

vanillaXML Webshop 1.1.0 - Handbuch

1. Allgemein

- 1.1.0. Die Benutzeroberfläche
- 1.1.1. Suchmaschinenfreundliche URL's
- 1.1.2. Über den Hersteller
- 1.1.3. Überprüfung der Systemanforderungen

2. Benutzer

- 2.1.0. Benutzer löschen
- 2.1.1. Login / Logout

3. Einstellungen

- 3.1.0. Allgemeine Einstellungen
- 3.1.1. Cache
- 3.1.2. Metatags

4. Module

- 4.1.0. Aufruf der Module
- 4.2. Statistik
- 4.3. Werbung
- 4.4. vanillaXML Webshop

5. Vorlagen

- 5.1.0. Allgemeines zu den Vorlagen-Sets
- 5.1.1. Alternative Seitenlayouts für einzelne Bereiche
- 5.1.2. Kompletter Wechsel zwischen den Vorlagen-Sets
- 5.1.3. Statische Seiten in vanillaXML Webshop einbinden
- 5.1.4. Stylesheets anpassen
- 5.1.5. Vorlage "index_member.htm"
- 5.1.6. Vorlage "index_visitor.htm"
- 5.1.7. Vorlage "main.htm"
- 5.1.8. Vorlage "print.htm"
- 5.1.9. Vorlage "table.htm"

6. Sonstiges

- 6.1.0. Copy & Paste
- 6.1.1. Filter
- 6.1.2. MD5-Passwortverschlüsselung
- 6.1.3. Partnerprogramm
- 6.1.4. Vorlagen-Service

7. {cp:tags}

- 7.1.0. {cp:content}
- 7.1.1. Allgemein
- 7.1.2. Aufbau der Tags
- 7.1.3. Sprachelemente in Vorlagen: {cpl:tags}
- 7.1.4. {cp:assign}
- 7.1.5. {cp:break}
- 7.1.6. {cp:buildtime}
- 7.1.7. {cp:configuration}
- 7.1.8. {cp:container module="fast_gallery"}
- 7.1.9. {cp:container module="navigation"}
- 7.1.10. {cp:container module="search"}
- 7.1.11. {cp:container module="statistic"}
- 7.1.12. {cp:container module="user"}
- 7.1.13. {cp:container module="webservices"}

- 7.1.14. {cp:date}
- 7.1.15. {cp:header}
- 7.1.16. {cp:help}
- 7.1.17. {cp:homepage}
- 7.1.18. {cp:ifnot}
- 7.1.19. {cp:if}
- 7.1.20. {cp:ignore}
- 7.1.21. {cp:image}
- 7.1.22. {cp:include}
- 7.1.23. {cp:ismember}
- 7.1.24. {cp:language}
- 7.1.25. {cp:lastlogin}
- 7.1.26. {cp:level}
- 7.1.27. {cp:linksnavigation}
- 7.1.28. {cp:link}
- 7.1.29. {cp:path}
- 7.1.30. {cp:pdf}
- 7.1.31. {cp:poster}
- 7.1.32. {cp:print}
- 7.1.33. {cp:random}
- 7.1.34. {cp:rating}
- 7.1.35. {cp:rewrite}
- 7.1.36. {cp:rights}
- 7.1.37. {cp:systemlogo}
- 7.1.38. {cp:top}
- 7.1.39. {cp:variable}
- 7.1.40. {cp:vote}

8. Entwickler-API

- 8.1.0. Beispiel-Modul
 - 8.1.1. cp_absolute_path()
 - 8.1.2. cp_array_average()
 - 8.1.3. cp_array_structure()
 - 8.1.4. cp_array_sum()
 - 8.1.5. cp_currency_info()
 - 8.1.6. cp_delete_cache()
 - 8.1.7. cp_get_cache()
 - 8.1.8. cp_lang_info()
 - 8.1.9. cp_mime_info()
 - 8.1.10. cp_module_exists()
 - 8.1.11. cp_random_string()
 - 8.1.12. cp_read_cache()
 - 8.1.13. cp_set_cache()
 - 8.1.14. cp_validate_color()
 - 8.1.15. cp_validate_email()
 - 8.1.16. cp_validate_number()
 - 8.1.17. cp_validate_string()
 - 8.1.18. cp_word_shorten()
- 8.2. Allgemeines zur Modulentwicklung
- 8.3. Benutzer
- 8.4. CSS
- 8.5. Datumsfunktionen
- 8.6. Formulare
- 8.7. Grafik
- 8.8. Mail
- 8.9. RTF
- 8.10. URL
- 8.11. Verschiedenes
- 8.12. XML
- 8.13. Zeichen & Texte
- 8.14. ZIP
- 8.15. FTP - File Transfer Protocol

9. FAQ - Häufig gestellte Fragen

- 9.1.0. Es treten beim Absenden von Formularen Fehler auf. Es wird immer ein Download-Fenster geöffnet!
- 9.1.1. Ich habe die Verschlüsselung per SHA1-Algorithmus aktiviert, jetzt treten Probleme auf!
- 9.1.2. Ich habe mein Passwort verloren, wie kann ich es zurückerhalten?
- 9.1.3. Ist die Installation von Software / Plugins auf dem lokalen Rechner notwendig?
- 9.1.4. Ist die Verwaltung großer Dateien (> 1 MB) möglich?
- 9.1.5. Ist ein Einsatz im Intranet oder Extranet möglich?

- 9.1.6. Kann ich meine Seite gezielt für Suchmaschinen optimieren?
- 9.1.7. Können {cp:tags} in PHP-Dateien verarbeitet werden?
- 9.1.8. Nach dem Ändern von Einstellungen werden diese nicht aktualisiert. Warum?
- 9.1.9. Sind fertige Vorlagen-Sets verfügbar?
- 9.1.10. Sind zusätzliche Schulungen erforderlich?
- 9.1.11. Warum werden nicht alle Inhalte nach dem Bearbeiten aktualisiert?
- 9.1.12. Welche Aufgabe hat die Datei ".htaccess"?
- 9.1.13. Welche Limitierungen bei der Erstellung des Layouts gibt es?
- 9.1.14. Wie gehe ich beim Update auf eine neue vanillaXML Webshop-Version vor?
- 9.1.15. Wie kann ich die Fehlerseiten ändern?
- 9.1.16. Wie sieht es mit der Sicherheit aus?
- 9.1.17. Wie werden eigene Vorlagen-Sets erstellt?
- 9.2. Installation / Systemanforderungen
- 9.3. Lizenzbedingungen
- 9.4. Vorlagen / Entwicklung

1. Allgemein

*In Zeiten, in denen der digitale Geschäftsverkehr immer präsenter wird und überall von Webservices der Zukunft die Rede ist, benötigen speziell klein- und mittelständische Unternehmen leistungsstarke Programme, die die tägliche Arbeit erleichtern. **Contentpapst** ist ein solches Programm für den Online-Bereich. Entdecken Sie die Möglichkeiten, welche sich durch **Contentpapst** eröffnen und Ihnen Zeit und damit Geld sparen helfen. Den Content-Management-Systemen gehört die Zukunft!*

1.1.0. Die Benutzeroberfläche

vanillaXML Webshop ist bewusst einfach und übersichtlich strukturiert. Nach dem Login über die Mitgliederverwaltung mit Ihren Benutzerdaten wechseln Sie in das Administrations-Menü des Systems. Hier finden Sie eine dreigeteilte Benutzeroberfläche vor.

Im Kopfbereich können Sie zu Ihrer Website zurückkehren, das Hilfe-System aufrufen oder sich aus dem System ausloggen. Wenn aktuelle Aufgaben anstehen, so erscheint ein entsprechender Hinweis-Button .

Auf der linken Seite sind die wichtigsten Bereiche von vanillaXML Webshop aufgelistet. Sie können hier direkt zu den Einstellungen, der Vorlagen-Verwaltung usw. wechseln. Die Inhalte der einzelnen Bereiche werden dann jeweils auf der rechten Seite ausgegeben. Wenn ein Bereich umfangreichere Funktionen beinhaltet, so wird über dem Inhaltsbereich eine Leiste mit weiteren Unterbereichen eingeblendet. Einzelne Bereiche wie die Module sind in noch weiter unterteilt und bieten reichhaltige Möglichkeiten zur Anpassung Ihrer Website.

1.1.1. Suchmaschinenfreundliche URL's

Alle Seiten werden bei vanillaXML Webshop dynamisch bzw. semi-dynamisch erzeugt und können dann temporär in einem Cache zwischengespeichert werden. Dies ermöglicht eine stets aktuelle Seite, gleichzeitig kann durch den Cache das Lastaufkommen drastisch gesenkt werden.

Diese dynamischen Seiten sind meist über URL's mit angehängten Parametern erreichbar, mit denen vanillaXML Webshop mitgeteilt wird, beispielsweise welche Seite eines Artikels gerade gelesen wird. Diese URL's (z.B. "index.php?file=article&mode=detail&number=32&page=5") können problematisch sein, wenn es um die Aufnahme in Suchmaschinen geht. Einige Suchmaschinen indizieren dynamische Seiten aus verschiedenen Gründen grundsätzlich nicht.

Für vanillaXML Webshop gibt es allerdings eine elegante Lösung, um diese Blockade zu umgehen. Mit Hilfe der Option "Suchmaschinenfreundliche URL's" in den Einstellungen, können URL's auf ein normales HTML-Format gebracht werden. Hierzu werden Veränderungen in der Datei ".htaccess" vorgenommen. Die weiter oben als Beispiel genannte Artikel-Seite wäre mit aktivierter Option auch unter "article-entry-32-5.html" abrufbar. Für Suchmaschinen und den Besucher augenscheinlich eine normale HTML-Seite, intern jedoch weiterhin PHP.

Um bei einem Apache-Webserver diese Option nutzen zu können, muss die Erweiterung "mod_rewrite" installiert sein. Für den Microsoft IIS-Webserver wird ebenfalls ein Zusatztool benötigt, hier kann unter anderem IIS Rewrite zum Einsatz kommen. Für den IIS müssen die in der ".htaccess" enthaltenen Regeln in einer Datei "rewrite.ini" abgelegt und ins Verzeichnis "%windir%\system32\inet\IISRewrite" verschoben werden. Im letzteren Fall sind zusätzlich absolute statt relative Pfade anzugeben.

1.1.2. Über den Hersteller

vanillaXML Webshop ist ein Produkt der sandoba.de medien agentur aus Berlin. sandoba.de entwickelt gezielt Lösungen für kleine und mittelständische Unternehmen. Zu den Kerngebieten zählt die Programmierung von dynamischen Webapplikationen auf PHP/MySQL-Basis, erweitert um Technologien wie XML/XSL, Datenbanksysteme (neben MySQL u.a. PostGreSQL, Oracle, ODBC) usw., sowie die Erstellung von Internetauftritten und sonstigen multimedialen Präsentationen. Die sandoba.de medien agentur ist unter www.sandoba.de zu erreichen.

1.1.3. Überprüfung der Systemanforderungen

vanillaXML Webshop wird direkt auf einem Webserver installiert, d.h. auf dem Server auf dem Sie auch normale HTML-Seiten lagern. Voraussetzung ist, dass der Webserver, also auch ihr Webespace über PHP (Programmiersprache, ab Version 4.06) und eine MySQL-Datenbank (ab Version 3.23) verfügen. Dies können Sie überprüfen, indem Sie folgenden Text in eine Datei mit dem Namen "phpinfo.php" kopieren, diese per FTP auf Ihren Webspace laden und über den Browser aufrufen:

```
<?php
phpinfo();
?>
```

Sofern PHP ordnungsgemäß installiert ist, finden Sie hier alle wesentlichen Informationen über Ihren Webserver, Installierte Erweiterungen von PHP usw.. Direkt am Anfang sehen Sie die Version von PHP, auch zu MySQL gibt es einen Abschnitt. Eine Übersicht zu den benötigten Systemanforderungen der aktuellen Version finden Sie immer unter www.sandoba.de/article-entry-85.html. Wenn Sie eine ältere PHP-Version als 4.06 angezeigt bekommen, dann kann vanillaXML Webshop auf Ihrem Webserver nicht installiert werden.

2. Benutzer

Die Benutzer- / Rechteverwaltung ermöglicht sowohl die Bearbeitung der Benutzer, als auch das Versenden von Emails an einzelne Benutzer. Wenn Sie auf Ihrer Website eine besondere Sicherheitsstruktur nutzen wollen, so sollten Sie hier die einzelnen Rechte für die verschiedenen Gruppen an Benutzern vergeben.

2.1.0. Benutzer löschen

Eingeloggte Benutzer können auf Wunsch ihren jeweiligen Account komplett aus der Datenbank löschen. Dies wird aufgrund der Datenschutzbestimmungen in einigen Ländern der EU notwendig. Eine Abmeldung aus dem Newsletter muss getrennt davon vorgenommen werden. Die Löschung des Accounts ist unter "Benutzerprofil bearbeiten" am Ende der Seite unter "index.php?file=user" möglich.

2.1.1. Login / Logout

Um als angemeldeter Benutzer in den Administrations-Bereich zu wechseln, muss der Login über "[index.php?file=user](#)" durchgeführt werden bzw. über den Container "{cp:container module=user}". Als Angaben sind der Benutzername und das Passwort notwendig, wie vorab bei der Registrierung festgelegt. Bei aktiviertem Sicherheitscode muss zusätzlich der Text aus der generierten Grafik in ein Eingabefeld übernommen werden.

3. Einstellungen

Beinahe alle Einstellungen des Systems können nach der Installation direkt über den Administrations-Bereich durch die berechtigten Nutzer angepasst werden. Dazu gehören z.B. Standard-Metatags, Cache und Logging.

3.1.0. Allgemeine Einstellungen

In den "Allgemeinen Einstellungen" finden Sie viele elementare Einstellungen, die für den Betrieb des Systems wichtig sind. Dazu gehören u.a. die Einstellungen der richtigen URL's und einer Standardsprache. Viele dieser Einstellungen sind direkt auf dieser Seite erklärt. Einige Einstellungen seien aber zusätzlich hier erwähnt:

- *Domain der Website* - Domain auf der die Website liegt (bspw. "http://www.meinedomai.de")
- *Pfad zum CMS* - Der Pfad ausgehend von der Domain (bei "http://www.domain.de/cp/" ist es "cp/")
- *GZIP-Komprimierung* - Stärke der Komprimierung der Inhalte der Website vor der Auslieferung
- *Datum / Datum (Kurzform)* - Formatierung des Datums auf der Website
- *Standardsprache* - Auswahl der Standardsprache aus den für das System verfügbaren Sprachen
- *Zeichensatz* - Verwendeter Zeichensatz (wird je nach Sprache automatisch gesetzt)

3.1.1. Cache

vanillaXML Webshop speichert viele Inhalte nach dem einmaligen Abruf in einem Cache. D.h. sobald z.B. ein Artikel durch einen Besucher aufgerufen wird und dieser noch nicht im Cache vorhanden ist, generiert vanillaXML Webshop automatisch eine Kopie, die im Cache auf dem Server abgelegt und an den Besucher ausgeliefert wird. Weitere Besucher erhalten bis zum Ablauf der Cache-Zeit automatisch die Cache-Datei. Dies spart Rechenzeit und erhöht die Performance Ihres Webservers erheblich.

Der Cache ist in vanillaXML Webshop u.a. für Module wie die Artikel standardmäßig vorhanden und kann dort jeweils einzeln gesetzt werden. Alle Angaben sind dabei in Sekunden zu machen (für einen Tag bspw. "86400" Sekunden). Bei "0 Sekunden" bzw. ohne Auswahl des Caches bleibt dieser deaktiviert. Dies kann beispielsweise bei Inhalten sinnvoll sein, die dynamische Elemente wie z.B. PHP beinhalten. Als Besonderheit führt vanillaXML Webshop im Hintergrund bei allen Aktionen automatische Aktualisierungen des Caches durch. Daher erscheinen Inhalte immer in der neuesten Fassung auf der Website, ohne dass erst der Ablauf der regulären Cache-Zeit abgewartet werden muss. Eine weitere Besonderheit ist der individuelle Cache für jeden einzelnen Container. Hier kann über den Parameter "cache" stets eine Cache-Zeit in Sekunden angegeben werden.

