}}'
    {{PHPi_images_arrows}}{{PHPi_gif_arrows;up_yellow}}
    alt='go to top' />
```

### 7.3.22 BBOOK

Ein sehr spezielles, für unser Online-Marketing wichtiges Alias. Bindet ein Thumb-Nail in form eines kleinen, roten Buches ein, welches auf den Artikel `con=29010202,clt=kh` (die Download-Seite des Bewerbungshandbuchs) verlinkt.

Genauer gesagt wird der Code ersetzt durch das in den Alias-Preferences festgelegte Code-Fragment

```
 {{CLTk;29010202}}<img class='thumb' src='{{PHPi_protocol}}
    {{PHPi_html}}{{PHPi_images_arrows}}
    {{PHPi_gif_arrows;book_b_small}}' alt='Bewerbungshandbuch' />{{ae}}
```

### 7.3.23 BROWSER

Fügt einen grafischen Strukturbrowser der Baum-Klasse ein.

Verpackt ihn in einem `<div class='browser'>..</div>`

`{{BROWSER;}}`

- `<imgSrc>`: Dateiname der aktuellen Browsergrafik wie sie von `Baum->createTree(..)` im Kurzzeit-Tmp-Verzeichnis angelegt wird.
- `<imgMap>`: Area Map-String für die Schaltflächen auf der Grafik wie er von `Baum->createTree(..)` zurückgeliefert wird.

Das Alias wird ersetzt durch

```
<div class='browser'>
<map name='chart'>..</map>
<img alt='Browser'; border='0'; usemap='#chart'; src='..' />
</div>
```

### 7.3.24 BY

`{{BY}}` „Bull Yellow“. Wird ersetzt durch

`<span class='gelb'>&bull;</span>`

und erzeugt ein gelbes Bull's Eye.

### 7.3.25 CHR

Ersetzt den als Parameter übergebenen Ascii-Code durch das zugehörige Zeichen.

```
{{CHR<(int)ord>}}
```

- <ord>:  $ord \in \{0, \dots, 255\}$ . Ordnungszahl aus der 8-Bit-ASCII-Tabelle.

*Hinweis:* Dieses Alias wird redundant von `staticTools::hackSecureEncode(...)` implementiert.

### 7.3.26 CLT

Aliaspräfix für das Opening Tag eines Anchors auf einen beliebigen Artikel innerhalb der KARRIEREhandbuch-Domain.

```
{{CLT[<(text)cltsuffix>[;<(int)cid>[;<(text)lng>[;<(text)title>
[;<(bool)plainUrl>]]]]}}}
```

- <cltsuffix>: Client-Suffix des Mandanten innerhalb dessen Sub-Site das Ziel liegt. Default ist der aktive Mandant.
- <cid>: Content id des Zielartikels. Default ist 1.
- <lng>: Kürzel der gewünschten Sprache. Per Default wird die aktuell eingestellte Sprache beibehalten.
- <title>:
  - Falls `true`, `t` oder `1` wird die ihrer Tags beraubte Überschrift des Zielartikels als Link Title verwendet. D.h.: Dieses Feature unterstützende Browser zeigen sie an, falls der Betrachter der Seite mit der Maus auf den Link zeigt.
  - Falls `false`, `f` oder `0` ist die Title-Funktion abgeschalten.
  - Falls `toc` wird ein vollständiger Table-Of-Contents-Link erzeugt. D.h.:
    - \* Es wird kein Title angezeigt.
    - \* Das Anker-Tag erhält die Eigenschaft `name='<cid>'`.
    - \* Dem fertigen Opening Tag wird als Link-Text die Überschrift des Zielartikels sowie das Closing-Tag `</a>` mitangehängt.

Die Defaulteinstellung ist `true`.

- <plainUrl>: Falls nicht leer angegeben wird das Alias nur durch die Ziel-Url ersetzt.

### 7.3.27 COMBO

Alias für eine Combobox.

