Joomla!
Mambo OS

Praxisbuch

Mambo OS Versionen 4.5.1 bis 4.5.3
Joomla Version ab 1.0

1. VORWORT 5

2. VON MAMBO ZU JOOMLA 6

3. TERMINOLOGIE 6

4. WAS IST EIN CONTENT MANAGEMENT SYSTEM 8

5. WAS IST MAMBO 9

5.1. VOR- UND NACHTEILE VOM CMS MAMBO OS 9
5.2. MAMBOVERSIONEN 9

6. INSTALLATION 9

6.1. VORAUSSETZUNGEN 9
6.2. DIE INSTALLATION 9
6.3. DIE LOKALE INSTALLATION 9
6.3.1. LOKAL → WEB 10
6.4. WELCHE SPRACHE SPRICHT MAMBO? 10
6.5. BEKANNTE FEHLER BEI DER INSTALLATION 10
6.5.1. ICH KANN MICH NICHT EINLOGGEN?! 10
6.6. UPDATE 10
6.6.1. BACKUP DER DATENBANK 10
6.6.2. BEKANNTE FEHLER BEIM BACKUP 10

7. ERSTE BEGEGNUNG NACH DER INSTALLATION 10

7.1. KURZÜBERBLICK - PULLDOWNMENÜS 11
7.2. KURZÜBERBLICK – SCHALTZENTRALE 12
7.3. KURZÜBERBLICK – ÜBERSICHTSFENSTER 12
7.4. WIE FUNKTIONIERT ABER NUN JOOMLA? EIN ERSTER ARTIKEL 12
7.5. ZUSAMMENFASSUNG 21

8. DIE MANAGER 22

9. BEREICHE UND KATEGORIEN 22

9.1. BEREICHE 22
9.2. KATEGORIEN 24
9.3. DER BEREICH MANAGER 26
9.4. DER KATEGORIEN MANAGER 28

10. ARTIKEL – INHALT – BEITRÄGE - CONTENT 28

10.1. INHALTE NACH BEREICH – NEUE KATEGORIE 28
10.2. INHALTE NACH BEREICH – ARTIKEL ANLEGEN 29
10.2.1. WYSIWYG – PRO UND CONTRA 29

10.2.1.1. Artikel formatiern mit WYSIWYG-Editoren 30
10.2.1.2. Probleme mit WYSIWYG 33
10.2.1.3. Artikel formatieren - ohne WYSIWYG-Editor 34
10.2.2. BILDER EINFÜGEN 34
10.2.2.1. Media Manager 35
10.2.2.2. mosimage 36
10.2.3. EIN BEITRAG ÜBER MEHRERE SEITEN - MOSPAGEBREAK 39
10.2.4. PARAMETERS 40
10.2.5. META INFO 42
10.2.6. PUBLISHING 43
10.2.7. LINK TO MENU 44
10.3. STATISCH ODER DYNAMISCH 44
10.3.1. STATISCHER INHALT 45
10.4. BEITRÄGE ERSTELLEN UND BEARBEITEN IM FRONTEND 45
10.5. "AUF STARTSEITE ANZEIGEN" ODER BESSER WOANDERS? 46
10.6. CHECKIN 47

11. MENÜS 47

11.1. MENÜ MANAGER 47
11.1.1. VERSION 4.5.1 48
11.1.2. VERSIONEN AB 4.5.2 48
11.1.3. UND WIEDER ALLE VERSIONEN ZUSAMMEN 48
11.2. NEUE MENÜEINTRÄGE 48
11.2.1. INHALT 49
11.2.2. KOMPONENTEN 50
11.3. MENÜ MODULE 50
11.4. TOP ODER SUB – UNTERMENÜS ERSTELLEN 50

12. GLOBAL CONFIG 51

13. TEMPLATES 51

13.1. EIN EIGENES TEMPLATE ERSTELLEN 51
13.2. EINE KLEINE EINFÜHRUNG IN CSS 51
13.2.1. BROWSERPROBLEME 53
13.3. EIGENE CSS-KLASSEN DEFINIEREN 54
13.4. PAGE CLASS SUFFIX 54
13.5. MODUL CLASS SUFFIX 54
13.6. ADMINISTRATOR-TEMPLATE 54

14. EIGENE KOMPONENTEN ENTWICKELN 54

14.1. BACKEND 54
14.1.1. STRUKTUREN 54
14.1.1.1. Verzeichnisstruktur 54
14.1.1.2. Dateienstruktur 54
14.1.1.3. Version und Copyright 55
14.1.2. DATEI: ADMIN.MEINEKOMPONENTE.HTML.PHP 55
14.1.3. DATEI: ADMIN.MEINEKOMPONENTE.PHP 55
14.1.4. DATEI: TOOLBAR.MEINEKOMPONENTE.HTML.PHP 57
14.1.4.1. Ereignisse auswerten 57
14.1.4.2. Mögliche Aufrufe (Kopie aus der Datei *menubar.html.php*) 58

14.1.5. DATEI: TOOLBAR.MEINEKOMPONENTE.PHP 63
14.1.6. ZUSAMMENFASSUNG 64
14.2. SYMBOLE UND GLOBALES 64
14.2.1. TABELLENLISTING 64
14.3. FRONTEND 67
14.3.1. STRUKTUREN 67
14.3.1.1. Verzeichnisstruktur 67
14.3.1.2. Dateienstruktur 67
14.3.2. DATEI: MEINEKOMPONENTE.PHP 68
14.4. INSTALLATION 68
14.4.1. XML-DATEI 68
14.4.2. INSTALL-FILE 70
14.5. UNINSTALL 70
14.6. INDEX.HTML 70
14.7. SPRACHDATEIEN 71

15. WAS GEHT WO 71

15.1. ORDNUNG MUß SEIN 71
15.2. FRONTPAGE EINSTELLEN 71
15.3. HAUPT- UND UNTERMENÜANZEIGE GETRENNT 71

16. ANHANG 73

16.1. WO STEHT WAS 73
16.1.1. DIE VERZEICHNISSTRUKTUR 73
16.1.2. DATEIEN 73
16.2. MEHRERE MAMBOS AUF EINER WEBSITE 74
16.3. WEBSEITEN 74
16.3.1. COMMUNITYS, HILFEN, DOWNLOADS 74
16.3.2. KOMMERZIELLE WEBSEITEN 74
16.3.3. DARÜBER HINAUS 74

1. Vorwort

Das Buch ist von Praktikern für Praktiker geschrieben. Deshalb geht es auch nicht unbedingt Menüpunkt für Menüpunkt vor, sondern baut mit den einzelnen Kapiteln langsam auf und schafft so ein immer umfangreiches Wissen über das Content Management System *Mambo Open Source*. Beginnend mit der Erstellung eines einfachen Artikels und dem Anlegen eines Menüpunktes ohne großes Bimborium wird zunächst die grundlegende Vorgehensweise in Mambo erklärt, ohne dabei den Leser mit zuvielen Details am Anfang zu überfordern. In den weiteren Kapiteln werden jedoch Stück für Stück die vielfältigen Möglichkeiten von Mambo an konkreten praktischen Beispielen erklärt.

Darüber hinaus wird ein Überblick über die gängigsten Module und Komponenten und deren Anwendung geschaffen. Ein paar Tipps und Tricks zu bekannten Fehlern oder Unwegsamkeiten runden dieses Buch ab.

Dabei richtet sich dieses Buch sowohl an Einsteiger, als auch an Fortgeschrittene. Und so mancher Profi wird sicherlich auch das eine oder andere Neue entdecken.

Als Leser dieses Buches kann es nicht schaden ein Grundwissen über HTML und CSS zu haben, aber wer es nicht hat, wird dennoch mit diesem Buch arbeiten können, denn überall wo es wichtig ist, werden Details klar besprochen.

Für dieses Buch wird schwerpunktmäßig die Mamboversion 4.5.3 als lokale Installation innerhalb des Webservers Xampp (Xampp-Mambo2 (siehe dazu auch weiteres weiter unten)) in deutscher Sprache verwendet. Gravierende Abweichungen zu den direkten Vorgängerversionen werden im Text erwähnt.
Ein Kapitel beschäftigt sich damit, wie die lokalen Projekte ins Internet gebracht werden und eventuelle Dinge, auf die es zu achten gilt, werden natürlich besprochen.

Wer eine englische Mamboversion sein eigen nennt, erfährt natürlich in diesem Buch auch, wie ein nachträgliches eindeutschen möglich ist – sowohl im Front- als auch im Backend.

Geschrieben ist dieses Buch für alle, die die Mamboversionen ab 4.5.1 benutzen.

Für die Screenshots und die Arbeit an diesem Buch wurden nachfolgende Browserversionen benutzt:

- Opera 8.0
- Mozilla Firefox 1.0.1
- Microsoft Internet Explorer 6.0
- Netscape 7.02

Ein paar Hinweise zur Benutzung:
Zusätzliche Informationen sind in roter Schrift gekennzeichnet.

Hinweis
Zu einem bestimmten Thema gibt es einen wichtigen Punkt, der hiermit herausgehoben wird.

Tipp
Tipps und Tricks, die sich zumeist aus der Praxis heraus bewährt haben

HTML

Für alle, die sich mit HTML und CSS nicht allzu gut auskennen, finden unter diesem Stichwort ein paar erklärende Hinweise. Allerdings würde es den Rahmen sprengen, wenn wir in diesem Buch HTML vermitteln wollten. Es soll lediglich als kleinere Hinweise verstanden werden, um zumindest zu wissen, um was es geht …

2. Von Mambo zu Joomla

Das Ende der Ära Mambo OS
Entstehung von Joomla

3. Terminologie

Backend
Der Administrationsbereich

Bereich
Inhaltlicher Hauptpunkt.

CHMOD
Innerhalb eines → FTP-Programms können Sie die Zugriffsrechte einer Datei oder eines Verzeichnisses verändern. Diese Zugriffsrechte werden CHMOD genannt. In vielen FTP-Programmen wird dieses zumeist über die rechte Maustaste gesteuert. Also Datei markieren und mit rechts anklicken. Im Menü dann CHMOD auswählen.

Content
→ Inhalt

FTP
FTP = File Transfer Protocol
FTP sorgt dafür das Dateien auf einen Server up- und downloadet werden können. Für diesen Vorgang gibt es verschiedene Programme. Am bekanntesten sind mittlerweile der WS-FTP und PSFtp. Beide Programme kosten Geld. Es gibt allerdings auch eine Vielzahl Freeware-Tools. Einfach mal die Suchmaschinen fragen …

Frontend
Der "sichtbare" Bereich. Also der Bereich, der dem User zugänglich ist. Wobei kein Unterschied dabei mit den einzelnen Rechten gemacht wird. Es ist also einerlei, ob eine Seite nur für Registrierte, Administratoren oder für alle User zugänglich ist.

HTML-Tag
HTML-Formatierungsbefehle werden als Tag (englisch ausgesprochen) bezeichnet. Also alles was innerhalb der eckigen Klammer steht . Beispielsweise: *
,* , etc.

Inhalt
Unter dem Begriff *Inhalt* wird in Mambo alles an Text und Bildern zusammengefaßt. Das können Artikel, Beiträge, Manuals, Tipps und Tricks, Informationen, Bilder und deren Beschreibungen, Geschichten und und und … sein

Kategorie
Inhaltlicher Unterpunkt zu einem Bereich.

Komponenten
Komponenten sind kleine fertige Programmteile, zu unterschiedlichen Themen, die in Mambo integriert werden. Komponenten lassen zumeist die Eingabe von Daten/Datensätzen zu.
Einige sind bereits bei der Installation vorhanden. Eine Vielzahl von Komponenten, die weit in die hunderte hineingeht, lassen sich zusätzlich in Mambo integrieren.
Komponenten bieten ihrerseits in aller Regel eigene Untermenüs an und lassen es zu eigene Daten einzugeben.

Mambots
Mambots sind Plugins, die manchmal im Hintergrund wirken, manchmal in seinen eigenen Inhalten integriert werden können oder sogar größere Programmteile, wie beispielsweise der WYSIWYG-Editor.
Einige sind bereits bei der Installation vorhanden. Eine Vielzahl von Mambots lassen sich zusätzlich in Mambo integrieren. Generell ist ein Mambot stehts ein kleines Programm, welches bestimmte Aufgaben übernimmt. Beispielsweise ist die Suchenfunktion auf einer Webseite ein Mambot.

Module
Module sind fertige Programmteile, die zumeist Formatierungsänderungen, aber keine Eingabe eigener Daten zulassen (Ausnahme können URL's oder ähnliche Quellenangaben sein).
Einige sind bereits bei der Installation vorhanden. Eine Vielzahl von Modulen, die weit in die hunderte hineingeht, lassen sich zusätzlich in Mambo integrieren.

Sektion
→ Bereich

SEF

Tag
→ HTML-Tag

WYSIWYG
What **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et
Was im Text eingegben wird, wird auch exakt so abgespeichert und am Bildschirm ausgegeben. Also eine grafische Oberfläche, die die ausgewählten Formatierungen direkt anzeigt und speichert.

4. Was ist ein Content Management System

Mambo ist ein komfortables, leistungsstarkes Content Management System (CMS). Doch was ist ein CMS genau?

Content Management bedeutet, dass ein System zur Verwaltung von Daten - in unserem Fall von Webseiten - eingesetzt wird.

Alle Inhalte, die auf den Webseiten erscheinen sollen, werden mit diesem System verwaltet. Das hört sich zunächst banal an. Das Entscheidene ist jedoch, dass nicht mehr wie zuvor einzelne Webseiten erstellt werden (wie z.B. statische HTML-Dateien, dynamische PHP-Dateien etc.), sondern dass lediglich die Dateninhalte eingegeben werden und das System dann dafür sorgt, was wo in welcher Form angezeigt wird.

Ob Artikel, Bilder, Diskussionen, Nachrichten oder Informationen - im Content Management System Mambo lassen sich alle Daten gleich gut, ohne großen nachträglichen Aufwand, einfach und bequem eingeben, darstellen, verwalten oder archivieren.

Mittels Rechtevergabe lässt sich ebenso leicht ein Workflow für neue Inhalte festlegen. Also wer was schreiben und veröffentlichen darf bzw. wer mehr Rechte und somit mehr Möglichkeiten innerhalb des CMS hat.

Damit ist Mambo u.a. hervorragend dazu geeignet, um in Betrieben die Erzeugung und Bereitstellung von Inhalten im Inter- und/oder Intranet zu regeln, in Communitys durch unterschiedliche Schreib-/Leserechte Zugriffe auf verschiedene Inhalte und Seiten anzubieten oder auf Webseiten z.B. Artikel oder auch komplette Menüs über das Rechtesystem abzuschirmen oder öffentlich zugänglich zu machen.

Ein weiterer Pluspunkt in der einfachen Bedienbarkeit von Mambo ist, dass die Eingabe von Inhalten ohne Kenntnisse einer Programmiersprache möglich ist. Ein festinstallierter WYSIWYG-Editor (What You See Is What You Get) erlaubt das Schreiben ähnlich wie in einer Textverarbeitung. Das System erzeugt automatisch den notwendigen Code - der Autor braucht sich um nichts zu kümmern, außer dass die gewünschte Zusammenstellung einigermaßen vernünftig aussieht.

Die Vorteile eines Einsatzes von Mambo zur Webseiten-Gestaltung liegt also auf der Hand:

- Datenmengen lassen sich übersichtlich verwalten
- aktuelle Anpassungen sind in Sekunden möglich
- alte Datenbestände können übersichtlich archiviert werden
- die Erzeugung dynamischer Webseiten
- die Pflege der Webseiten ohne das Kenntnisse über HTML oder anderer Codes erforderlich sind
- jederzeit problemlose Erweiterung (oder Verkleinerung) des Webauftritts durch das De-/Installieren von zusätzlichen Komponenten
- Änderung des Designs ohne die Daten (Texte, Bilder etc.) separat anpassen zu müssen

5. Was ist Mambo

Web-CMS, kein Webdesigner oder ähnliches …

5.1. Vor- und Nachteile vom CMS Mambo OS

Vorteile: Leichte Bedienbarkeit, Templateerstellung, kostenlos (GNU/GPL) …
Nachteile: Rechtevergabe, Unterverschachtelung bei Kategorien, …

5.2. Mamboversionen

Die Versionspolitik ist etwas unglücklich und verwirrend.
Mit der Entstehung dieses Buches ist die **Mamboversion 4.5.3** und die **Joomlaversion 1.0.3** aktuell.

Verwirrend ist, dass es lange Zeit die "Hauptversion" 4.5 gab (und gibt), deren Releas bis zur 1.0.9 lief. Geschrieben wurde dieses dann: *Mamboversion 4.5–1.0.9*
Danach gab es eine komplett überarbeitete und in weiten Teilen auch wirklich **neue** Version: *Mambo 4.5.1*
Zwischen diesem kleinen Schritt (4.5-1.0.9 auf 4.5.1) lag in Wirklichkeit ein sehr großer Schritt. In Fachforen wird daher auch strikt zwischen diesen beiden "Hauptversionen" getrennt.
Für den unbedarften Nutzer erscheint es aber oftmals so, als wenn die Version *4.5-1.0.9* eine Weiterentwicklung der *4.5.1* sei. Zumal es in der "1er-Gruppe" auch noch solch verwirrende Versionen, wie *Mambo 4.5.1a stable* und andere Merkwürdigkeiten gab.
Wichtig ist also genau zu schauen, ob nach der "5" ein Punkt oder ein Bindestrich folgt. Ein "Punkt" bringt einem auf die sichere, sprich: aktuelle, Seite!

6. Installation

6.1. Voraussetzungen

6.2. Die Installation

6.3. Die lokale Installation

Xampp und Co.

6.3.1. Lokal → Web

Wie bekomme ich meine lokale Mamboinstallation ins Inet …

6.4. Welche Sprache spricht Mambo?

Mamboversionen
Nachträgliche Installation eines deutschen Backends und Frontend.
Unterschiede Sprache im Backend und Frontend wegen Übersetzungsfehler und –vergesslichkeiten, etc. …

6.5. Bekannte Fehler bei der Installation

u.a. Safemode

6.5.1. Ich kann mich nicht einloggen?!

6.6. Update

Von Version xx auf Version xx

6.6.1. Backup der Datenbank

- Auch beschreiben wie ein Backup durchgeführt wird.
- Welche Einstellungen gibt es beim Backup und was bedeuten diese
- Wie wird das Backup "zurückgeschrieben"

6.6.2. Bekannte Fehler beim Backup

7. Erste Begegnung nach der Installation

Nachdem Mambo frisch installiert wurde und man sich das erste Mal einloggt, findet man einen Administrationsbereich – Backend genannt - vor, der in verschiedene übersichtliche Bereiche gegliedert ist:

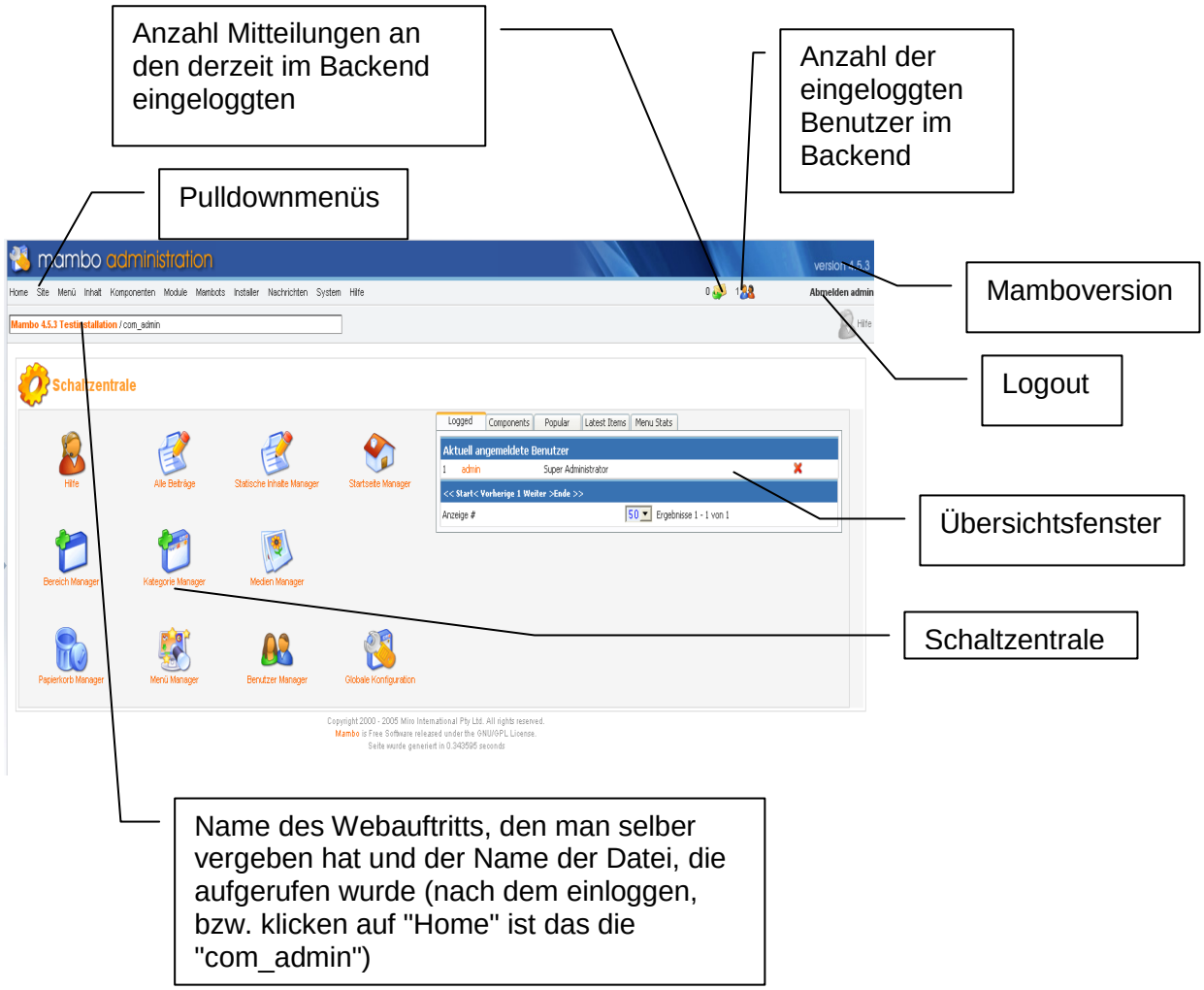


Abbildung: Home → Schaltzentrale

7.1. Kurzüberblick - Pulldownmenüs

Mit dem Pulldownmenü *Home* kommt man jederzeit auf obige Ansicht zurück.
Mit dem Menüpunkt "Site" erreicht man Menüpunkte, die sich auf die Darstellung und Benutzung des ganzen Webauftritts beziehen.

Menü bezieht sich auf das oder die Menüs des eigenen Projekts. Hier wird also die Navigation seiner Website festgelegt und verwaltet.

Bei *Inhalt* dreht sich alles um den eigentlichen Inhalt, der erzeugt wird. Also Artikel, Texte, etc.

Unter *Komponenten* finden sich die installierten Komponenten und deren spezifischen Menüpunkte wieder.

Unter *Module* lassen sich vorhandene Module editieren und einstellen.

Mambots verwaltet die Plugins (siehe Terminologie).

Ab der Mamboversion 4.5.3 gibt es einen neuen Menüpunkt: Den *Installer*. Hier sind alle Installationsbereiche zusammengefaßt. In den Verionen vor der 4.5.3 sind diese Installationsroutinen in den jeweiligen Bereichen untergebracht: *Templates* und *Sprachen* unter dem Menüpunkt *Site*, Komponenten unter *Komponenten* und Module unter *Module*.

Unter *System* findet sich jetzt nur noch das "Einchecken" – in früheren Versionen konnten hier auch Datenbankbackups erstellt werden.

Hilfe ruft eine englischsprachige Hilfe zu Mambo auf.