In den Einstellungen haben Sie die Möglichkeit, den Cache insgesamt zurückzusetzen. Dann wird der gesamte Inhalt des Ordners "cache" entfernt. Dies kann sinnvoll sein, wenn Änderungen an genutzten Vorlagen vorgenommen wurden etc.

3.1.2. Metatags

vanillaXML Webshop unterstützt zur Optimierung der Seite für Suchmaschinen etc. auch die Vergabe von individuellen Metatags. Voraussetzung ist, dass in der Vorlage "main.htm" des genutzten Vorlagen-Sets {cp:tags} im Format "{cp:meta:*}" eingesetzt wurden.

In den Einstellungen können Sie im gleichnamigen Bereich allgemeingültige Metatags angeben. Diese werden immer dann zum Einsatz gebracht, wenn es für den jeweils angezeigten Inhalt keine eigenen Metatags gibt. U.a. sind als Metatags Angaben zum Titel der Seite, zu Beschreibung, Schlüsselwörtern, dem Verhalten der Suchmaschinen etc. vorhanden.

Weiterhin gibt es auf dieser Seite einen Bereich "Neue Metatags erstellen" über den Sie allgemein den HTML-Code für die Metatags erstellen können. Hier können Sie alternativ direkt Texte eingeben. Wenn Sie diese jedoch im Vorlagen-Set verwenden, findet ein Austausch je nach Seite nicht mehr statt.

Neben diesen allgemeinen Metatags gibt es dann für Module wie die Artikel jeweils eigene Eingabefelder. So können Sie beispielsweise einem Artikel gezielt die passenden Schlüsselwörter mitliefern und so Ihre Seite weiter optimieren.

4. Module

*Module sind eigenständige Teile des Systems. Dies sind beispielsweise Kalender, ein Online-Shop, Foren oder Galerien. Module greifen u.a. auf zentrale Dateien von **Contentpapst**, die Entwickler-API, sowie auf die jeweils genutzten Templates und Sprachdateien zu. Aufgrund dieser Modularisierung lässt sich **Contentpapst** schnell und einfach erweitern. Bei der Grundversion nicht enthaltene Module, können auch später eingebaut werden.*

4.1.0. Aufruf der Module

Fast alle Module innerhalb des vanillaXML Webshop-Frameworks können über die URL "index.php?file=[MODULNAME]" für den öffentlichen Bereich bzw. "admin.php?file=[MODULNAME]" für den Administrationsbereich aufgerufen werden. Voraussetzung hierfür ist, dass die entsprechenden Dateien vorhanden sind und das Modul als Unterordner im Ordner "modules" liegt. An der Stelle von "[MODULNAME]" ist der Name des Modules (d.h. der Ordnername) einzusetzen.

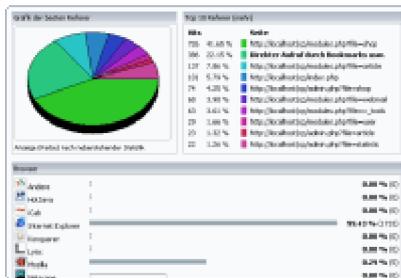
Viele Module bieten darüberhinaus eigene {cp:tags}. Die Inhalte des Modules müssen also nicht unbedingt nur über die direkte URL aufgerufen werden, sondern es ist auch eine Verwendung auf anderen Seiten möglich.

4.2. Statistik

Wenn Sie genauere Informationen über Ihre Besucher erhalten wollen, so sind Sie hier richtig. Sie finden in diesem Bereich Informationen über die genutzten Browser und Betriebssysteme sowie darüber, aus welchen Ländern Ihre Besucher stammen, welche Seiten wie oft aufgerufen wurden usw.

4.2.1. Allgemein

Das Statistik-Modul zeigt Ihnen im oberen Bereich die Gesamtanzahl der Seitenaufrufe innerhalb der bisherigen Laufzeit der Statistik, sowie die durchschnittlichen Seitenaufrufe pro Tag und Stunde. Daneben befinden sich die Optionen der Statistik. Alle Darstellungen werden in Echtzeit aus der Datenbank generiert.



Die darunter liegende Tabelle stellt die Seitenaufrufe der letzten 30 Tage zusammen mit prozentualen und absoluten Angaben dar. Weiterhin gibt es auf dieser Seite noch die Auflistung der Referer. Wenn die Option "Eigene Seiten als Referer aufzeichnen" aktiviert ist, können Sie hier ablesen, welche Ihrer Seiten am meisten aufgerufen wurden. Die grafische Darstellung verdeutlicht zusammen mit der Farbkodierung noch einmal den Anteil der einzelnen Seiten.

Die folgenden Tabellen geben Ihnen Informationen über die genutzten Browser und Betriebssysteme der Besucher. Sollte es sich dabei um Suchmaschinen, Crawler usw. handeln, so werden diese getrennt aufgezeichnet. Genauere Angaben über die Seitenaufrufe über einen

Zeitraum (Monate, Wochentage, Stunden) hinweg erhalten Sie in den entsprechenden Tabellen. Wenn Sie die Herkunftsländer der Besucher aufzeichnen lassen, so steht Ihnen auf einer dritten Seite zusätzlich noch eine Übersicht über diese Länder zur Verfügung.

4.2.2. Eigene Seiten als Referer aufzeichnen

Optional ist die Funktion "Eigene Seiten als Referer aufzeichnen". Diese bewirkt, dass einzelne Seiten Ihrer Website als Referer in die Datenbank aufgenommen werden. Dadurch erhalten Sie Informationen über die von Ihren Besuchern am häufigsten aufgerufenen Seiten.

4.2.3. Herkunftsländ aufzeichnen

Das Herkunftsland Ihrer Besucher wird anhand des Server-Namens ermittelt, über welchen diese auf Ihre Website gelangen. Ist diese Option aktiviert, so wird für jeden Seitenaufruf das Herkunftsland protokolliert, mit den vorhandenen Einträgen verglichen und in die Datenbank eingetragen. Allerdings verlängert sich durch die Überprüfung des Server-Namens die Ladezeit der einzelnen Seiten.

4.2.4. Statistik aufräumen

Die von der Statistik gesammelten Daten wachsen mit der Laufzeit der Website immer weiter an. Besonders Daten über die Herkunftsseiten ("Referer") Ihrer Besucher (sowie die Anzahl der Zugriffe von diesen Seiten) belegen in der Datenbank zunehmend einen größeren Speicherplatz.

Jedoch werden nicht alle diese Informationen für die grafische oder tabellarische Aufbereitung benötigt. Für die Herkunftsseiten sind nur die 100-1000 Seiten mit den meisten Zugriffen auch bedeutsam. Alle weiteren Einträge können mit dieser Funktion automatisch aus der Datenbank gelöscht werden.

4.2.5. Statistik neustarten

Mit einem Klick auf "Statistik jetzt neustarten" lässt sich die Statistik komplett neustarten. Alle bisherigen Einträge werden dadurch zurückgesetzt bzw. aus der Datenbank gelöscht. Wenn Sie vorher ein Backup durchführen wollen, so können Sie dies mit Hilfe des Modules "Sicherung" durchführen. Wählen Sie hier alle Tabellen mit dem Namen "_stats_" aus und speichern Sie Struktur und Daten dieser Tabellen auf Ihrer Festplatte.

4.3. Werbung

Schalten Sie Werbeeinblendungen auf Ihrer Website und verdienen Sie. Sie können verschiedene Werbeformen parallel laufen lassen, indem Sie den Tag `{cp:banner}` (Anleitung) auf der Seiten-Vorlage einbauen. Derzeit sind neben den Standardformaten für Bannerwerbung auch Werbeschaltungen als Flashfilm oder per Text/HTML möglich.

4.4. vanillaXML Webshop

Diese Anleitung soll Ihnen die zahlreichen Funktionen von vanillaXML Webshop näher bringen. Dies beinhaltet sowohl den öffentlichen Bereich und den Administrations-Bereich, als auch die direkte Arbeit mit der Entwickler-API von vanillaXML Webshop.

4.4.1. Allgemeines zum Einsatz der Container

Die Container können Sie im normalen Rahmen der vorhandenen Vorlagen einsetzen oder Sie platzieren diese in einer beliebigen HTML-Datei, die Sie unterhalb des Hauptverzeichnisses ablegen und dann über "index.php?static=[PFAD ZUR DATEI.HTM]&clean=yes" aufrufen. Über diesen Weg ist es möglich, Produkte von Amazon z.B. innerhalb anderer Texte, einem Forum etc. anzuzeigen. An ein festes Layout sind Sie durch die freie Gestaltung der HTML-Vorlagen nicht gebunden.

4.4.2. Anpassung der Vorlagen

Fast alle Elemente des vanillaXML Webshop können über Vorlagen an den individuellen Geschmack angepasst werden. Für vanillaXML Webshop kommen dabei zusätzlich zu den normalen Vorlagen des Vorlagen-Sets im Ordner "themes" weitere Vorlagen im Ordner "modules/amazon/templates/" zum Einsatz. Dies sind im einzelnen die folgenden Vorlagen:

- **"browsenode.htm"**:
Diese Vorlage kommt in der Detailansicht zum Einsatz in der Auflistung der verwandten Kategorien.
- **"cart.htm"**:
Der Warenkorb mit der Auflistung aller abgelegten Produkte.
- **"details.htm"**:
Detailansicht eines Produktes mit Anzeige von Tracks, Funktionen etc. je nach dem Katalog des Produktes
- **"error_noitems.htm"**:
Vorlage, die angezeigt wird, wenn innerhalb einer Kategorie keine Produkte vorhanden sind.
- **"error_noresults.htm"**:
Vorlage, die angezeigt wird, wenn bei einer Suchanfrage kein Ergebnis geliefert wird.
- **"img_full.htm"**:
Anzeige des vergrößerten Bildes nach Klick auf das Bild in der Detailansicht eines Produktes.
- **"list.heavy.htm"**:
Vorlage für die Auflistung von Produkten mit erweiterten Angaben.
- **"list.lite.htm"**:
Vorlage für die Auflistung von Produkten mit einfachen Angaben (Standardvorlage für die Auflistung).
- **"node_show.htm"**:
Anzeige einer einzelnen Kategorie mit Anzahl der enthaltenen Produkte etc.
- **"reviews.htm"**:
Auflistung der einzelnen Kundenmeinungen zu einem Produkt.
- **"search_full.htm"**:
Vorlage für die Darstellung des Suchergebnisses (Übersicht über die einzelnen Kataloge).
- **"search_show.htm"**:
Vorlage für die Darstellung des Suchergebnisses (Detailansicht einer einzelnen Suche).
- **"similar.htm"**:
Auflistung der verwandten Produkte in der Detailansicht eines Produktes.

- **"sitemap_standard.htm"**:
Standardvorlage für die Erstellung einer Sitemap über das Sitemap-Tool.

4.4.3. Container: Beliebteste Kategorien

Um Ihren Kunden weitere Kaufempfehlungen zu geben, können Sie die aus den Statistiken im Administrationsbereich bekannte Auflistung der beliebtesten Kategorien auch im öffentlichen Bereich über einen Container anzeigen lassen:

```
{cp:container module="amazon" type="favouritecategories" [limit="5"]}
```

Über den Parameter "limit" ist eine Eingrenzung der maximalen Kategorieanzahl möglich. Ist "limit" nicht gesetzt, wird automatisch nur eine Auflistung der ersten 10 Kategorien ausgeliefert. Um durch diesen Container nicht die Ladezeit der Website zu erhöhen, sollten Sie diesen über den Parameter "cache" zusätzlich für eine längere Zeit zwischenspeichern (z.B. "86400" Sekunden für einen Tag).

4.4.4. Container: Beliebteste Produkte

Um Ihren Kunden weitere Kaufempfehlungen zu geben, können Sie die aus den Statistiken im Administrationsbereich bekannte Auflistung der beliebtesten Produkte auch im öffentlichen Bereich über einen Container anzeigen lassen:

```
{cp:container module="amazon" type="favouriteproducts" [limit="5"]}
```

Über den Parameter "limit" ist eine Eingrenzung der maximalen Produktanzahl möglich. Ist "limit" nicht gesetzt, wird automatisch nur eine Auflistung der ersten 10 Produkte ausgeliefert. Um durch diesen Container nicht die Ladezeit der Website zu erhöhen, sollten Sie diesen über den Parameter "cache" zusätzlich für eine längere Zeit zwischenspeichern (z.B. "86400" Sekunden für einen Tag).

4.4.5. Container: Besondere Eigenschaften (bei Elektronik, DVD und Zeitschriften)

Vergleichbar zu den Plattformen bei Software etc. gibt es für die Kataloge Elektronik, DVD und Zeitschriften zusätzliche Informationen in Form der detaillierten Eigenschaften (bspw. die max. Auflösung bei einem Computer-Monitor). Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="features" file="abc.xml"}  
// Besondere Eigenschaften, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt. Auch diese Informationen können wieder per CSS angepasst werden.

4.4.6. Container: Letzte Suchanfragen

Um Ihre Besucher bei der Suche ein wenig zu inspirieren, bietet es sich an, die letzten Suchanfragen an geeigneter Stelle einzublenden. Durch einen Klick auf die einzelnen Begriffe ist dann schnell ein Wechsel zur jeweiligen Ergebnisseite der Suche möglich. Häufig nachgefragte Produkte können so im Absatz noch weiter gestärkt werden.

Über die Parameter "limit" und "catalog" kann zusätzlich die Ausgabe beeinflusst werden. "limit" gibt hier die maximale Anzahl der angezeigten Suchanfragen an, "catalog" begrenzt hingegen die Ausgabe auf einen bestimmten Produktkatalog wie z.B. Musik (Katalog "music").

```
{cp:container module="amazon" type="last_searches" limit="10" cache="180"}  
// Beispiel mit 10 Suchanfragen und aktiviertem Cache
```

4.4.7. Container: Musiktitel (nur bei Musik)

Nur beim Katalog Musik stehen bei vielen Produkten genaue Informationen über die einzelnen Musiktitel zur Verfügung. Diese Informationen sind für Ihre Besucher natürlich besonders wichtig. Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="tracks" file="abc.xml"}  
// Besondere Eigenschaften, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt. Auch diese Informationen können wieder per CSS angepasst werden.

4.4.8. Container: Navigation einblenden (Produktkategorien)

Die Navigation aller Produktkategorien wird von vanillaXML Webshop automatisch in Abhängigkeit der gerade geöffneten Seite und der ausgewählten Sprache aufgebaut. Die Integration der Navigation in die Website geschieht über einen einfachen {cp:tag} in einer beliebigen Vorlage:

```
{cp:container module="amazon" type="navigation"}
```

4.4.9. Container: Plattformen (bei Software und Spielen)

Für die Kataloge Software und Computer- & Videospiele werden zusätzliche Informationen über die unterstützten Plattformen, d.h. die Betriebssysteme geliefert. Für vanillaXML Webshop steht zur Einblendung dieser Informationen ein weiterer Container zur Verfügung:

```
{cp:container module="amazon" type="plattformen" file="abc.xml"}  
// Produkt-Plattformen, generiert aus der Datei abc.xml
```

Damit die richtige Datei als Datenquelle genutzt wird, muss diese über den Parameter "file" angegeben werden. Allgemein sollte dieser Container nicht manuell gesetzt werden, sondern wird direkt in der Detailansicht eines Produktes eingesetzt.