```
{{{COMBO[<(text)hashName>[;<(text name='combo'>
    [;<(bool)multi>[;<(int)size>[;<(bool)selfass>
        [;<(text)selVal>[;<(text)add>]]]]]]]}}}
```

- <hashName>: Name einer globalen Variablen die einen Hash enthält. Die Schlüssel dieses Hashs stellen die **values**, die Werte die einzelnen **options** der Combobox dar.
- <name>: Name für die Combobox. Default ist 'combo'.
- <multi>: Boolean. Falls **true** wird die Box als **multichoose** definiert. Sonst nicht. Default ist 0.
- <size>: size für die Combobox. Wird nicht angegeben falls 0. Default ist 0.
- <selfAss>: Boolean. Falls **true** wird der übergebene Hash „selbst-assoziativ“. D.h. sich wiederholende Werte werden entfernt und die Anweisung **\$hash = array\_combine(\$hash,\$hash)** wird ausgeführt. Dies erlaubt die Verwendung eines normalen Arrays dessen Werte in der Combobox sowohl als **values** als auch als **options** dienen.
- <selval>: „Selected Value“. Kann einen der Schlüssel des die Optionen erzeugenden Hashs enthalten. Dieser Wert wird dann die Voreinstellung der Combo-Box. Per Default ist diese Teifunktion inaktiv.
- <add>: Zusätzlicher Code für das <select ..>-Tag

### 7.3.28 CONTACTNOW

ToDo!

### 7.3.29 CORPCOL

{{{CLTCOL<cltsuffix>}}} wird ebenso behandelt wie {{{ADD..}}}}. Bezieht sich aber auf die Standardfarbe des Zielmandanten. Zurückgeliefert wird ein String der Länge 6. Dieser codiert die Farbe hexadezimal in der Form **RRGGBB**.

### 7.3.30 D

{{{D}}} wird ersetzt durch <span class='dist' />  
Kann über CSS dazu benutzt werden, um einen horizontalen Textabstand zu erzwingen.

### 7.3.31 DATE

Wandelt einen Unix-Timestamp in ein Datum um und zeigt es an.

`{{DATE[<time>[;<shift>]]}}`

- **<time>**: Unix-Timestamp<sup>8</sup>. Default ist der Zeitpunkt zu dem die Seite angefordert wird.
- **<shift>**: Boolean. Falls `true` wird der angegebene erste Parameter `time` dem aktuellen Zeitwert aufaddiert und das Ergebnis als Datum dargestellt. Sonst nicht. Default ist 0.

`{{DATE}}` wird durch die Rückgabe von `date('d.m.Y', ...)` ersetzt.

*Siehe auch: TIME*

### 7.3.32 DD

`{{DD}}` wird ersetzt durch `<span class='dist' /><span class='dist' />` Kann über CSS dazu benutzt werden, um einen horizontalen Textabstand zu erzwingen.

### 7.3.33 DWN

Alias für das Opening-Tag eines Download-Anchors.

`{{DWN<did>[;<targetClient>[;<noTag>]]}}`

- **<did>**: Download id des gewünschten Downloads innerhalb der Download-Tabelle des aktiven Mandanten.
- **<targetClient>**: Optional. Client in dessen Download-Tabelle und unter dessen Download-Pfad die Zielfile registriert ist. Per Default der aktive Client.
- **<noTag>**: Optional. Boolsch. Falls `true` wird nur die Url auf die Zielfile zurückgeliefert. Sonst ein komplettes HTML-Anchor-Tag. Per Default `false`.

`{{DWN..}}` wird ersetzt durch einen mit `title` versehenen Anchor-Opening Tag der Form

`<a title='..' href='..'>`

Der Wert der `href` wird vorher durch eine regular Expression gefiltert.

---

<sup>8</sup>Die Unix-Zeit zählt die Sekunden seit dem 1. Januar 1970, 0.00 Uhr, UTC

### 7.3.34 DWN\*

Variation von {{DWN}}. Diese Version des Aliases ist für PDF-Dokumente gedacht, die in einem neuen Fenster geöffnet werden sollen. Dafür kenne ich im Augenblick leider noch keine W3C-XHTML1.0/STRICT standardkompatible Möglichkeit.

### 7.3.35 EML

{{EML<cltsuffix>}} wird ebenso behandelt wie {{ADD..}}. Bezieht sich aber auf die E-Mail-Adresse des Zielmandanten.

### 7.3.36 FAX

{{FAX<cltsuffix>}} wird ebenso behandelt wie {{ADD..}}. Bezieht sich aber auf die FAX-Nummer des Zielmandanten.

### 7.3.37 FILE

Vereinfachte Version des Aliases {{DWN..}}. {{FILE..}} kommt ohne Download-Tabelle aus und ist für einmalig auftretende Gelegenheitsdownloads wie die Druckversion eines Artikels als statisches PDF-Dokument gedacht.

`{{FILE<fname>[;<cltSuffix>[;<title>]]}}`

- <fname>: Unterpfad und Dateiname der Zielfile im Unix-Format, wo bei direkt mit dem ersten Pfadnamen oder dem Dateinamen begonnen wird. Also `Unterpfad/text.pdf` anstelle von `./Unterpfad/text.pdf`. Außerdem wird davon ausgegangen, dass der Startpfad das Download-Stammverzeichnis des unter `cltSuffix` angegebenen Mandanten ist.
- <cltsuffix>: Kürzel des Mandanten in dessen Download-Pfad das Ziel liegt. Per Default ist der aktive Mandant eingestellt.
- <title>: Title für den Link. Per Default leer.

Das Alias wird ersetzt durch einen Opening Tag der Form