7.2. Kurzüberblick – Schaltzentrale

Alle Icons können auch über das Pulldownmenü erreicht werden.
Einige der häufiger benutzten Menüpunkte sind hier als Icons dargestellt und können somit als "Direktauswahl" angesehen werden.
Der Nachteil ist allerdings, dass diese Darstellung nur auf der "Home"-Seite, bzw. direkt nach dem Einloggen zusehen ist. Um über die "Schaltzentrale" zu arbeiten ist es daher notwendig, stets zuerst auf "Home" zu klicken, um in diese Ansicht zu kommen.

7.3. Kurzüberblick – Übersichtsfenster

Auf einem Blick informiert dieses Fenster über verschiedene Bereiche seiner installierten oder erzeugten Inhalte.

Unter dem Register *Logged* finden wir einen Überblick über alle zur Zeit eingeloggten Personen. Nach der Installation stehen vermutlich nur Sie dort und beim ersten Mal auch als "admin". Wie wir später sehen, läßt sich natürlich all das auch ändern …

Das Register *Components* gibt nicht nur einen Überblick über die derzeit installierten Komponenten, sondern auch über dessen Untermenüpunkte, die selbstverständlich auch in diesem Übersichtsfenster direkt ausgwählt werden können.
Komponenten, die keinen eigenen Menüpunkt haben, können direkt angewählt werden.
Alles was sich anklicken läßt – also sich um einen Verweis handelt – ist Orange, der Mambofarbe, dargestellt.

Unter dem Register *Popular* finden sich Artikel, die für alle lesbar sind.

Das Register *Latest Items* gibt eine Übersicht über die zuletzt angelegten Artikel wider.

Unter *menustats* finden sich die Hauptmenüs, nicht jedoch die einzelnen Menüpunkte.

7.4. Wie funktioniert aber nun Joomla? Ein erster Artikel

Zunächst erstmal einen kleinen groben Einblick über die Funktionsweise von Joomla, bevor wir uns in die Details und vielfältigen Möglichkeiten wagen:

Wesentliches Merkmal ist, dass ein Artikel zu einem bestimmten selbstbestimmten Thema angelegt wird und dann dieser Artikel im Menü aufgerufen wird.
Wobei die Reihenfolge immer die ist, dass zunächst ein Thema bestimmt wird, dann ein Unterthema. Erst danach wird der Artikel erzeugt und dann erst kann dieser im Menü verlinkt werden.

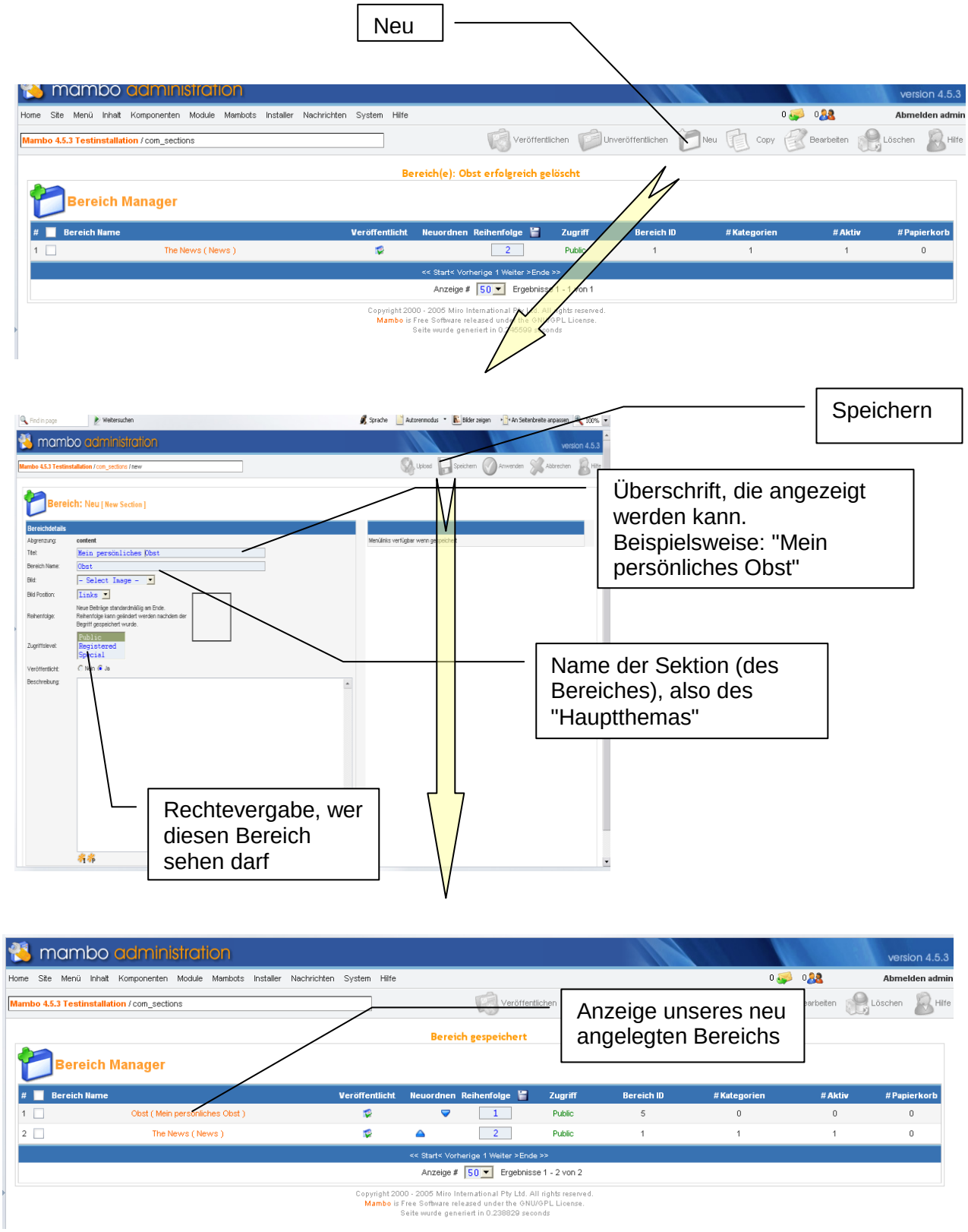
Beispiel:
Wir wollen verschiedene Artikel zu verschiedenen Obstsorten schreiben.
Dazu wählen wir zunächst als Hauptthema "Obst" aus. Als Unterthemen nehmen wir "Zitrusfrüchte" und "Exotische Früchte".
Nun können wir einen Artikel über Apfelsinen schreiben, den wir dem Unterthema "Zitrusfrüchte" zuordnen und einen Artikel über "Papayas", der dem Unterthema "Exotische Früchte" zugeordnet wird.
Verlinken können wir nun beispielsweise alle Artikel über Zitrusfrüchte (sofern wir weitere Artikel zu diesem Thema schreiben – also beispielsweise Artikel über Zitronen, Limonen und Mandarinen) und ebenso einen eigenen Menüpunkt für alle exotischen Früchte.

Und vermutlich ahnen Sie schon, dass es noch eine Vielzahl weitere Verlinkungsmethoden und damit Menüauswahlen und Darstellungen gibt, doch wollen wir uns in diesem Beispiel mit der genannten Art begnügen.

- Obst *(Bereich)*
  - Zitrusfrüchte *(Kategorie 1)*
    - Apfelsinen *(Artikel)*
    - Mandarinen *(Artikel)*
    - Zitronen *(Artikel)*
    - Limonen *(Artikel)*
  - Exotische Früchte *(Kategorie 2)*
    - Papayas *(Artikel)*
    - Ananas *(Artikel)*
    - Babaco *(Artikel)*
    - Datteln *(Artikel)*

Schauen wir nun, wie das konkret in Joomla abläuft:

Zunächst legen wir unter dem Menüpunkt *Inhalt/Bereich Manager* unseren Hauptpunkt – in Joomla auch **Sektion** oder **Bereich** genannt - an. Wir klicken oben rechts in der Iconleiste auf "Neu". Danach erhalten wir ein neues Fenster, in dem wir unseren Obstbereich anlegen. Für unser kleines Beispiel reicht es, wenn wir "Titel" und "Bereichsname" ausfüllen. Wobei "Titel" erfoderlich ist. Alles andere ist optional. Den "Zugriffslevel" lassen wir auf **Public** stehen, damit alle unsere Obstartikel nachher auch sehen können.



Abbildungen: Bereich erstellen
1. Bereich Manager – 2. Bereich erstellen – 3. Bereich Manager mit neu angelegten Bereich

Nach dem Speichern kommen wir also wieder zu unserem ersten Bildschirm zurück, nur das wir jetzt neben den Bereich "News" (der sich übrigens nicht löschen läßt!) auch unseren Bereich "Obst" angezeigt bekommen.

Nun rufen wir über das Pulldownmenü den Menüpunkt *Inhalt/Kategorie Manager* auf.

Das Fenster unterscheidet sich nicht sonderlich von dem des "Bereich Managers". Auch hier klicken wir wieder auf "Neu" bei den großen Icons und können nun eine Kategorie anlegen. Der entscheidene Unterschied zum Bildschirm "Bereich anlegen" ist, das wir hier einen Pulldownbutton haben, bei dem wir einen Bereich auswählen müssen.

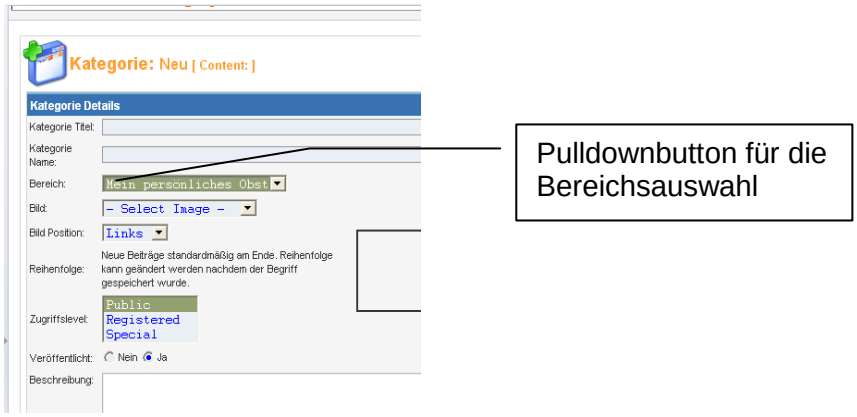


Abbildung: Kategorie anlegen

Wir erhalten mindestens die "News" und den Titel, den wir zuvor für unser Obst eingegeben haben, angezeigt.
Nachdem wir in Kategorie-Titel und Kategorie-Name unsere "Zitrusfrüchte" eingetragen haben, wählen wir als Bereich natürlich unser Obst aus.
Nun noch auf Speichern drücken und das ganze für die exotischen Früchte wiederholen:
Neu → In Kategorie-Titel und Kategorie-Name "Exotische Früchte" eintragen → "Obst" als Bereich auswählen → Speichern

Als "Belohnung" erhalten wir im Managerfenster unsere beiden Kategorien (zusammmen mit den bereits angelegten Kategorien (also mindestens der nicht löschbaren News-Kategorie)) angezeigt:

Abbildung: Kategorie Manager nach Eingabe der Kategorien

Wenn wir nun zum Pulldownmenü *Inhalte* gehen und mit dem Mauszeiger über *Inhalte nach Bereich* gehen, sehen wir unser Obst, zusammmen mit den News und eventuell noch anderen Bereichen angezeigt.
Wollen wir auch sogleich einen Artikel anlegen. Also rufen wir nachfolgenden Menüpunkt auf:

Inhalt / Inhalt nach Bereich / Mein persönliches Obst / Mein persönliches Obst Einträge

In dem sich nun öffnenden Fenster steht zwar nichts drin, unterscheidet sich vom Aufbau aber nicht von den bereits kennengelernten Managerfenstern.
Klicken wir also auf das große Icon "Neu" um nun endlich einen Artikel in unser Content Management System zu schreiben.

Das nun folgende Fenster deutet bereits an, dass es eine Vielzahl von Einstellungs- und Anzeigemöglichkeiten für Artikel gibt. Für unser erstes Beispiel wollen wir uns zunächst auf das Wesentliche konzentrieren - um Details kümmern wir uns später …

Unter **Titel** lege ich erstmal die Überschrift des Artikels fest. Unter **Titel Alias** wird zumeist eine Kurzform (sofern die Überschrift länger ist) eingetragen. Dieser Alias wird innerhalb des Backends verwendet um den Artikel zu kennzeichnen.
Danach wählen wir unter **Bereich** "Mein persönliches Obst" und unter den **Kategorien** "Zitrusfrüchte" aus.
Also all das, was wir zuvor an anderer Stelle angelegt haben an Bereichen und Kategorien taucht hier wieder auf und damit kann ein Artikel einwandfrei zugeordnet werden.
Es entsteht also somit eine Hirarchie für Artikel.

Im **Intro Text** geben wir nun unseren Artikel ein. Also wissenwertes rund um Apfelsinen oder anderen Zitrusfrüchten.
Nun noch rechts unter dem Register **Publishing** einen Haken neben "Veröffentlicht" setzen (zumeist voreingestellt) und darauf achten das die **Zugriffsrechte** auf Public stehen. Nun noch auf das große Icon **Speichern** klicken und fertig ist unser erster Artikel!

Weitere Artikel erstellen wir nach dem gleichen Prinzip.

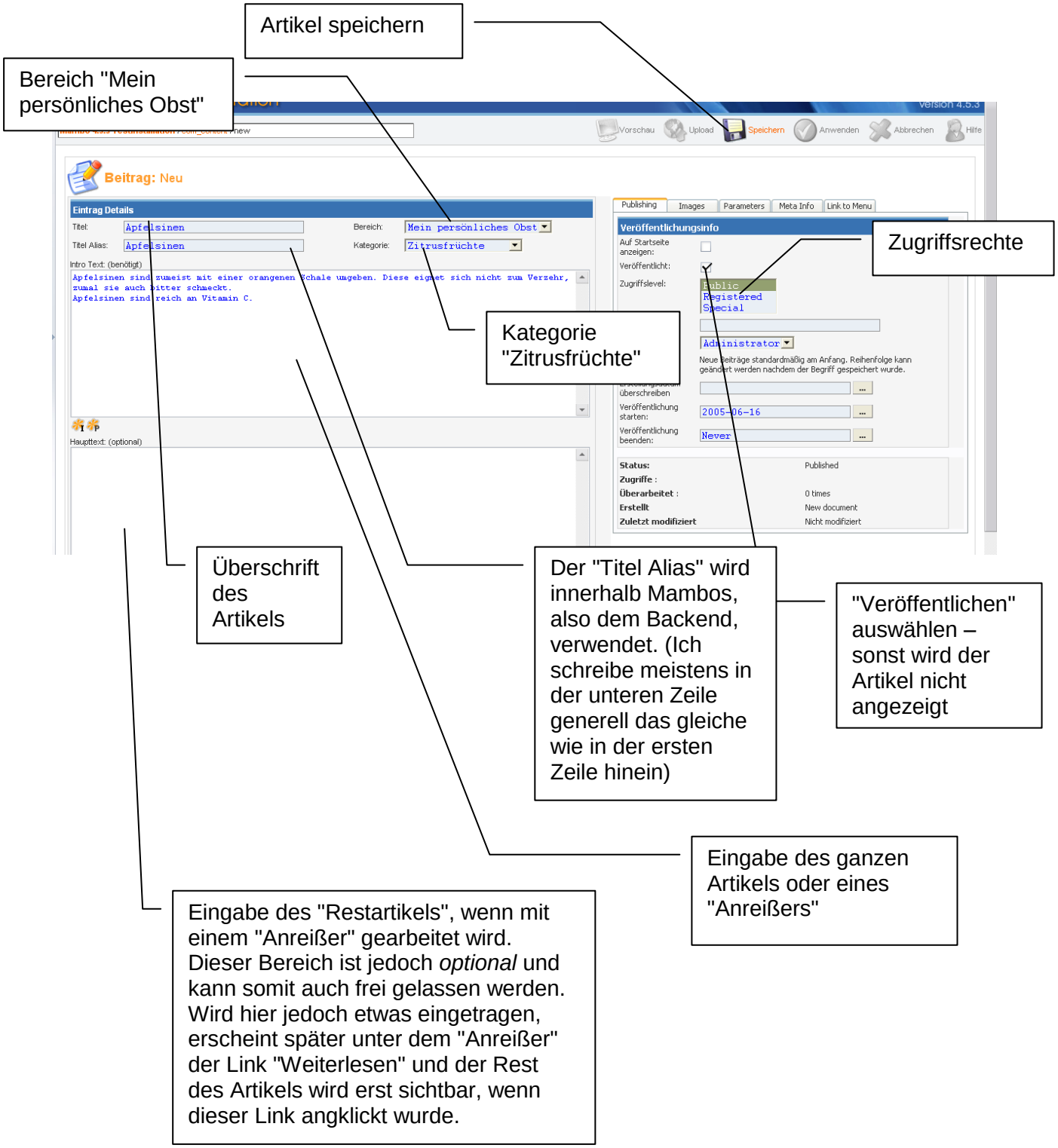


Abbildung: Artikel anlegen

Nun müssen wir noch dafür sorgen, dass unsere Obstartikel auch über die Navigation auf unserer Webseite gefunden werden.

Dazu legen wir in unserem Beispiel ein komplett neues Menü an.

Über das Pulldownmenü rufen wir den **Menümanager** auf:
Menü / Menü Manager
Im Manager werden alle angelegten Hauptmenüs angezeigt. Voreingestellt sind die Menüs

- mainmenu
- othermenu
- topmenu
- usermenu

Diese Voreinstellungen kann man übernehmen oder aber auch löschen und sich eigene Menüs anlegen.

Wir wählen, wie gesagt, letzteres:

Zunächst klicken wir wie üblich auf das große Icon **Neu**.

Abbildung: Anlegen eines neuen Hauptmenüs

In *Menu Name* tragen wir ein, wie wir unser Menü innerhalb Mambos nennen und in *Menu Titel* wie unser Menü auf unserer Webseite angezeigt werden soll. Danach auf *Save* klicken und wir haben ein neues Hauptmenü.

In der Version 4.5.3 können wir im Menümanager direkt auf das kleine Icon **Menüeinträge** klicken, um nun Untermenüeinträge erstellen zu können. Desgleichen können wir aber auch über das Pulldownmenü aufrufen:
Menu / Obst

Wieder haben wir das nun bereits bekannte Managerfenster vor uns, in dem natürlich noch nichts steht. Schließlich haben wir ja noch keine Menüpunkte anglegt.

Ein Klick auf das große Icon **Neu** bringt uns in das nächste Fenster, in dem wir angeben müssen, in welcher Art wir was einbinden möchten.

Auch hier wollen wir uns zunächst auf unser Beispiel konzentrieren und wählen einfach mal **Blog Content Category** aus. Damit werden alle Artikel, die einer bestimmten Kategorie zugeordnet sind, angezeigt.
Was die anderen Auswahlmöglichkeiten bedeuten werden wir weiter unten ansprechen. Zunächst soll uns dieses genügen, schließlich wollen wir ganz schnell unsere Artikel sichtbar machen …
Anschließend klicken wir auf das große Icon oben rechts "Next" (wahlweise direkt auf "Blog – Content Category").

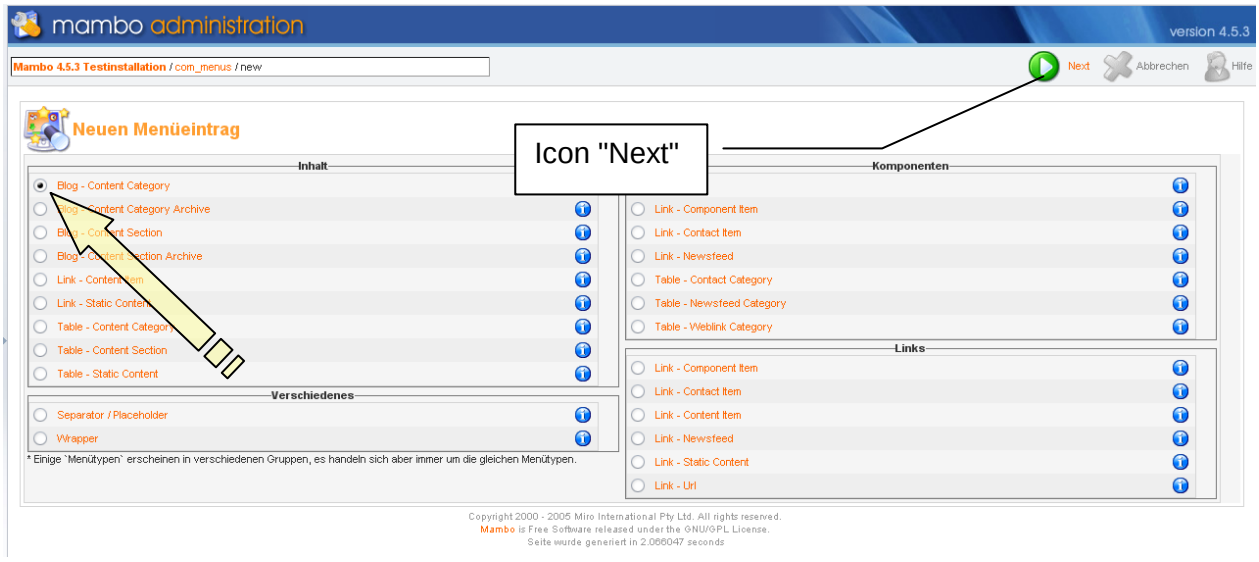


Abbildung: Neuer Menüeintrag – Art des Eintrages

Wir kommen nun zu dem Fenster, in dem wir nähere Angaben zu unserer vorangegangenen Wahl der Art des Eintrages machen müssen.
Dieses Fenster sieht schon richtig kompliziert aus - aber keine Angst, wir nähern uns langsam der Materie …

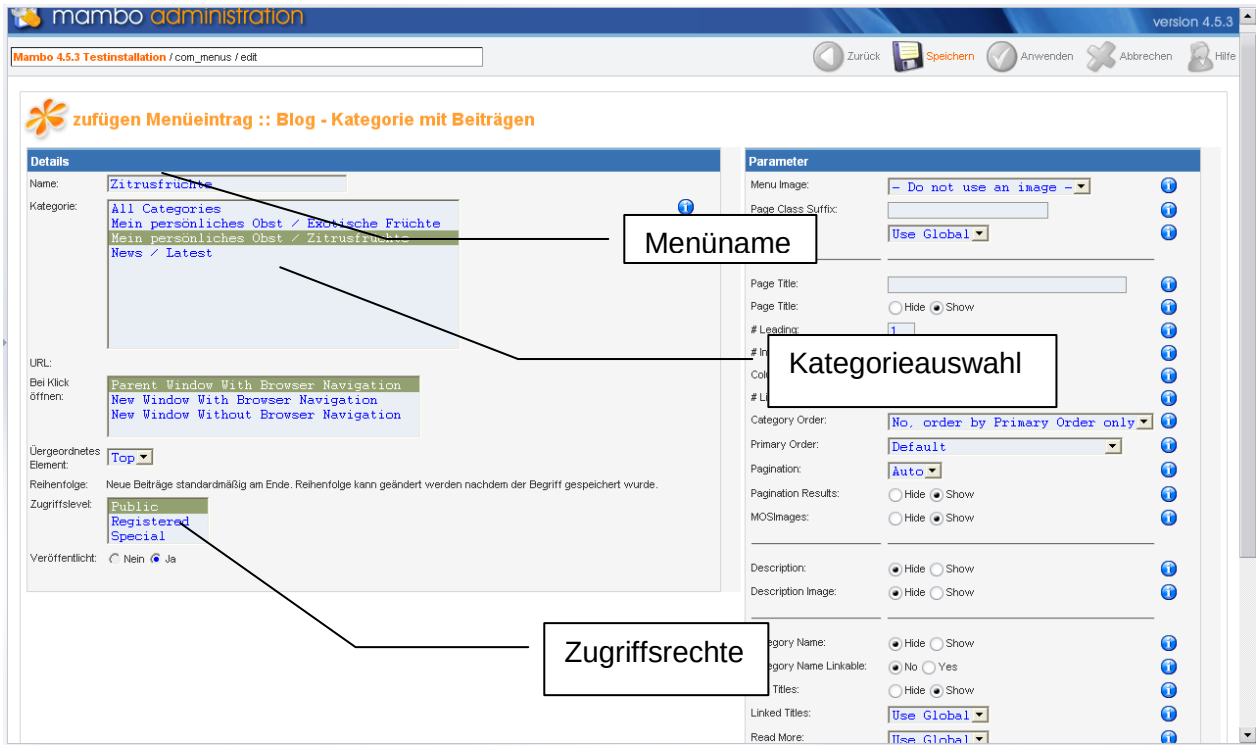


Abbildung: Menüeintrag zufügen

Was unter **Name** steht, wird dann auch im Menü angezeigt. Also beispielsweise "Zitrusfrüchte"
Unter **Kategorie** finden wir unter anderem unsere Obstkategorien wieder. Hier wählen wir "Mein perönliches Obst/Zitrusfrüchte" für unser Beispiel aus.