Auch diese Informationen können wieder per CSS angepasst werden. Soweit vorhanden wird für die Plattformen (u.a. Windows XP, Windows 2000, Gameboy, Playstation 2, X-Box etc.) das passende Logo daneben eingeblendet.

4.4.10. Container: Produktinformationen abrufen

Wenn Sie einzelne Produkte innerhalb anderer Seiten dynamisch einbinden möchten, können Sie dies direkt über einen {cp:tag} als Container realisieren. Der Container ermöglicht es Ihnen zusätzlich die Produkte beliebig oft in unterschiedlichen Layouts darzustellen.

```
{cp:container module="amazon" type="products" asin="B0000AJ595" catalog="music" template="abc.htm"}  
// Musik-CD Dido "Life For Rent" dargestellt über die Vorlage "abc.htm".
```

Wenn im obigen {cp:tag} der Parameter "template" nicht angegeben wird, verwendet vanillaXML Webshop automatisch die Vorlage "modules/amazon/templates/details.htm". Für eine manuelle Anpassung der Abfrage über die Programmiersprache PHP empfiehlt sich ein Blick in die Datei "modules/amazon/container/products.php".

4.4.11. Container: Suchabfragen durchführen

Besonders interessant - gerade beim Einsatz von vanillaXML Webshop als vanillaXML Webshop-Modul - sind die Container für die Durchführung von Suchabfragen. In einem Text zum Thema Tauchsport bspw. könnte direkt ein Container platziert werden, der thematisch passende Bücher bei Amazon heraussucht und diese dort einblendet. Der Absatz lässt sich durch derartige Verknüpfungen stark steigern.

Im Folgenden werden eine Reihe von Containern vorgestellt, jeweils mit einem Beispiel. Die Parameter "query" (bspw. das Suchwort), "catalog" (der Produktkatalog), "complexity" (Komplexität der Abfrage, "lite" oder "heavy"), "sort" (Sortierung), "format" (standardmäßig "xml") und "page" (aktuelle Seite) sind jeweils allgemeingültig. Über den Parameter "limit" muss zusätzlich die gewünschte Anzahl der Suchergebnisse angegeben werden. Maximal sind dies 10 Suchergebnisse pro Seite.

Je nach Einstellung des Parameters "complexity" wird die Vorlage "modules/amazon/templates/list.heavy.htm" oder die Vorlage "modules/amazon/templates/list.lite.htm" genutzt. Alternativ kann über den Parameter "template" eine eigene Vorlage eingesetzt werden.

Suche nach Schlüsselwörtern:

```
{cp:container module="amazon" type="searchkeyword" query="Tauchen"}  
// einfache Suche nach Büchern mit dem Begriff "Tauchen"
```

```
{cp:container module="amazon" type="searchkeyword" query="Fussball" catalog="vhs" sort="+title" page="2"}  
// Suche nachc Fussball-Videos, Seite 2 geordnet nach dem Titel
```

Suche nach Autoren:

```
{cp:container module="amazon" type="searchauthor" query="Dan Brown"}  
// einfache Suche nach Büchern mit dem Autor "Dan Brown"
```

Suche nach ähnlichen Produkten:

```
{cp:container module="amazon" type="searchsimilarity" query="B00005Q66T"}  
// Suche nach ähnlichen Produkten zum Produkt mit der ASIN "B00005Q66T"
```

Suche nach Regisseuren / Produktionsleitung:

```
{cp:container module="amazon" type="searchdirector" query="Andrew Stanton"}  
// einfache Suche nach Büchern mit dem Regisseur "Andrew Stanton"
```

Suche nach Künstlern:

```
{cp:container module="amazon" type="searchartist" query="Madonna" catalog="music"}  
// einfache Suche nach Musik-CD's von "Madonna"
```

Suche nach Hersteller:

```
{cp:container module="amazon" type="searchmanufacturer" query="Sony" catalog="electronics"}  
// Suche "Sony" im Katalog "Elektronik"  
// unterstützt werden hier "electronics", "kitchen", "videogames", "software", "photo" und "pc-hardware"
```

Suche nach Schauspielern:

```
{cp:container module="amazon" type="searchactor" query="Renée Zellweger" catalog="dvd"}  
// einfache Suche nach DVD's mit der Schauspielerin "Renée Zellweger"
```

Suche nach Produktkategorien:

```
{cp:container module="amazon" type="searchbrowsenode" query="288100"}  
// Produkt in der Kategorie "Fachbücher" werden hier gesucht
```

Suche nach Produkten per UPC (Unique Product Code):

```
{cp:container module="amazon" type="searchupc" query="B0000CFW87"}  
// Suche nach einem UPC (momentan nicht in Deutschland möglich)
```

Aufgrund der Ladezeit, die teilweise durch die Amazon-Server entstehen kann, empfiehlt es sich über den Parameter "cache" die einzelnen Suchabfragen in den Zwischenspeicher legen zu lassen (bspw. "86400" für 24 h).

4.4.12. Container: Warenkorb mit Produktinformationen

Über die Einblendung des Warenkorbs als Container haben Ihre Besucher stets im Blick, welche Produkte bereits dort abgelegt wurden und für die Bestellung bereitgestellt sind. Der Container kann wie üblich innerhalb einer beliebigen Vorlage platziert werden. Der Aufbau wird direkt von vanillaXML Webshop realisiert, kann aber mittels CSS angepasst werden.

```
{cp:container module="amazon" type="cart"}
```

Im Container werden zu jedem Produkt Menge und Produktname eingeblendet, sowie der Gesamtpreis (ohne event. Versandgebühren). Sollten keine Produkte im Warenkorb liegen wird eine Alternativmeldung ausgegeben.

4.4.13. Fehlercodes / -meldungen

vanillaXML Webshop liefert bei verschiedenen Methoden der API Fehlercodes / -meldungen zurück. Die wichtigsten dieser Meldungen sind hier aufgelistet, zusammen mit den jeweiligen Methodennamen:

- 0 - Keine Affiliate-ID angegeben. (get_affiliate_link/get_affiliate_id/set_affiliate_id)

- 1 - Keine Länderkennung angegeben. (get_locality/set_locality)
- 2 - Notwendige Angaben fehlen.
(get_dev_key/set_dev_key/set_cache_time/get_browse_node_link/get_keyword_link)
- 3 - Keine Amazon-Webservices in diesem Land verfügbar.
- 4 - Dieser Modus existiert nicht.
- 5 - Status kann nur "open" oder "closed" sein.
- 6 - Daten konnten erfolgreich geholt werden.

4.4.14. Suchmaschinenfreundliche URL's aktivieren

Der vanillaXML Webshop ist bereits standardmäßig für optimale Suchmaschinenfreundlichkeit vorbereitet. Dazu gehört u.a. auch die Simulation von statischen Seiten mit der Endung ".html". Dies wird über eine Erweiterung des Apache-Webservers mit dem Namen "mod_rewrite" möglich. Ausführliche Informationen finden Sie u.a. in "Allgemein: Suchmaschinenfreundliche URL's" und "FAQ: Kann ich meine Seite gezielt für Suchmaschinen optimieren?".

Zusätzlich zu den dort genannten Schritten zur Aktivierung der suchmaschinenfreundlichen URL's benötigt der vanillaXML Webshop einige weitere Regeln, die in der Datei ".htaccess" zu platzieren sind:

```
RewriteRule ^(de)/([a-z]+)/([0-9]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=catalog&catalog=$2&page=$3&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/search/([a-z]+)/([^\w]+)/([0-9]+)/([^\w]+),([a-z]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=search&q=$3&form[catalog]=$2&form[sort]=$5&form[complexity]
=$6&page=$4&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/search/([a-z]+)/([^\w]+)/([0-9]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=search&q=$3&form[catalog]=$2&page=$4&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/(search|cart)/ \http://[DOMAIN]/index.php?
file=amazon&mode=$2&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/(author|artist)/([^\w]+)/ \http://[DOMAIN]/index.php?file=amazon&mode=search&form[type]
=$2&q=$3&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/([a-z]+)/details/([A-Za-z0-9]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=details&number=$3&catalog=$2&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/([a-z]+)/([A-Za-z0-9]+)/([^\w]+)/([0-9]+)/(\?) \http://[DOMAIN]/index.php?
file=amazon&mode=node&number=$3&page=$5&catalog=$2&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/([a-z]+)/([A-Za-z0-9]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=node&number=$3&catalog=$2&language=$1&mod_rewrite=1 [L]
```

```
RewriteRule ^(de)/([a-z]+)/ \http://[DOMAIN]/index.php?
file=amazon&mode=catalog&catalog=$2&language=$1&mod_rewrite=1 [L]
```

"[DOMAIN]" ersetzen Sie bitte durch Ihren vollen Domainnamen sowie gegebenenfalls den Pfad zum Verzeichnis, in dem sich vanillaXML Webshop befindet.

Die URL's werden dann vom System meist automatisch auf das neue Format geändert. Wenn Sie in den Vorlagen bisher die statischen URL's genutzt haben, dann müssen Sie diese entsprechend der Regeln anpassen. U.a. sind die Verlinkungen von Grafiken, CSS-Dateien etc. auf absolute Verlinkungen mit "/" am Anfang zu ändern.

4.4.15. Übersicht über die Methoden der Entwickler-API

Die Entwickler-API ist die direkte Schnittstelle von vanillaXML Webshop zu Amazon. Auf der Basis einiger grundlegenden Methoden werden Daten von Amazon geholt, verarbeitet und ausgegeben. Mit ein wenig Eingewöhnung werden Sie schnell eigene Klassen ableiten können, die die von Ihnen benötigten Funktionen optimal kombinieren.

Folgende Methoden gehören aktuell zur Entwickler-API:

- *cache_images()* - speichert ein Bild lokal ab
- *check_availability()* - überprüft, ob das Produkt auch verfügbar ist bzw. nachbestellt werden kann
- *check_format()* - überprüft die Gültigkeit des Formats ("xml")
- *check_type()* - überprüft die Gültigkeit des Typs ("lite"/"heavy")

- `create_product_list()` - generiert mittels einer Vorlage eine Produktaufistung (z.B. für Übersichtsseiten)
- `create_review_list()` - erstellt eine Liste der Rezensionen eines Produkts mittels einer Vorlage
- `create_similiar_list()` - erstellt eine Liste der verwandten Produkte zu einem oder mehreren Produkten
- `db_query()` - hält die Verbindung zur Datenbank (für die Statistik)
- `form_add_items()` - Formular zum Hinzufügen von Produkten
- `form_add_marketplace()` - Formular zum Hinzufügen von Marktplatz-Produkten
- `form_add_wishlist()` - Formular zum Hinzufügen von Produkten zur Wunschliste
- `get_affiliate_id()` - Affiliate-ID erhalten
- `get_affiliate_link()` - Liefert anhand von \$type einen Affiliate-Link
- `get_asin_link()` - Generiert Link zu Produktseite mit Affiliate-ID anhand einer ASIN (mit \$direct direkt zur Produktseite)
- `get_dev_key()` - Developer-Key erhalten
- `get_browse_node_link()` - generiert Link zu einer bestimmten Kategorie mit Affiliate ID
- `get_content()` - holt den XML-Inhalt aus einer Suchabfrage ab und speichert diesen
- `get_info()` - gezielt eine einzelne Information zu einem Produkt abfragen
- `get_keyword_link()` - Generiert Link zu Produktsuche mit verschiedenen Parametern
- `get_link()` - Generiert einen beliebigen Link zu Amazon (immer ab "tg/browse/..." angeben)
- `get_locality()` - Landeskennung erhalten
- `get_search_url()` - setzt die URL für die Abfragen zusammen
- `get_rewrite_name()` - stellt für get_url den Namen im mod_rewrite-Format zusammen
- `get_url()` - erzeugt die URL zu einem Artikel / einer Kategorie
- `local_data()` - liefert landesspezifische Inhalte zu Amazon
- `num_results()` - gibt die Anzahl der Ergebnisse einer Sucheabfrage zurück
- `reset()` - Setzt die temporären Daten zurück (intern genutzt)
- `save()` - Speichert je nach Parameter XML oder den Inhalt als Datei ab (intern genutzt)
- `search_actor()` - Suche nach Schauspielern (liefert passende DVDs)
- `search_artist()` - Suche nach Künstlern (liefert passende CDs)
- `search_asin()` - Details über ein bestimmtes Produkt heraussuchen
- `search_author()` - Suche nach Autoren (liefert passende Bücher)
- `search_blended()` - Suche über alle Kategorien hinweg
- `search_browse_node()` - Suche nach Produktkategorien
- `search_director()` - Suche nach Produktionsleitung (liefert passende DVDs)
- `search_exchange()` - Finds a given exchange
- `search_isbn()` - Greift auf search_asin zurück
- `search_keyword()` - Führt eine Stichwort-Suche durch
- `search_listmania()` - Sucht Listen nach Listmania-IDs
- `search_manufacturer()` - Suche nach Hersteller
- `search_power()` - Erweiterte Suche ("PowerSearch")
- `search_seller()` - Findet Produkte zu einer Seller-ID
- `search_seller_profile()` - Liefert das Profil zu einer Seller-ID
- `search_similarity()` - Sucht ähnliche Produkte zu einer ASIN
- `search_upc()` - Suche nach UPC (nur Music)
- `search_wishlist()` - Sucht Wunschlisten nach ID
- `set_affiliate_id()` - Affiliate-ID festlegen
- `set_cache_time()` - legt Cache-Zeit in Sekunden fest (anhand der Amazon-Regeln)
- `set_dev_key()` - Developer-Key festlegen
- `set_locality()` - Landeskennung festlegen
- `set_price()` - Preisgrenzen festlegen (noch nicht überall bei Amazon möglich)
- `set_options()` - Detaileinstellungen verändern
- `set_show_all()` - zeigt alle Produkte an, egal ob Verfügbar oder nicht

5. Vorlagen

Die komplette Gestaltung der Website kann in **Contentpapst** mittels des Vorlagen-Modules festgelegt werden. Stellen Sie hier die Farben und sonstige Eigenschaften der einzelnen Seitenelemente zusammen. Vorlagen trennen die technischen Grundlagen des Systems von grafischen und sonstigen Inhalten.

5.1.0. Allgemeines zu den Vorlagen-Sets

Die Vorlagen-Sets beinhalten jeweils alle unterstützten Vorlagen und stellen somit ein gesamtes Design der Website dar. Diese werden im Ordner "themes" als eigener Unterordner eingebunden und stehen automatisch im System zur Verfügung. Wenn ein Wechsel zwischen den Vorlagen-Sets vorgenommen werden soll, kann dieser in den Einstellungen unter "Aktuelles Vorlagen-Set" vollführt werden.

Neue Vorlagen-Sets lassen sich hier ebenso erstellen. Geben Sie dazu einen neuen Namen ein und vanillaXML Webshop legt automatisch ein neues Vorlagen-Set auf der Basis des aktuell ausgewählten Vorlagen-Sets an.

5.1.1. Alternative Seitenlayouts für einzelne Bereiche

Einzelne Bereiche können in vanillaXML Webshop mit einem komplett anderen Seitenlayout dargestellt werden. Die gewünschte Seite muss dazu mit dem Parameter "set_template" aufgerufen werden. D.h. wenn der Artikel mit der ID "84" im Seitenlayout "alternative.htm" aufgerufen werden soll (diese Datei liegt im Ordner des verwendeten Vorlagen-Sets und ist mit der "main.htm" vergleichbar) muss dieser per Link wie folgt aufgerufen werden:

*/index.php?file=article&mode=entry&number=84&set_template=alternative

Aus dem Parameter "set_template" wird der Dateiname zusammengesetzt und um die Dateierweiterung ".htm" erweitert. Ist diese Datei vorhanden, wird sie zur Darstellung des Layouts genutzt, andernfalls kommt das Standardlayout zum Einsatz.