```
<a class='file' href='http..' title='.. '>
```

### 7.3.38 FORM

Alias für das Opening-Tag einer Formularumgebung im eingestellten Standard-Charset (UTF-8). Die Defaulteinstellungen der Parameter erzeugen ein Formular für die Redaktion.

`{{FORM[<action>[;<method>[;<suffixes>[;<class>[;<add>]]]]]}}`

- <action>: action-Wert für das <form>-Tag. Default ist {{PHPself}}

- <method>: method-Wert für das <form>-Tag. Default ist post
- <suffixes>: String. Wird an die Action des Formulars angehängt. Falls 'this' übergeben wird wird der \$\_GET-Hash als GET-Parameter aufbereitet und angehängt. Default ist 'this'.
- <class>: HTML-Klassenbezeichner für die Formularumgebung. Per Default FORM.
- <add>: Zusätzliches HTML für die das <form ..>-Tag. Per Default leer.

Das Alias wird ersetzt durch

```
<form action='..' method='..' accept-charset='..>
```

### 7.3.39 FORMe

{{FORMe}} wird ersetzt durch </form>

### 7.3.40 GOOGLEMAP

Alias für die Erzeugung eines GOOGLE MAP-Fensters. Dieses wird in ein <div> der Klasse GOOGLEMAP eingebunden. Die Einbindung mehrerer solcher Fenster in einem Artikel ist möglich.

Die Karten werden auf Geokoordinaten im Gradmaß zentriert. Diese haben die Form (*lat*, *lon*) mit *lat* ∈ [−90, 90] und *lon* ∈ [−180, 180]. Hierbei werden negative Breite für die Südhalbkugel der Erde und negative Länge für ihre westliche Hemisphäre verwendet.

```
 {{GOOGLEMAP<(mixed)where>[;<(int)height>[;<(string)id>]]}}
```

- <where>: Dieser Parameter ist eine ','-Liste und kann eine der beiden folgenden Formen haben:

1. <cltSuffix>[,<Punktnummer>]: In den Kunden-Tabellen können mehrere Koordinatenpaare hinterlegt sein. Diese werden für den angegebenen Client ausgelesen und der angegebene Punkt wird verwendet. Die Zählung der Punkte beginnt bei null. 0 ist auch der Default für Punktnummer.

Bsp.: kh oder auch kh,0 beschreibt den Ort des Hauptsitzes von Koch Management Consulting.

2. <lat>,<lon>: Ein Koordinatenpaar, wie eingangs beschrieben. Achten Sie darauf, dass die Zahlen amerikanisch notiert sind. Dezimalen werden also nicht, wie Deutschland üblich, mit einem Komma, sondern mit einem Punkt abgetrennt.

Bsp.: 48.7419,9.1061 beschreibt einen Punkt in Stuttgart.

- **height**: Google-Zoom-Wert. Per Default 13. Auf dieser Auflösung kann man bereits Straßen und einzelne Gebäude gut erkennen. Höhere Werte zoomen noch näher heran (soweit bei Google Maps verfügbar), niedrigere zoomen heraus. Ein Wert von 8 etwa gibt einen guten Überblick über die Region.
- **id**: Das `<div>` in welches die Karte eingebunden wird benötigt eine eindeutige HTML-id. Per Default wird diese fortlaufend als `GOOGLEMAP0`, `GOOGLEMAP1`,... nummeriert. Mit diesem Parameter können Sie die id selbst wählen.

### 7.3.41 GOTONEXT

Alias für einen Link auf den „nächsten“ Artikel in Gestalt eines gelben, abwärts gerichteten, mit einem statischen Inhalt der Form „Zum nachfolgenden Artikel“ beschrifteten Pfeils. Sowohl der Pfeil als auch der statische Inhalt ist von einem Div umgeben.