Die **Zugriffsrechte** belassen wir auf das voreingestellte *Public* (sollte das nicht markiert sein, bitte ändern).
Jetzt noch auf das große Icon **Speichern** und wir landen wieder im Menümanager, wo jetzt unsere Zitrusfrüchte als Menüeintrag angezeigt werden.

Um den Menüeintrag "Exotische Früchte" anzulegen verfahren wir noch einmal wie beschrieben:
Neu → Blog Content Category → Next → Name: "Exotische Früchte" → Dazugehörige Kategorie auswählen → Zugriffslevel: Public → Speichern

Nun haben wir zwei Menüpunkte. Der eine mit Namen "Zitrusfrüchte", der andere mit Namen "Exotische Früchte".

Wenn wir nun die Vorschau aufrufen würden, würden wir unser schönes Menü nicht zu Gesicht bekommen. Denn zuvor müssen wir das Menü selber noch veröffentlichen.
Unser Hauptmenü ("Obst") hat Mambointern den Status eines Moduls. Und ein Modul finden wir stets im Modulmanager und dort muß es auch explizit veröffentlicht werden.

Klicken wir uns also im Pulldownmenü zum Modulmanager:
Module / Site Module

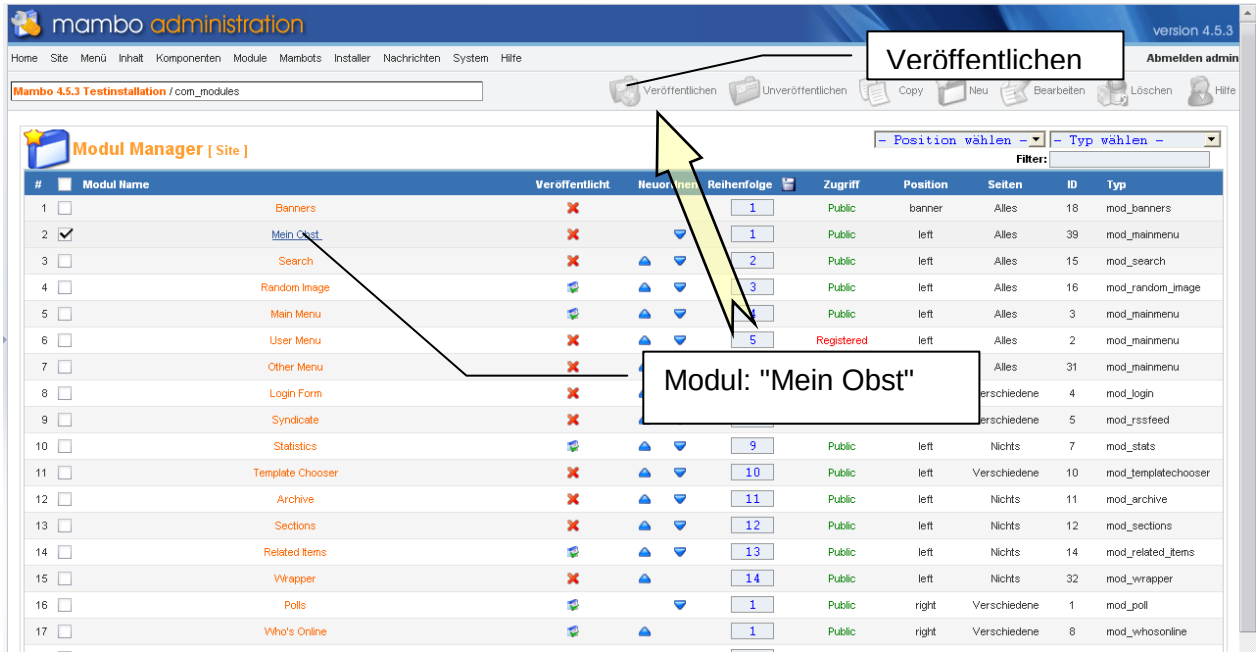


Abbildung: Modul Manager

Nun markieren wir einfach unser Modul, in dem wir ein Häkchen setzen und wählen bei den großen Icons oben rechts gleich das erste aus: **Veröffentlichen**.

Natürlich gibt es auch hier eine Vielzahl weiterer Einstellungsmöglichkeiten, aber für unser Beispiel genügt dieses zunächst.

Jetzt können wir uns unser Menü mit den dazugehörigen Artikeln anschauen!
Dazu rufen wir die Vorschau auf:
Site / Vorschau / In neuem Fenster

Wenn Sie alles so gemacht haben, wie im Beispiel, sehen Sie nun Ihr Menü mit den beiden Menüeinträgen "Zitrusfrüchte" und "Exotische Früchte" und wenn Sie einen der Menüpunkte anklicken, erhalten Sie die dazugehörigen Artikel.

Da es mittlerweile sehr viele unterschiedliche Mamboinstallationsroutinen gibt und somit auch viele unterschiedliche Webtemplates in der Standardausführung, kann Ihr Aussehen gravierend von meiner Abbildung abweichen. Die Funktion allerdings ist überall die gleiche. Und damit haben Sie gleich einen wesentlichen Punkt von Joomla: Egal wie Ihr Template aussieht – Menüs, Komponenten und Module können in jedem Design exakt übernommen / dargestellt werden.

Abbildung: Website mit Obstmenü (Beispielmenü)

7.5. Zusammenfassung

Wollen wir uns die letzten Seiten noch einmal als kleine Zusammenfassung anschauen:

1. Anlegen von Hauptthemen (Bereiche)
2. Anlegen von Unterthemen (Kategorien) zum jeweiligen Bereich
3. Artikel erstellen und Bereiche und Kategorien zuordnen
4. Menühauptthema erstellen
5. Menüunterthemen erstellen und verlinken
6. Das Menü im Modulmanager freischalten

Wenn wir das aufmerksam lesen, werden wir feststellen, dass Menüs sehr ähnlich wie ein Artikel erstellt werden. Statt Bereiche und Kategorien müssen Hauptmenüpunkte und Untermenüpunkte erstellt werden. Vom Prinzip her sehr ähnlich.

8. Die Manager

[Die Manager werden hier grundsätzlich vorgestellt. Funktionsweise und Überblick.]

9. Bereiche und Kategorien

Bereiche und Kategorien dienen natürlich nicht nur zur einwandfreien Kategorisierung von Artikeln, sondern bieten darüber hinaus auch unterschiedliche Methoden der Anzeige.

Allerdings gibt es hier einen kleinen Wermutstropfen:
Es ist in Mambo nicht möglich Unterkategorien anzulegen. Also stets nur ein Bereich mit einer oder mehreren Kategorien, aber diese Kategorien können keine Unterkategorien enthalten.

9.1. Bereiche

Gehen wir in den *Bereich Manager* (erreichbar über das Pulldownmenü *Inhalt / Bereich Manager*) und editieren unseren "Bereich Obst". Dazu markieren wir entweder den Bereich, in dem wir ein Häkchen setzen und über das Icon-Menü **Bearbeiten** aufrufen oder ganz einfach auf den orangenfarbenden Text von dem Bereich Obst klicken.
Nun haben wir wieder das gleiche Fenster, wie wir es beim Anlegen des Bereiches hatten, nur das natürlich schon da steht, was wir eingetragen hatten.

Nun können wir zusätzlich ein Bild auswählen, welches zusammen mit unserem Bereich angezeigt werden soll. Dazu einfach im Combobutton ein Bild auswählen. Eigene Bilder können in das Verzeichnis *images* übertragen werden.

Mit **Bild Position** legen wir den Textumfluß fest. Wobei *links* bedeutet, dass der Text rechts um das Bild fließt und umgekehrt. Mitte richtet das Bild mittig über den Text stehend aus.

Über **Reihenfolge** können wir festlegen an welcher Stelle im Manager unser Bereich angezeigt wird.
Wer viele Bereiche hat möchte diese vielleicht im Manager entsprechend sortieren, um den Überblick zu behalten. Wenn bereits viele Bereiche vorhanden sind und ein neuer Bereich hinzugefügt wird, kann es sehr mühselig sein, die Pfeiltasten im Manager zur Positionierung zu benutzen. Allerdings steht diese Zuweisung erst nach dem ersten Speichern zur Verfügung, bzw. nach dem ersten editieren. Dabei gilt, dass der Bereich VOR dem ausgewählten angezeigt wird.

Wirklich Interessant ist allerdings der rechte Teil dieses Fensters: **Verlinken im Menü**.
Auch dieses steht uns erst nach dem ersten Speichern und editieren zur Verfügung.

Abbildung: Bereich bearbeiten → Verlinken im Menü

Wir können nun direkt von hier aus einen Menüpunkt anlegen und dabei alle Arten der Menüverlinkung auswählen, die für einen Bereich möglich sind.

Um dieses zu tun, wird zunächst ein Menü ausgewählt, in dem der Menüeintrag erscheinen soll. Da wir zuvor ein Hauptmenü mit Namen "Obst" angelegt haben, wählen wir auch dieses Menü aus. Darunter geben wir nun den Typ an:

- Bereichsliste
  - Gibt den Bereich mit allen seinen Kategorien aus und zeigt dabei die Anzahl der Artikel an, die in der zugehörigen Kategorie geschrieben wurden.
- Bereich Blog
  - Gibt alle Artikel aus, die zu dem Bereich geschrieben wurden, ungeachtet der Kategorien.
- Bereich Archiv Blog
  - Gibt alle archivierten Artikel zu dem Bereich aus.

Um sich die unterschiedlichen Bereichsanzeigen mal anzuschauen, legen wir einfach nacheinander jede Menüart an.
Also im Menü "Obst" wählen wir zuerst die Menüart *Bereichsliste*. Der Einfachheit halber nennen wir unseren Menüpunkt am besten auch genauso: "Bereichsliste". Schließlich wollen wir ja sehen, was passiert. Das alles wird nun gespeichert indem wir den Button **Verlinken im Menü** anklicken. Damit wird nun automatisch der entsprechende Menüpunkt nach unseren Vorgaben angelegt, ohne das wir uns weiter darum kümmern müssen.
Unten wird unsere Auswahl noch einmal zur Kontrolle und zum Überblick angezeigt: Welches Menü wir augewählt hatten, die Art, den Namen und für wen der Menüpunkt zugänglich ist. Wenn wir nichts verändert haben, dann müßte dort "Public" stehen – also für jeden User, der unsere Seite besucht, ist dieser Link benutzbar. Eingestellt wird der **Zugriffslevel** im linken Fensterteil.
Danach wählen wir die Menüart *Bereich Blog* aus und nennen unseren Menüeintrag am besten auch genau so und klicken wieder auf den Button **Verlinken im Menü**.
Unten wird nun ebenfalls unsere Auswahl angezeigt.
Das ganze machen wir noch ein drittes Mal und wählen dieses mal *Bereich Archiv Blog* aus.

Abermals sehen wir unten eine Zusammenfassung unserer Auswahl und können so aber vor allem jederzeit erkennen, ob und wie wir diesen Bereich per Menü verlinkt haben.

Jetzt können wir einfach die Vorschau aufrufen (Pulldownmenü: *Site / Vorschau / In neuem Fenster*).

Wenn wir alles richtig gemacht haben, sehen wir drei neue Menüpunkte, die unseren Bereich unterschiedlich darstellen.

Abbildung: Die drei Bereichmenüs

Wenn wir allerdings auf "Archiv Blog" klicken sehen wir lediglich den Hinweis, dass es keine archivierten Beiträge gibt.
Wollen wir das zunächst so stehen lassen und erst weiter unten, wenn wir uns näher mit den Artikeln befassen, wieder hierher zurückkehren. Nur soviel sei gesagt: So mancher Artikel ist aus irgendwelchen Gründen nicht mehr aktuell. Er kann dann ganz einfach archiviert werden und braucht nicht gelöscht zu werden (wer weiß, wozu wir den Artikel noch einmal brauchen) Unter anderem kann er in dieser Art und Weise wieder angezeigt werden. Damit ist es möglich, auch auf "alte" Artikel jederzeit zugreifen zu können.

Wenn wir nun im Pulldownmenü das Menü "Obst" aufrufen (*Menü / Obst*), dann finden wir selbstverständlich dort auch unsere drei angelegten Bereichs-Menüpunkte wieder.
Das wir diese natürlich dort editieren und alles einstellen können, was man für Menüs einstellen kann, brauche ich vermutlich nicht extra zu erwähnen …

9.2. Kategorien

Gehen wir in den Kategorie Manager (Pulldownmenü *Inhalt / Kategorie Manager*) und editieren eine Katagorie. Nehmen wir zunächst die Zitrusfrüchte.

Das Fenster für die Kategorien sieht eigentlich genauso aus, wie das Fenster für die Bereiche. Lediglich bei den Menüarten gibt es ein paar Unterschiede. Außerdem steht in der Menüübersicht ("Existierende Menülinks") bereits unser angelegter Menüpunkt mit dem

Namen "Zitrusfrüchte". Vom Typ wird uns *Kategorie Blog* angezeigt. Als wir den Menüpunkt anlegten, wählten wir "Blog – Content Category" aus. Leider bietet uns Mambo für identische Dinge manchmal unterschiedliche Namen an. Und leider, das werden Sie sicherlich längst bemerkt haben, werden wir immer wieder einen Mischmasch aus deutschen und englischen Bezeichnungen haben.

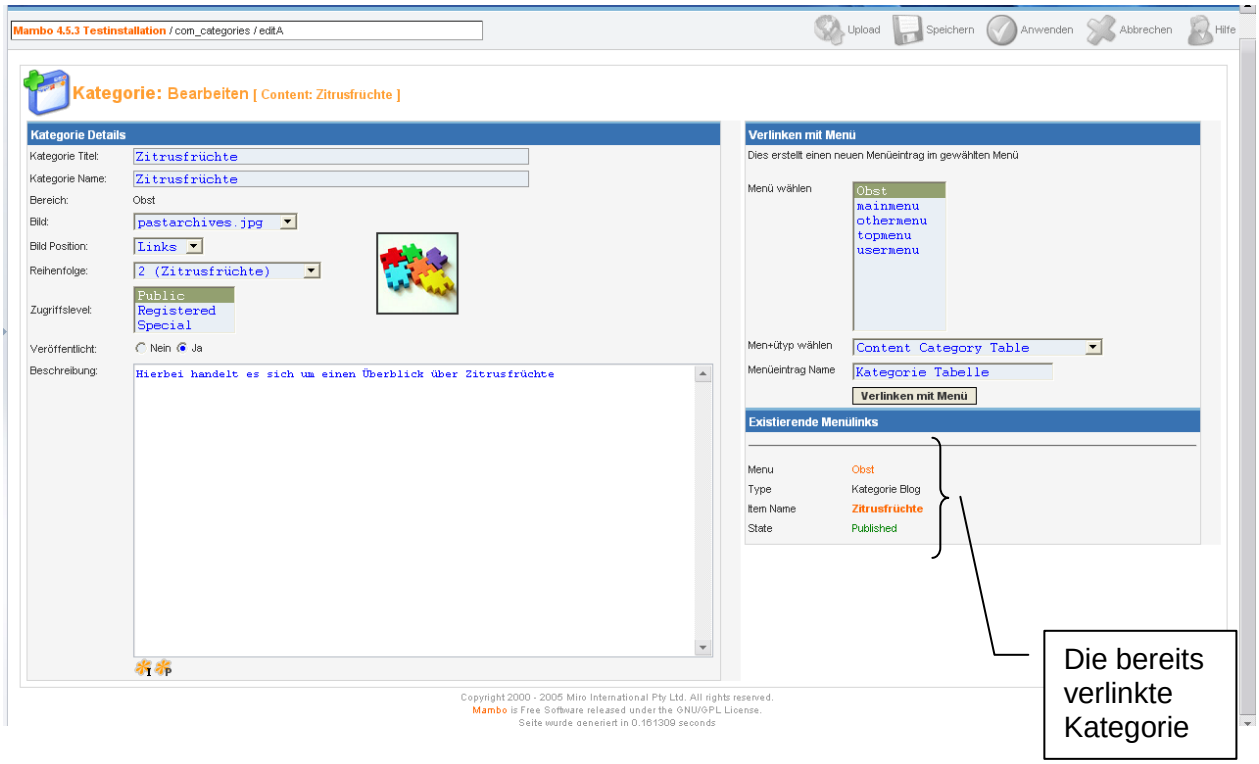


Abbildung: Kategorie bearbeiten

Der Menütyp **Content Category Table** bedeutet, dass im Frontend zunächst eine Übersicht aller Artikel, die unter dieser Kategorie geschrieben wurden, angezeigt wird. Durch klicken auf einen Artikelnamen wird dieser angezeigt. Außerdem wird dem User die Möglichkeit gegeben, die Artikel nach unterschiedlichen Kriterien zu sortieren.
Wer umfangreichen Inhalt hat, kann hiermit dem User eine interessante Möglichkeit bieten nach bestimmten Artikeln zu suchen.

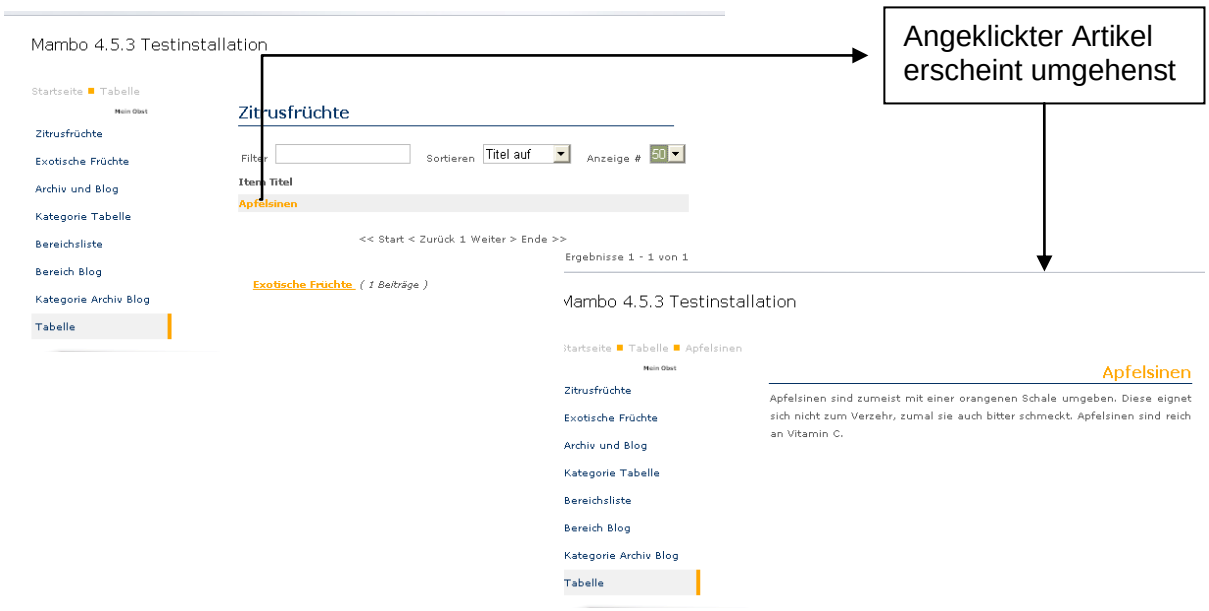


Abbildung: "Content Category Table" im Menü eingebunden und nach Aufruf eines Artikels aus der Tabelle

Der **Content Category Blog** zeigt alle Artikel aus dieser Kategorie untereinander an.
Und mit **Content Category Archiv Blog** werden alle bislang archivierten Artikel aus dieser Kategorie angezeigt.

9.3. Der Bereich Manager

Der Bereich Manager bietet uns nicht nur einen Überblick über unsere Bereiche, sondern beinhaltet noch weitere Informationen:

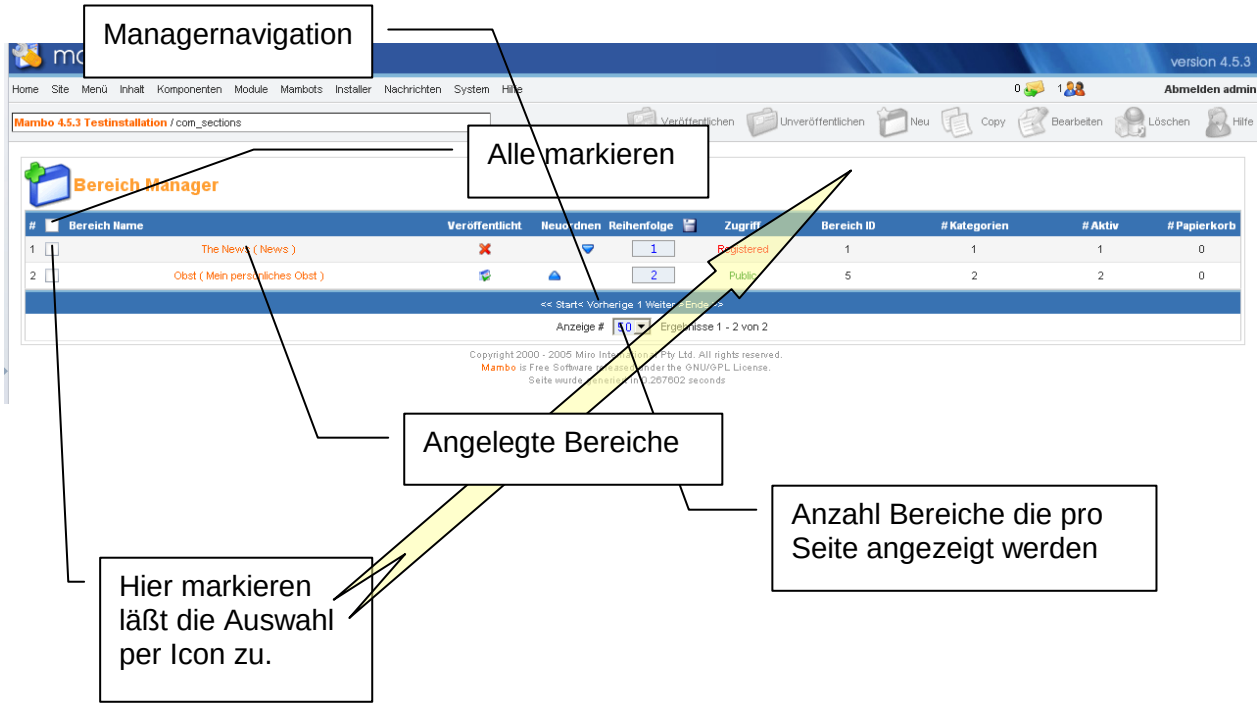


Abbildung: Bereich Manager

Die Spalte **Veröffentlicht** gibt uns Aufschluß, ob der Bereich überhaupt veröffentlicht wurde oder zwar angelegt, aber für niemanden zugänglich ist. Um dieses zu ändern kann das Icon (Kreuz- oder Fenstericon) angeklickt werden. Es wird dann der jeweils nächste Zustand eingestellt.

> *Hinweis*: Sind zwar die zugehörigen Kategorien oder Artikel auf *Veröffentlicht* gesetzt, aber die Bereiche auf *unveröffentlicht*, dann werden die Kategorien und Artikel nicht angezeigt!

In der Spalte **Zugriff** ist auf einen Blick ersichtlich, welcher Bereich für wen benutzbar ist, also welcher Zugriffslevel eingestellt ist. Auch hier kann direkt das Wort angeklickt werden, um den jeweils nächsten Zustand einzustellen.