5.1.2. Kompletter Wechsel zwischen den Vorlagen-Sets

Ein kompletter Wechsel zwischen den Vorlagen-Sets kann ebenfalls ermöglicht werden, beispielsweise um registrierten Nutzern eine alternative Website mit angepasster Oberfläche zu präsentieren. Bauen Sie hierzu auf einer beliebigen Seite bzw. direkt im Inhalt (Artikel/News etc.) einen Link nach folgendem Muster ein:

set_options.php?mode=template&template=alternative

"alternative" muss dabei durch den Namen des gewünschten Vorlagen-Sets ersetzt werden. Für registrierte Nutzer werden die Einstellungen im Profil gespeichert, andernfalls wird eine Speicherung in der Session vorgenommen.

5.1.3. Statische Seiten in vanillaXML Webshop einbinden

Speziell bei älteren Projekten existieren oftmals Daten in unterschiedlichen Formaten, welche z.B. in Form von statischen HTML-Dateien vorliegen. Generell empfiehlt es sich, diese Inhalte in vanillaXML Webshop zu importieren, um auf erweiterte Funktionen wie die Rechteverwaltung, die Volltextsuche oder Caching-Verfahren zurückgreifen zu können.

Sollte dies jedoch nicht gewünscht werden, können die Vorteile des Vorlagen-Systems von vanillaXML Webshop dennoch genutzt werden. Hierzu wird die statische Seite über die Datei "index.php" mittels des Parameters "\$file" aufgerufen und durch vanillaXML Webshop verarbeitet.

In der HTML-Datei können alle in vanillaXML Webshop vorhandenen {cp:tags} genutzt werden. Dies erlaubt u.a. die Einbettung von Artikeln, die Anzeige von Forenbeiträgen, eines Datums oder die Anzeige bestimmter Bereiche nur für Mitglieder. Über den optionalen Parameter "\$clean" lässt sich die Anzeige des allgemeinen Seitenlayouts ein- und ausschalten.

Alle HTML-Dateien müssen aus Sicherheitsgründen innerhalb des vanillaXML Webshop-Verzeichnisses (bzw. in einem der Unterverzeichnisse) abgelegt werden.

*/index.php?static=faq.htm

Ruft die Datei "faq.htm" aus dem Hauptverzeichnis auf und verarbeitet alle in dieser Datei vorhandenen {cp:tags}. Die gesamte Datei wird in das Layout der Website integriert.

*/index.php?static=lokalinformation/steuerabgaben.htm&clean=yes

Über diese URL wird die Datei "steuerabgaben.htm" aus dem Verzeichnis "lokalinformation" aufgerufen. Auch hier werden die enthaltenen {cp:tags} verarbeitet, eine Integration in das Layout der Website unterbleibt jedoch. Dies erlaubt es, ein komplett eigenständiges Layout aufzubauen.

5.1.4. Stylesheets anpassen

Wesentliche Teile der grafischen Gestaltung werden über die CSS-Stylesheets gesteuert. Diese sind in der Datei "style.css" des jeweils genutzten Vorlagen-Sets integriert. Sie legen den genutzten Schrifttyp mit einer Standardgröße und -farbe, sowie die Gestaltung der Scrollleisten fest. Darüberhinaus werden alle Formular-Elemente wie Eingabefelder, Checkboxes etc. durch die Stylesheets formatiert.

Eine Umgestaltung aller Textboxen beispielsweise kann dann zentral in dieser Datei vorgenommen werden und gilt für alle Bereiche, welche dieses Vorlagen-Set einsetzen. Auch Navigationselemente werden in der Datei "style.css" gestaltet. CSS-Angaben, welche im Vorlagen-Set bereits genutzt werden, sollten nicht entfernt werden. Hier ist nur eine Anpassung an das Layout der Website ratsam.

5.1.5. Vorlage "index_member.htm"

Über die Vorlage "index_member.htm" eines Vorlagen-Sets wird die Startseite für eingeloggte Mitglieder gestaltet. Eine derartige Startseite sollte im Gegensatz zur Startseite für Besucher alle Neuerungen auf der Seite prägnant präsentieren, damit Mitglieder schnell informiert sind. FAQ's und Rundgänge etc. sind hier nicht angebracht.

5.1.6. Vorlage "index_visitor.htm"

Über die Vorlage "index_visitor.htm" steuern Sie die Ansicht für die Besucher. Besucher verlangen meist eine übersichtliche Startseite mit Informationen über die Benutzung der Seite. Hier können Sie z.B. gezielt einen schnellen Zugriff auf einen Rundgang oder die FAQ's platzieren.

5.1.7. Vorlage "main.htm"

Die Hauptseite eines Vorlagensets steuert die grundlegende Darstellung der Website. In diese werden mittels der {cp:tags} im Nachhinein alle Inhalte eingebettet. Die Gestaltung ist dementsprechend frei und unterliegt keiner Eingrenzung durch interne oder externe Scripte. Durch die Tags legen Sie hier sogenannte "Container" für Navigationsmenüs, Werbeplätze, Umfragen, Artikel usw. an, welche später dynamisch mit Inhalten gefüllt werden. Ausführliche Informationen über die möglichen Tags finden sich in der Anleitung unter "{cp:tags}".

Für den Inhaltsbereich müssen Sie in dieser Vorlage den Tag "{cp:content}" einmalig einfügen. Zusätzlich sollten Sie die CSS-Datei "style.css" innerhalb des Ordners des Vorlagen-Sets einbinden, damit Formulare etc. auf der Website korrekt dargestellt werden können.

Besondere {cp:tags}:
{cp:content}

5.1.8. Vorlage "print.htm"

Viele Inhalte können direkt in einem druckfreundlichen Format ausgegeben werden. Die Vorlage "print.htm" wird hierfür mit Daten gefüllt, welche vorher über den Tag {cp:print} übermittelt worden. Es sollte eine Breite von 600 Pixeln bei der Gestaltung der Vorlage nicht überschritten werden, damit beim Druck alle Inhalte korrekt ausgegeben werden können.

Besondere {cp:tags}:
{cp:header}, {cp:content}

5.1.9. Vorlage "table.htm"

Die Vorlage "table.htm" dient zur Anpassung der allgemein auf der Website genutzten Tabellen. Alle Tabellen welche nicht über eine externe Vorlage angepasst werden können, werden auf der Basis dieser Datei aufgebaut. U.a. sind dies Teile des Mitgliedersystems. Innerhalb der Vorlage sind die Tags {cp:header} (für den Titel der Tabelle) und {cp:content} (für den Inhalt) zu verwenden.

Besondere {cp:tags}:
{cp:header}, {cp:content}

6. Sonstiges

*In diesem Bereich sind alle Funktionen, Angebote usw. zusammengefasst, welche zusätzliche Dienste ansprechen. Dies können z.B. Informationen zum Vorlagen-Service und zum Partnerprogramm für **Contentpapst** oder aber auch zur Generierung von Newsfeeds sein.*

6.1.0. Copy & Paste

Das System von **vanillaXML Webshop** ist offen für textbasierte Inhalte aus anderen Anwendungen. Sie können diese Inhalte aus Textbearbeitungsprogrammen, WYSIWYG-Editoren usw. einfach herauskopieren und in **vanillaXML Webshop** direkt einfügen, ohne Dateien verschieben zu müssen. Für den WYSIWYG-Editor gibt es (in der erweiterten Ansicht) zusätzlich einige Funktionen um per Copy & Paste eingefügte Inhalte zu säubern (damit vanillaXML Webshop eine valide Ausgabe generieren kann).

6.1.1. Filter

Gerade in öffentlich zugänglichen Gästebüchern, Foren und anderen interaktiven Elementen einer Website kann es oftmals zu Äußerungen der Mitglieder kommen, welche nicht gesetzeskonform sind oder gegen den guten Geschmack verstoßen. Diese Äußerungen lassen sich durch den in vanillaXML Webshop integrierten Filter entschärfen.

Der Filter erlaubt es, einzelne Worte aus von Mitgliedern eingegebenen Texten systematisch durch "*" ersetzen zu lassen. Die Abfrage ist dabei "case-insensitive", d.h. Wörter in unterschiedlicher Groß- und Kleinschreibung werden automatisch erkannt ("Sport", "sport", "SPoRT" usw.).

6.1.2. MD5-Passwortverschlüsselung

Alle sensitiven Daten der Mitglieder, Redakteure und Administratoren sind durch Benutzername und Passwort gesichert. Das Passwort wird zusätzlich mittels des MD5-Algorithmus verschlüsselt und liegt auch in der Datenbank nicht in entschlüsselter Form vor. Der MD5-Algorithmus erlaubt höchste Sicherheit, bisher ist eine Entschlüsselung nicht möglich. Für vergessene Passwörter bietet vanillaXML Webshop eine Passwort-Recovery-Funktion per Email an.

6.1.3. Partnerprogramm

Wenn Sie - wie wir es Ihnen wünschen - mit dem System von **vanillaXML Webshop** zufrieden sein sollten, können Sie uns gerne weiterempfehlen. Und dabei können Sie auch zusätzlich verdienen! Die sandoba.de medien agentur bietet für **vanillaXML Webshop** ein Partnerprogramm mit fester Provision am durch Sie erzielten Umsatz durch Käufe des Systems an. Informieren Sie sich unter www.sandoba.de über das Partnerprogramm und wie Sie Ihre Provisionserlöse durch Werbung steigern können.

6.1.4. Vorlagen-Service

Die sandoba.de medien agentur erstellt mit einem erfahrenen Team von Freelancern auf Wunsch komplette Vorlagen-Sets für Ihre Website. Neben den standardmäßig integrierten Vorlagen können Sie so die Funktionen, die Ihnen vanillaXML Webshop bietet, noch besser einsetzen. Setzen Sie sich mit uns in Kontakt und wir machen Ihnen ein unverbindliches Angebot. Informationen über den Vorlagen-Service erhalten Sie direkt auf unserer Website.

7. {cp:tags}

Die {cp:tags} dienen der einfachen Implementierung dynamischer Seitenbestandteile in die Vorlagen. Mit Hilfe dieser Tags lassen sich so beispielsweise einzelne Seiten mit einer Auflistung der News oder verschiedenen Werbeformen versehen. Auch die Navigation kann in ihrer Art und Position unterschiedlich angeordnet werden. Die Möglichkeiten sind vielfältig. Alle im System nutzbaren {cp:tags} werden hier kurz erläutert.

7.1.0. {cp:content}

Dieser Tag steht nur eingeschränkt innerhalb der Vorlagen zur Verfügung. Er kennzeichnet den Bereich, in welchem später der Inhalt eingesetzt wird. Der Tag kann u.a. innerhalb der Vorlagen für "Links - Allgemein", "Galerie - Allgemein", "Blöcke", "News" und "Artikel" zum Einsatz kommen.

7.1.1. Allgemein

Die {cp:tags} sind eine in vanillaXML Webshop integrierte Metasprache, durch welche sich bestimmte Elemente innerhalb der Vorlagen oder auch in Artikel einfach positionieren lassen. Dazu gehören die Anzeige der neuesten Links, eine Möglichkeit zum Wechsel der Sprache usw. Die Syntax beginnt dabei immer mit "cp:" und wird in geschweifte Klammern eingefasst. Einige {cp:tags} müssen zusätzlich geschlossen werden.

Die Syntax baut auf folgender Schreibweise auf (am Beispiel von "{cp:container}"):

```
{cp:container module="news" start="0" limit="5" [order=""] [dir=""] [category=""]}
```

Der Name des Tags ist hierbei "container". Alle Angaben rechts davon innerhalb des Tags sind Parameter, welche z.B. die Art der Daten oder die Darstellung dieser Daten beeinflussen. Die Attribute dieser Parameter sind generell in Hochkommata eingeschlossen, Leerzeichen sind nur zwischen den Parametern zulässig.

Parameter, welche in eckigen Klammern stehen sind optional und können je nach Bedarf hinzugefügt werden.

7.1.2. Aufbau der Tags

Es wird unterschieden zwischen Tags, welche die weitere Verarbeitung der Vorlagen beeinflussen (bspw. Container, Variablen-Bearbeitung) und Tags, welche lediglich einzelne Inhalte ausgeben. Alle Tags des ersten Typs werden einzeln beschrieben und sind in allen Artikeln/Vorlagen verwendbar, für den zweiten Typ der Tags gibt es einen festgelegten Aufbau. Diese Tags können nur in den Vorlagen des jeweiligen Modules verwendet werden.

Am Beispiel einer Vorlage für eine News (z.B. die Vorlage "news_text.htm" in Ihrem Vorlagen-Set) sehen Sie die Verwendung der Tags. Der Name setzt sich stets zusammen aus "{cp:", dem Namen des Modules sowie dem Namen der Datenbankspalte, in welcher sich der gewünschte Inhalt befindet. Soll beispielsweise der Titel einer News ausgegeben werden, müssen Sie den Tag "{cp:news:title}" in Ihre Vorlage integrieren. Ein Tag diesen Typs kann mehrfach innerhalb der Vorlage verwendet werden.

7.1.3. Sprachelemente in Vorlagen: {cpl:tags}

Um Sprachelemente in den Vorlagen nutzen zu können und dennoch diese dynamisch nach der eingestellten Sprache zu ersetzen, wurden die sogenannten {cpl:tags} eingeführt ("vanillaXML Webshop-Language-Tags"). Im Namen des Tags {cpl:*}) wird * durch den Inhalt der entsprechenden Variable aus den Sprachdateien ersetzt, zusammen mit dem Namen des Moduls.

Werden Sprachelemente aus der Hauptdatei (Ordner "/languages/**") genutzt, so ist als Modul "main" anzugeben. Beinhaltet eine Vorlage beispielsweise den Tag {cpl:main:0469} und es ist als Sprache Deutsch ("de") eingestellt, so wird aus dem Ordner "languages/de" die Datei "index.php" geöffnet. In dieser wird nach der Variable \$lang_main_0469 gesucht. Deren Inhalt (in diesem Falle "Pfad zum CMS") ersetzt in der Vorlage dann die ursprünglich vorhandene Angabe {cpl:main:0469}.

Bei den Modulen verhält es sich ähnlich. Wenn ein Sprachelement aus dem Modul "CP::Webmail" verwendet werden soll, so schreibt man beispielsweise "{cpl:webmail:0083}". Hier würde dann die Variable "\$lang [{"\$module_name}"]["0083"]" aus der Datei "modules/webmail/languages/de/index.php" angefordert werden. Voraussetzung ist ebenfalls, dass "de" als Sprache voreingestellt ist und das Modul im Ordner "modules/webmail" abgelegt ist.

7.1.4. {cp:assign}

{cp:assign} dient der Zuweisung von Variableninhalten innerhalb einer Vorlage. Vorhandene Variableninhalte können damit überschrieben werden. In Kombination mit anderen {cp:tags} stellt {cp:assign} eine Möglichkeit dar, z.B. ein Passwort nur verschlüsselt anzuzeigen, wenn dieses vorhanden ist. {cp:assign} erhält zur Verarbeitung den Namen der zu überschreibenden Variablen und deren neuen Inhalt übermittelt.

```
{cp:assign variable="password" value="*****"}  
// das Passwort aus der Variable "$password" wird mit "*****" überschrieben
```

7.1.5. {cp:break}

Dieser spezielle Tag dient der Arbeitserleichterung bei der Erstellung von Artikeln. Er steht nicht innerhalb der Vorlagen zur Verfügung. In einem Artikel eingesetzt, erzeugt er einen Seitenumbruch, d.h. der Artikel wird auf mehrere Seiten aufgeteilt. Mittels {cp:pages} kann dann in "article_list.htm" eine automatisch generierte Navigation zum Umblättern eingeblendet werden.