```
 {{GOTOTOP [<(char)type>[;<(string)c1t>[;<(string)con>]]]}}
```

- **<plain>**: Falls `b` (both sides) Es wird auf den vorangehenden und den nachfolgenden Artikel verlinkt.  
Falls `bf` (both sides first) Es wird nur auf den nachfolgenden verlinkt.  
Falls `bl` (both sides last) Es wird nur auf den den vorangehenden Artikel verlinkt.  
(both sides) ignoriert die Parameter `c1t` und `con`.  
Falls `p` (text free) Gelber Pfeil nach unten ohne statischen Inhalt.  
Falls `n` (narrow) werden ein nach unten weisender Keil und das Wörtchen „weiter“ eingebunden. In diesem Fall erhält das Alias ein Inline-Div.  
Per Default hat der Pfeil die oben in der Kurzbeschreibung genannte Form.
- **<c1t>**: Falls der Zielartikel nicht automatisch bestimmt werden soll, kann er auch explizit angegeben werden. `c1t` steht hierbei für den Target Client. Default ist der aktive Client.
- **<con>**: Index Id des zu verlinkenden Artikels. Per Default der von `Kdb->getNextId(..)` zurückgelieferte Wert.

### 7.3.42 GOTOTOP

Alias für einen Link zum Anker `#top` in Gestalt eines gelben, aufwärts gerichteten, mit einem statischen Inhalt der Form „Zum Seitenanfang“ beschrifteten Pfeils.

```
 {{GOTOTOP [<(char)plain>[;<(mixed)force>]]}}
```

- <plain>: Falls p wird nur ein gelber, aufwärts weisender Pfeil eingebunden, der zum Anker #top verlinkt.
- <force>: Falls f wird der Pfeil angezeigt. Falls eine Zahl wird der Pfeil nur angezeigt, falls diese Zahl  $> 15$  ist. Die Stilklassen standardDisp übergibt an dieser Stelle ein Maß für die Artikellänge. Das führt dazu, dass der Pfeil nur dann angezeigt wird, falls der Artikel eine gewisse Mindestlänge überschreitet.

### 7.3.43 GTEXT

Schreibt Texte als png-Grafik mit transparentem Hintergrund.

```
 {{GTEXT<text>[;<color>[;<size>[;<width>[;<height>]]]]}}}
```

- <text>: Darzustellender Text. Damit auch das Semikolon als Zeichen möglich wird, wird das erste Zeichen des Textes als Delimiter verwendet. Der zu schreibende Text endet, sobald dieses Zeichen ein zweites mal auftaucht.  
Z.B. 'Hallo Welt!' würde somit als Hallo Welt! interpretiert werden.
- <color>: Ein hexadezimaler RGB-Farbcodes der Form RRGGBB. Per Default 000000.
- <size>: PHP-Schrift-Font. Ein Wert aus {1, ..., 5}. Per Default 4.
- <width>: Breite der zu erstellenden Grafik in Pixeln. Per Default 8 mal so viele wie der darzustellende Text Zeichen hat.
- <height>: Höhe der darzustellenden Grafik in Pixeln. Per Default 16.

### 7.3.44 HIDDEN

Alias für eine versteckte Inputbox. Wird von der Redaktion verwendet, um unter anderem den action-Parameter zu posten.

```
 {{HIDDEN<value>[;<name>]}}
```

- <value>: value der Box.
- <name>: name der Box. Default ist 'action'.

Ersatz durch <input type='hidden' name='...' value='...' />

### 7.3.45 HOME

`{{HOME...}}` wird ersetzt durch eine Linkadresse auf eine Homepage des dem `cltsuffix` zugeordneten Kunden. Diese ist in der Client-Tabelle hinterlegt.

`{{HOME<cltsuffix>[;<id>]}}`

- `<cltsuffix>`: Client Suffix des Kunden dessen Url ausgegeben werden soll.
- `<id>`: Ganze Zahl. Beginnend bei 0 gibt sie die Nummer der Ziel-Url an. Das betreffende Feld in der Client-Tabelle ist eine `\r\n`-Liste in deren Zeilen verschiedene URLs stehen können. Dabei gilt im Karrierehandbuch die Konvention, dass die nullte Zeile auf die externe Homepage des Unternehmens und die erste ggf. auf die firmeneigene Stellenbrse verweist. Default ist 0.

### 7.3.46 IMG

Alias für die Einbindung einer Mandantengrafik. Diese Grafik muss im Images-Unterverzeichnis des Mandanten liegen.

`{{IMG<fname>[;<cltsuffix>[;[div][;<url>]]]}}`

- `<fname>`: File Name der Grafik. Kann aber auch ein String der Form `subpath/fname` sein. Der Parameter wird durch eine Regular Expression gefiltert. Insbesondere Zeichenfolgen '..' sind nicht zugelassen.
- `<cltsuffix>`: Die Grafik wird im Verzeichnis `../Images/<cltSuffix>` gesucht. Per Default wird das Clientsuffix des aktiven Mandanten verwendet. Eine Ausnahme entsteht, wenn ein bestimmter Darstellungs-GET-Parameter `media` (z.B. `media=handheld`) verwendet wird. In diesem Fall prüft das Alias, ob die gesuchte Datei auch in einem Verzeichnis `../Images/<cltSuffix>/<media>` vorhanden vorhanden ist und lädt bevorzugt diese.
- `div`: Wird als dritter Parameter die case insensitive Zeichenfolge `div` übergeben wird die Ausgabe zusätzlich in `<div class='IMG'>..</div>` eingebunden.
- `<url>`: Falls angegeben wird die Grafik gleichzeitig ein Link auf diese URL.

Falls alle Optionen wahrgenommen werden wird das Alias ersetzt durch