Die **Bereich ID** wird von Mambo zur Identifikation des Bereichs benutzt und ist z.B. für Programmierer interessant.
Unter **#Kategorien** wird uns angezeigt, wieviele Kategorien zu diesem Bereich gehören. Sollten wir also mal vergessen eine Kategorie zu einem Bereich anzulegen, können wir das hier ganz schnell ersehen.
So gibt uns dann **#Aktiv** an, wieviele dieser Kategorien auch aktiv sind oder derzeit nicht "veröffentlicht", also nicht "published" sind.
Unter **#Papierkorb** findet sich die Anzahl der Kategorien, die von uns mal gelöscht wurden, allerdings noch im Papierkorb liegen (siehe weiter unten).

Desweiteren können wir unter **Neuordnen** die Reihenfolge der Anzeige der Bereiche im Bereich Manager verändern. Wer viele Bereiche hat, möchte diese vielleicht sortieren. Desgleichen können wir auch mit den Ziffern unter **Reihenenfolge** machen. Dazu einfach die Zahlen mit der neuen Reihenfolge in die Kästchen schreiben und dann auf das "Speichernsymbol" neben der Spaltenüberschrift klicken.

> *Hinweis*: Nicht zwischendurch ENTER drücken – denn dann steht wieder alles in der ursprünglichen Reihenfolge da!

Um einen Bereich zu editieren entweder direkt den Bereichsnamen anklicken oder vorn ein Häkchen setzen und dann in der Iconleiste *Bearbeiten* auswählen.

Desweiteren gibt es auch die Möglichkeit auf diese Art vorhandene Bereiche zu kopieren oder zu verschieben. Also den zu kopierenden oder verschiebenen Bereich mit einem Häkchen markieren und das entsprechende Icon auswählen.
Beim **Copy** eines Bereichs wird ein neues Fenster geöffnet, in dem die Information steht, welche Artikel und welche Kategorien dem zu kopierenden Bereich zugeordnet ist und es kann ein neuer Bereichsname eingetragen werden.

> *Hinweis*: Werden mehrere Bereiche ausgewählt, die kopiert werden sollen, werden zwar alle Kategorien und Artikel angezeigt, allerdings wird ein leerer Bereich erstellt. Offenbar ein Bug in Mambo …

Möchte man einen Bereich löschen, geht dieses nur, wenn zuvor alle Artikel und dann alle zugehörigen Kategorien gelöscht werden.

> *Hinweis*: Ein Bereich mit noch mindestens einer zugehörigen Kategorie kann nicht gelöscht werden!

Ansonsten kann man aber mehrere oder alle Bereiche auswählen und diese dann alle *veröffentlichen* oder *unveröffentlichen*.
Um **alle Bereiche auszuwählen** kann in der ersten Spalte das Kästchen neben dem Nummernzeichen ("#") angeklickt werden – es werden dann alle Bereiche, die auf der Seite angezeigt werden, ausgewählt. Nicht jedoch Bereiche, die sich auf einer anderen Seite befinden. Also: Sind beispielsweise 15 Bereiche angelegt, es ist aber eingestellt, dass nur 10 Bereiche pro Seite angezeigt werden sollen, dann werden auch nur die ersten 10 oder befindet man sich auf der 2. Seite, die letzten fünf Bereiche mit einem Häkchen markiert.

Ganz unten kann eingestellt werden, wieviele Einträge auf der Seite angezeigt werden sollen.
Wenn mehr Bereiche vorhanden sind, als auf einer Seite angezeigt werden sollen/können, dann läßt sich unten vor- oder zurückblättern.

9.4. Der Kategorien Manager

Für den Kategorie Manager gilt im großen und ganzen das Gleiche, wie für den Bereich Manager. Allerdings gibt es die Zusatzinformation, zu welchem Bereich eine Kategorie gehört.
Desweiteren gibt es bei den Kategorien noch ein Icon zusätzlich: **Move**
Move arbeitet ähnlich wie **Copy**. Bei Move werden alle Artikel und deren Kategorie, zu einem bereits bestehenden anderen Bereich verschoben.
Und bei **Copy** werden die Kategorie und die zugehörigen Artikel in einen vorhandenen Bereich kopiert.
Bei der Mambo-Version 4.5.3 findet sich beim "Move-Fenster" ein kleiner Übersetzungsfehler im Hilfetext. Aber ich denke, es ist dennoch klar, was hier gemeint ist.
Mit "gelistete" sind veröffentlichte Beiträge gemeint.

Um eine Kategorie zu **löschen**, müssen zunächst alle zugehörigen Artikel gelöscht werden!
Eine weitere Navigationshilfe ist, dass im Kategorie Manager per Pulldownbutton der Bereich vorgewählt werden kann. Es werden dann nur die Kategorien angezeigt, die zum ausgewählten Bereich gehören.

Abbildung: Kategorie Manager

10. Artikel – Inhalt – Beiträge - Content

Das ist das eigentliche Herzstück von Mambo: Der Artikel.
Wobei unter Artikel alles verstanden wird, was mit Text zu tun hat. Mit Ausnahme der Menüs. Alles andere findet sich hier wider. Manchmal als "Content" bezeichnet, manchmal lapidar ins Deutsche übertragen als "Inhalt" oder auch als "Beiträge" – letztlich ist es immer eine Form von Artikel. Und wenn auch "nur" der Copyrightvermerk dort drin steht …
Diese Artikel können wir auf unterschiedlicher Art und Weise einbinden. Eine Variante haben wir uns bereits angeschaut, nun wird es Zeit, tiefer in die Materie einzutauchen.

10.1. Inhalte nach Bereich – Neue Kategorie

Bleiben wir zunächst bei der Art von Artikeln, wie wir sie bereits kennengelernt haben und legen uns ein neues Beispiel an. Unter den nicht löschbaren "News" erstellen wir eine neue Kategorie und legen uns dort neue Inhalte (Artikel) an.
Anders als wir es zuvor gelernt haben, wollen wir nun die neue Kategorie nicht über den "Kategorie Manager" anlegen, sondern direkt im "News-Bereich". Dazu rufen wir den nachfolgenden Menüpunkt über das Pulldownmenü auf:

Inhalt / Inhalte nach Bereich / News / Hinzufüren/Bearbeiten News Kategorien

Im sich nun öffnenden Kategorie Manager steht bereits ein Eintrag – *Latest News (News)*
Wobei dieser Kategorie Manager indentisch mit dem, der über den direkten Aufruf sich öffnet, ist, allerdings nur und ausschließlich die Kategorien anzeigt, die zum ausgewählten Bereich gehören.
Auf diese Art lassen sich schnell alle zu einem Bereich gehörenden Kategorien anzeigen und für einen entsprechenden Überblick sorgen.
Wählen wir nun über das große Icon *Neu* aus, erscheint wieder das bekannte Fenster zum Kategorieanlegen, allerdings ist der Bereich in diesem Fall ebenfalls vorgegeben. Wer viele Bereiche hat, läuft mit diesem Aufruf nicht Gefahr, eine Kategorie für den falschen Bereich anzulegen.
Legen wir nun also, wie oben beschrieben, eine neue Kategorie mit dem Namen "Meine News" an.

10.2.Inhalte nach Bereich – Artikel anlegen

Nun gehen wir per Pulldownmenü zu den Einträgen (also: *Inhalt / Inhalt nach Bereich / News / News Einträge*). Im **Beitrag Manager [Sektion: News]** angekommen, wird dort vermutlich schon ein Willkommenstext stehen. Wir wollen jedoch einen neuen Artikel anlegen und klicken bei den großen Icons das Icon **Neu** an.

Wir wollen einen Artikel anlegen, den wir "Seite Online" nennen und unter der Kategorie "Meine Neuigkeiten" ablegen des Bereiches "News".
Obwohl wir einen neuen Artikel im Bereich Manager "News" augerufen haben, haben wir dennoch die Möglichkeit den Artikel einem anderen Bereich zuzuordnen. Also bietet uns Mambo hier einen sehr großen Komfort an.
Wenn wir einen anderen Bereich auswählen, als dem, in dem wir uns befinden, dann wird unser Artikel nach dem Speichern nicht angezeigt, da Mambo zum zuvor ausgewählten Bereich zurückkehrt. Würden wir also beispielsweise als Bereich "Unser Obst" auswählen, würde der Artikel natürlich dann auch diesem Bereich zugeordnet und nur dort im Manager angezeigt. Wir befinden uns aber derzeit im Bereich "News". Ergo müßten wir dann den Bereich Manager wechseln, um unseren Artikel wiederzufinden.

Aber wir wollen unseren Artikel im Bereich "News" anlegen und der Kategorie "Meine Neuigkeiten" zuordnen.

10.2.1. WYSIWYG – Pro und Contra

Es gibt zwei Möglichkeiten in Mambo Artikel zu formatieren. Die eine ist mit einem WYSIWYG-Editor (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) – also einem Editor, der direkt im Eingabefeld anzeigt, wie die Formatierung exakt aussieht oder man verzichtet auf einen Editor und gibt die Formatierungen per HTML-Tags selber ein.

Welche Form benutzt wird, hängt sowohl von logischen, als auch individuellen Entscheidungen ab.
Wenn viele User Artikel schreiben, sollte ein WYSIWYG-Editor benutzt werden. Wenn der Admin selber in erster Linie für die Artikel zuständig ist und man sich mit HTML nicht so gut auskennt, empfiehlt sich ebenfalls der Editor. Wer aber mit HTML einigermaßen sicher ist,

findet bei Verzicht eines Editors mehr Möglichketen zur Formatierung. Außerdem ist es dann auch einfach, eigenen Code, wie beispielsweise JavaScript oder PHP-Elemente in seinen Artikel einzufügen. Wird dieses über den WYSIWYG-Editor gemacht (per umschalten auf Code-Eingabe), wird beim erneuten aufrufen des Artikels ein Teil oder auch der ganze Code der eigenen Elemente vom WYSIWYG-Editor entfernt, weil er nicht in der Lage ist, diesen zu interpretieren. Statt ihn dann einfach unangetastet zu lassen, wird dieser Code weitgehenst entfernt.
Ebenfalls erzeugen die WYSIWYG-Editoren sehr viel "Code-Müll". Insbesondere bei vielen längeren Artikeln entstehen auf diese Art lange Files, die schon so manche Website zu einer langsam ladenen Site gemacht haben.

Wer also weiterreichende Möglichkeiten in seinen Artikeln einbauen oder schlanke Files erzeugen möchte und weitgehnst allein an der Pflege der Seiten arbeitet, sollte sich sehr genau überlegen, ob er einen WYSIWYG-Editor benutzt.
Wer sich einmal die Mühe macht und in einem der Fachforen das Suchwort *editor* eingibt, wird auch schnell erkennen, dass es nicht wenige Fehler im Zusammenhang der WYSIWYG-Editoren gibt …

10.2.1.1. Artikel formatiern mit WYSIWYG-Editoren

Trotz oben genannter Einschränkungen sind die derzeit vorhandenen WYSIWYG-Editoren, die es für Mambo gibt, sehr leistungsstark. Direkt nach der Installation steht der TinyMCE-Editor zur Verfügung. Aktiviert wird er über die Globale Konfiguration. Die öffnen wir über das Pulldownmenü: *Site / Globale Konfiguration*.
Das erste Register "Site" läßt uns die WYSIWYG-Editoren einstellen

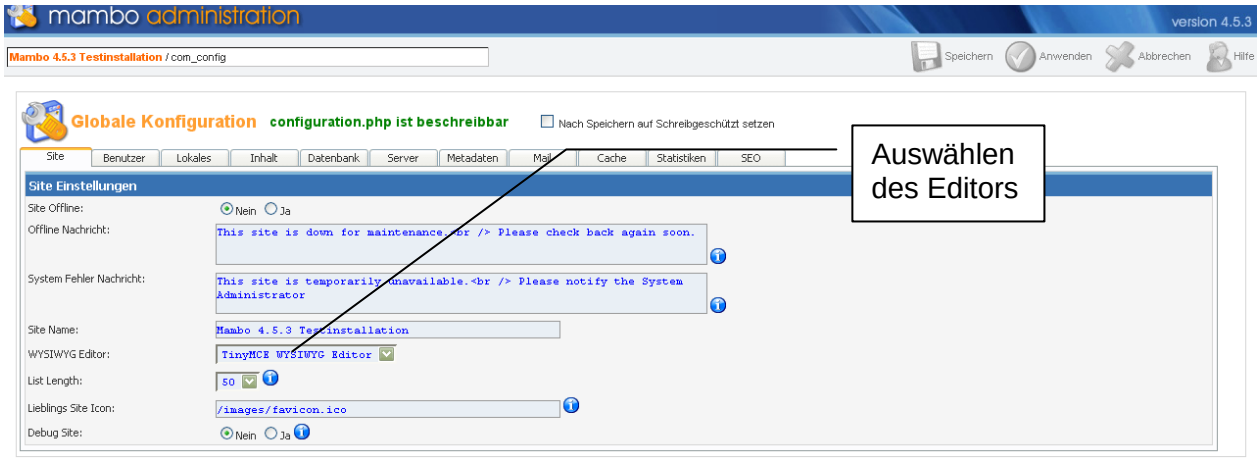


Abbildung: Globale Konfiguration

Nachdem wir den "TinyMCE WYSIWYG Editor" ausgwählt haben und das Icon "Speichern" angeklickt haben, rufen wir nun den Artikel über "die Ananas" auf. Falls Sie den Weg dahin noch nicht verinnerlicht haben: Pulldownmenü *Inhalt / Alle Beiträge* oder wahlweise über *Inhalt / Inhalt nach Bereich / Mein persönliches Obst / Mein persönliches Obst Einträge* oder Sie gehen über die Schaltzentrale: Pulldownmenü: *Home* und dann das große Icon "Alle Beiträge".
Viele Wege – alle führen zum Ziel.

Jetzt sieht unser "Beitragfenster" schon recht bunt und professionell aus. Mit den vielen Icons können wir unsere Texte an Ort und Stelle gestalten.

Abbildung: Icons des Tiny-Editors

Fett (Bold), Kursiv (Italic), Unterstrichen (Underline), Durchgestrichen, Ausrichtung und Aufzählung spricht sicherlich für sich selbst und ist auch von einer Textverarbeitung her bekannt. Und wer ein wenig Spaß am rumspielen hat, wird sich die anderen Icons schnell erschließen.
Dennoch wollen wir uns einige Icons und Vorgehensweisen genauer ansehen:
Um einen Verweis einzufügen muß zunächst ein Wort markiert werden. Erst dann wird das Icon mit den Kettengliedern anklickbar. Das klicken auf den "**Anker**" führt zu einem **Dokumentenverweis**. Und dieser ist als solches im WYSIWYG-Modus nicht sichtbar.

HTML: Für die, die sich mit HTML nicht allzu gut auskennen:
Ein **Dokumentenverweis** ist beispielsweise der Link, der bei langen Texten ganz unten steht und beim anklicken einen wieder nach ganz oben bringt. Also ein Verweis innerhalb einer Webseite. Der "Anker" fügt an Cursorposition den Punkt ein, zudem gesprungen werden soll.

Interessant ist, dass man im Assistenten für den Verweis auch ein Popup-Fenster bestimmen kann, welches dann mittels JavaScript geöffnet wird. Wobei man sich über das JavaScript keine Gedanken machen muß –das ist bereits fest in Mambo integriert!
Der "Verweis-Assistent" bietet dabei einfache und für sich selbst sprechende Möglichkeiten das Aussehens des Popup-Fensters, bzw. des Verweises zu gestalten.

Mit dem "**zerissenen Kettenglied**" kann man einen Verweis wieder entfernen.

Spannend ist auch der "**Baum**" neben dem "Anker". Denn es kann hier nicht nur einfach ein Bild eingefügt werden, sondern auch gleich ein Rollover-Effekt erzeugt werden.
Im Verzeichnis *images/stories/fruit/* des Mamboroots liegen ein paar Früchte, die wir hier zur Veranschaulichung nehmen wollen.
In die erste Zeile "Image URL" tragen wir die Kirsche ein: *images/stories/fruit/cherry.jpg*
Wenn wir mit unserem Cursor die Zeile verlassen, zeigt der Assistent unser Bild an.
Unter "Image description" lassen sich kleine Texte eingeben, die nach dem kurzen Verweilen auf dem Bild angezeigt werden. Menschen, die blind sind und ein Lesegerät benutzen, denen wird vorgelesen, was an dieser Stelle eingetragen ist. Nachdem das Kästchen von "Alternative Image" aktiviert wurde, lassen sich auch die unteren beiden Zeilen benutzen. Wobei unter "for mouse out" bereits die zuvor ausgewählte Grafik automatisch eingetragen wird, denn in der Regel soll ja nur eine andere Grafik angezeigt werden, wenn die Maus über das Bild fährt. Aber das kann man natürlich auch ändern.
Unter "for mouse over" tragen wir folgenden Link ein: *images/stories/fruit/peas.jpg*.
Dann noch unten links auf den Button "Insert" klicken und schon haben wir einen Rollover-Effekt in unseren Text eingetragen (eventuell fünf Sekunden warten, bis die Grafiken geladen sind).
Wenn Sie das Bild nun markieren im Text (einmal draufklicken) und dann das Icon mit dem Kettenglied anklicken, können Sie dem Rollover-Effekt einen Link zuweisen.

Mit dem Auswahlbutton **Paragraph** können Sie einige Grundformate direkt zuweisen. Einen Bereich markieren und dann beispielsweise *Heading 3* auswählen und Sie haben die Formatierung für Überschriften der Größe 3 zugewiesen.

Interessant ist der Auswahlbutton **– Styles –** direkt daneben. Dort sind alle verfügbaren CSS-Klassen für das ausgewählte Template auswählbar. Wie im Kapitel über Templateerstellung genauer erklärt, hat jedes Design seine eigene CSS, in der die zugehörigen Formatierungen definiert sind. Über den Auswahlbutton können diese direkt für den eigenen Text genutzt werden.
Das kann ganz einfach funktionieren, weil in das Template die zugehörige CSS geladen wird und in diesem Template letztlich der Inhalt erzeugt wird. Somit sind die CSS-Klassen jederzeit verfügbar.
Somit lassen sich natürlich auch eigene Klassen in der CSS definieren, die dann im Text eingesetzt werden können.

> *Hinweis*: Damit die eigene Klasse im Auswahlbutton angezeigt wird, muß zuvor der Browsercache geleert und sich ab- und angemeldet werden.

Diese eigene Klasse kann ich auch für das **Icon Tabelle** nutzen. Denn wenn auf das Icon geklickt wird (das mit dem Schreibstift – das andere, ähnlich aussehende Icon hat eine andere Bedeutung), dann wird gefragt, wie die Tabelle aussehen soll. Es lassen sich zwar nur Standards einer Tabellenformatierung über den Assistenten angeben, aber eben auch eine zuvor definierte Klasse. Über Insert wird dann die angegebene Tabelle an Curcorposition dargestellt.
Bei mir ist das Fenster mit dem Tabellenassistenten etwas zu klein, so dass die rechten Felder aus dem Fenster hinausragen, bzw. verschwinden – aber es läßt sich dennoch alles benutzen.
Wenn man jetzt in ein Tabellenfeld klickt erscheinen kleine waagerechte und vertikale Pfeile, die einen Kreis mit Kreuz flankieren.
Beim klicken auf die Pfeile erscheinen davor oder dahinter, bzw. links oder rechts neue Tabellenzellen. Beim klick auf das Kreuz verschwindet die Zeile oder Spalte. So läßt sich die Tabelle jederzeit nach Bedarf direkt im Textfenster erweitern oder verkleinern. Desweiteren bieten zusätzliche Icons die Möglichkeit einzelne Zellen oder Zeilen zu formatieren oder eben auch eine neue Zeile oder Spalte einzufügen, bzw. zu löschen.
Mit dem weiter links stehenden Icon, welches auch ein Tabellensymbol anzeigt, kann ich die Tabellengitterlinien ein- oder ausblenden.

Um ein **Sonderzeichen** einzufügen, einfach auf das Icon mit dem Ohmzeichen klicken. Im dann erscheinen Assistenten das Zeichen aussuchen und anklicken – es erscheint dann umgehenst an Cursorposition. Der Assistent wird nicht geschlossen – es können weitere Sonderzeichen ausgewählt werden. Geschlossen wird der Assistent durch beherztes klicken auf das Kreuz oben rechts.

Das "**Besen"-Icon** ersetzt ggfls. veralteten HTML-Code durch neueren nach dem 3W-Standard. Beispielsweise wird der Tag "<b>" in "" umgewandelt.

Um eine Formatierung zu löschen, den Bereich mit der Maus markieren und dann das Icon mit dem **Radiergummi** anklicken.

Horizontale Linien lassen sich entweder als einfache Standard-Bar mit dem linken Icon, mit dem Strich, einfügen oder etwas kompfortabler mit einigen Einstellmöglichkeiten mit dem rechten Icon, welches den "breiten" Strich hat.

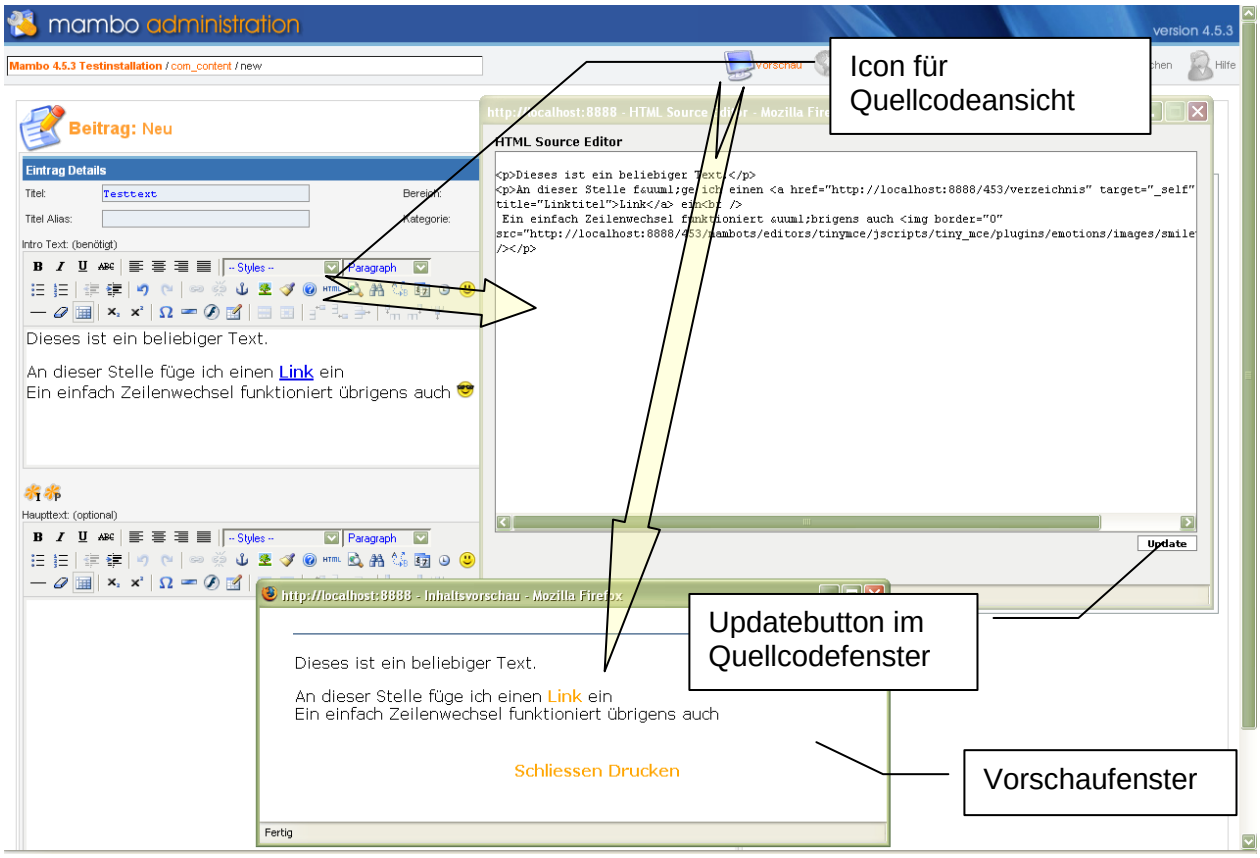


Abbildung: Artikeleingabe mit dem TinyMCE WYSIWYG-Editor (Advanced-Modus)

Mit dem Icon **"HTML"** kann ich mir den Quellcode von meinem Artikel anschauen. Dort kann ich selbstverständlich auch direkt Code eingeben und mittels des Update-Buttons mir das Ergebnis dann sogleich im WYSIWYG-Editor anschauen.