7.1.6. {cp:buildtime}

Dieser Tag gibt die Zeit an, welche zum Generieren der aktuellen Seite benötigt wurde. Vor allem in der Test-Phase kann man mit {cp:buildtime} Fortschritte bei der Optimierung der Seiten feststellen. Die einzelnen Seiten werden vom Browser der Reihe nach von oben nach unten generiert. Dementsprechend sollte {cp:buildtime} auch nur am Ende der Seite zur Anzeige der benötigten Zeit genutzt werden.

7.1.7. {cp:configuration}

Ähnlich wie "{cp:variable}" sorgt dieser Tag für die Integration von Systemeinstellungen in die Vorlagen. Kann {cp:variable} lediglich normale String- und Integer-Variablen in die Vorlagen einsetzen, so greift {cp:configuration} direkt auf das Konfigurations-Array zu, in dem alle Einstellungen gespeichert werden. Der Tag baut sich folgendermaßen auf:

```
{cp:configuration:[MODUL]:[EINSTELLUNG]}
```

Anstelle von "[MODUL]" setzen Sie den Namen des gewünschten Moduls ein, bei "[EINSTELLUNG]" entsprechend den Namen der Einstellung. Intern greift vanillaXML Webshop hier auf "\$configuration [{"MODUL}"]["EINSTELLUNG"]" zu.

Welche Namen Sie für die beiden Parameter einsetzen können, erfahren Sie in der Datei "[config.db.php](#)" im vanillaXML Webshop-Hauptordner.

7.1.8. {cp:container module="fast_gallery"}

Neben dem standardmäßig in vanillaXML Webshop integrierten Galerie-Modul wurde das Modul "fast_gallery" zur unkomplizierten Anzeige von Grafiken in einem Verzeichnis entwickelt. "fast_gallery" wird über Container direkt in eine beliebige Vorlage eingebettet. Dies ermöglicht beispielsweise die Anzeige einer Galerie mit themenbezogenen Bildern direkt in einem Artikel, ohne dass alle Bilder einzeln hochgeladen und eingefügt werden müssen. Ebenso nimmt einem die Galerie Arbeiten wie die Größenanpassung und Optimierung der Grafiken usw. ab.

Folgende Angaben sind zulässig / erforderlich:

- *path* - Der Pfad zum anzuzeigenden Verzeichnis
- *start* - Anzeige ab Bild "n" (optional)
- *width* - Breite der Grafiken (optional)
- *height* - Höhe der Grafiken (optional)
- *cols* - Spaltenanzahl der Tabelle
- *rows* - Zeilenanzahl der Tabelle
- *type* - Zugelassene Dateiformate (komma-separiert, optional)
- *thumbnails* - Vorschau-Bilder generieren (0 = nein, 1 = ja)
- *cache* - Ergebnis zwischenspeichern (0 = nein, 1 = ja)
- *popup* - Popup auf Klick (0 = nein, 1 = ja)

Zusammenfassung (mit Beispiel):

```
{cp:container module="fast_gallery" path="__VERZEICHNIS__" [start="0"] [width="200"] [height="100"] cols="3" rows="3" [type="jpg|gif"] [popup="1"] [thumbnails="1"] [cache="1"]}
```

Anwendungsbeispiele:

```
{cp:container module="fast_gallery" path="images/reisen" start="0" width="150" cols="2" rows="4" type="png|jpg|swf" popup="1"}
```

Zeigt insgesamt 8 Bilder (2 Spalten à 4 Zeilen) aus dem Verzeichnis "images/reisen" an, angefangen vom ersten Bild. Gelistet werden nur Bilder in den Formaten "png", "jpg" und "swf". Die Bilder werden auf eine Breite von 150 Pixel skaliert, zwischengespeichert und mit einem Link zum Popup versehen.

Einsatz:

Der Tag `{cp:container module="fast_gallery"}` kann für jegliche Arten von Galerien / Bildübersichten genutzt werden, für die die Funktionen des normalen Galerie-Moduls zu umfangreich sind. Denkbar wäre z.B. eine Übersicht mit Produktscreenshots zu einer Werbepäsentation, deren Vollansicht erst nach einem Klick auf eines der Vorschau-Bilder sichtbar wird.

7.1.9. `{cp:container module="navigation"}`

Die Navigation ist das Herzstück einer jeden Website. Sie kann äußerst vielgestaltig sein und an verschiedenen Stellen der Website in jeweils unterschiedlichen Stilen benötigt werden. Über einen Container lassen sich in vanillaXML Webshop fast beliebige Navigationen mit einer gewünschten Tiefe realisieren. Ausführliche Anleitungen gibt es auch in der Knowledge Base.

`{cp:container module="navigation"}` enthält Angaben zu Kategorie, Cache-Zeit, den verwendeten Vorlagen, dem Stil usw.:

- *block* - Liste der nicht sichtbaren Navigations-Punkte (kommasepariert)
- *cache* - Cache-Zeit der Navigation in Sekunden
- *style* - Stil (nur wenn Standardnavigation genutzt wird)
- *category* - Übergeordnete Kategorie (Hauptmenü ist "0")
- *number* - alternativ zu *category* für die direkte Anzeige eines Unterpunktes
- *order* - Links ordnen (möglich sind "date", "hits", "number" und "title")
- *type* - Verwendete Tabelle ("news", "article", "navigation" oder leer)
- *template* - Pfad zur einer Vorlage
- *template_selected* - Pfad zur Vorlage, wenn Punkt ausgewählt *1
- *template_subnodes* - Pfad zur Vorlage, wenn Unterpunkte vorhanden

*1 Für "template_selected" muss zusätzlich das Feld angegeben werden, welches die URL enthält. Bei der Navigationstabelle muss dies also "selected_field="nav_url"" sein.

7.1.10. `{cp:container module="search"}`

Mit `{cp:container module="search"}` lässt sich ein Container für die Volltextsuche an einer beliebigen Stelle der Website platzieren. Über einen optionalen Parameter "template" kann angegeben werden, welche Vorlage für den Container verwendet werden soll. Ist dieser Parameter nicht vorhanden, wird automatisch auf die Vorlage "modules/search/templates/search_form.htm" zurückgegriffen.

Dies ermöglicht die Darstellung verschiedener Suchcontainer je nach Bereich der Website. Das Suchformular kann zusätzlich über ein Feld "category" den Suchbereich beeinflussen. Möglich sind hier standardmäßig "news" und "articles".

Anwendungsbeispiele:

```
{cp:container module="search" template="suche_artikel.htm"}
```

Bindet den Suchcontainer mit der Vorlage "suche_artikel.htm" ein. Der Bereich der Suche ist in dieser Vorlage automatisch ausgewählt über ein verstecktes Formular-Feld.

```
{cp:container module="search"}
```

Bindet den Suchcontainer mit der Standard-Vorlage (siehe oben) auf der Website ein.

7.1.11. `{cp:container module="statistic"}`

In einigen Fällen kann es angebracht sein, Besucher auch öffentlich über die Statistik der Website zu informieren. `{cp:container module="statistic"}` kann hierzu in das Layout einer beliebigen Vorlage integriert werden. Der Tag liefert Angaben über die aktuellen Besucher sowie die Hits der Seite heute, gestern und insgesamt.

Optional kann man sich über die Angabe "visitors" (z.B. `{cp:container module="statistic" visitors="today"}`) auch nur bestimmte Angaben über die Statistik zurückliefern lassen. Mögliche Attribute sind hierbei "today" (Seitenaufrufe heute), "yesterday" (Seitenaufrufe gestern), "total" (Seitenaufrufe insgesamt) und "online" (Anzahl der aktuellen Besucher).

7.1.12. {cp:container module="user"}

Dieser Tag integriert ein Formular für das Einloggen im handlichen Format in die Seite. Ist das Mitglied bereits eingeloggt wird eine Liste mit Links zur Mitgliederliste, zur Bearbeitung des Profiles sowie zur Abmeldung eingeblendet. Genutzt werden kann der Tag in den allgemeinen Seitenvorlagen.

7.1.13. {cp:container module="webservices"}

Viele Websites und Blogs, welche auf Content-Management-Systemen basieren, bieten mittlerweile Inhalte für andere Websites an. Auch bei vanillaXML Webshop ist dies als Funktion integriert.

Mit {cp:container module="webservices"} können diese externen Inhalte in den Formaten XML, RSS und RDF in vanillaXML Webshop importiert und angezeigt werden. Die Gestaltung ist mittels Vorlagen realisierbar und alle Inhalte können zwischengespeichert werden, so dass nicht bei jedem Aufruf der Seite ein erneutes Herunterladen der externen Datei notwendig wird. Der Tag {cp:container module="webservices"} besteht im Wesentlichen aus einer Reihe von Angaben, welche Typ und Ort der anzuzeigenden Datei beschreiben:

- *type* - Typ der Datei ("RSS", "RDF", "XML")
- *file* - Externe Datei mit den Inhalten
- *template* - Vorlage für die Gestaltung
- *limit* - Begrenzung der Anzeige auf "n"-Datensätze

Wenn Sie eine Begrenzung nach einer Kategorie vornehmen möchten, erhalten Sie die entsprechende Kategorienummer unter "Artikel > Kategorien".

Anwendungsbeispiele:

```
{cp:container module="webservices" type="rdf" file="www.heise.de/newsticker/heise.rdf "  
template="templates/heise.htm" limit="10"}
```

Gibt die ersten Zehn Artikel des RDF-Feeds von www.heise.de aus. Die Gestaltung der Artikel wird über die Vorlage "templates/heise.htm" realisiert..

7.1.14. {cp:date}

Mit {cp:date} können Sie ein formatiertes Datum auf Ihrer Website anzeigen lassen. Im Gegensatz zu {cp:variable:date} und {cp:variable:dateshort} wird hier kein festgelegtes Format genutzt. Vielmehr wird über den Parameter "format" das gewünschte Format des Datums angegeben. Genauere Informationen zu den möglichen Zeichenketten erhalten Sie unter "<http://www.php.net/date>". Als zweiter Parameter kann mit "time" ein Datum im Timestamp-Format angegeben werden. Ist dieser Parameter nicht gesetzt, wird das aktuelle Datum verwendet.

```
{cp:date format="d.m.Y"}  
// liefert Datumsangabe, z.B. 23.05.2003
```

```
{cp:date format="H:i" time="1075754007"}  
// liefert Zeitangabe, z.B. 18:32 (hier für "1075754007")
```

7.1.15. {cp:header}

{cp:header} kommt in verschiedenen Vorlagen (u.a. "print.htm") zum Einsatz und wird bei der Ausführung in der Regel durch den Titel der Seite bzw. des jeweiligen Eintrages ersetzt.

7.1.16. {cp:help}

{cp:help} erzeugt einen Button, welcher auf der Seite platziert werden kann und durch einen Klick zur Hilfe bzw. auf die Sitemap springt. Standardmäßig wird hierfür die Grafik "images/cms_help.gif" in dem ausgewählten Vorlagen-Set genutzt.

7.1.17. {cp:homepage}

{cp:homepage} erzeugt einen Button, welcher auf der Seite platziert werden kann und durch einen Klick zurück auf die Startseite springt. Standardmäßig wird hierfür die Grafik "images/cms_home.gif" in dem ausgewählten Vorlagen-Set genutzt.

7.1.18. {cp:ifnot}

Mittels des Konstrukts {cp:ifnot} lässt sich überprüfen, ob eine Variable NICHT vorhanden ist. Dies können Variablen aus dem Modul oder auch aus den Konfigurationseinstellungen sein. Variablen sind in PHP durch ein "\$" am Anfang des Variablennamens gekennzeichnet (z.B. "\$symbol"). Ein Beispiel soll den Sinn von {cp:ifnot} verdeutlichen:

```
{cp:if:username}
<table border="0"><tr bgcolor="#A32333"><td><font color="#FFFFFF">Sie Sind nicht eingeloggt, bitte <a
href="index.php?file=user">hier</a> einloggen.</font></td></tr></table>
{/cp:if:username}
```

In diesem Beispiel wird geprüft, ob die Variable "\$username" NICHT vorhanden ist. Ist dies der Fall, dann wird eine Tabelle eingeblendet mit innenliegendem Text und einem Link zur Login-Seite.

7.1.19. {cp:if}

Mittels des Konstrukts {cp:if} lässt sich das Vorhandensein einer Variablen direkt in einer Vorlage überprüfen. Dies können Variablen aus dem Modul oder auch aus den Konfigurationseinstellungen sein. Variablen sind in PHP durch ein "\$" am Anfang des Variablennamens gekennzeichnet (z.B. "\$symbol"). Ein Beispiel soll den Sinn von {cp:if} verdeutlichen:

```
{cp:if:email}
<table border="0"><tr bgcolor="#A32333"><td><font color="#FFFFFF">Unser Support: {cp:variable:email}
</font></td></tr></table>
{/cp:if:email}
```

In diesem Beispiel wird geprüft, ob die Variable "\$email" vorhanden und mit Inhalt gefüllt ist. Ist dies der Fall, dann wird eine Tabelle eingeblendet mit innenliegendem Text und der Email-Adresse. Diese wird dann über {cp:variable} eingeblendet.

7.1.20. {cp:ignore}

Wenn Sie in Artikeln oder Vorlagen {cp:tags} einsetzen wollen, welche nicht verarbeitet werden sollen, so können Sie dies dem Parser mitteilen, indem Sie die betroffenen Bereiche mit {cp:ignore} ... {/cp:ignore} umgeben. Dies ist z.B. bei Anleitungen zu den {cp:tags} nützlich, wie diese unter www.sandoba.de in der Knowledge Base zu finden sind.

7.1.21. {cp:image}

{cp:image} kommt momentan nur in den Vorlagen zur Anzeige einer News zum Einsatz. Hier wird der Tag für die Einblendung einer Grafik zur ausgewählten News genutzt.

7.1.22. {cp:include}

vanillaXML Webshop stellt Ihnen eine Vielzahl an Funktionen für Ihre Website bereit. Benötigen Sie jedoch eine spezielle Funktion und wollen nicht bis zur nächsten Version warten, so können Sie auch andere, bereits vorhandene Skripte einbinden. Hierfür wurde mit {cp:include} ein eigener Tag angelegt, welcher einfach in den Vorlagen eingesetzt wird.

Zur Einbindung des Skripts "weblog.php" ist Folgendes in die Vorlage einzubauen:

{cp:include}weblog.php{/cp:include}

Wenn das Script in einem anderen Ordner liegt, so geben Sie zusätzlich den Pfad zu dieser Datei ausgehend vom vanillaXML Webshop-Stammverzeichnis an. Für PHP-Dateien sollten Sie zusätzlich den Parameter "type" mit dem Wert "php" anhängen ("... type="php" ..."). Dann kann der Inhalt immer korrekt verarbeitet werden.

7.1.23. {cp:ismember}

Gezielt können in den Vorlagen einzelne Inhalte nur für registrierte und eingeloggte Mitglieder sichtbar gemacht werden. Dazu zählen alle Mitglieder-Gruppen inklusive der Redakteure usw. Der Tag muss nach den zu schützenden Inhalten wieder geschlossen werden.

{cp:ismember}Dieser Text wird nur Mitgliedern gezeigt.{/cp:ismember}

7.1.24. {cp:language}

Wenn Ihre Website mehrsprachig aufgebaut wird, so müssen Sie Ihren Besuchern die Möglichkeit geben, die Sprache zu wechseln. Dies ist mit dem Tag {cp:language} möglich, durch welchen die Flaggen für die verfügbaren Sprachversionen eingeblendet werden.

7.1.25. {cp:lastlogin}

Mit diesem Tag können Sie in einer beliebigen Vorlage der Website das Datum des letzten Logins eines Benutzers einblenden lassen. Standardmäßig wird dies in der Vorlage des Administrations-Bereiches vorgenommen.

7.1.26. {cp:level}

{cp:level} zeigt die Position der aufgerufenen Seite innerhalb der Struktur der Website an. Beispielsweise kann dies bei der Galerie oder im Link-Modul genutzt werden und sieht dann folgendermaßen aus: "Galerie > Kategorienname > Bildname". Alle Kategorien sowie die Hauptseite sind mit Links versehen, so dass ein flexibles Wechseln möglich ist.