```
<div class='IMG'><a class='IMG' href='...'>
  <img class='IMG' alt='...' src='...' /></a></div>
```

Als `alt` wird der übergebene Dateiname verwendet.

### 7.3.47 JOB

Erstellt einen Link zu einer aus einer job\_<lng>\_<clt>-Tabelle heraus automatisch erzeugten Stellenausschreibung.

```
 {{JOB[<clt>[;<id>[;<lng>[;<type>[;<title>]]]]]}}
```

- <clt>: Client-Suffix des Mandanten, dem die auszuwertende job-Tabelle zugeordnet ist. Per Default der aktive Client.
- <id>: Primary id der auszuwertenden Stellenbeschreibung. Per Default die id der ersten Zeile in der Tabelle.
- <lng>: Sprachsuffix der Sprache die der auszuwertenden Tabelle zugeordnet ist. Per Default die aktive Sprache.
- <type>: Art des Links. Bislang implementiert:
  - link: Kompletter Link mit Opening-Tag, Closing-Tag und Linktext (Default).
  - opening: Nur der Opening-Tag für den Link.
  - plain: Nur der Title. Kein Link wird gesetzt.
- <title>: Text für den Link. Per Default der Title zu verlinkenden Jobs.

### 7.3.48 JOBLIST

Erstellt {{JOB..}}-Links für alle aktiven Stellenangebote einer Sprache eines Mandanten.

```
 {{JOBLIST[<clt>[;<lng>[;<type>[;<sep>]]]]}}
```

- <clt>: Client-Suffix des Mandanten, dem die auszuwertende job-Tabelle zugeordnet ist. Per Default der aktive Client.
- <lng>: Sprachsuffix der Sprache die der auszuwertenden Tabelle zugeordnet ist. Per Default die aktive Sprache.
- <type>: Type für die darzustellenden {{JOB..}}-Links. Wird direkt in die erzeugten {{JOB..}}-Aliase eingefügt. Per Default leer.
- <sep>: Separator der zwischen die einzelnen {{JOB..}}-Aliase eingefügt wird.  
*Achtung:* Die Syntax verlangt eine Begrenzung durch Hochkommas: '<sep>' und erlaubt somit das Verwenden von Semikoli ; im Separator-Code. Default ist \r\n für einen Zeilenumbruch um Carriage Return.

*Beispiel:* {{JOBLISTkoch;;' {{WRY}} '} } erzeugt eine durch gelbe Keile getrennte Liste der Jobs des Mandanten koch aus Daten der der aktiven Sprache zugeordneten Job-Tabelle.

### 7.3.49 JUS

`{{JUS<cltsuffix>}}` wird ebenso behandelt wie  `{{ADD..}}`. Bezieht sich aber auf die juristische Bezeichnung des Zielmandanten.

### 7.3.50 KBOOK

Analog zu  `{{BBOOK}}`, wird allerdings durch ein grünes Buch-Thmub-Nail ersetzt, welches auf den Artikel `con=29010201,clt=kh` (die Download-Seite des Karrierehandbuchs) verlinkt.

### 7.3.51 KH\_URL

`{{KH_URL}}` wird ersetzt durch

`{{PHPi_protocol}} {{PHP_SERVER;HTTP_HOST}}/{{PHPi_main_page}}`

### 7.3.52 MPFORM

Multipart-Formular-Umgebung. Im Grunde das Alias  `{{FORM..}}` mit der zusätzlichen Option `enctype='multipart/form-data'`.

Wird für File-Upload-Formulare benötigt.

### 7.3.53 NAME

`{{NAME<cltsuffix>}}` wird ebenso behandelt wie  `{{ADD..}}`. Bezieht sich aber auf den Namen des Zielmandanten.

### 7.3.54 OBJECT

`{{OBJECT..}}` erzeugt eine Umgebung in der Applets wie z.B. ein Youtube-Video eingebunden werden können. Bedient sich sowohl HTML-<object..> als auch <param..>-Tags.

`{{OBJECT[<nogo>[;<add>[;<param>]]]}}`

- **nogo:** Code der hinter die optionalen Parameter im Object-Tag abgelegt werden soll.
- **add:** Code der Innerhalb des Attributbereichs des Object-Tags abgelegt werden soll.
- **param:** ',' -Liste von Schlüssel-Werte-Paaren der Syntax `key=>value`. Diese Paare werden HTML-in param-Tags umgesetzt mit dem Name `key` und dem Value `value`.

*Beispiele:*