Um zu sehen, wie alles anschließend im Browser ausschaut, kann ich ganz oben in der Mambo-Iconleiste, das Icon **Vorschau** anklicken und erhalte dann ein Vorschaufenster.

Ich denke, die anderen Icons sprechen für sich und sollten keine Schwierigkeiten machen.

10.2.1.2. Probleme mit WYSIWYG

WYSIWYG-Editoren wandeln den eingegebenen Inhalt in HTML-Code um und speichern diesen dann ab.
Problematisch wird das, wenn Code eingegeben wird, den ein WYSIWYG-Editor nicht interpretieren kann. Und das ist eine ganze Menge. Wer beispielsweise über die HTML-Code-Ansicht solchen Code eingibt, wird sich wundern, wenn dieser Code nach erneuten Aufruf des Inhalts nicht mehr angezeigt wird.
Im Bestreben den vorhanden Code direkt und fertig formatiert darzustellen, löschen die WYSIWYG-Editoren unbekannten, nicht darstellbaren Code schlicht und einfach.

Ich selber habe manchmal einen externen Statistikzähler in manchen Artikeln per PHP-Code integriert. Solange ich den Artikel kein zweites Mal editiere passiert nichts. Aber bei einem erneuten editieren, wird dieser Code eben vom WYSIWYG-Editor gelöscht.

10.2.1.3. Artikel formatieren - ohne WYSIWYG-Editor

All das geht natürlich auch ohne einen WYSIWYG-Editor. Und wer sich ein wenig mit HTML auskennt, wird auch viel mehr Möglöichkeiten haben, seine Texte zu formatieren.
Möglich ist dabei alles, was in HTML erlaubt ist!

Wir können also im Text selber jede Art von HTML-Tags unterbringen oder auch Scripte, wie JavaScript oder PHP, ebenso auch Java, CSS, Perl oder eine beliebige andere Sprache, sofern sie internettauglich und vom Webserver erlaubt ist (ggfls. den Hoster des Webspace fragen, welche Sprachen möglich sind). Flash, Shockwave und andere DHTML-Elemente sind natürlich auch möglich.

> *Tipp*: Insbesondere längere oder aufwenderisch gestaltete Artikel bieten sich an, zunächst in einem "richtigen" HTML-Editor ausgearbeitet zu werden und dann den Quellcode per Copy/Paste in Mambo einzufügen.

Gerade in der Tabellengestaltung bieten sich Möglichkeiten an, die einen WYSIWYG-Editor innerhalb eines CMS letztlich überfordern müssen. Würde ein sehr umfangreicher WYSIWYG-Editor in Mambo integriert, würde vermutlich die Schnelligkeit des Systems nicht unerheblich darunter leiden. Zumindest beim anlegen oder editieren eines Artikels.

Gute Webdesigner testen ihre Webseiten auch auf den gängigen Internet-Browsern ab. Schon oft hat uns ein User stolz seine frisch designte Website zeigen wollte und wunderte sich dann darüber, wenn Opera, Firefox und Netscapebenutzer oder, je nach dem auf welchem Browser das Design entwickelt wurde, Internet Explorer Nutzer was ganz anderes sahen, als vom Webdesigner auf seinem System entwickelt.
Desweiteren sollte auch immer auf Browserkompatibilität geachtet werden – und das umfaßt die gängigen Browser (Opera, Firefox, Netscape und MS-Internet Explorer). Bereits eine einfache Tabelle kann auf manchem Browser anders dargestellt werden, als es zu Hause am heimischen System entwickelt wurde.

Ein großer Vorteil, wenn ohne WYSIWYG-Editor gearbeitet wird, ist, dass der Quellcode nicht unnötig aufgebläht wird. Insbesondere bei genauen Positionierungen oder auch bei häufigen Schriftarten/-größen/-farbwechsel wird oft viel unnötiger Code bei WYSIWYG-Editoren erzeugt. Das mag bei wenigen Artikeln nicht weiter ins Gewicht fallen, aber bei vielen und/oder längeren Artikeln kann das schon (insbesondere bei Modemnutzern) durchaus ins Gewicht fallen.

10.2.2. Bilder einfügen

Um Bilder in Beiträgen einzufügen, stehen gleich mehrere Möglichkeiten in Mambo zur Verfügung.

Die eine besteht einfach darin den HTML-Tag für Grafiken zu benutzen. Also innerhalb seines Artikels
****
einzufügen.

Wenn mit einem WYSIWYG-Editor gearbeitet wird, dann kann natürlich das zugehörige Icon zum Bildereinfügen, wie oben bereits beschrieben, benutzt werden.

Über das große **Icon: Upload** kann ein Bild nach Mambo geladen werden. Die Grafik wird in das Verzeichnis *images / stories* abgelegt. Über dem Register "Images" kann man das Bild jedoch erst sehen, wenn man den Editbereich des Artikel verläßt und erneut aufruft, weil die Bilderliste in den Cache vorgeladen wird und nicht automatisch erneuert.

10.2.2.1. Media Manager

Die zweite Variante arbeitet ebenfalls mit dem img-Tag, jedoch etwas kompfortabler:
Dazu müssen wir zunächst über das Pulldownmenü den Media Manager aufrufen:
Site / Media Manager (wahlweise über die Schaltzentrale)

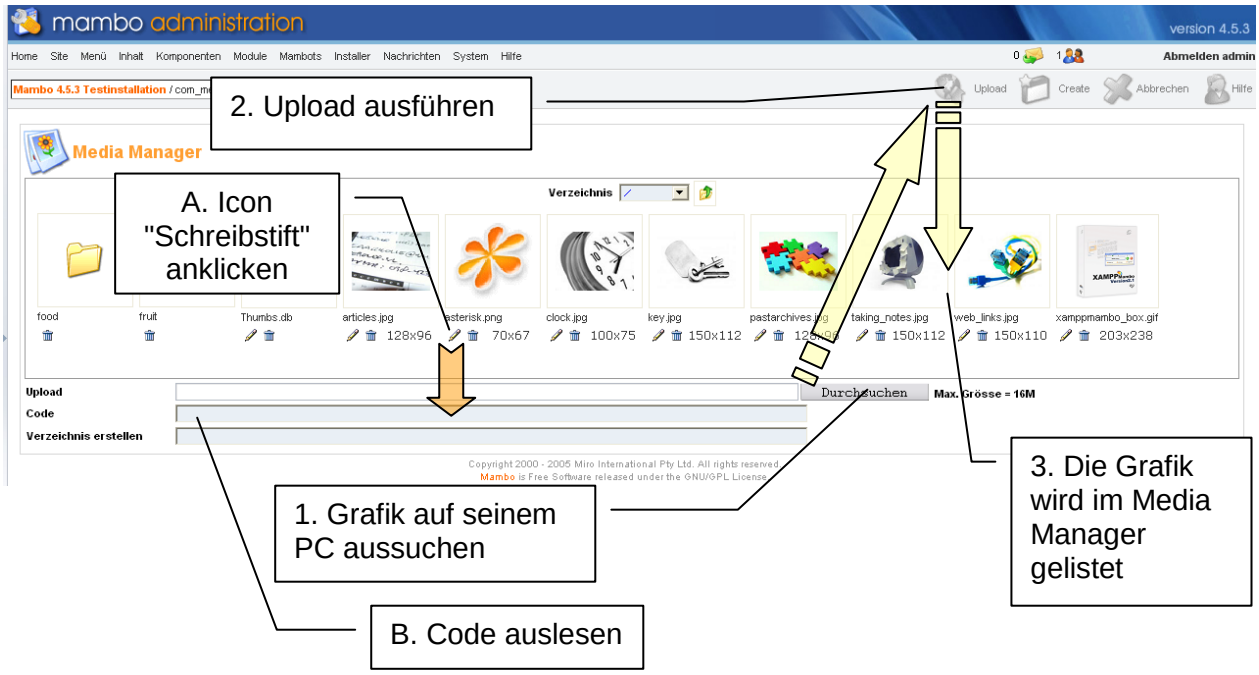


Abbildung: Media Manager

Hier haben wir einen Überblick über unsere Grafiken innerhalb Mambos. Wobei grundsätzlich vom "Grundverzeichnis" für Grafiken ausgegangen wird:

Mamboroot/images/stories

Ein Durchsuchen oberhalb dieses Pfades ist nicht möglich!

Um die Übersicht zu behalten, empfiehlt es sich ein eigenes Verzeichnis anzulegen. Dazu einfach den Verzeichnisnamen, beispielsweise: "MeinVerzeichnis", in das Feld neben **Verzeichnis erstellen** schreiben und oben das große Icon **Create** anklicken. Und umgehend wird das neu erstellte Verzeichnis im Media Manager angezeigt.
In ein Verzeichnis kann entweder per Doppelklick auf das Verzeichnissymbol oder über den Pulldownbutton *Verzeichnis* gewechselt werden.

Eine neue Grafik kann folgendermaßen eingefügt werden:
Zunächst wird die heimische Festplatte mittels des *Durchsuchen-Buttons* durchsucht und ein Bild mit den Dateiendungen " gif, png, jpg, bmp, pdf, swf, doc, xls oder ppt" ausgewählt.
Danach wird oben auf das Icon **Upload** geklickt und der Media Manager zeigt umgehenstt

die Grafik an, sofern nicht ein Fehler beim Uploaden passiert ist. Sollte das der Fall sein, wird eine entsprechende Fehlermeldung ausgegeben.

Im Media Manager wird nun der Name der Grafik angezeigt und in welcher Größe es vorliegt. Desweiteren finden wir unter jedem Bild zwei kleine Icons: einen Mülleimer und einen Schreibstift.
Mittels des *Mülleimers* kann ich das Bild löschen.
Wenn wir auf den *Schreibstift* klicken, wird uns unten der lorrekte HTML-Code angezeigt.
Beispielsweise: ****
Diesen Code nun können wir markieren und dann per Copy/Paste in unseren Artikel einfügen. Wobei ggfls. der Pfad angepaßt werden muß – im obigen Beispiel ist ersichtlich, dass es sich um eine lokale Installation handelt. Für's Web würde ich einfach den ersten Teil des Pfades weglassen und lediglich stehen lassen: **images/stories/articles.jpg**
Was sich unbedingt empfiehlt, ist eine Änderung des alt-Attributs (**alt="Code"**). Menschen, die ein Lesegerät benutzen (sehschwache und blinde Menschen) können logischerweise ein Bild nicht "lesen". Diese Lesegeräte lesen jedoch das vor, was innerhalb des alt-Attributs steht. Aus diesem Grunde sollte dort immer eine kurze Beschreibung des Bildes stehen oder, wenn es sich um einen grafischen Verweis handelt, einen kurzen Hinweis, dass man dort klicken muß/kann, um weiter zu kommen.

Tipp: Bei mir war es so, dass ich zunächst den Media Manager schließen und neu starten mußte, um in mein eigenes Verzeichnis Bilder uploaden und um diese auch wieder löschen zu können. Vorher gab es immer den Hinweis, dass Bild und Verzeichnis nicht verlinkt seien.

Tipp: Eventuell schleicht sich bei der Pfadangabe ein Slash zuviel ein. Also die Pfade immer auf korrekte Syntax überprüfen!

10.2.2.2. mosimage

Eine sehr interessante Variante ist das arbeiten mit dem Platzhalter *mosimage*. Zumal es auch hiermit sehr einfach ist, zu einem späteren Zeitppunkt Grafiken zu verändern ohne dabei direkt in den Artikel eingreifen zu müssen.

Innerhalb des Artikels wird an der Stelle, in der die Grafik eingefügt werden soll mosimage in geschweiften Klammern geschrieben:
{mosimage}
Statt es per Hand zu schreiben kann auch auf das Icon mit dem "I" ganz unten geklickt werden (siehe Abbildung).

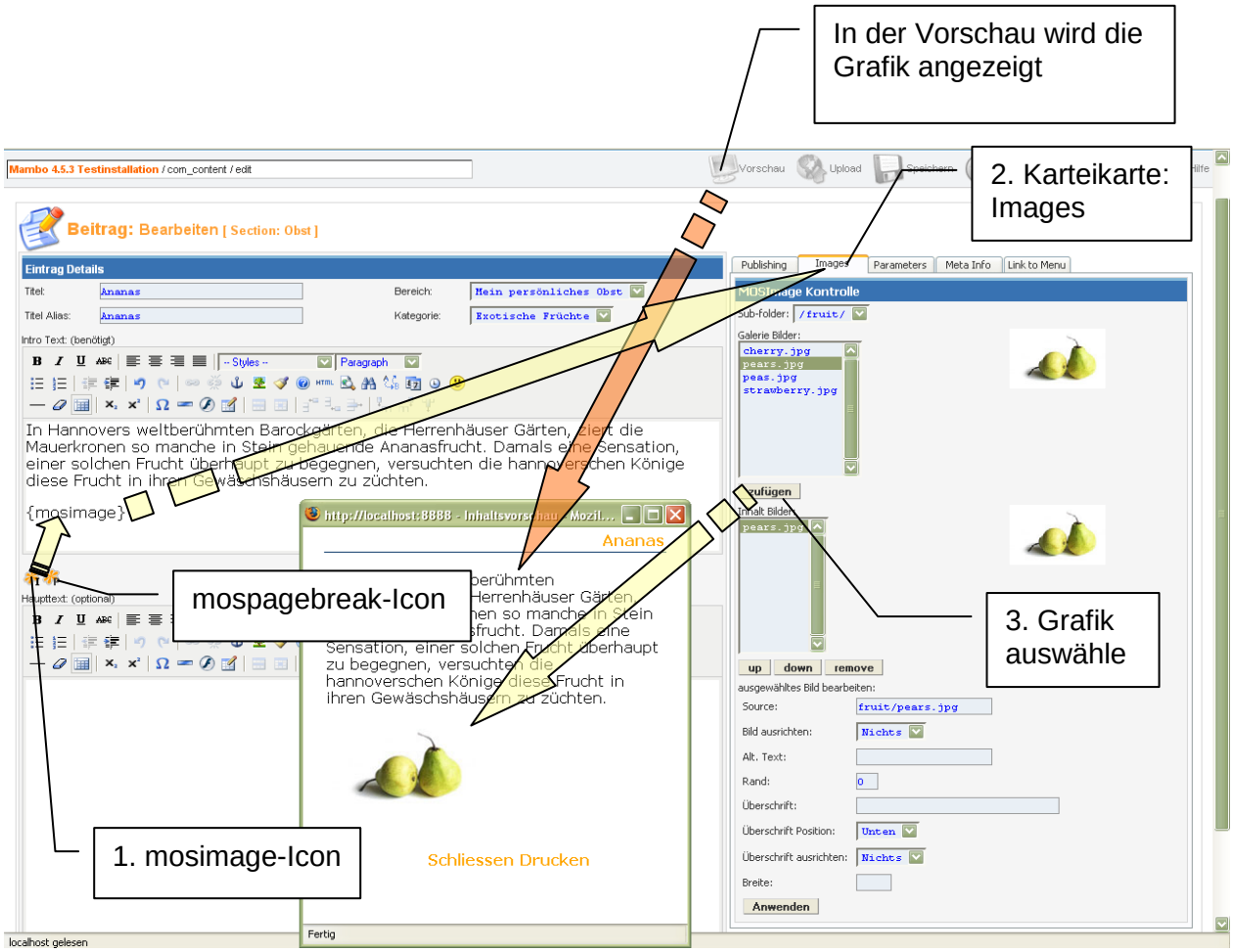


Abbildung: Mit "mosimage" arbeiten

Hinweis: Im Opera funktioniert der mosimage-Button nicht.

Nachdem *mosimage* eingefügt wurde, muß rechts das Register **Images** ausgewählt werden. Dort kann ich zunächst das richtige Verzeichnis beim *Sub-folder* und dann aus der Liste der *Galerie Bilder* das einzufügende Bild auswählen. Dieses Bild erscheint als Thumbnail rechts neben der Auswahlbox.
Nun muß der **Zufügen-Button** angeklickt werden. Das Bild erscheint nocheinmal neben der Auswahlbox *Inhalt Bilder*. Wenn mehrere Bilder ausgewählt werden, kann hier die Reihenfolge festgelegt werden. Pro Bild wird im Artikel ein *{mosimage}* eingetragen. Die ausgewählten Grafiken werden von oben nach unten in der Reihenfolge, wie sie in der *Inhalt Bilder-Box* stehen in den Artikel anstelle der *mosimage's* eingefügt.

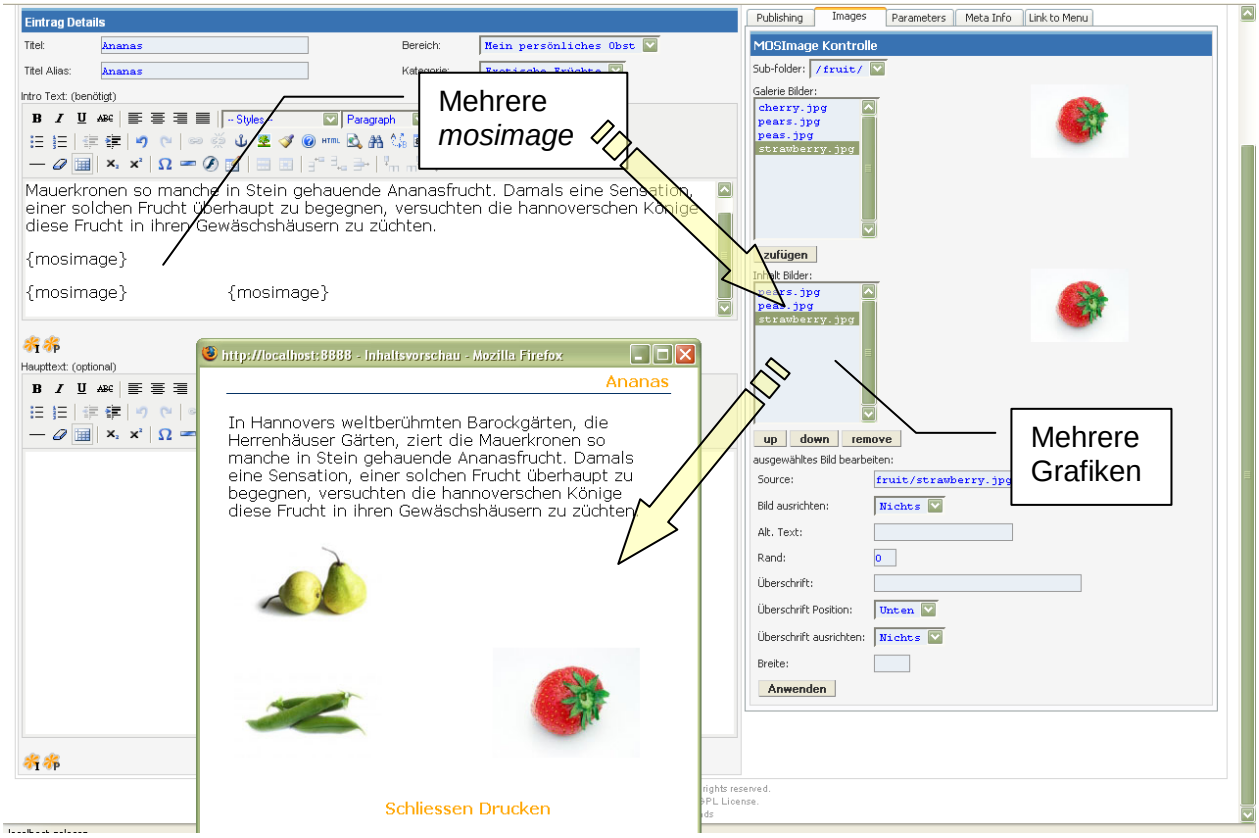


Abbildung: Mehrere Bilder mit *mosimage*

Wenn jetzt die *Vorschau* ausgeweählt wird, werden die ausgewählten Bilder in unserem Artikel angezeigt.

Wir können aber noch mehr machen! Um ein Bild in einem Artikel zu integrieren, können wir unten das Bild ausrichten. *Links* bedeutet, dass der Text rechts um das Bild herumfließt, *rechts* entsprechend umgekehrt. *Mitte* bedeutet, dass die nachfolgende erste Zeile hinter dem mosimage-Tag mittig rechts neben dem Bild steht und der restliche Text unter dem Bild weiterfließt. *Nichts* läßt die erste Zeile des Textes unten direkt neben dem Bild beginnen. Wobei diese Ausrichtung natürlich nur Sinn ergibt, wenn das mosimage-Tag direkt im Text eingefügt wird.

Unter **Alt. Text** wird das eingetragen, was im Quelltext im alt-Attribut eingetragen wird. An dieser Stelle sei auch noch einmal auf blinde und sehschwache Menschen hingewiesen, die ein Lesegerät benutzen und auf diese alternativ Bezeichnungen angewiesen sind!

Wenn ich mein Bild mit einem Rahmen umgeben möchte, kann hier ein Wert zwischen 0 und 9 beim **Rand** eingetragen werden. Wobei "0" kein Rand bedeutet. Dieser Wert ist standardmäßig vorgegeben.

Überschrift ist eine recht interessante Sache. Denn ich kann einem Bild sowohl eine Überschrift, als auch eine Bildunterschrift zuweisen.Der Text den ich hier eingebe, kann mit den beiden nächsten Buttons positioniert und ausgerichtet werden.
Mit **Breite** kann die Bildüberschrift/-unterschrift begrenzt werden.

Nachdem auf **Anwenden** geklickt wurde, werden die Einstellungen, die in diesem unteren Bereich gemacht wurden zu dem ausgewählten Bild intern gespeichert. Sehen kann ich das

erst, wenn ich die Seitenvorschau aufrufe – also nach dem Speichern des ganzen Artikels über das Pulldownmenü *Site / Vorschau / In neuem Fenster*
Wenn die Maus kurze Zeit über dem Bild verharrt, wird auch der Alternativtext angezeigt. Dieser wird übrigens auch angezeigt, wenn aus irgendwelchen Gründen einmal die Grafik nicht geladen werden kann.

10.2.3. Ein Beitrag über mehrere Seiten - mospagebreak

Wenn Artikel zu lang werden oder verschiedene Themenbereiche berühren, kann es Sinn machen, einen Artikel in mehrere Seiten zu gliedern.

Um dieses bewerkstelligen zu können, bietet uns Mambo das Mambot **mospagebreak** an.

Als Button auch direkt neben dem *mosimage-Button* oder direkt in geschweiften Klammern im Beitrag eingeben: **{mospagebreak}**. *mospagebreak* fügt dann an der Stelle einen Seitenwechsel ein.
Mambo bietet uns für das Aussehen dieses Wechsels verschiedene Möglichkeiten, wie in der Abbildung zu erkennen.

Abbildung: mospagebreak – oben: Table of Contents, mitte: Page Navigation (zuerst mit, dann ohne "Site&Page Titel"), unten: Individuelle Seitenüberschriften

Da *mospagebreak* ein Mambot ist, wird es auch genau dort eingestellt. Also rufen wir über das Pulldownmenü *Mambots / Site Mambots* auf und befinden uns nun im *Mambot Manager*. Auf den Manager kommen wir später noch einmal zu sprechen. Hier wollen wir uns nur den

zuständigen Mambot für den Seitenumbruch anschauen. Vermutlich weit oben (bei mir ist es die neunte Position) steht der Mambot **Mos Pagination**. Den wollen wir bearbeiten.

Zunächst erhalten wir informationen über den Mambot selber. Neben dem Namen, erfahren wir auch, auf welche PHP-Datei das Mambot zugreift und in welchem Verzeichnis des Mambot-Ordners die Datei liegt. Wer mit PHP programmieren kann, kann somit auf die Datei zugreifen und sie verändern.
Auch wenn es beim *mospagebreak* nicht allzu viel Sinn macht, so können wir dennoch einstellen, welche Usergruppe mit diesem Mambot arbeiten darf, bzw. sichtbar ist.