Für die Artikelvorlagen kann der zusätzlich um die Variable "base" erweitert werden. Ist "base" nicht gewählt oder "base" = 0, wird der Link zur Artikel-Hauptseite mit angezeigt. Bei "base" = 1 jedoch ist die oberste Ebene die Hauptkategorie des Artikels.

7.1.27. {cp:linksnavigation}

Der Tag {cp:linksnavigation} zeigt im Link-Modul eine horizontale Navigationsleiste an, mit folgenden Auswahlmöglichkeiten:

- Neue Links
- Beliebteste Links
- Favoriten
- Link anmelden

7.1.28. {cp:link}

Der Tag {cp:link} ist äusserst vielfältig einsetzbar. Bei den News-Übersichten kann dieser beispielsweise zum Aufruf der Detail-Übersicht genutzt werden...

7.1.29. {cp:path}

{cp:path} fügt immer den relativen Pfad zum aktuell genutzten Vorlagen-Set ein. Hiermit können Sie beispielsweise auf Grafiken zugreifen, welche in den Vorlagen-Ordern liegen, obwohl diese generell nicht die

beste Lösung für die Lagerung von Grafiken ist.

7.1.30. {cp:pdf}

Für Artikel, News und Bilder in der Galerie können in Echtzeit Dateien im PDF-Format von Adobe® erzeugt werden. Diese zeichnen sich durch überragende Fähigkeiten beim Druck bzw. dem Austausch von Daten mit anderen Systemen aus. Eine Vorlage für PDF-Dateien kann in vanillaXML Webshop momentan noch nicht angelegt werden, manuelle Anpassungen an den generierten Dateien können jedoch von erfahrenen Nutzer in der Datei "print.pdf.php" im Hauptverzeichnis vorgenommen werden.

7.1.31. {cp:poster}

{cp:poster} zeigt den Verfasser eines Eintrages bei den News sowie in der Gallery an. Der Name des jeweiligen Mitglieds ist zugleich ein Hyperlink zum öffentlich einsehbaren Profil. Hier kann sich das Mitglied in aller Ausführlichkeit präsentieren.

7.1.32. {cp:print}

Der Print-Tag blendet bei News oder Artikeln ein Icon ein, welches durch einen Klick ein Popup öffnet. Dieses Popup enthält eine speziell für den Druck aufbereitete Version des ausgewählten Eintrages. Einzelne enthaltene HTML-Tags werden dabei bewusst entfernt, um eine optimale Kompatibilität sicherzustellen. Die Druck-Vorlage kann in der Datei "print.htm" angepasst werden.

7.1.33. {cp:random}

Mit {cp:random} integrieren Sie die im Modul "Zufall" angegebenen Inhalte in die Website. Hierzu wird mittels des Parameters "type" der Typ bzw. die Kategorie des Inhaltes angegeben. Aus allen entsprechenden Inhalten wird dann jeweils ein Eintrag ausgewählt und auf der Website angezeigt.

```
{cp:random type="1"}  
// zeigt Inhalte des Typs mit der ID "1" an
```

7.1.34. {cp:rating}

Blendet die aktuelle Bewertung des angewählten Elementes ein. Genutzt werden hierfür die Grafiken "rating_a.gif" und "rating_b.gif" im Ordner "images/" eines Vorlagen-Sets. Die Bewertung setzt sich aus insgesamt 10 Stufen zusammen, da die maximale Bewertung von 5.0 zur besseren Verdeutlichung verdoppelt dargestellt wird. {cp:rating} kann u.a. bei Artikeln, im Gästebuch, bzw. in der Galerie eingesetzt werden.

7.1.35. {cp:rewrite}

Besonders wichtig für die Suchmaschinen-Optimierung der Seite sind URL's, die Schlüsselwörter enthalten, welche auch im Text vorkommen. Mit Hilfe des Tags {cp:rewrite} können derartige URL's leicht erzeugt werden. Über den Parameter "text" wird dem Tag ein kurzer Text bis ca. 50 Zeichen übergeben. Zurückgeliefert wird der Text in einem Format, wie dieses in einer URL verwendet werden kann.

```
{cp:rewrite text="Mit vanillaXML Webshop in die Suchmaschinen"}  
// gibt "mit-vanillaxml-in-die-suchmaschinen" zurück
```

7.1.36. {cp:rights}

Der Tag "{cp:rights}" überprüft, ob ein Benutzer über ein bestimmtes Zugriffsrecht verfügt. Wenn dies der Fall sein sollte, dann wird der durch den Tag umschlossene Raum angezeigt. Hierzu muss zusätzlich der Name des Zugriffsrechtes bzw. ein Array mit Namen übergeben werden (letzteres bewirkt eine "UND"-Verknüpfung). "{cp:rights}" kann mit der Angabe "type" auch verschachtelt eingesetzt werden (zwei Ebenen). Eine "OR"-Verknüpfung ist durch das Hinzufügen von "connection="or"" möglich.

Anwendung:

```
{cp:rights name="options"} ... {/cp:rights}
```

Gibt "..." aus, wenn das Zugriffsrecht "options" vorhanden ist.

```
{cp:rights name[0]="article_edit" name[1]="article_delete"}...{/cp:rights}
```

Bei diesem Beispiel wird "..." ausgegeben, wenn sowohl das Zugriffsrecht "article_edit" als auch "article_delete" dem Benutzer zugeteilt sind. Die Namen der möglichen Zugriffsrechte entsprechen dabei dem "Internen Namen", welcher bei einem Zugriffsrecht in der Rechteverwaltung eingegeben wurde.

7.1.37. {cp:systemlogo}

Der Tag {cp:systemversion} erlaubt die Anzeige des Logos von vanillaXML Webshop. Dieses ist per Link mit der Website www.sandoba.de verbunden.

7.1.38. {cp:top}

{cp:top} erzeugt einfach einen Button, welcher am Ende einer Seite platziert werden kann und durch einen Klick zurück zum Anfang der Seite springt. Standardmäßig wird hierfür die Grafik "images/cms_top.gif" in dem ausgewählten Vorlagen-Set genutzt.

7.1.39. {cp:variable}

Mit "{cp:variable}" können Sie Umgebungsvariablen aus eingebundenen PHP-Dateien (beispielsweise aus der Konfigurations-Datei) in den Vorlagen verarbeiten. Die Anzeige der Email-Adresse des Webmasters ist z.B. mit "{cp:variable:email}" möglich. In diesem Fall wird auf die Variable "\$email" zugegriffen. Sofern eine Variable mit diesem Namen existiert und einen Inhalt besitzt, wird der Tag mit dem entsprechenden Inhalt ersetzt. Beispiele für die Anwendung:

- {cp:variable:email} - Email des Webmasters
- {cp:variable:fullurl} - URL der Website
- {cp:variable:language} - genutzte Sprache
- {cp:variable:REQUEST_URI} - "\$REQUEST-URI"
- {cp:variable:systemname} - Name des CMS
- {cp:variable:systemversion} - Version des CMS
- {cp:variable:title} - Titel der Website
- {cp:variable:version} - PHP-Version
- {cp:variable:log_username} - Name des eingeloggten Nutzers
- {cp:variable:date} - Datum in Standardform

7.1.40. {cp:vote}

Dieser Tag bindet bei einem Artikel ein Formular zur Bewertung ein. Er muss durch den Tag {cp:votaset} komplettiert werden. Dazwischen sind zusätzliche Texte, Grafiken usw. einblendbar.

8. Entwickler-API

Mit der Entwickler-API gibt es einen Überblick über die öffentlichen Funktionen, welche bei der Entwicklung eigener Module für "**Contentpapt**" genutzt werden können. Dies sind Funktionen zur Interaktion mit anderen Modulen sowie auch mit externen Ressourcen. Beachten Sie zusätzlich die "Entwickler-Informationen" im Kapitel "Sonstiges".

8.1.0. Beispiel-Modul

Im Ordner "modules/example" ist ein Beispiel für ein Modul abgelegt. Dieses beschreibt den grundsätzlichen Aufbau des Modules mit einzelnen Funktionen. Es wird in der "index.php" gezeigt, welche Dateien allgemein eingebunden werden müssen, um einen ordnungsgemäßen Lauf des Modules zu gewährleisten.

Bilder, die durch das Modul genutzt werden, werden in einem Unterordner "images" abgelegt. Vorlagen werden im Unterordner "templates" platziert, Sprachelemente im Unterordner "languages/SPRACHE".

8.1.1. cp_absolute_path()

Mit der Funktion "cp_absolute_path" kann der absolute Pfad des jeweiligen Systems festgestellt werden. Der absolute Pfad, setzt sich dabei aus dem Basispfad und (eventuell) einem relativen Pfad zusammen. Dieser wird u.a. benötigt, wenn auf installierte Software oder bestimmte Systemdateien zugegriffen werden muss.

8.1.2. cp_array_average()

Mit "cp_array_average" lässt sich der Durchschnitt von Werten in einem numerischen Array berechnen. Dies kann u.a. bei der Erstellung von Graphen zum Einsatz kommen. Der Funktion werden neben dem Array ein Start- und ein Endwert übergeben. Die beiden letzten Parameter sind optional.

Wenn der Startwert nicht angegeben ist, wird der erste Wert des Arrays genutzt. Bei einem fehlenden Endwert wird hingegen der letzte Wert des Arrays genutzt. Der Durchschnittswert kann erweitert werden, indem man die Differenz zwischen Start- und Endwert erhöht.

```
$var = cp_array_average($array,"5","15");  
// liefert den Durchschnittswert der Werte 5-15 des Arrays
```

8.1.3. cp_array_structure()

In vanillaXML Webshop-Modulen wird häufig mit komplexen Arrays gearbeitet, da diese eine sehr effiziente Abwicklung von Prozessen ermöglichen. Allerdings kann es passieren, dass man als Entwickler den Überblick über den Umfang eines Arrays verliert. Mit der Funktion "cp_array_structure" gehört dieses Problem der Vergangenheit an.

"cp_array_structure" gibt das übermittelte Array auf dem Bildschirm in einer beliebigen Tiefe aus. Die Funktion wird hierzu rekursiv durchlaufen. Beinhaltet das Array weitere Arrays, so werden diese ebenfalls durchlaufen.

```
cp_array_structure($test_array);  
// das Array "$test_array" wird angezeigt
```

8.1.4. cp_array_sum()

Mit "cp_array_sum" lässt sich die Summe von Werten in einem numerischen Array berechnen. Dies kann u.a. bei der Erstellung von Statistiken zum Einsatz kommen. Der Funktion werden neben dem Array ein Start- und ein Endwert übergeben. Die beiden letzten Parameter sind optional.

Wenn der Startwert nicht angegeben ist, wird der erste Wert des Arrays genutzt. Bei einem fehlenden Endwert wird hingegen der letzte Wert des Arrays genutzt. Die Summe kann erweitert werden, indem man die Differenz zwischen Start- und Endwert erhöht.

```
$var = cp_array_sum($array,"5","15");  
// das liefert die Summe der Werte 5-15 des Arrays
```

8.1.5. cp_currency_info()

Die Funktion "cp_currency_info" gibt Auskunft über den Namen einer (Landes-)Währung, nachdem ihr ein dreistelliges Währungskürzel übermittelt wurde. Die Überprüfung ist dabei größtenteils konform zur ISO-Regelung "4217", erweitert um einige Industriestandards.

```
$var = cp_currency_info("EUR");  
// sucht den Namen der Währung zum Kürzel "EUR"
```

8.1.6. cp_delete_cache()

Natürlich müssen im Cache abgelegte Inhalte auch vor Ablauf der festgelegten Cache-Zeit wieder gelöscht werden können, z.B. wenn eine Aktualisierung der Inhalte vorgenommen wurde. Hierzu muss der Funktion "cp_delete_cache" der Anfang des Namens der Cache-Datei übermittelt werden sowie der Ordner. Alle Dateien mit diesem Anfang werden dann gelöscht.

```
cp_delete_cache("cache/", "test.tmp");
```

8.1.7. cp_get_cache()

Für umfangreichere Inhalte, welche nicht regelmäßig aktualisiert werden, empfiehlt es sich, diese in einem lokalen Cache zwischenspeichern. Die Funktion "cp_get_cache" ermöglicht dabei den Abgleich der gewünschten Daten mit Inhalten im Cache. Der Funktion werden dazu als Parameter das Verzeichnis des Caches, der Dateiname und die maximale Cache-Zeit (in Sekunden) übergeben. Bei Erfolg wird TRUE zurückgegeben, andernfalls FALSE. Als Standard-Cache-Zeit werden 3600 Sekunden (d.h. 1 Stunde verwendet).

```
$var = cp_get_cache("cache/", "test.tmp", 3600);
```

Prüft, ob die Datei "test.tmp" im Ordner "cache" vorhanden ist (diese darf nicht älter als 3600 Sekunden sein)

8.1.8. cp_lang_info()

Die Funktion "cp_lang_info" gibt Auskunft über den Namen einer Sprache, nachdem ihr ein zweistelliges Namens Kürzel übermittelt wurde. Die Überprüfung ist dabei konform zur ISO-Regelung "639-1".

```
$var = cp_lang_info("eo");  
// sucht den Namen der Sprache zum Kürzel "eo" (Ergebnis ist "Esperanto")
```

8.1.9. cp_mime_info()

Die Funktion "cp_mime_info" gibt Auskunft über den MIME-Typ eines Dateiformates, nachdem ihr die entsprechende Dateiendung übergeben wurde. Ist diese Dateiendung nicht vorhanden, wird ein FALSE zurückgegeben. Ohne Angabe der Dateiendung wird hingegen ein Array zurückgegeben, welches alle Einträge enthält (Dateiendungen jeweils als Schlüssel und MIME-Typ als Wert)

```
$var = cp_mime_info("mp3");  
// gibt den MIME-Typ für "mp3" zurück (Ergebnis ist "audio/mpeg")
```

8.1.10. cp_module_exists()

Oftmals soll zwischen verschiedenen vanillaXML Webshop-Modulen eine Verbindung hergestellt werden. Da jedoch nicht immer gewährleistet werden kann, dass diese Module auch vorhanden sind, muss vorab ihre Verfügbarkeit überprüft werden. Hierfür kann die Funktion "cp_module_exists" genutzt werden. Der Funktion wird der Modul-Name (d.h. der Namen des Ordners, in dem das Modul liegt) übergeben. Als Ergebnis werden TRUE oder FALSE zurückgeliefert.

```
$var = cp_module_exists("forum");  
// prüft, ob das Foren-Modul vorhanden ist
```

8.1.11. cp_random_string()

"cp_random_string" ist eine Funktion zur Erzeugung von zufallsgenerierten Werten. Diese bestehen aus einer Kombination von Buchstaben und Zahlen und können beispielsweise bei der Generierung von Passwörtern zum Einsatz kommen. Über den Parameter "length" kann die gewünschte Länge vorgegeben werden.

```
$var = cp_random_string("10");  
// liefert einen 10-stelligen Wert zurück (z.B. "A94ZUAIQ0P")
```

8.1.12. cp_read_cache()

Nach der Abfrage mittels "cp_get_cache", ob eine Seite im Cache vorhanden ist, kann mit "cp_read_cache" bei Erfolg diese Seite aus dem Cache geladen und angezeigt werden.

```
$var = cp_read_cache("cache/", "$full_name");  
// liest den Cache von "$full_name" aus dem Ordner "files"
```

8.1.13. cp_set_cache()