```
 {{OBJECTFlash-Video von Harald Lesch;codetype="application/x-shockwave-flash" height="344" width="425"; movie=>http://www.youtube.com/v/MbvMVRkhaRo&amp;hl=de&amp;fs=1, allowFullScreen=>true,allowScriptAccess=>always}}
```

wird zu

```
<object codetype="application/x-shockwave-flash" height="344" width="425"><param name='movie' value='http://www.youtube.com/v/MbvMVRkhaRo&amp;' /> Flash-Video von Harald Lesch</object>
```

Oder

```
 {{OBJECTJava-Rubik-Cube von Karl Hoernell; codetype="application/java" classid="java:rubik.class" standby="One moment, Sir." codebase='Extern/Rubik'}}}
```

Wird zu

```
<object codetype="application/java" classid="java:rubik.class" standby="One moment, Sir." codebase='Extern/Rubik'> Java-Rubik-Cube von Karl Hoernell</object>
```

Die Zeilenumbrüche sind hierbei nachträglich eingefügt worden.

### 7.3.55 PAR

{{PAR}} wird ersetzt durch das Opening-Tag eines Anchors auf die url des ersten Parent-Artikel des aktiven Artikels. Außerdem wird die id des aktiven Artikels als Sub-Anchor verwendet.

```
<a href='url#id'>
```

### 7.3.56 PDF

Hat im HTML-Modus keine Auswirkung. Sein Parameter ist aber ein Steuerbefehl für den PDF-Konverter.

```
 {{PDF<command>}}
```

Bislang unterstützte Kommandos:

- newpage: Erzwingt einen Seitenumbruch.

Hinweis: Das PDF-Alias bildet einen eigenen Block. Dies kann zu unerwünschten Effekten führen, falls dadurch kohärente Textblöcke zerteilt werden. Fügen Sie das Alias nach Möglichkeit nur zu Zeilenbeginn oder -ende ein.

### 7.3.57 PHP

Wird ersetzt durch den Wert einer als Parameter übergebenen Variablen.

`{{{PHP<varName>[;<key>]}}}`

- `<varName>`: Key eines Wertes der entweder im Hash `$GLOBALS` oder in einem `htDisplay->replaceAliases(..)` als `$extra` übergebenen Hash zu finden ist.
- `<key>`: Optionaler Zusatzparameter. Falls angegeben geht das Programm davon aus, dass die Zielvariable selbst ein Hash ist und gibt denjenigen Wert zurück der unter `<varName>[<key>]` gefunden wird.

Falls der gesuchte Wert nicht gefunden werden kann wird das Alias einfach entfernt.

Dieses Alias stellt natürlich ein gewisses Sicherheitsrisiko dar. Aus diesem Grund werden einige als *besonders vertraulich* eingestufte Variablen vor der Auswertung gesondert herausgefiltert und sind über `{{{PHP..}}}` nicht darstellbar.

### 7.3.58 REM

`{{{REM<irgend_ein_Kommentartext>}}}` wird als 'remark' ersatzlos entfernt.

### 7.3.59 SA

Das Alias wird ersetzt durch das HTML einer Short Advertisement Visitenkarte.

`{{{SA[<clt>[;<saId>]]}}}`

- `<clt>`: Client Suffix des Kunden dem die darzustellende SA zugeordnet ist. Per Default ist der aktive Kunde eingestellt. Wird hier `RAND` angegeben, wird ein Zufallskunde aus den Standard-Clients ausgewählt.
- `<saId>`: Id der darzustellenden Visitenkarte innerhalb der SA-Tabelle des Zielkunden.

### 7.3.60 SATAR

Dieses Alias liefert eine Liste von Links auf Ziele von Visitenkarten zurück.

`{{{SATAR[clt[;sid[;demo]]]}}}`

- `<clt>`: Suffix des Mandanten für den die Liste erstellt werden soll. Default ist `kh`.

- <sid>: Id der Visitenkarte innerhalb der Short-Ad-Tabelle des Mandanten. Falls nicht angegeben oder mit all belegt werden alle Visitenkarten gleichermaßen berücksichtigt. Default ist all.
- <demo>: Optional. Kann einen in Apostrophe '...' geklammerten cid-String im Stil der sa\_<lng>\_<clt> Tabellen enthalten. In diesem Fall werden die tatsächlichen TargetIds ignoriert und stattdessen die Demo-Liste verwendet. Dies ist sinnvoll für die Demonstration einer Probe-Site bei einem potentiellen Neukunden. Per Default ist dieser Feature deaktiviert.

Zurückgeliefert wird ein zeilenweise organisierter String der Form