Unter Beschreibung steht, in welcher Form *mospagebreak* individuell erweitert werden kann. Dazu etwas weiter unten.
Wollen wir zunächst die rechte Seite betrachten – die Parameter.

Zunächst können wir einstellen, ob wir ein Inhaltsverzeichnis anzeigen wollen oder nicht. **Table of Contents** zeigt in einem Kasten den Inhalt der einzelnen Seiten an und bietet mit einem Direktklick auf den Seitennamen eine schnelle Navigation zwischen den Seiten an. Zumal so auch ein direkter Überblick über den Beitragsinhalt gegeben wird.
Wobei die Voreinstellung zunächst die Überschrift als Headline für die erste Seite nimmt und darunter lediglich mit dem Schlüsselwort *Page* die Seiten durchnummeriert. Was natürlich nicht allzu viel Übersicht bietet.
Dieses können wir aber natürlich ändern! Dazu müssen wir aber wieder zurück zu unserem Artikel gehen. Am besten zunächst *speichern* anklicken und dann über das Pulldownmenü wieder zum *Beitrag Manager* zurück und unseren Artikel mit dem Seitenumbruch editieren.
Die Mambots werden auch als Tag ('Täck' gesprochen) bezeichnet. Also wie auch ein HTML-Befehl. Und viele Tags erlauben die zusätzliche Angabe von Attributen. So auch das Tag *mospagebreak*. Wie können den einzelnen Seiten eigene Inhalts-Überschriften zuweisen.

Hinweis: Das Attribut "title" muß klein geschrieben werden, sonst funktioniert es nicht!

Hinweis: Im Opera funktioniert der mospagebreak-Button nicht.

10.2.4. Parameters

Unter dem Register "Parameters" lassen sich verschiedene Einstellungen vornehmen, die teilweise auch in der globalen Konfiguration vorgenommen werden können, hier jedoch für jeden einzelnen Artikel individuell angepaßt werden kann.

Der erste Punkt *Page Class Suffix* wird unter dem Kapitel "Template selber erstellen" näher beschrieben.

Mit **Back Button** wird eingestellt, ob ein "Zurück"-Verweis unter dem Artikel angezeigt werden soll.
Wobei generell die drei Einstellungen "Anzeigen", "Verstecken", "benutze die globale Einstellung" getätigt werden können.

Manchmal kann es ratsam sein, die Artikelüberschrift nicht anzuzeigen. Unter **Page Titel** kann die Überschrift – also der Text, der links unter *Titel* steht – angezeigt oder ausgeschaltet werden.

Mit **Linked Titels** können Sie die Überschrift des Artikels anklickbar machen und haben dann den gleichen Effekt, wie mit dem "Weiterlesen…"-Button. Also beim klicken auf die

Überschrift wird der gesamte Artikel angezeigt (Intro Text und Haupttext (es sei denn Sie haben andere Einstellungen vorgenommen [*Intro Text = Hide*])).
Um einer Verwirrung vorzubeugen: Selbstverständlich funktioniert der anklickbare Titel nur in der Übersicht. Also dort, wo lediglich der Intro Text angezeigt wird.

Mit der Einstellung beim **Intro Text** wird angegeben, ob der Intro Text nach klick auf den "Weiterlesen-Button" noch einmal angezeigt wird oder nur noch der Haupttext.
Beispiel: Sie geben einen Text im Intro-Text-Feld ein und einen weiterführenden Text im Haupttext-Feld. Zunächst wird nur der Text vom Intro-Text-Feld angezeigt und unten drunter steht der Verweis "*Weiterlesen…*". Wenn SIe dort nun draufklicken und unter **Intro Text** *Show* aktiviert haben, wird der gesamte Artikel nun auf einer gesonderten Seite allein stehend angezeigt. Also der **Intro Text** <u>und</u> der **Haupttext**. Haben Sie jedoch *Hide* aktiviert, wird nur noch der **Haupttext** angezeigt und der **Intro Text** nicht mehr.

Die Einstellung unter **Section Name** wird auch erst sichtbar, wenn es einen "Weiterlesen-Button" gibt. Wenn dann der *Haupttext* erscheint, wird ober drüber der Bereich (Sektion) – Name angezeigt oder eben nicht – je nach dem wie Sie es eingestellt haben.

Wenn SIe den **Section Name** auf *Show* eingestellt haben, dann können Sie unter dem Punkt **Section Name Linkable** angeben, ob der Bereichname anklickbar ist. Also ob er gleichzeitig als Verweis dienen soll. Wenn ja, hat ein klick auf den Bereichnamen den gleichen Effekt, als wenn Sie über Menü die *Bereichliste* aufrufen würden.

Category Name und **Category Name Linkable** hat den gleichen Effekt, wie unter **Section Name** beschrieben, nur das eben statt dem Bereichnamen die Kategorie angezeigt wird. Allerdings immer erst, wenn auf "Weiterlesen …" geklickt wird – also ein Text sowohl im Intro-Text-Feld, als auch im Haupttext-Feld steht.

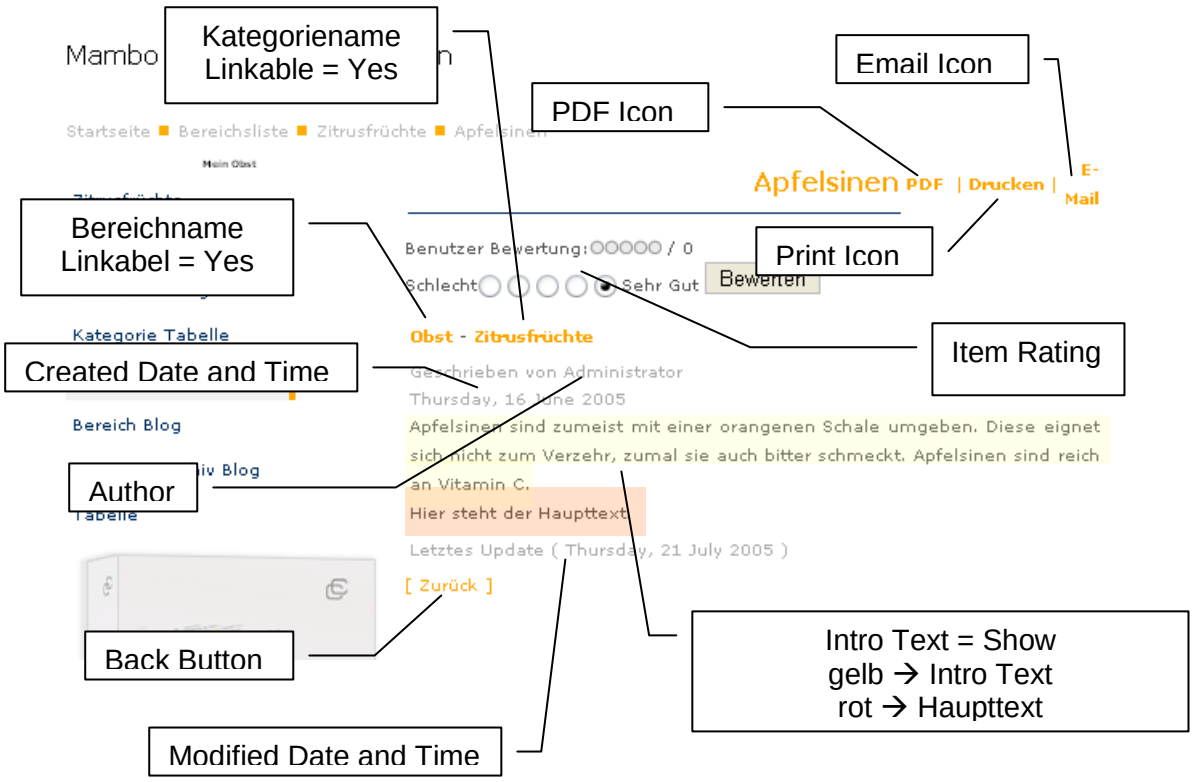


Abbildung: Parametereinstellungen Artikel

Wenn Sie das **PDF Icon** aktivieren, erlauben Sie dem Anwender den Artikel als PDF-Datei anzuzeigen.

Mit dem **Print Icon** erlauben Sie Ihren Artikel Druckergerecht auszugeben. Dabei wird dann beispielsweise auch das Rating (sofern aktiviert) ausgeblendet.

Bei aktivierten **Email Icon** rufen Sie beim klick auf dieses Icon/Textverweis ein neues Fenster auf, indem die Möglichkeit besteht, einem Freund den Link zu diesem Artikel zu senden.

Wenn Sie erfahren möchten, wie gut Ihre Artikel bei den Usern ankommen, stellen Sie das **Item Rating** auf *Show*. Die User haben dann die Möglichkeit Ihre Artikel zu bewerten. Das Ergebnis – Durchschnittsbewertung und Anzahl der abgestimmten User wird mit angezeigt.

Schreiben nur Sie oder auch andere Autoren Artikel? Wenn letzteres möchten Sie vielleicht den Namen der Autoren im Artikel angeben. Wobei nicht der Nickname, mit dem der **Autor** eingeloggt ist, angezeigt wird, sondern der Name, der in der Benutzerverwaltung in der gleichnamigen Spalte steht.

Um immer auf dem laufenden zu sein, wann ein Artikel geschrieben oder eventuell verändert wurde, haben Sie die Möglichkeit dieses in **Created** (Erstellt) und **Modified** (Verändert) **Date and Time** einzustellen.

Tipp: Fügen Sie am Ende des Intro Textes einen Zeilenumbruch (<br>) oder im Haupttext einen Absatz (<p>) ein, sonst steht der Haupttext ohne Übergang direkt neben dem Intro Text.

10.2.5. Meta Info

Meist wird angenommen, dass Einträge an dieser Stelle Einfluß auf Suchmaschinen haben. Also die **Meta Infos** identisch mit den Meta-Tags von HTML sind. Sind sie aber nicht. Diese **Meta Infos** geben die Möglichkeit verwante Artikel zu kennzeichnen.

Beispiel: Tragen Sie in unsere zwei Obst- Artikel (Ananas und Apfelsine) jeweils das Schlüsselwort "Saft" unter **Meta Info** ein und speichern Sie jeweils. Nun müssen Sie noch in den **Modul Manager für die Seiten** gehen → Pulldownmenü *Module / Site Module* und das Modul **Related Items** editieren (direkt auf den Namen klicken). Dieses Modul ist dafür verantwortlich, dass Artikel mit gleichen Schlüsselwörtern als Auswahlliste angezeigt werden.
Voreingestellt ist bei diesem Modul, dass es zwar veröffentlich ist, jedoch nirgends angezeigt wird (eine vollkommen widersinnige Einstellung …). Geben Sie bitte rechts unter *Menüeinträge* **All** an und speichern Sie.
Wenn Sie nun im Frontend den Artikel über "Apfelsinen" aufrufen, erscheint links eine Liste, in der in unserem Beispiel die Artikelüberschrift "Ananas" angezeigt wird. Also alle Artikel, die das gleiche Schlüsselwort (mindestens eins) beinhaltet, wie der Artikel, den ich ausgwählt habe. Bei unserem Beispiel ist das das jeweilige Schlüsselwort "Saft".

Durch die Vergabe verschiedener Schlüsselwörter können auf dieser Art und Weise verschiedene Artikel zueinander kombiniert und weiterführende Informationen angeboten werden:
Liste alle Artikel, aus denen Saft hergestellt werden kann. Wenn Artikel xy ausgewählt wird, zeige zusätzlich auch alle Artikel an, die "bitter" schmecken. Wird hingegen Artikel xyz angezeigt, liste alle Artikel, die neben "Saft" auch noch "gekocht" werden können. Usw.

Hinweis: Wenn ein Artikel aus der "Verwandschaftsliste" ausgewählt wird, wird stets der gesamte Artikel angezeigt. Also der Teil, der beim anklicken des "Weiterlesen-Buttons" angezeigt werden würde. Ist der *Intro Text* auf *Hide* gestellt, wird dieser nicht mehr angezeigt.

Durch anklicken des Buttons **Bereich/Kat/Beitrag hinzufügen** können Sie den zugehörigenBereich, die Kategorie und die Artikelüberschrift als Schlüsselwörter hinzufügen.

10.2.6. Publishing

Dort haben Sie die Möglichkeit verschiedene Eigenschaften, die mit der Veröffentlichung des Artikels zu tun haben, einzustellen.

Wenn Sie möchten, dass der Artikel auf der Startseite – also der ersten Seite, die geladen wird, die Home-Seite – angezeigt wird, dann aktivieren Sie das Kästchen **auf Startseite anzeigen**.
Der Artikel wird dann aber auch weiter unter dem Bereich/der Kategorie sichtbar sein und unter entsprechenden Menüpunkten auch sichtbar sein. Auf der Startseite sichtbar ist also ein zusätzlicher Ort der Anzeige. Es sei denn, Sie haben weiter keine Menüauswahl für diesen Artikel vorgesehen.
So können Sie also auch eine eigene Kategorie, bzw. Bereich anlegen, für Artikel, die ausschließlich für die Startseite bestimmt sind (häufig ist das die Einstellung für *Kden Bereich News*).

Wenn Sie **Veröffentlicht** deaktivieren, wird Ihr Artikel nicht angezeigt.

Mit **Zugriffslevel** legen Sie fest, wer diesen Artikel lesen darf:

- Public = Alle
- Registered = nur registrierte User
- Special = nur Administratoren

Unter **Autor Alias** können Sie einen Namen eingeben, der vom Namen der Benutzereinstellungen abweicht. Dieser wird statt dessen dann angezeigt. Vorraussetzung ist allerdings, dass Sie unter **Parameters** *Authors Names = Show* eingestellt haben. Somit können sich Ihre Redakteure auch spezielle Namen für die Artikelerstellung zulegen ("Der rasende Netzreporter" beispielsweise).

Überarbeiten Sie manchmal Texte anderer und möchten, dass Sie als der Verfasser augegeben werden oder hat jemand versehentlich den falschen Namen als Autor eingetragen? Unter **Ersteller ändern** können Sie dieses alles wieder richten.

Die **Reihenfolge** der Artikel können Sie im gleichnamigen Feld festlegen …

Sollte das **Erstellungsdatum** nicht korrekt sein, ändern Sie es hier.
Und wenn Sie möchten, dass dcer Artikel erst näächste Woche erscheint, dann geben Sie das Erscheinungsdatum an (**Veröffentlichung starten**). Mambo sorgt dafür, dass der Artikel pünktlich erscheint! Beispielsweise Heiligabend der "Frohe Weihnachten"-Artikel oder pünktlich zu Ostern die "Frohe Ostern"-Botschaften.
Oder wenn Sie in Urlaub fahren, dann geben Sie doch einfach hier an, wann der Artikel mit Ihren Urlaubsgrüßen erscheint und wie lange er veröffentlicht bleibt.

Hinweis: Leider funktioniert im Opera der Kalender nicht.

Im unteren Teil haben Sie einen schnellen Überblick über die Eigenschaften des Artikels. Also ob der Artikel veröffentlich ist, wann erstellt und von wem und vor allem, wieviel **Zugriffe** bislang auf den Artikel stattfanden.
Bedenken Sie jedoch, dass Artikel auf der Startseite nicht gezählt werden. Da die Artikel immer angezeigt werden, wenn Ihre Site aufgerufen wird und der Artikel nicht explizit vom Anwender ausgewählt wird, taucht er als solches nicht in der Statistzik auf. Es kann also gut sein, dass Ihre Startseite wie verrückt frenquentiert wird, jedoch der Artikel auf der Frontseite null Zugriffe verbucht.

Gerade in der Entstehungszeit kann es gut sein, dass eine Website immer und wieder aufgerufenb wird, aber eigentlich nur, um sich die Vorschau anzusehen. Um aber einen echten Überblick über die Zugriffe auf eine Site, bzw. einen Artikel zu haben, erscheint es vielleicht sinnvoll, den **Zugriffszähler zurückzustellen**. Klicken Sie einfach auf den gleichnamigen Button und die Zugriffszahlen werden auf null gesetzt. Außerdem wird **Zuletzt modifiziert** aktualisiert.

10.2.7. Link to Menu

Mit dem letzten Register "Link to Menu" können Sie ebenso verfahren, wie oben bei den Bereichen und Kategorien bereits erklärt. Nur dass Sie hier direkt auf den Artikel verlinken können.
Dazu einfach ein Menü auswählen, den Menüpunkt-Namen eintragen und auf **Verlinken mit Menü** klicken. Und schon wurde automatisch ein Menüpunkt angelegt, der direkt auf den Artikel verweist.

10.3.statisch oder dynamisch

Bislang haben wir unsere Texte eingebunden, in dem wir zunächst einen Bereich und eine Kategorie festgelegt haben. Aber natürlich geht es auch anders. Dabei wird sowohl auf einen Bereich, als auch auf eine Kategorie, die dem Text zugeordnet ist, verzichtet. Der Text wird direkt eingegeben und direkt veröffentlicht.

Manche mögen jetzt denken, warum dann erst den komplizierten oder zumindest umständlichen Weg über Bereich und Kategorie, wenn es doch auch so geht?

Wer nur drei, vier Artikel auf seiner Webseite hat, die sich zudem selten ändert, braucht in der Tat keine Bereiche und auch keine Kategorien. Wobei die Frage dann natürlich auch erlaubt sein muß, ob dieser überhaupt ein Content Management System braucht?

Content Management System setzt zunächst vorraus, dass überhaupt Content – also Inhalt – vorhanden ist, der verwaltet werden muß. Und grundsätzlich kann ich diesen besser verwalten, wenn ich den Inhalt sortiere.

Statischen Inhalt kann ich aber nicht sortieren. Ich kann ihn nur erstellen und direkt per Link anzeigen lassen.
Wenn ich sehr vielen statischen Inhalt erzeuge, verliere ich sehr schnell den Überblick und die nachträgliche Änderung oder Neuerstellung von Artikeln wird schwieriger. Und noch schwieriger wird es, wenn an meinem Inhalt mehrere Menschen beteiligt sind – zum Beispiel in einem Intranet oder einer Community.

Aber natürlich gibt es dennoch ettliche sinnvolle Bereiche, in denen statischer Text Sinn macht: Zum Beispiel der bereits erwähnte Copyrighthinweis. Oder kurze Texte, die sich nicht ändern und grundsätzliche Dinge regeln (Disclaimer, Kontaktdaten, Firmenslogan u.ä.). Jedoch sollten Sie sich sehr genau überlegen, ob dabei nicht Artikel sind, die innerhalb von Bereichen und Kategorien sinnvoller untergebracht wären und diese Artikel somit dynamisch angelegt werden.
Dynamisch bedeutet stets auch wesentlich mehr Möglichkeiten zur Anzeige des Artikels und eine wesentlich bessere Übersicht, wenn viele Artikel erzeugt werden.

10.3.1. Statischer Inhalt

Rufen wir also über das Pulldownmenü *Inhalt / Statische Inhalte Manager* auf.
Die Oberfläche sieht sehrt ähnlich aus, wie der für den dynamischen Inhalt. Ausnahme auf der linken Seite: Es finden sich keine Auswahlbuttons für Kategorie und Bereich. Auch läßt sich kein Intro Text/Haupttext gesondert eingeben. Somit ist immer der gesamte Text sichtbar. Was aber auch logisch ist, da statische Artikel immer einzeln angezeigt werden und niemals innerhalb einer Liste, wie es bei Kategorien oder Bereichen der Fall ist. Somit wäre eine Aufsplttung in Intro und Haupttext nicht sinnvoll.
Folglich gibt es auf der rechten Seite, unter dem Register **Publishing,** auch keine Eingabemöglichkeit für die Reihenfolge.
Statischer Inhalt läßt sich auch nicht auf der Frontseite veröffentlichen, sondern muß immer mit einem Menüpunkt aufgerufen werden.
Ebenfalls folgerichtig entfallen unter dem Register **Parameters** alle Einstellmöglichkeiten in denen es um Kategorien und Bereiche geht.
Allerdings gibt es statt dessen die Möglichkeit ein Menüimage auszuwählen. Nur leider wird es nicht angezeigt – dieses ist ein Fehler in Mambo. Aber über einen andren Weg, direkt in der Menügestaltung haben wir eine funktionierende Möglichkeit ein Bild neben dem Menüpunkt anzuzeigen. Näheres siehe unter dem Kapitel *Menügestaltung*.

Alle anderen Einstellmöglichkeiten sind identisch mit einem dynamisch erzeugten Artikel und dort erklärt.

10.4. Beiträge erstellen und bearbeiten im Frontend

Natürlich können neue Artikel erstellt oder bestehende überarbeitet werden, ohne sich erst im Administrationsbereich einzuloggen.

Um vorhandene Artikel zu bearbeiten, genügt ein klick auf das Icon zum editieren des Artikels, welches direkt neben der Überschrift abgebildet ist (siehe Abbildung).

Abbildung: Artikel editieren im Frontend

Vorraussetzung dafür ist allerdings, dass man sich zuvor im Frontend einloggt. Was im übigen auch gilt, wenn man bereits im Backend eingeloggt ist!
Der darauffolgende Editierbereich unterscheidet sich von der Artikeleingabe im Backend lediglich insofern, dass er etwas einfacher aufgebaut und teilweise auch anders angeordnet ist. Ebenfalls wird er im bestehenden Design eingefügt und nimmt damit die Farben des Frontend-Templates an.

Im Benutzermenü (Backend: Pulldownmenü: *Site/Benutzer Manager*) können Sie bei jedem Benutzer unter *Parameters* festlegen, ob und wenn ja, welchen WYSIWYG-Editor er benutzen darf.
Wenn also kein WYSIWYG-Editor angezeigt wird, haben Sie dort *Kein WYSIWYG-Editor* eingestellt, andernfalls wird ein WYSIWYG-Editor angezeigt.

Der untere Bereich mit der Bildereingabe ist identisch mit der *mosimage-Eingabe* im Backend. Der einzige Unterschied ist, dass die einzelnen Bereiche nicht untereinander, sondern nebeneinader stehen.

Was Sie allerdings nicht machen können (unabhängig von der Rechtevergabe) ist, den Artikel direkt als Menü einzubinden oder weitere individuelle Angaben zu einem Artikel einzustellen. Dieses geht ausschließlich im Backend. Ebenso wie das Neuanlegen von Kategorien und Bereiche.

Normalerweise sind die Artikel aber bereits irgendwie im Menü verlinkt und werden dann auch angezeigt.
Unter dem Register "Veröffentlicht" kann auch eingestellt werden, ob der Artikel auf der Startseite, also der "Home"-Seite, angezeigt werden soll.

10.5."Auf Startseite anzeigen" oder besser woanders?

Bei jedem Artikel kann angegeben werden, ob er auf der "Startseite" stehen soll (Ausnahme Statische Beiträge) Das bedeutet, dass ein User, der Ihre Webseite besucht, dann auch diesen Artikel direkt lesen kann (sofern der Artikel auf "Public" gesetzt wurde). Wenn Sie zusätzlich noch Bereich oder Kategorie, die dem Artikel zugeordnet wurden per Menü verlinkt haben, dann erscheint der Artikel auch unter diesem Menüpunkt.
Damit bieten sich einige interessante Möglichkeiten in der Gestaltung einer Website. So können Sie einige Artikel aus verschiedenen Bereichen und/oder Kategorien auf der Startseite veröffentlichen, um Ihren Besucher Lust auf mehr zu machen und ihn zu "animieren", Ihre Website zu durchstöbern. Oder Sie bringen neue Artikel auf die Startseite, die Sie nach einer vorgegeben Zeit oder Anzahl von Artikeln dann von der Startseite entfernen, aber diese weiterhin über entsprechende Menüpunkte aufrufbar sind.

So können Sie auch mit Beiträgen verfahren, die durch Rechtevergabe entsprechend eingeschränkt sind, aber einer speziellen Usergruppe (Registrierte oder Administratoren) dann direkt nach dem Einloggen angezeigt werden. Also "Auf Startseite veröffentlichen" aktivieren und bei den Rechten auf *Registered* oder *Special* stellen (Details hierzu siehe unter "Rechte")

10.6.Checkin

Wenn ein Artikel bearbeitet oder auch erstellt wird, ist es wichtig, einen Artikel davor zu schützen, dass er plötzlich von zwei Autoren gleichzeitig bearbeitet wird.