Für umfangreichere Inhalte, welche nicht regelmäßig aktualisiert werden, empfiehlt es sich, diese in einem lokalen Cache zwischenspeichern. "cp_set_cache" dient der Speicherung von Inhalten im Cache. Der Funktion wird dabei neben dem Inhalt noch ein Ablageort mitgeteilt. Dieser setzt sich aus dem Verzeichnis (standardmäßig ist dies der Ordner "cache") und dem Dateinamen zusammen. Letzterer sollte optimal aus einer md5-Verschlüsselung des originalen Dateinamens samt Endung ".tmp" bestehen.

```
$var = cp_set_cache($data, "cache/", "$output");  
// speichert "$data" unter dem Namen "$output" im Ordner "cache"
```

8.1.14. cp_validate_color()

Eine ganz besondere Funktion ist "cp_validate_color". Mit dieser Funktion lässt sich die Gültigkeit eines Farbwertes im HTML-Schema ermitteln. Ein Farbwert muss dabei mit einer Raute "#" beginnen und zusammen mit dieser Raute insgesamt 7 Zeichen lang sein. Wenn der Farbwert gültig ist, so wird TRUE zurückgegeben, andernfalls ein FALSE.

```
$var = cp_validate_color("#604FFFF");  
// testet, ob der Farbwert gültig ist (Ergebnis: FALSE, da zu lang)
```

8.1.15. cp_validate_email()

Überprüfen Sie Email-Adressen auf ihre Gültigkeit mit der Funktion "cp_validate_email". Die gewünschte Email-Adresse wird der Funktion als Parameter übergeben und dort anhand eines regulären Ausdrucks überprüft. Es wird nur die Struktur überprüft, nicht ob diese Email-Adresse auch genutzt wird. Ist die Email-Adresse korrekt, so wird ein TRUE zurückgegeben, andernfalls ein FALSE.

```
$var = cp_validate_email("contact@sandoba.de");  
// testet auf die Email-Adresse "contact@sandoba.de"
```

8.1.16. cp_validate_number()

Mit "cp_validate_number" überprüfen Sie, ob Inhalte andere Zeichen als die Zahlen 0-9 enthalten. Ist dies der Fall, so wird FALSE zurückgegeben, andernfalls TRUE.

```
$var = cp_validate_number("19. August 2003");  
// überprüft den übergebenen Inhalt (Ergebnis: "FALSE")
```

8.1.17. cp_validate_string()

Wenn Sie ein Mitgliedersystem oder ähnliches nutzen und hierfür per Formular übermittelte Inhalte überprüfen müssen, können Sie die Funktion "cp_validate_string" benutzen. Diese erlaubt nur alphanumerische Zeichen. Der Funktion kann außer dem String noch eine maximale bzw. minimale Länge übergeben werden.

```
$var = cp_validate_string($string, "5", "20");  
// "$string" muss min. 5 und darf max. 20 Zeichen lang sein
```

8.1.18. cp_word_shorten()

Gerade in Foren, Gästebüchern oder anderen Modulen, welche die Interaktion mit dem Besucher einbeziehen

kann es oftmals zu Problemen kommen, wenn beispielsweise das Layout der Seite durch zu lange Wörter aufgesprengt wird. Dies lässt sich mit der Funktion "cp_word_shorten" hervorragend verhindern.

Übermitteln Sie dieser Funktion einfach ein Wort bzw. einen Text und geben Sie als zweiten Parameter die maximale Länge eines Wortes an. Standardmäßig sind dies 50 Zeichen. Längere Wörter werden auf (n-3) Zeichen verkürzt. Diesen wird anschließend "..." angehängt.

```
$var = cp_word_shorten("dies_ist_nur_ein_simpler_test","20");  
// verkürzt den Text auf "dies_ist_nur_ein_..."
```

8.2. Allgemeines zur Modulentwicklung

Diese Informationen beschreiben die Entwicklung neuer Module für **Contentpapst** auf Basis des neuer objektorientierten Ansatzes. **Contentpapst** verwendet hier Klassen und deren Vorteile wie Ableitung, Vererbung und Wiederverwendbarkeit. Sie sollten sich also gut mit Klassen auskennen, um damit arbeiten zu können.

8.2.1. Datei "class.admin.php": Der Administrationsbereich

Der Administrationsbereich eines Modules wird über die Datei "class.admin.php" erstellt. Auch dies ist eine Klasse (genauere Informationen auch unter "Aufbau des Konstruktors").

```
<?php  
class van_gui_example extends van_example {  
  
    var $module_configuration;  
    var $module_language;  
  
    /*  
    class van_gui_example::van_gui_example  
    Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.  
    */  
  
    function van_gui_example() { }  
  
    /*  
    class van_gui_example::install  
    SQL- und Dateianweisungen zur Installation des Modules.  
    */  
  
    function install() { }  
  
    /*  
    class van_gui_example::uninstall  
    SQL- und Dateianweisungen zur Deinstallation des Modules.  
    */  
  
    function uninstall() { }  
  
}  
?>
```

8.2.2. Datei "class.index.php": Der öffentliche Bereich

Der öffentliche eines Modules wird über die Datei "class.index.php" erstellt. Auch dies ist eine Klasse (genauere Informationen auch unter "Aufbau des Konstruktors").

```
<?php  
class van_gui_example extends van_example {  
  
    var $module_configuration;  
    var $module_language;  
  
    /*  
    class van_gui_example::van_gui_example  
    Konstruktor, dient der Initialisierung der Klasse und einiger Variablen.  
    */  
  
    function van_gui_example() {
```

```

GLOBAL $file, $lang, $mode;

$this->module_language = module_language($file) + $lang;

$this->module_configuration["functions"] = array(
    "main"=>array("do_main", array("show"))
);

}

/*
class van_gui_example::metatags
Behandelt die Einblendung der Metatags.
*/

function metatags() {}

/*
class van_gui_example::search
Stellt die Verbindung zur Volltextsuche her.
*/

function search() {}

#####

/*
class van_gui_example::do_main
*/

function do_main($show) {

    // Einzelne Methoden...

}

}
?>

```

8.2.3. Datei "class.webservices.php": Zugriff auf Inhalte über Webservices

Über die Datei "class.webservices.php" können Sie das Modul um eine XML-Ausgabe erweitern. D.h. Inhalte können als XML exportiert und andere Systeme importiert bzw. beliebig dargestellt werden. Hier sehen Sie die Basis-Datei, die dann je nach Modul erweitert werden muss:

```

<?php
class van_webservices_example {

    /*
class van_webservices_example::process
Methode, dient der Verarbeitung der ankommenden Informationen.
*/

function process($command, $form) {

    include("config.php");

    include_once("functions/cp.string.php");
    include_once("functions/cp.url.php");
    include_once("functions/cp.xml.php");

    GLOBAL $xml, $file, $sql;

    $content = "";

    $xml->set_content("example/error", "Sorry, there are no exact matches for your request.");

    $content = $xml->show("xml");

```

```

        RETURN $content;
    }
}
?>

```

8.2.4. Datei "config.php": Die Konfigurationsdatei

In der zentralen Konfigurationsdatei für jedes Modul werden wichtige Einstellungen und Informationen festgehalten. Die Datei muss als "config.php" im Hauptordner des Modules abgelegt werden.

```

<?php
$module_config["example"]["name"] = "example";
// Interner Name des Modules (wie der Ordnername)

$module_config["example"]["version"] = "1.0";
// Versionsnummer des Modules

$module_config["example"]["date"] = "01.05.2004";
// Datum der letzten Arbeiten an diesem Modul

$module_config["example"]["author"] = "sandoba.de medien agentur";
// Name des Modul-Entwicklers

$module_config["example"]["email"] = "contact@sandoba.de";
// Email-Adresse des Modul-Entwicklers

$module_config["example"]["website"] = "http://www.sandoba.de";
// Website des Modul-Entwicklers

$module_config["example"]["image"] = "";
// URL zu einem Bild des Modules (für die Übersicht im Administrations-Bereich)

$module_config["example"]["status"] = "1";
// Status des Modules (1 für Aktiviert, 0 für Deaktiviert, -1 für Deaktiviert in Admin- und Benutzer-Bereich)

$module_config["example"]["install"] = "0";
// Bei "install" = 1 muss das Modul erst noch per Admin-Menü installiert werden

$module_config["example"]["show_list"] = "0";
// Im Admin-Menü als Modul sichtbar (1 für Aktiviert, 0 für Deaktiviert)

?>

```

8.3. Benutzer

Die Datei "cp.user.php" stellt Funktionen rund um die Benutzer/Mitglieder des Systemes zur Verfügung. Dies schließt u.a. Funktionen zur Gewinnung von IP-Adressen und zur Validierung der Zugriffsrechte mit ein. Um diese Funktionen nutzen zu können, muss die Datei "functions/cp.user.php" integriert werden.

8.3.1. cp_get_userip()

Die Funktion "cp_get_userip" ermittelt die IP-Nummer des aktuellen Besuchers und kann u.a. bei der Speicherung von Beiträgen oder bei Abstimmungen (Umfrage-Modul) genutzt werden.

8.3.2. cp_get_username()

"cp_get_username" liefert einen Benutzernamen zurück. Dazu wird der Funktion die Nummer des jeweiliger Benutzers übergeben. Ist kein Benutzer mit dieser Nummer vorhanden, dann wird FALSE zurückgegeben.

8.3.3. cp_user_info()

Mit der Funktion "cp_user_info" lassen sich gezielt anhand der Nummer eines Benutzers weitere Informationen über diesen Benutzer ermitteln. Der Funktion wird zuerst die Nummer übergeben, als zweiter Parameter folgt

der Name des gewünschten Feldes (z.B. "username", "street", "email"). Die Funktion liefert den Inhalt dieses Feldes dann zurück. Ist kein Benutzer mit dieser Nummer vorhanden, dann wird FALSE zurückgegeben.

8.3.4. cp_user_lastlogin()

Mit dem integrierten Logging-System ist es jetzt möglich, gezielt bestimmte Aktionen eines Benutzers aus der Vergangenheit herauszugreifen und diese darzustellen. Die Funktion "cp_user_lastlogin" liefert für den aktuellen Benutzer das Datum des letzten Logins zurück. Standardmäßig ist der Rückgabewert im Format "d.m.Y", über einen optionalen Parameter kann dies jedoch geändert werden. So liefert "cp_user_lastlogin ("d.m.y - H:i")" beispielsweise Datum und Uhrzeit des letzten Logins.

8.3.5. cp_user_rights()

Diese Funktion ist besonders wichtig für den Zugriff von Benutzern auf geschützte Bereiche. Der Funktion wird der interne Name eines Rechtes aus der Rechteverwaltung übergeben. Anschließend wird überprüft, ob die Gruppe des aktuellen Benutzers auf diesen Bereich Zugriff hat. Ist dies der Fall, wird TRUE zurückgegeben, andernfalls FALSE.

8.4. CSS

Mit den CSS-Funktionen können Sie vorhandene CSS-Dateien auslesen, bearbeiten und wieder speichern. Auch das Anlegen neuer CSS-Dateien ist möglich. Um die Funktionen in Module einzubinden, muss die Datei "functions/cp.css.php" per "include" oder "include_once" eingebunden werden.

8.4.1. cp_css_input()

Eine CSS-Datei kann mit der Funktion "cp_css_input" eingelesen und in ein Array umgewandelt werden. Dazu muss der Funktion lediglich der Pfad zur Datei übergeben werden.

```
$var = cp_css_input("stylesheet.css");  
// Datei "stylesheet.css" einlesen
```

8.4.2. cp_css_output()

Die Funktion "cp_css_output" verwandelt ein zweidimensionales Array in eine fertige CSS-Datei. Das Array kann aus einer durch "cp_css_input" eingelesenen Datei stammen bzw. manuell erstellt werden. Neben dem Array muss der Funktion der Pfad zur neuen Datei übergeben werden.

```
cp_css_output("style.css",$array);  
// erstellt die Datei "style.css" mit den Daten aus "$array"
```

8.5. Datumsfunktionen

Mit diesen Methoden der Date-Klasse ist das Berechnen beliebiger Datumsangaben möglich. Ausgehend von Parametern können Schaltjahre, Wochentage usw. genau ermittelt werden. Um die Methoden in Module einzubinden, muss die Datei "functions/cp.date.php" per "include" oder "include_once" eingebunden werden.

8.5.1. check_date()

Alternativ zur Methode "is_valid_date" kann zur Überprüfung eines Datums die Methode "check_date" eingesetzt werden. Diese nutzt intern die Funktion "checkdate" von PHP. Für die Überprüfung wird der Methode ein Datum im Format TT.MM.YY bzw. TT.MM.YYYY übergeben. Rückgabewerte sind true (1) und false (0).

```
$date->check_date("30.07.2005");  
// liefert true zurück für den 30.07.2005
```

8.5.2. days_per_month()

"days_per_month" wird auch von anderen Datumsfunktionen eingesetzt und ermittelt die Anzahl der Tage in einem Monat in einem bestimmten Jahr. Dabei werden auch Schaltjahre mit beachtet. Der Rückgabewert ist die entsprechende Anzahl an Tagen.

```
$var = $date->days_per_month("4","2007");  
// Anzahl der Tage im Monat 04/2007 (Ergebnis: "30")
```

8.5.3. first_monday()

Die Methode "first_monday" ermittelt den Montag der ersten Kalenderwoche eines Jahres. Dieser muss allerdings nicht immer innerhalb des Jahres, d.h. im Januar, sondern kann auch im Dezember liegen (z.B. am 29.12. für das Jahr 2004). Die Methode erhält als einzigen Parameter ein Jahr übergeben (4-stellig). Wird dieser Parameter nicht übergeben, dann wird das aktuelle Jahr verwendet. Der Rückgabewert ist das Datum des ersten Montags des Jahres im Timestamp-Format.

```
$date->first_monday("2005");  
// liefert für "2005" den Timestamp "1104274800" zurück
```

8.5.4. is_leap_year()

"is_leap_year" prüft, ob das übergebene Jahr ein Schaltjahr ist. Wird der Methode kein Jahr übergeben, so wird das aktuelle Jahr als Parameter genommen. Bei Erfolg gibt die Methode "true" zurück, andernfalls "false".

```
$var = $date->is_leap_year("2009");  
// prüft das Jahr 2009 (Ergebnis: "false")
```

8.5.5. is_valid_date()

Mit dieser Methode kann geprüft werden, ob ein Datum auch wirklich existiert. D.h. Tag, Monat und Jahr müssen vorhanden sein und bestimmte Begrenzungen einhalten (Tage pro Monat, Schaltjahr usw.). Der Methode "cp_valid_date" werden hierzu Tag, Monat und Jahr übergeben. Als Ergebnis wird "true" oder "false" zurückgegeben.

```
$var = $date->is_valid_date("29","3","2003");  
// prüft den 29.03.2003 (Ergebnis: "true")
```

8.5.6. mysql_timestamp()

Diese Methode benötigt keinen Parameter, sondern liefert einfach nur die aktuelle Zeit bzw. das aktuelle Datum im von MySQL benötigten Timestamp-Format zurück.

```
$var = $date->mysql_timestamp();  
// liefert das aktuelle Datum als MySQL-Timestamp
```

8.5.7. week_monday()

Als Erweiterung der Methode "first_monday" wird "week_monday" eingesetzt. Dieser Methode werden eine Kalenderwoche (1-52) sowie optional ein Jahr (4-stellig) übergeben. Als Rückgabewert erhalten Sie im Timestamp-Format das Datum des Montags in dieser Kalenderwoche.

```
$date->week_monday("3", "2005");  
// liefert den Timestamp für den Montag der 3. Kalenderwoche 2005
```

8.6. Formulare

*Alle Formulare in **Contentpapist** werden über die API erzeugt, da sich hier zentral schnell Überprüfungen der Formulare, Veränderungen der Felder usw. einbauen lassen. Die Formular-Funktionen stehen direkt in jeder Datei zur Verfügung, welche als Modul integriert ist, eine Einbindung weiterer Dateien ist nicht notwendig.*

8.6.1. form_start_wysiwyg()

Wird in einem Formular ein WYSIWYG-Editor eingesetzt, so muss dieses Formular mit der Funktion "form_start_wysiwyg" begonnen werden. Der Funktion wird hierzu beim Aufruf ein Formular-Ziel und die gewünschte Übergabemethode ("get"/"post") übergeben. Standardmäßig wird "post" genutzt "form_start_wysiwyg" nimmt zusätzlich eine Überprüfung des Browsers des Benutzers vor.

```
form_start_wysiwyg("admin.php?file=controlling&mode=add_info","post");
// Formular wird per "post" an das Modul "controlling" gesendet
```

8.7. Grafik

Die Grafik-Klasse der Entwickler-API ermöglicht die interaktive Erstellung von Kreis- und Balkengrafiken für Statistiken, die Anwendung verschiedener Bildeffekte usw. Es muss vorher die Datei "functions/cp.image.php" eingebunden werden. Zur Anwendung ist eine neue Instanz der Klasse zu initialisieren.