```
 {{CLT<clt>;<cid>;0}}<tclt==kh?'Kapiteltitel - ':'><title(cid)>{{ae}}
 ...

```

### 7.3.61 STATIC

Dieses Alias erlaubt die Verwendung von Inhalten aus den Static-Tabellen.

```
 {{STATIC<handle>[;<context>[;<args>]]}}
```

Die Parameter werden direkt an `staticTools::fmtStatCon(..)` weitergeleitet. Insbesondere darf der dritte Parameter, `args`, Semikoli enthalten.

### 7.3.62 TEL

`{{TEL<cltsuffix>}}` wird ebenso behandelt wie  `{{ADD..}}`. Bezieht sich aber auf die Telefonnummer des Zielmandanten.

### 7.3.63 TIME

`{{TIME<timestamp>[;<shift>]}}` wird ebenso behandelt wie  `{{DATE..}}`. Bezieht sich aber auf die Uhrzeit, die in der Form HH:MM:SS dargestellt wird.

### 7.3.64 TITLE

`{{TITLE}}` wird durch den `title`-Inhalt eines Contents ersetzt.

```
 {{TITLE[<tclt>[;<id>]]}}
```

- <tclt>: Target Client. Default ist der aktive Client.
- <id>: Index-Id des Zielartikels. Default ist der aktive Artikel.

### 7.3.65 TITLELINE

Fügt einen Überschriftenlink auf den ersten Parent-Knoten des aktiven Contents ein.

`{{TITLELINE[<title>]}}`

- `<title>`: Text `title` für den Überschriftenlink. Default ist '`{{TITLE}}`'

Das Alias wird ersetzt durch

```
<div class='titleline'><a href='..'>
  <span class='i'>|</span>&nbsp;&nbsp;
    title
  &nbsp;&nbsp;<span class='i'>|</span>
</a></div>
```

### 7.3.66 TITLELINE\*

Variation von `{{TITLELINE}}`. Verhält sich identisch zum Original, legt aber keinen Link an.

### 7.3.67 TOC

Inline Table of Contents. Erzeugt einen als `CLmi` formatierten Table of Contents der Tiefe 2. Bezieht sich auch auf den aktiven Client.

`{{TOC[<id>[;<includeRoot>[;<doClFrame>[;<firstNoLink>]]]]}}`

- `<id>`: Index-Id des Artikels innerhalb der Indextabelle des aktiven Clients. Per Default die aktive Id.
- `<includeRoot>`: Boolsch. Soll die Wurzel bereits Teil des TOC sein. Per Default `false`
- `<doClFrame>`: Boolsch. Soll eine Checklist-Umgebung um die Ausgabe gesetzt werden? Dies ist erforderlich, falls das Alias nicht bereits in einer Checkliste platziert ist. Default ist `false`.
- `<firstNoLink>`: Boolsch. Soll die erste Ebene des Verzeichnisses nur Plaientext ohne Link sein? Per Default `false`.

*Beispiele:*

1. Die Code-Fragmente

```
CLmi
{{TOC}}
CLe
```

und

```
{{TOC;;true}}
```

erzeugen einen Standard-ToC, aufbauend auf dem aktiven Artikel.

## 2. Die Code-Fragmente

```
CLmi
{{TOC100;true}}
CLe
```

und

```
{{TOC100;true;true}}
```

erzeugen jeweils ein Inhaltsverzeichnis mit dem Artikel 100 und seinen eventuell vorhandenen Kids als einzigen Ast.

*Anmerkung an den Admin: Damit dieses Alias richtig arbeitet, muss es vor dem Aufruf von htDisplay->std2html() ausgewertet werden.*

*htDisplay->importContents() sorgt dafür. Das soll so bleiben!*

### 7.3.68 TR

Tabellenzeilenumgebungseröffnungstag.