Ein neuer Artikel taucht nach dem ersten Speichern in der Artikelliste auf und kann somit auch von einem weiteren Autoren aufgerufen werden.

Artikel werden dadurch geschützt, in dem sie für andere Autoren gesperrt werden. Normalerweise wird der Artikel von Mambo wieder freigegeben, wenn der Artikel ordnungsgemäß gespeichert wird.

Manchmal jedoch verläßt man einen Artikel nicht ordnungsgemäß – klickt beispielsweise einfach auf ausloggen oder schlimmer noch, schließt einfach das Browserfenster. Manchmal auch ist man länger eingeloggt ohne etwas zu tun und Mambo verlangt ein neues einloggen, jedoch der Artikel ist weiterhin gesperrt für andere Autoren.

Um nun wieder Zugang zu den Artikeln zu erhalten, muß der Artikel eingecheckt werden. Um dieses zu tun, gibt es zwei Möglichkeiten.
Befinden Sie sich noch im Frontend, gibt es im Benutzermenü den Menüpunkt *Check-In My Items*. Damit können die eigenen Artikel, also die, die der angemeldete Benutzer selber geschrieben hat, eingecheckt werden.
Die zweite Möglichkeit besteht im Backend unter dem Pulldownmenü *System / Globales Einchecken*. Vorraussetzung ist allerdings, dass man als *SuperAdministrator* eingeloggt ist, denn andernfalls steht das Menü nicht zur Verfügung.

Mit dem Einchecken kann allerdings nicht das Freigeben von Dateien gesteuert werden. Speichert ein Benutzer seine Datei ab, wird diese auch sofort veröffentlicht.

Interessant wäre sicherlich das Schützen bestimmter Beiträge durch bewußtes auschecken, schließlich kann ja nur der SuperAdmin Beiträge im Backend einchecken. Leider ist diese Funktion in Mambo aber nicht vorgesehen.

11. Menüs

Nun haben wir soviel über das Erstellen von Beiträgen geschrieben, nun wird es Zeit die vielfältigen Anzeigemöglichkeiten unseres Inhalts genauer unter die Lupe zu nehmen.

Ein erster Blick in das Erstellen eines Menüs verwirrt zumeist und was nun das Richtige für die eigene Homepage ist, wird oft erst nach langem Ausprobieren erkennt.

Wollen wir also ein wenig Licht ins Dunkel bringen …

11.1.Menü Manager

Um ein neues Menü anzulegen oder den Namen eines bestehenden zu verändern, muß der Menü Manager aufgerufen werden (Pulldownmenü: *Menu / Menu Manager*).

Nachdem Sie auf das Icon *Neu* geklickt haben, geben Sie zunächst den Namen Ihres Menüs ein.
Danach gibt es einen kleinen Unterschied in den Versionen:

11.1.1. Version 4.5.1

Unter *menu_type* soll nun der *menu_style* eingegeben werden. Die Bezeichnung bereits ist etwas verwirrend. Am besten Sie tragen hier einen eigenen Namen, der Einfachheit den gleichen, wie Sie ihn im Menünamen eingtragen haben, ein.
Wenn Sie später eine andere Menükomponente installieren, die Ihnen mehr Möglichkeiten bietet, als vorhanden, dann können Sie dort diesen *menu_style* zuweisen und der Inhalt Ihres Menüs wird dann ausgegeben.
Gleiches können Sie bereits jetzt schon in den Parametern der Menü-Module machen. Weiter unten dazu mehr.

In den neueren Versionen wird nicht mehr ein Extra Style Name angegeben, sondern direkt der Name des Menüs in den Auswahllisten angezeigt. Die Mambo-Programmierer haben also erkannt, dass dieser zweite Eintrag ansich identisch mit dem ersten ist.
Aber Eintragen müssen sie dort etwas!

11.1.2. Versionen ab 4.5.2

In diesen Versionen tragen Sie unter *Modul Titel* den Namen ein, mit dem Ihr Menü im Modul Manager gelistet werden soll.

11.1.3. Und wieder alle Versionen zusammen

Jedes neue Menü erstellt selbstständig ein neues Modul. Alle Module für die Menüs fußen auf das Grundmodul *mod_mainmenu*, bzw. greifen darauf zu. So muß also das Menü selber noch im *Modul Manager* freigeschaltet werden. Auch ist es dort möglich, weitere Einstellungen vorzunehmen. Doch dazu weiter unten.

Die Menüs, die hier angelegt werden, sind die Hauptmenüs. Um diese nun mit Leben zu füllen, müssen Menüeinträge erstellt werden.

11.2.Neue Menüeinträge

Jedes zuvor angelegte *Hauptmenü* kann Menüeinträge bekommen. Diese Menüeinträge unterscheiden sich in *Top-Menüeinträge* und *Untermenüeinträge*. Desweiteren kann der zuvor erstellte Inhalt auf vielfältige Art per Menü angezeigt werden.

Wählen Sie im Pulldownmenü *Menü* ein bereits vorhandenes Menü aus und klicken Sie im darauf folgenden *Menü Manager [Menüname]* auf das Icon *Neu*. Sie erhalten nun verschiedene Angebote welche Art von Inhalt wie verlinkt werden soll.

Wobei sogleich unterschieden wird, ob es sich um erzeugten Inhalt (Artikel, Beiträge, etc.) oder Komponenten oder Links handelt. Und Sonstiges ist neben dem Seperator für externe Webseiten zuständig. Betrachten wir die einzelnen Bereiche etwas genauer:

Hinweis: Sie können erst dann ein Menü anlegen, wenn Sie den zugehörigen Inhalt erstellt haben. Wobei der Inhalt auch aus Bereichen und Kategorien, denen noch kein Artikel zugeordnet ist oder aus Komponenten bestehen kann. Was jedoch nicht geht, ist erstmal Menüeinträge anzulegen und anschließend sich Gedanken über die Struktur und den Inhalt zu machen.
Also

- zuerst eine Komponente installieren
- zuerst einen Bereich erstellen
- zuerst eine Kategorie erstellen
- zuerst einen Artikel erstellen

bevor Sie diese entsprechend im Menü verlinken können.

11.2.1. Inhalt

Unter Inhalt schlüsseln sich die Einträge in den drei Arten **Blog, Link** und **Table** auf.

Unter **Blog** werden alle Beiträge nach dem ausgewählten Bereich oder der ausgewählten Kategorie sortiert ausgegeben. Wobei jeweils immer der *Introtext* der Beiträge und, sofern ein Haupttext vorhanden ist, darunter der "Weiterlesen-Button" angezeigt wird.
Somit gibt es also je zwei Möglichkeiten Bereiche und Kategorien anzuzeigen. Wobei mit dem **Archive Blog** zuvor archivierte Beiträge monatsweise angezeigt werden. Womit sich im laufe der Zeit ein ansehnliches Archiv zusammentragen läßt und jederzeit auf alte Beiträge zugegriffen werden kann.

Mit **Link-Content Item** kann ein einzelner Beitrag verlinkt werden.

Hinweis: Wenn in erster Linie diese Variante der Menüeinträge gewählt wird – also eine Direktverlinkung der zuvor angelegten Beiträge, erscheint es unsinnig zu sein, vorher Bereiche und Kategorien anzulegen. Die Logik des CMS erlaubt aber keinen anderen Weg, da es schließlich zur Verwaltung von vielen Beiträgen programmiert wurde. Allerdings bleibt als Alternative, dass Sie statt dessen mit statischen Inhalten arbeiten. Denn für diese Art von Beiträgen gibt es bekanntlich weder Bereiche, noch Kategorien. Allerdings ist die Verwaltung von statischem Inhalt auch stark eingeschränkt.

Wenn Sie statische Beiträge erstellt haben, erlaubt Ihnen **Link-Static Content** diesen einzubinden.

Mit **Table** schließlich, werden die Beiträge untereinander mit ihrer Überschrift dargestellt und der User kann alle Beiträge zu einer Kategorie, einem Bereich oder alle statischen Beiträge als Liste untereinander sehen und kann einen Artikel per Mausklick auswählen.

Abbildung: Verlinkung **Table-Content Category**

Wenn nun auf *Next* oder direkt die Auswahl angeklickt wird, werden im nächsten Fenster nur noch die möglichen Auswahlmöglichkeiten angezeigt. Wenn also beispielsweise *Link-Static Content* ausgewählt wurde, werden anschließend als weitere Auswahlmöglichkeit ausschließlich die *statischen Beiträge* angezeigt.

11.2.2. Komponenten

11.3. Menü Module

11.4. Top oder Sub – Untermenüs erstellen

12. Global Config

13. Templates

Das Gestalten der Website mit eigenen Templates ist ein sehr mächtiger Bereich in Mambo. Neben dem grundlegenden Aussehen der Website, bietet Mambo auch die Möglichkeit mehrere Templates in einem Webauftritt zu benutzen, sowie eigene Formatierungen für einzelne Module, Komponenten und einzelne Seiten.
Damit bietet Mambo fast unbeschränkte Gestaltungsmöglichkeiten, die darüber hinaus auch einfach zu handhaben sind und sich problemlos in beliebig viele Webauftritte benutzen lassen.

Zur Gestaltung einer Website werden HTML- und CSS-Kenntnisse benötigt. PHP ist nicht unbedingt erforderlich, denn die wenigen Vorgaben, die Mambo dafür macht, lassen sich leicht einfügen, ohne Wissen über PHP haben zu müssen.

13.1. Ein eigenes Template erstellen

13.2. Eine kleine Einführung in CSS

Die Syntax einer CSS-Anweisung besteht immer aus dem sogenannten Selektor und der Deklaration, bestehend aus mindestens einer Eigenschaft und zugehörigen Wert, der per Doppelpunkt getrennt werden. Alles steht innerhalb geschweifter Klammern. Eine Deklaration wird immer mit einem Semikolon abgeschlossen.

Abbildung: CSS-Syntax

Selektoren können direkt HTML-Tags (Elemente) sein, Klassen, Kontextsensitiv oder ID-Selektoren.

Leerezeichen werden ignoriert (mit Ausnahme alter Netscape-Versionen 4.x). Somit werden nachfolgende Anweisungen einwandfrei abgearbeitet:

p {color:red;font-size:12pt;}
p{ color : red ; font-size: 12pt; }

Das gleiche gilt für Zeilenumbrüche.

Klassenselektoren

Klassenselektoren werden mit einem Punkt und einem frei vergebbaren Klassennamen eingeleitet. Der Klassennamen darf keine deutschen Umlaute und Leerzeichen enthalten und darf nicht mit einer Zahl oder einem Bindestrich beginnen.

.MeineKlasse { color:green; }
h1.MeineKlasse { color:green; }

Ohne Element gilt die Klasse für alle Tags – mit einleitenden Element ausschließlich für den vorangestellten HTML-Tag.
Der Aufruf erfolgt fogendermaßen:

** … **
<h1 class="MeineKlasse">…</h1>

Wobei für ** ** jeder andere Tag angegeben werden kann.

ID-Selektoren

ID's werden mit einem Gatterzeichen eingeleitet:

#MeineID { color:green; }
Der Aufruf dazu: ** … **

Für den p-Tag kann jeder andere HTML-Tag angegeben werden.

Klasse oder ID?

Eigentlich gibt es zunächst keinen erkennbaren Unterschied zwischen Klasse und ID-Definitionen.
ID's werden für den gesamten Webauftritt zur Definition von zentralen Formaten eingesetzt und sind in aller Regel nicht an einem bestimmten HTML-Tag gebunden (was auch mit CSS 2.0 überhaupt erst möglich ist).
Wohingegen die Klassendefinitionen meist an einem oder wenigen Tags gebunden sind.

Kontextsensitive Selektoren

Es können Formatdefinitionen festgelegt werden, die nur in einem bestimmten Kontext auftreten.
Beispielsweise kann ich mit nachfolgender Anweisung festlegen, dass nur dann, wenn der **b-Tag** (also Fettschrift) innerhalb eines **p-Tags** (Paragraph/Absatz) steht, die Deklaration ausgeführt wird.

p b {color:green;}

HTML: Dieses ist ein beliebiger Beispieltext, der einen kontextsensitiven Selektor beschreibt.
Ausgabe: Dieses ist ein **beliebiger Beispieltext**, der einen kontextsensitiven Selektor beschreibt.

HTML: Dieser Text gibt alles wie üblich aus.
Ausgabe: Dieser Text **gibt alles wie üblich** aus.

13.2.1. Browserprobleme

Leider werden CSS-Definitionen nicht immer von allen Browsern gleich interpretiert. Das liegt zum einen daran, dass 1 Pixel keine klar definierte Größe ist und wenn, sich auch nicht einwandfrei auf einem Bildschirm darstellen ließe. Wieviel mag 1 cm auf einem Bildschirm und der zugehörigen Website sein? Aus diesen Gründen fallen Schriftgrößen bei manchen Browsern anders aus. Für einen Designer eine sehr schwierige Angelegenheiten. So manch perfekt gestylte Site sieht auf einem anderen Browser katastrophal aus!
Hinzu kommt, dass HTML-Anweisungen generell anderes interpretiert werden. Nicht in ihren Grundzügen (eine Überschrift bleibt eine Überschrift), aber in Details gibt es manchesmal Abweichungen. Was aber auch meist an der Pixelproblematik liegt.
Leider ist es auch nicht so, dass von vornherein gesagt werden kann, ein bestimmter Browser fällt regelmäßig aus dem Rahmen. Jeder Browser hat seine eigenen Schwierigkeiten und mal verflucht der Designer diesen, mal den anderen Browser.

Für Webdesigner gilt daher auch, dass eine Website, **bevor(!)** sie der Öffentlichkeit vorgestellt wird, mit den gängigen Browsern abgeprüft wird!

Gängige Browser sind die aktuellen oder fast aktuellen Browserversionen von
- Micorsoft Internet Explorer
- Firefox
- Opera

In letzter Zeit setzen sich auch zunehmend die beiden Browser
- Safari
- Konquerror

durch.

Der Marktanteil von Microsoft legt es nahe, seine Seiten zunächst für diesen Browser zu designen und anschließend,so gut es möglich ist, die Seiten für die anderen Browser anzupassen.

Einen Überblick, welcher Browser was und wie interpretiert gibt es sowohl bei SelfHTML (Adresse im Anhang) als auch bei anderen Webseiten, die sich über Suchmaschinen sicherlich finden lassen.

13.3. Eigene CSS-Klassen definieren

13.4. Page Class Suffix

13.5. Modul Class Suffix

13.6. Administrator-Template

14. Eigene Komponenten entwickeln

14.1. Backend

14.1.1. Strukturen

14.1.1.1. Verzeichnisstruktur

/administrator/components/com_MeineKomponente
Eventuell eigene Sprachdateien, Images oder Hilfedateien in entsprechenden Unterverzeichnissen:
/administrator/components/com_MeineKomponente/help
/administrator/components/com_MeineKomponente/languages
/administrator/components/com_MeineKomponente/images
…

14.1.1.2. Dateienstruktur

Im Backend sind folgende Dateien notwendig:
admin.MeineKomponente.html.php
admin.MeineKomponente.php
toolbar.MeineKomponente.html.php
toolbar.MeineKomponente.php
Dazu können natürlich weitere Dateien entwickelt werden, die dann entsprechend der eigenen Verarbeitung dienen.

14.1.1.3. Version und Copyright

Im Kopf jeder Datei sollte – muß allerdings nicht – folgender Kommentar zu Anfang eingebunden werden (wobei weitere Hinweise natürlich möglich sind):

```
<?php
/**
* @version 1.0.0
* @package MeineKomponente
* @copyright Copyright (C) 2005 IhrName. All rights reserved.
* @license http://www.gnu.org/copyleft/gpl.html GNU/GPL, see LICENSE.php
*/
?>
```

14.1.2. Datei: admin.MeineKomponente.html.php

Diese Datei ist in erster Linie dafür verantwortlich, was im Backend angezeigt wird.

```
<?php

// Datei wird korrekt von Mambo/Joomla aufgerufen oder eben nicht und dann beendet
defined( '_VALID_MOS' ) or
    die( 'Direct Access to this location is not allowed.' );

// Funktion kann jederzeit wieder aufgerufen werden und somit die
// Administrationsebene  der Komponente angezeigt werden (Name beliebig. Wobei
// dieser Teil bei vielen Komponenten auch innerhalb einer Klasse eingebunden ist.).

        function anzeigeBackend() {
?>
```

...
Hier dann der eigene (HTML-)Code, der im Backend angezeigt wird.
...

```
// Abschluß der ‚anzeigeBackend'-Funktion
<?php
}
?>
```

14.1.3. Datei: admin.MeineKomponente.php

Der eigentliche Code. Also was passiert, wenn irgendwo geklickt, etc. wird? In dieser Datei wird alles verarbeitet. Wobei hier alles steht, was nicht der unmittelbaren Anzeige dient (z.B. DB-Abfragen)

```
// Der eigentliche Code muß zunächst in ein Formular eingeschlossen werden, da
sonst die Icons nicht reagieren (siehe auch Punkt 1.4.1)

<form name="adminForm">

<?php
```

```
defined( '_VALID_MOS' ) or die( 'Direct Access to this location is not allowed.' );

// Handling der Tasks – Einbindung der Datei zum Handling der Ereignisse
require_once($mainframe->getPath('admin_html'));

// Abfrage der Icons oben rechts und Zuweisung an $task und die Abfrage der
Formulardaten in $option (siehe unten)
$task=mosGetParam($_REQUEST,'task','');
$option = mosGetParam( $_REQUEST, 'option', '' );

?>

...
Platz für eigenen Code
...

// Die Auswertung von $task und damit der Icons
<?php
switch ($task) {
        // Beispiel:
        case 'save':
                save();
                break;


        default:
                vorbereitungBackend ();
                break;
}

function vorbereitungBackend()
{
SQL-Abfragen
…
// Aufruf der Funktion in admin.MeineKomponente.html.php
anzeigeBackend ();
}

function save()
{
...
}

…
Funktionen zur Verarbeitung von $task – wahlweise auch in anderen Dateien (ggf.
includen)
...

?>


// Das Ende des einschließenden Formulars. Unsichtbar, da es nur zur Auswertung der
Benutzereingaben gebraucht wird.
```

```
// Ohne dieser Zeile springt die Seite immer auf die Admin-Home-Seite zurück (in $option steht 'com_MeineKomponente').
<input type="hidden" name="option" value="<?php echo($option)?>" />
//Ohne dieser Zeile wird auf kein Ereignis reagiert
<input type="hidden" name="task" value="" />
</form>
```

14.1.4. Datei: toolbar.MeineKomponente.html.php

In dieser Datei wird die Iconleiste (Toolbar) oben rechts definiert.
Jede Funktion ermöglicht ein anderes Aussehen – also unterschiedliche Seiten, die ausgewählt wurden (Beispielsweise eine andere Toolbar, wenn ein Datensatz editiert wurde, etc.).

Bei einigen Methodenaufrufen lassen sich weitere Argumente übergeben. Eine detailierte Liste aller möglichen Icons und Übergabe-Parameter finden sich in der Datei (siehe auch Punkt 1.4.2):
/administrator/includes/menubar.html.php

14.1.4.1. Ereignisse auswerten

Um die einzelenen Toolbars (Icons) auswerten zu können, ist es erforderlich, die Datei, die Funktionen mit den Toolbars enthält, in einem Formular zu umschließen.

Die erste Zeile:

```
<form name="adminForm">
```

Und am Ende der Datei diese Zeilen einfügen:

```
<input type="hidden" name="option" value="<?php echo($option)?>" />
<input type="hidden" name="task" value="" />
</form>
```

Dazwischen kann dann beliebiger Code eingefügt werden.

Beispiel:

```
<form name="adminForm">
<?php
defined( '_VALID_MOS' ) or
   die( 'Direct Access to this location is not allowed.' );
function anzeigeBackend()
{
?>
```

(HTML-)Ausgabe

```
<?php
}
?>
```

```
<?php
function irgendwas()
{
?>
```

Andere Ausgabe

```
<?php
}
?>
<input type="hidden" name="option" value="<?php echo($option)?>" />
<input type="hidden" name="task" value="" />
</form>
```

Wichtig ist dabei, dass diese Zeilen am Anfang und Ende des Auswertungsbereichs stehen. Den Versuch, diese Zeilen lediglich in die einzelnen Funktionen zu schreiben, führte bei mir zu einem Fehlverhalten.

14.1.4.2. Mögliche Aufrufe (Kopie aus der Datei *menubar.html.php*)

Den einzelnen Toolbar-Icons können teilweise zusätzliche Parameter übergeben werden. Nachfolgend eine zusammenfassende Kopie aus der *menubar.html.php*. Beachten Sie auch den abschließenden Text zu 'help'.

```
/**
* Writes the start of the button bar table
*/
function startTable()

// Mit custom und customX lassen sich sehr einfach eigene Icons definieren
/**
* Writes a custom option and task button for the button bar
* @param string The task to perform (picked up by the switch($task) blocks
* @param string The image to display
* @param string The image to display when moused over
* @param string The alt text for the icon image
* @param boolean True if required to check that a standard list item is checked
*/
function custom( $task='', $icon='', $iconOver='', $alt='', $listSelect=true )

/**
* Writes a custom option and task button for the button bar.
* Extended version of custom() calling hideMainMenu() before submitbutton().
* @param string The task to perform (picked up by the switch($task) blocks
* @param string The image to display
* @param string The image to display when moused over
* @param string The alt text for the icon image
* @param boolean True if required to check that a standard list item is checked
*/
function customX( $task='', $icon='', $iconOver='', $alt='', $listSelect=true )

/**
```

```
* Writes the common 'new' icon for the button bar
* @param string An override for the task
* @param string An override for the alt text
*/
function addNew( $task='new', $alt='New' )

/**
* Writes the common 'new' icon for the button bar.
* Extended version of addNew() calling hideMainMenu() before submitbutton().
* @param string An override for the task
* @param string An override for the alt text
*/
function addNewX( $task='new', $alt='New' )

/**
* Writes a common 'publish' button
* @param string An override for the task
* @param string An override for the alt text
*/
function publish( $task='publish', $alt='Publish' )

/**
* Writes a common 'publish' button for a list of records
* @param string An override for the task
* @param string An override for the alt text
*/
function publishList( $task='publish', $alt='Publish' )

/**
* Writes a common 'default' button for a record
* @param string An override for the task
* @param string An override for the alt text
*/
function makeDefault( $task='default', $alt='Default' )

/**
* Writes a common 'assign' button for a record
* @param string An override for the task
* @param string An override for the alt text
*/
function assign( $task='assign', $alt='Assign' )


/**
* Writes a common 'unpublish' button
* @param string An override for the task
* @param string An override for the alt text
*/
function unpublish( $task='unpublish', $alt='Unpublish' )

/**
* Writes a common 'unpublish' button for a list of records
* @param string An override for the task
* @param string An override for the alt text
*/
function unpublishList( $task='unpublish', $alt='Unpublish' )
```

```
/**
* Writes a common 'archive' button for a list of records
* @param string An override for the task
* @param string An override for the alt text
*/
function archiveList( $task='archive', $alt='Archive' )

/**
* Writes an unarchive button for a list of records
* @param string An override for the task
* @param string An override for the alt text
*/
function unarchiveList( $task='unarchive', $alt='Unarchive' )

/**
* Writes a common 'edit' button for a list of records
* @param string An override for the task
* @param string An override for the alt text
*/
function editList( $task='edit', $alt='Edit' )

/**
* Writes a common 'edit' button for a list of records.
* Extended version of editList() calling hideMainMenu() before submitbutton().
* @param string An override for the task
* @param string An override for the alt text
*/
function editListX( $task='edit', $alt='Edit' )

/**
* Writes a common 'edit' button for a template html
* @param string An override for the task
* @param string An override for the alt text
*/
function editHtml( $task='edit_source', $alt='Edit HTML' )

/**
* Writes a common 'edit' button for a template html.
* Extended version of editHtml() calling hideMainMenu() before submitbutton().
* @param string An override for the task
* @param string An override for the alt text
*/
function editHtmlX( $task='edit_source', $alt='Edit HTML' )

/**
* Writes a common 'edit' button for a template css
* @param string An override for the task
* @param string An override for the alt text
*/
function editCss( $task='edit_css', $alt='Edit CSS' )

/**
* Writes a common 'edit' button for a template css.
* Extended version of editCss() calling hideMainMenu() before submitbutton().
* @param string An override for the task
```

```
* @param string An override for the alt text
*/
function editCssX( $task='edit_css', $alt='Edit CSS' )

/**
* Writes a common 'delete' button for a list of records
* @param string  Postscript for the 'are you sure' message
* @param string An override for the task
* @param string An override for the alt text
*/
function deleteList( $msg='', $task='remove', $alt='Delete' )

/**
* Writes a common 'delete' button for a list of records.
* Extended version of deleteList() calling hideMainMenu() before submitbutton().
* @param string  Postscript for the 'are you sure' message
* @param string An override for the task
* @param string An override for the alt text
*/
function deleteListX( $msg='', $task='remove', $alt='Delete' )

/**
* Write a trash button that will move items to Trash Manager
*/
function trash( $task='remove', $alt='Trash', $check=true )

/**
* Writes a preview button for a given option (opens a popup window)
* @param string The name of the popup file (excluding the file extension)
*/
function preview( $popup='', $updateEditors=false )

/**
* Writes a preview button for a given option (opens a popup window)
* @param string The name of the popup file (excluding the file extension for an xml file)
* @param boolean Use the help file in the component directory
*/
function help( $ref, $com=false )

/**
* Writes a save button for a given option
* Apply operation leads to a save action only (does not leave edit mode)
* @param string An override for the task
* @param string An override for the alt text
*/
function apply( $task='apply', $alt='Apply' )

/**
* Writes a save button for a given option
* Save operation leads to a save and then close action
* @param string An override for the task
* @param string An override for the alt text
*/
function save( $task='save', $alt='Save' )

/**
```

```
* Writes a save button for a given option (NOTE this is being deprecated)
*/
function savenew()

/**
* Writes a save button for a given option (NOTE this is being deprecated)
*/
function saveedit()

/**
* Writes a cancel button and invokes a cancel operation (eg a checkin)
* @param string An override for the task
* @param string An override for the alt text
*/
function cancel( $task='cancel', $alt='Cancel' )

/**
* Writes a cancel button that will go back to the previous page without doing
* any other operation
*/
function back( $alt='Back', $href='' )

/**
* Write a divider between menu buttons
*/
function divider()

/**
* Writes a media_manager button
* @param string The sub-drectory to upload the media to
*/
function media_manager( $directory = '', $alt='Upload' )

/**
* Writes a spacer cell
* @param string The width for the cell
*/
function spacer( $width='' )

/**
* Writes the end of the menu bar table
*/
function endTable()
```