```
 {{TR[<class>]}}
```

- **<class>**: CSS-Klasse der Zeile. Per Default ein bei 'even' beginnender Wechsel zwischen 'even' und 'odd'. Die Buchführung funktioniert über eine statische Variable. Es wird auch umgeschalten falls ein Wert für **<class>** angegeben wurde.

Das Alias wird ersetzt durch `<tr class='..>`

### 7.3.69 TRe

`{{TRe}}`  wird ersetzt durch `</tr>`

### 7.3.70 USER

Wird durch öffentliche Informationen über einen bestimmten User ersetzt.

```
 {{USER<login>[;<datakeys>]}}
```

- **<login>**: Login des Users.

- <datakeys>: Eine ; -Liste anzuzeigender Daten. Zugelassene Schlüssel sind
  - **address**: Anrede. Meist „Herr“ oder „Frau“.
  - **firstname**: Vorname.
  - **lastname**: Nachname.
  - **tel**: Gegebenenfalls die *öffentliche* Telefonnummer des Users.
  - **appmail**: Gegebenenfalls die *öffentliche* E-Mail-Adresse des Users.
  - **times**: Gegebenenfalls Sprechzeiten.
  - **url**: Die in der Kontakttabelle veröffentlichte Url.
  - **urlcaption**: Der zu obiger Url gehörige Link-Title.
  - **onlinetext**: Der Online-Text des Users.
  - **offlinetext**: Der Offline-Text des Users.

Werden mehrere Schlüssel übergeben, wird das Alias durch eine durch Spaces getrennte Liste der geforderten Daten ersetzt.

Beispiel:

`{{USERkochmn;firstname;lastname}}` wird zu „Markus-Hermann Koch“.

### 7.3.71 WRY

’Wedge Right Yellow’. `{{WRY}}` wird ersetzt durch

`<span class='gelb'>#9658;</span>`

`#9658` erzeugt eine ausgefüllte Version des Symbols ▷.

### 7.3.72 WDY, WLY, WUY

Analog zu ’Wedge Right Yellow’: ’Wedge Down Yellow’, ’Wedge Left Yellow’ und ’Wedge Up Yellow’.

## 8 Häufig benötigte Aliase

Wenn Sie das Aliaskonzept einsetzen, aber nicht die gesamte Alias-Dokumentation durcharbeiten wollen, finden Sie hier eine Kurzbeschreibung der nach Ansicht des Autors dieser Dokumentation nützlichsten Aliase.

Dieses Kapitel ersetzt nicht die ausführliche Dokumentation der angeführten Aliase. Seine Aufgabe erschöpft sich in der Auflistung der wichtigsten Aliase und einer Kurzbeschreibung ihrer Funktionen.

- **CLT:** Dient der Verlinkung zweier Artikel der Site untereinander. Wird in den meisten Fällen lediglich durch das Opening-Tag des Links ersetzt.

```
{{{CLT[<(text)cltsuffix>[;<(int)cid>[;<(text)lng>[;<(text)title>
[;<(bool)plainUrl>]]]]]}}}
```

- **FILE:** Clients haben auf dem Server eine Quota von 30 MB und einen eigenen Verzeichnisbaum, in dem Sie Dateien zum herunterladen durch Besucher ihrer Seiten bereitstellen können. Diese Dateien können mit Hilfe des Aliases `FILE..` verlinkt werden.

```
{{{FILE<fname>[;<cltSuffix>[;<title>]]}}}
```

`FILE..` wird lediglich durch das Opening-Tag eines Links ersetzt.

- **ae:** `ae..` wird ersetzt durch ein Linkabschluss-Tag `</a>`.
- **IMG:** Alias für die Einbindung einer Mandantengrafik. Diese Grafik muss im Images-Unterverzeichnis des Mandanten liegen.

```
{{{IMG<fname>[;<cltsuffix>[;[div][;<url>]]]]}}}
```

- **bf:** *Bold Face*. Setzt seinen Parameter als fett dargestellten Text.  
*Beispiel:* `bfBold Face..` wird ersetzt durch **Bold Face**.  
Analog funktionieren die Aliases `em..` für *kursive* Texte,  
`underline..` für unterstrichene Texte, sowie `smaller..` und  
`larger..` für mit den gleichnamigen CSS-Eigenschaften versehene Texte.