Eine interessante Besonderheit stellt die Klassenmethode 'help' dar:
mosMenuBar::help('hilfe', true);
bedeutet das die Datei "hilfe.html"aufgerufen wird. Allerdings muß zuvor ein Unterverzeichnis *help* angelegt werden und die Datei muß sich in diesem Verzeichnis befinden.
Wird *false* angegeben, wird die "Joomla.org" Hilfe aufgerufen und das Schlüsselword *hilfe* übergeben. Befindet sich in der "globalen" Hilfe ein Eintrag zu dem Wort, wird dieser angezeigt.

```
<?php
```

```
defined( '_VALID_MOS' ) or
   die( 'Direct Access to this location is not allowed.' );

// Klasse mit Methoden – Namen frei vergebbar
class toolbarProgramm {
	function EditMenu()
	{
		// Start der Iconleiste
		mosMenuBar::startTable();
		// Speichern-Icon
		mosMenuBar::save();
		// Abbrechen-Icon
		mosMenuBar::cancel();
		// Neu-Icon
		mosMenuBar::addNew();
		// Editieren-Icon
		mosMenuBar::editList();
		// Löschen-Icon
		mosMenuBar::deleteList();
		// Ende der Iconleiste
		mosMenuBar::endTable();
	}

	function WeiteresMenu()
	{
		mosMenuBar::startTable();
		// Zurück-Icon
		mosMenuBar::back();
		// Kleiner Zwischenraum zwischen den Icons
		mosMenuBar::spacer();
		// Senkrechter Trennstrich zwischen den Icons
		mosMenuBar::divider();
		// Übernehmen-Icon
		mosMenuBar::apply();
		mosMenuBar::endTable();
	}
}

 ?>
```

14.1.5. Datei: toolbar.MeineKomponente.php

In dieser Datei wird die case-Anweisung zur Auswertung der Variablen ***$task*** geschrieben und damit festgelegt, welche Toolbar wann aufgerufen wird.
Die eigentliche Verarbeitung der einzelnen Icons findet in der Datei *admin.MeineKomponente.php* statt.

```
<?php
defined( '_VALID_MOS' ) or
   die( 'Direct Access to this location is not allowed.' );

// include support libraries
```

```
require_once( $mainframe->getPath( 'toolbar_html' ) );

// Handling der Tasks
$task = mosGetParam( $_REQUEST, 'task', '' );

switch ($task) {
	// Beispielaufruf
	case 'list'
	toolbarProgramm::WeiteresMenu();
	break;

	// Hier ist der Aufruf zur Standard-Toolbar der Komponente
	default:
	toolbarProgramm::EditMenu ();
	break;
}


?>
```

14.1.6. Zusammenfassung

Die beiden toolbar-Dateien regeln die Iconleiste oben rechts. Wobei die *toolbar.meine.**html**.php* für die sichtbare Anzeige zuständig ist und die *toolbar.meine.php* für die Verarbeitung der Iconereignisse insofern zuständig ist, dass hier festgelegt wird, welche Iconleiste bei welcher Auswahl angezeigt wird.

Ähnlich verhält es sich bei den beiden admin-Dateien.
Die *admin.meine.**html**.php* ist dafür verantwortlich, was im Backend angezeigt wird und die Datei *admin.meine.php* ist für die Verarbeitung der Ereignisse zuständig.

Wird mit Hilfedateien gearbeitet, dann müssen diese in einem Unterverzeichnis Namens *help* innerhalb des Admin-Komponenten-Verzeichnisses liegen und mit der Suffix *html* enden.

Bei eigenen Sprachdateien empfiehlt sich ein eigenes Unterverzeichnis Namens *languages*.

14.2.Symbole und Globales

14.2.1. Tabellenlisting

Um ein Tabellenlisting in gewohnter Form auszugeben, wird mit der Admin-CSS-Datei gearbeitet und die entsprechenden Klassen aufgerufen. Die CSS-Dateien liegen in:
administrator/templates/…/css/
Nur vosrichtshalber sei darauf hingewiesen, dass diese CSS-Datei Bestandteil des Admin-Templates ist und somit individuell aussehen kann. Ergo würde es nichts bringen, ein Backend zu entwickeln, welches bestimmte Farben in der CSS-Datei vorraussetzt!

Für den Kopf des "Managers" ist die CSS-Klasse *adminheading* verantwortlich, für die oberste Zeile des eigentlichen Listings *adminlist*. Die einzelnen Zeilen werden mit den beiden tr-Klassen *row0* und *row1* gekennzeichnet.

```
<?php
function TabList(
<table class="adminheading">
	<tr>
		<th>
			// Das Icon vor der Tabellenüberschrift wird automatich
			   hinzugefügt (über der CSS-Datei)
			Beispiel Listing
		</th>
		<td>
			// Das Suchenfeld
			Filter: <input type="text" name="search" value="<?php echo
			$search;?>" class="text_area"
			onChange="document.adminForm.submit();" />
		</td>
	</tr>
</table>

// Das folgende Code-Fragment habe ich aus der admin.modules.html.php entnommen
(also der Datei, die u.a für die Anzeige des Modulmanagers zuständig ist) und leicht
modifiziert
<table class="adminlist">
	<tr>
		<th width="20px">
			#
		</th>
		<th width="20px">
			// Checkbox mit der alle Checkboxen ausgewählt werden können
			<input type="checkbox" name="toggle" value=""
			onclick="checkAll(<?php echo count( $rows );?>);" />
		</th>
		<th class="title">
			Name
		</th>
		<th nowrap="nowrap" width="10%">
			Published
		</th>
		<th colspan="2" align="center" width="5%">
			Reorder
		</th>
		<th width="2%">
			Order
		</th>
		<th width="1%">
			// Das Diskettensymbol zum speichern der Reihenfolge
			<a href="javascript: saveorder( <?php echo count( $rows )-1; ?> )">
			<img src="images/filesave.png" border="0" width="16" height="16"
			alt="Save Order" /></a>
		</th>
		<th nowrap="nowrap" width="7%">
			Access
		</th>
		<th nowrap="nowrap" width="5%">
			Pages
		</th>
```

```
        <th nowrap="nowrap" width="5%">
                ID
        </th>
</tr>
// ab hier die einzelnen Tabellenzeilen mit der Auflistung der erstellten Objekte
<?php
        // Zunächst eine Schleife, um die einzelnen Tabellenzeilen farblich zu
        unterscheiden – also die beiden row-CSS-Klassen zu erzeugen (wobei
        $rows zuvor übergeben werden muß! – Kann beispielsweise mit
        folgenden Zeilen erzeugt werden:
        $query=[SQL-ANWEISUNG]
        $database->setQuery( $query );
        $rows = $database->loadObjectList();
        Diese Zeilen sollten in der admin.MeineKomponente.php stehen und
        dann per Funktionsaufruf entsprechen übergeben werden.
        $k = 0;
        for ($i=0, $n=count( $rows ); $i < $n; $i++) {

        // Initialisierung der einzelnen Zeile
        $row = &$rows[$i];

        // Leider habe ich nachfolgende Klasse niergends finden können und
        kann daher nichts weiter zu ihr sagen …
        $access         = mosCommonHTML::AccessProcessing( $row, $i );
        $checked        = mosCommonHTML::CheckedOutProcessing( $row, $i );
        $published      = mosCommonHTML::PublishedProcessing( $row, $i );
?>
<tr class="<?php echo "row$k"; ?>">
        <td align="right">
                // Ausgabe: # (Laufende Nummer)
                <?php echo $pageNav->rowNumber( $i ); ?>
        </td>
        <td>
                 // Ausgabe: Checked-Formularfeld
                <?php echo $checked; ?>
        </td>
        <td>
                // Ausgabe: Name
                <?php echo $row->title;?>
        </td>
        <td align="center">
                // Ausgabe: Modul/Komponente veröffentlicht/unveröffentlicht
                <?php echo $published;?>
        </td>
        // Ausgabe: Icons für Sortierung aufwärts/abwärts
        <td>
                <?php echo $pageNav->orderUpIcon( $i, ($row->position ==
                @$rows[$i-1]->position) ); ?>
        </td>
        <td>
                <?php echo $pageNav->orderDownIcon( $i, $n, ($row->position ==
                @$rows[$i+1]->position) ); ?>
        </td>
        <td align="center" colspan="2">
                // Ausgabe: Zahlenfeld für Sortierung
```

```
                <input type="text" name="order[]" size="5" value="<?php echo
                $row->ordering; ?>" class="text_area" style="text-align: center" />
        </td>
        <td align="center">
                // Ausgabe: Berechtigungsstufe
                <?php echo $access;?>
        </td>
        <td align="center">
                // Ausgabe: Komponente wird auf einer/mehreren/allen Seiten
                angezeigt
                <?php
                if (is_null( $row->pages )) {
                        echo 'None';
                        } else if ($row->pages > 0) {
                                echo 'Varies';
                        } else {
                                echo 'All';
                        }
                        ?>
        </td>
        <td align="center">
                // Ausgabe: ID
                <?php echo $row->id;?>
        </td>
        </tr>
                // Abschluß der "row-Schleife"
                <?php
                $k = 1 - $k;
        }
        ?>
</table>
```

14.3. Frontend

14.3.1. Strukturen

14.3.1.1. Verzeichnisstruktur

Das eigene Verzeichnis sollte, wie auch schon im Adminbereich mit *com_* beginnen. Das ist zwar nicht zwingend notwendig, erleichtert jedoch immens den Überblick …
/components/com_MeineKomponente/

14.3.1.2. Dateienstruktur

Im obigen Verzeichnis muß eine php-Datei mit dem Komponentennamen liegen. Mehr ist nicht notwendig. Eventuelle weitere Dateien ergeben sich aus der individuellen Programmierung.
MeineKomponente.php

14.3.2. Datei: MeineKomponente.php

Diese Datei ist für die Anzeige im Frontend verantwortlich und wird letztlich auch per Menüpunkt oder Verlinkung aufgerufen. Einfacher HTML-Code reicht völlig aus. Aber zur Anzeige von Datenbankausgaben, etc. ist natürlich etwas mehr erforderlich ;)

```
// Neben der bereits oben erwähnten PHP-Kommentar-Einbindung zu Beginn der Files,
// kann hier auch mit dem 'mos:comment'-Tag ein Kommentar eingebunden werden:
<mos:comment>
@version 1.0.0
@package MeineKomponente
@copyright Copyright (C) 2005 IhrName. All rights reserved.
@license http://www.gnu.org/copyleft/gpl.html GNU/GPL, see LICENSE.php
</mos:comment>

// Hier folgt der (HTML-)Code zur Anzeige der Komponente im Frontend
```

14.4.Installation

Zur Installation wird unbedingt eine XML-Datei benötigt. Bei Bedarf kann auch ein Install-PHP-File erstellt werden, welches aus der XML-Datei aufgerufen wird. Alles muß am Ende, einschl. der richtigen Verzeichnisse, in der die Dateien liegen werden, gepackt werden. Entweder als ZIP- oder als TAR-Datei.

14.4.1. XML-Datei

```
<?xml version="1.0" ?>
<mosinstall type="component">
Der Name Ihrer Komponente
  <name>MeineKomponente</name>
Erstellungsdatum
  <creationDate>29.12.2005</creationDate>
Ihr Name oder Firmenname
<author>Axel Tüting</author>
Lizensrechtliches. Wenn die Komponente kostenpflichtig ist, dann sollten Sie hier
einen entsprechenden Vermerk eingeben
<copyright>This component is released under the GNU/GPL License</copyright>
<authorEmail>tueting@time4mambo.de</authorEmail>
<authorUrl>www.time4mambo.de</authorUrl>
Version der Komponente
<version>1.0</version>
```

Wenn keine Kurzbeschreibung vorgesehen ist, dann kann dieser XML-Tag auch weggelassen werden

```
<description>Diese Komponente verdient auch eine Kurzbeschreibung, die dann hier steht.</description>
```

Alle Files die im Frontend-Bereich installiert werden. Wenn Images oder Dateien, die in Unterverzeichnisse liegen, installiert werden, dann müssen sowohl hier die Pfade entsprechend angegeben werden, als auch im gepackten File diese Dateien in den entsprechenden Verzeichnissen liegen

```
<files>
	<filename>MeineKomponente.php</filename>
	<filename>index.html</filename>
</files>
	Wenn Grafiken vorhanden sein sollten – kann ansonsten weggelassen werden
	<images>
		<filename>images/meineGrafik.jpg</filename>
</images>
<install>
	<queries>
		Soviele query wie Tabellen, bzw. Inserts
		<query>DROP TABLE IF EXISTS `jom_MeineKomponente_settings`;</query>
		<query>CREATE TABLE `jom_MeineKomponente_settings` (…)</query>
		<query>INSERT INTO `jom_MeineKomponente_settings`(…);</query>
	</queries>
</install>
<uninstall>
	<queries>
		<query>DROP TABLE IF EXISTS 'jom_MeineKomponente_settings`;</query>
</queries>
</uninstall>
```

Wenn kein Install-File vorhanden ist, kann dieser XML-Tag auch weggelassen werden

```
<installfile>
	<filename>install.MeineKomponente.php</filename>
</installfile>
```

Wenn kein Uninstall-File vorhanden ist, kann dieser XML-Tag auch weggelassen werden

```
<uninstallfile>
	<filename>uninstall. MeineKomponente.php</filename>
</uninstallfile>
<administration>
	Zunächst das Menü im Backend:
	<menu>MeineKomponente</menu>
	Und das oder die Untermenüs, sofern vorhanden
	<submenu>
		task muß von Ihnen im Programmcode entsprechend verarbeitet sein
		<menu task="aufruf">Administration</menu>
	</submenu>
	<files>
		Hier müssen alle Dateien aufgeführt sein, die im Admin-Bereich stehen.
		Dabei gilt gleiches mit den Verzeichnissen wie weiter oben unter <files>
		<filename>index.html</filename>
		<filename>admin. MeineKomponente.php</filename>
		<filename>admin. MeineKomponente.html.php</filename>
		<filename>toolbar. MeineKomponente.html.php</filename>
```

```
		<filename>toolbar. MeineKomponente.php</filename>
	</files>
	Wenn Grafiken vorhanden sein sollten – kann ansonsten weggelassen werden
	<images>
		<filename>images/meineGrafik.jpg</filename>
	</images>
</administration>
</mosinstall>
```

Wenn Sie den obigen Code für Ihre Entwicklung kopieren, denken Sie bitte daran, meinen Namen durch Ihre Daten zu ersetzen ;)
So tauchen immer mal wieder Templates auf, in deren template-xml mein Name steht, weil der entsprechende Template-Designer achtlos den Code aus meinem Template-Tutorial kopiert hat …

14.4.2. Install-File

Eine extra Install-Datei ist nicht notwendig. Manche schreiben dort jedoch ein Dankeschön für die Installation der Komponente rein, andere schreiben dort aber auch Zusatzinstallationen rein, wie beispielsweise das Anhängen eigener Sprach-DEFINEs an den vorhandenen Sprachfiles. Weitere Möglichkeiten sind natürlich auch möglich.

```
<?php
function com_install()
{
...
}
?>
```

14.5.Uninstall

Für das Uninstall-File gilt das gleiche, wie für das Install-File. Eine Datei, die nicht vorhanden sein muß. Meist eine Ausgabe mit dem Hinweis, dass alles sauber deinstalliert wurde oder dem Dankeschön, fürs ausprobieren der Komponente.

```
<?php
function com_uninstall()
{
...
}
?>
```

14.6.Index.html

In den Verzeichnissen empfiehl sich eine einfache index.html zu legen, um unerlaubtes Aufrufen der Verzeichnisse zumindest mit einfachen Mitteln zu erschweren.

```
<html><body bgcolor="#FFFFFF"></body></html>
```

Diese Zeile als *index.html* abgespeichert reicht vollkommen.

14.7.Sprachdateien

Zum einen gibt es die Möglichkeit eigene Sprachdateien zu erzeugen, die dann im Admin-Verzeichnis oder/und im Komponentenbereich des Frontend der Komponente liegen sollten. Dann muß aber an eine Auswahlmöglichkeit innerhalb der Konfiguration für die Komponente gedacht werden.

Besser (meiner Meinung nach) ist es allerdings, seine eigenen DEFINEs an die vorhandene german.php und/oder english.php dran zu hängen, da dann der User automatisch bei Aufruf der Komponente gleich die richtige Sprache sieht.

15. Was geht wo

Manches mal sucht man etwas und weiß auch ganz genau, dass das irgendwie geht – doch wo es steht, weiß man nicht mehr.
Und obwohl Mambo recht logisch aufgebaut ist, muß einem schon dieser logische Aufbau bekannt sein, um es auch tatsächlich zu finden.

So erhalten wir immer wieder in unserem Forum Anfragen, wo steht, wie was sortiert wird und wie man die Startseite seinen eigenen Bedürfnissen anpassen kann.

Im nachfolgenden Kapitel will ich daher den Versuch unternehmen, bestimmte Fragen, die immer mal wieder auftauchen, nach anderer Sortierung zu beantworten. Wie es dann im Einzelnen funktioniert, lesen Sie bitte in den zugehörigen Kapiteln nach.

15.1.Ordnung muß sein

Wenn Sie die Reihenfolge der Artikel innerhalb eines Menüpunktes sortieren wollen, dann schauen Sie doch mal beim zuständigen Menüpunkt vorbei. Dort finden Sie unter *Parameter* im zweiten Abschnittsbereich die beiden Punkte *Category Order* und *Primary Order*. Ebenfalls können Sie dort einstellen, wieviele Spalten angezeigt werden und einiges mehr.

15.2.Frontpage einstellen

15.3.Haupt- und Untermenüanzeige getrennt

16. Anhang

16.1.Wo steht was

16.1.1. Die Verzeichnisstruktur

Im Mambo **root** (also der ersten Verzeichnisebene) liegen folgende Dateien direkt nach der Installation:

configuration.php-dist
index.php
htaccess.txt

> *Hinweis*: Pfade, die innerhalb seiner eigenen Templatedatei, also der index.php des Templates benutzt werden, gehen stets vom Root-Verzeichnis aus.

images
Hier befinden sich die Bilder, die bei den Bereichen und Kategorien ausgewählt werden können

images/stories
Ab diesem Unterverzeichnis befinden sich alle weiteren Bilder, die in den Modulen, Komponenten (soweit vorgesehen) und Artikeln angezeicht werden können. Wobei es ab hier auch die Möglichkeit gibt, eigene Verzeichnisse anzulegen. Alle Bilder können über den Mediamanager eingebunden werden.

16.1.2. Dateien

configuration.php
In dieser Datei befinden sich alle relevanten Daten, um Mambo zum laufen zu bekommen. Dazu gehören die Pfade bestimmter Verzeichnis (z.B. wo Mambo auf dem Server genau liegt), Die Datenbank-Daten und ein paar weiterer Daten. Näheres wird weiter unten besprochen.
index.php

htaccess.txt
Wenn Sie Ihr System in der Globalen Configuration auf SEF umstellen, müssen Sie diese Datei in *.htaccess* umändern (den Punkt am Anfang nich vergessen!). In einigen FTP-Programmen ist diese Datei dann nicht mehr sichtbar.

16.2. Mehrere Mambos auf einer Website

Eine DB – mehrere Mambos
Mehrere DB's – mehrere Mambos
Unterschiedliche Versionen (siehe auch XamppLite)

16.3.Webseiten

16.3.1. Communitys, Hilfen, Downloads

Eines der größten deuschsprachigen Supportforen mit großer Filebase, in dem die Autoren als Webmaster fungieren:
http://www.mamboportal.ch

16.3.2. Kommerzielle Webseiten

Axel Tüting seine Site: time4Mambo – Programmierung, Administration, Support, Design, Text. Mit freien Downloads:
http://www.time4mambo.de

Marcus Riedewald seine Site zum Webhosting, auch mit freien Downloads:
http://www.mambohosting.org

16.3.3. Darüber hinaus

DIE Website zum Thema HTML, JavaScript und mittlerweile auch PHP – SelfHTML:
http://selfhtml.teamone.org

Index

Archiv 24
Artikel **12**, **29**
Assistenten 31
Backend 6, **10**
Bereich 6, **13**, 16
Bereich Archiv Blog **23**
Bereich Blog **23**
Bereich Manager **26**
Bereich Managers 15
Bereiche **22**
Bereichsliste **23**
Blog Content Category 18
Components 12
Content 6
Content Category Archiv Blog 26
Content Category Blog 26
Content Category Table 25
Content Management System 8
formatieren **34**
Frontend 6
Globale Konfiguration 30
Hilfe 12
Home **11**
HTML-Tag 6
HTML-Tags **34**
Inhalt 7, 11, **28**
Inhalte 16
Inhalte nach Bereich 16
Installer 12
Intro Text 16
JavaScript 31
Kategorie 7, **28**
Kategorie Blog **25**
Kategorie Manager 15
Kategorien 16, **24**
Kategorien Manager **28**
Komponenten 7, 11
Latest Items 12
Logged **12**
Mambots 7, 11
Media Manager **35**
Menü 11
Menüeinträge 18
Menümanager 18
menustats 12
Module 7, 11
News 15
Popular 12
Popup **31**
Public 13
Publishing 16
Pulldownmenü **11**
Sektion 7, **13**
statisch 44, **45**
System 12
Tag 7
Templates **51**
TinyMCE **30**
Upload **36**
Verweis **31**
WYSIWYG 7
Zugriffslevel 13, 23
Zugriffsrechte *Siehe* Zugriffslevel