8.7.1. colorize()

"colorize" ist eine Methode, mit welcher Grafiken eingefärbt werden können. Der Methode müssen als Angaber eine Farbangabe (RGB, als Array) und eine Prozentangabe (Stärke der Einfärbung) übergeben werden.

```
$image->colorize(array(238,243,245),"50");
// färbt eine Grafik zu 50 % gelb ein
```

8.7.2. gd_version()

Einige spezifische Methoden/Funktionen zur Arbeit mit Grafiken stehen nur bei einer GD-Version > 2.0 zur Verfügung (z.B. "imagecreatetruecolor()"). Diese Methode überprüft, welche GD-Version vorhanden ist und gibt das Ergebnis als Array mit den Werten "short" (Kurze Nummer, z.B. "2") sowie "complete" (komplette Versionsnummer, z.B. "2.0.15") und "truecolor" (Unterstützung für Truecolor-Bilder [0/1]) zurück.

```
$image->gd_version();
// liefert Informationen zur GD-Version zurück
```

8.7.3. get_colorcode()

Die Methode "get_colorcode" ermöglicht die Umwandlung von Farbangaben in ein anderes Format. Übergeber werden dazu das Ausgangsformat, sowie die Angabe, ob es sich um ein Hex- oder Dezimalformat handelt. Die Rückgabe ist der entsprechend umgewandelte Wert. So lassen sich mit der Methode einfach RGB-Farbangaben in HTML-Farbangaben wandeln.

```
$image->get_colorcode("#0F07FF",1);
// liefert eine RGB-Angabe als Array zurück

$image->get_colorcode(array("120","86","42"),0);
// liefert eine HTML-Angabe als String
```

8.7.4. monochrome()

Mit der Methode "monochrome" wird eine Schwarz-Weiss-Version des aktuellen Bildes erzeugt. Diese verfügt über keinerlei Zwischenfarben. Als Speicherung empfehlen sich hier generell PNG und GIF aufgrund der geringeren Dateigrößen.

```
$image->monochrome();
// führt die Konvertierung durch
```

8.7.5. piechart()

Mit "piechart" lassen sich Statistiken in Form eines kreisförmigen Diagrammes darstellen inkl. Beschriftungen. Die Methode erhält Parameter für den Radius, die Beschriftung und den Wert der einzelnen Elemente sowie die Farben. Letztere werden per Array im RGB-Format übergeben.

```
$image->piechart(200,
array('Montag','Dienstag','Mittwoch','Donnerstag','Freitag'),
array(760,430,1508,980,1220),
array(
array('200','40','50'),
array('50','60','70'),
array('100','90','80'),
array('130','140','150'),
array('15','9','255'),
```

```
)  
);
```

Zeigt eine Grafik mit Angaben zu einzelnen Wochentagen, beispielsweise die Einnahmen eines Einzelhändlers. Im letzten Array werden die Farben zugewiesen (R,G,B).

8.7.6. rotate()

Für die Bildbearbeitung per Browser ist eine Methode zum Drehen von Bildern sehr nützlich. "rotate" berechnet alle Pixel eines Bildes für eine bestimmte Gradanzahl neu und liefert das Bild zurück. Der Methode wird hierzu lediglich die Stärke der Drehung in Grad übergeben.

```
$image->rotate("93");  
// aktuelles Bild um 20° drehen
```

8.7.7. show()

Mittels "show" wird ein erstelltes Bild an den Browser ausgegeben. Hierzu muss dieses als Bild aufgerufen werden (über den IMG-Tag). Es wird automatisch der Typ des Bildes benutzt, welcher bei der Erstellung angegeben bzw. beim Öffnen des Bildes erkannt wurde (unterstützt werden momentan nur JPG und PNG).

```
$image->show();  
// gibt das Bild aus
```

8.8. Mail

Mit den Methoden dieser Klasse können Sie HTML- oder Textemails zusammenstellen und versenden. Auch das Anhängen beliebiger Dateien oder der Versand an mehrere Personen in Form eines Newsletters ist möglich. Um diese Methoden in Module einzubinden, muss die Datei "functions/cp.mail.php" per "include" oder "include_once" eingebunden werden.

8.8.1. Beispiel 1

Mit diesem Beispiel soll die Leistungsfähigkeit der Klasse demonstriert werden. Hier wird eine neue Email im HTML-Format zusammengestellt. Sie erhält einen Anhang und wird anschließend an verschiedene Empfänger (TO, CC, BCC) versendet.

```
<?  
// ...  
// vanillaXML Webshop-Framework  
// ...  
  
include("functions/cp.mail.php");  
// Einbindung der Klasse  
  
include("functions/cp.mime.php");  
// Einbindung der MIME-Typen  
  
$mail = new van_mail;  
  
$mail->set_type("html");  
// Typ wird auf HTML gesetzt  
  
$mail->set_mime_types($cp_mime_types);  
  
$mail->set_subject("Einführung der neuen Klasse");  
// Titel der Email  
  
$mail->set_to("harry@domain.de","Harry Solid");  
// Empfängeremail und -name  
  
$mail->set_from("contact@sandoba.de","sandoba.de medien agentur");  
// Absenderemail und -name (TO)
```

```

$mail->add_cc(array("test_a@sandoba.de","test_b@sandoba.de"));
// CC-Empfänger in Array-Form

$mail->add_bcc(array("test_c@sandoba.de","test_d@sandoba.de"));
// BCC-Empfänger in Array-Form

$mail->set_ip();
// IP-Adresse des Versenders wird integriert

$mail->set_priority(3);
// Priorität normal

$mail->set_reply_to("reply_to@sandoba.de");
// Antworten gehen an diese Email-Adresse

$content = "Dies ist ein kleiner \TE<em>ST</em>\!!! ;-)";
// HTML-Inhalt der Email

$mail->set_body($content);

$mail->receive_receipt();
// Lesebestätigung anfordern

$mail->add_attachment("ordner/bild.gif","","","inline");
// Bild anfügen (wird in der Email angezeigt)

$mail->add_attachment("ordner/tool.zip");
// Datei anhängen

$mail->set_x_mailer($systemname." ".$systemversion." (".$systemurl.")");
$mail->set_x_bounced($systemname." ".$systemversion);

$mail->send();
// Email wird versandt

?>

```

8.9. RTF

Mit den Methoden dieser Klasse können Sie vorhandene RTF-Dateien auslesen und umwandeln lassen. Auch das Anlegen neuer RTF-Dateien ist möglich. Um diese Methoden in Module einzubinden, muss die Datei "functions/cp.rtf.php" per "include" oder "include_once" eingebunden werden.

8.9.1. Beispiel 1 (RTF-Datei anlegen)

In diesem Beispiel wird Ihnen gezeigt, wie Sie eine RFT-Datei anlegen und mit Inhalten füllen können. Nach der Erzeugung einer neuen Instanz der Klasse werden die notwendigen Daten übergeben, automatisch aufbereitet und als RTF gespeichert.

```

<?
// ...
// vanillaXML Webshop-Framework
// ...

include("functions/cp.rtf.php");
$rtf = new van_rtf;

$rtf->set_title("Meine RTF-Datei");
$rtf->set_author("Max Mustermann");
$rtf->set_company("Mustermann Ltd.");

$rtf->set_create_time();
// Erstellungsdatum in Sekunden (im Timestamp-Format)

```

```
$rtf->set_rev_time(time()-20000);  
// Datum der letzten Bearbeitung
```

```
$rtf->set_content("Hier können Sie <b>beliebige HTML- oder Text-Inhalte<<b> einfügen, alle <font size=\"25  
\>unterstützten <em>Tags</em></font> werden umgewandelt, andere automatisch <strike>gelöscht</strike>  
entfernt.<br/><br/>So sind Sie nicht nur auf einfache Textformierungen beschränkt...");
```

```
$content = $rtf->set_capitals();  
// Alles in Großbuchstaben (ja/nein)
```

```
// RTF erzeugen und als "myrtf.rtf" abspeichern
```

```
$rtf->create();  
$rtf->save("myrtf.rtf");  
?>
```

8.9.2. Beispiel 2 (RTF-Datei konvertieren)

Hier wird eine (zuvor erzeugte) RTF-Datei eingelesen und verarbeitet. Wichtige RTF-Formatierungen werden dabei nach Möglichkeit in HTML konvertiert.

```
<?  
// ...  
// vanillaXML Webshop-Framework  
// ...
```

```
include("functions/cp.rtf.php");  
$convert = new van_rtf;
```

```
$convert->open("myrtf.rtf");  
$convert->set_content();  
$convert->convert();  
?>
```

8.10. URL

Diese Funktionen ermöglichen beispielsweise das Auslesen von Titel und Metatags einer Website oder die Auflösung der URL in einzelne Teile. Um die Funktionen in Module einzubinden, muss die Datei "functions/cp.url.php" per "include" oder "include_once" eingebunden werden.

8.10.1. cp_url_element()

Diese Funktion liefert ein bestimmtes Element aus einer vorgegebenen URL zurück. Als Element können "scheme", "host", "port", "user", "pass", "path", "query", und "fragment" genutzt werden.

Anwendung:

```
cp_url_element("http://user@www.url.de:80/ordner/datei.php?variable=15","query")
```

- *scheme* - "http"
- *host* - "www.url.de"
- *port* - "80"
- *user* - "user"
- *pass* - ""
- *path* - "/ordner/datei.php"
- *query* - "variable=15"
- *fragment* - ""

8.10.2. cp_url_frameset()

Eine weitere Funktion zur Zerlegung von HTML-Seiten ist "cp_url_frameset". Mit dieser Funktion lassen sich

Framesets in die einzelnen Framequellen aufteilen. Alle Seiten, zu denen das Frameset verlinkt, werden in einem Array abgelegt mit den möglichen Attributen "href", "name" und "scolling". Auch hier ist die direkte Übergabe von Daten möglich.

```
$var = cp_url_frameset("frameset.html");  
// Frame-URLs in "frameset.html" ermitteln
```

8.10.3. cp_url_header()

Der Funktion "cp_url_header" wird eine URL übermittelt. Daraufhin wird eine Verbindung zu dieser URL aufgebaut und der Header ausgelesen. Dieser Header wird anschließend zurückgeliefert. Werden die Daten als Array benötigt, so kann dies als zusätzlicher Parameter angegeben werden. Über weitere Funktionen können genauere Daten ausgelesen werden.

```
$var = cp_url_header("http://www.url.de");  
// liefert den Header von "www.url.de"
```

```
$var = cp_url_header("http://www.url.de", "array");  
// liefert den Header von "www.url.de" als Array zurück
```

- *type* - Verbindungs-Typ
- *version* - HTTP-Version
- *status* - HTTP-Statuscode
- *description* - Meldung zum Statuscode
- *server* - Serverangaben
- *date* - Datum des Zugriffs
- *content-length* - Länge der Datei in Bytes
- *content-type* - MIME-Typ der Datei

8.10.4. cp_url_images()

Die Funktion "cp_url_images" dient der Gewinnung von Grafiken von einer vorgegebenen URL. Dies kann z.B. für die Archivierung von ganzen Websites nützlich sein. Die Seite wird hierzu eingelesen und durchläuft eine Routine, welche die Grafiken herausfiltert und die einzelnen Attribute geordnet in einem Array ablegt. Alternativ kann auch direkt ein Text übergeben werden, indem ein zweiter Parameter mit dem Wert "1" angehängt wird.

Die Rückgabe ist ein Array, in welchem als Attribute eines Bildes "width", "height", "border", "src" und "alt" abgelegt sein können. Grafiken, welche mehrfach auftreten, werden ignoriert.

```
$var = cp_url_images("datei.htm");  
// filtert alle Bilder aus "datei.htm"
```

8.10.5. cp_url_is_host()

Die Funktion "cp_url_is_host" überprüft, ob eine Datei auf dem lokalen bzw. einem anderen Server liegt. Dies kann u.a. zur Sicherung bei der Arbeit mit Dateien genutzt werden. Der Funktion wird hierzu als Parameter die URL der Datei übergeben. Im Erfolgsfall (Datei ist lokal vorhanden) wird TRUE zurückgegeben, andernfalls FALSE.

```
$var = cp_url_is_host("http://www.yahoo.com/index.php");  
// URL nicht auf lokalem Server vorhanden (Ergebnis: FALSE)
```

8.10.6. cp_url_links()

Die Funktion "cp_url_links" dient der Gewinnung von Links von einer vorgegebenen URL. Dies kann z.B. für die Archivierung von ganzen Websites nützlich sein. Die Seite wird hierzu eingelesen und durchläuft eine Routine, welche die Links herausfiltert und die einzelnen Attribute geordnet in einem Array ablegt. Alternativ kann auch direkt ein Text übergeben werden, indem ein zweiter Parameter mit dem Wert "1" angehängt wird.

Die Rückgabe ist ein Array, in welchem als Attribute eines "href" und "target" abgelegt sein können. Links, welche mehrfach auftreten, werden ignoriert.

```
$var = cp_url_links("datei.htm");  
// filtert alle Links aus "datei.htm"
```

8.10.7. cp_url_lookup()

Sie können mit der Funktion "cp_url_lookup" überprüfen, ob eine bestimmte URL auch tatsächlich verfügbar ist. Intern wird auf die Funktionen "cp_url_extend" und "cp_url_format" zurückgegriffen, um die Konformität der URL zu gewährleisten.

```
$var = cp_url_lookup("http://www.url.de");  
// testet, ob die URL "www.url.de" verfügbar ist
```

8.10.8. cp_url_metatags()

Bauen Sie einen eigenen Spider und / oder Link-Kataloge auf. Die Funktion "cp_url_metatags" liest alle Meta-Tags aus einer vorgegebenen URL aus und liefert diese als Array zurück. Sind bestimmte Meta-Tags nicht vorhanden, wird kein Array-Schlüssel erzeugt.

Ausnahmen sind "title" (wird ersetzt durch dem Titel der Website) und "description" (liest 300 Zeichen aus dem Textbereich der Website). Diese Meta-Tags werden ausgelesen:

- *audience* - Zielgruppe der Website (Erwachsene, Kinder usw.)
- *author* - Autor/Webmaster der Seite
- *content_language* - Sprache (nach ISO 639-1)
- *content_type* - MIME-Typ der Datei
- *copyright* - Copyright-Inhaber der Website
- *date* - Veröffentlichungsdatum
- *description* - Beschreibung der Website (max. 1000 Zeichen)
- *distribution* - Beschreibung des Verbreitungsgebietes
- *expires* - Verfallsdatum
- *keywords* - Stichwörter, Schlüsselwörter
- *language* - Sprache (nach ISO 639-1)
- *page_topic* - Stichwort für Seiteninhalt
- *pragma* - Cache-Verhalten für Proxy-Server
- *publisher* - Verantwortlicher für Veröffentlichung
- *refresh* - Aktualisierung nach "x" Sekunden
- *resource_type* - Typ der Website
- *revisit_after* - Website alle "x" Tage besuchen
- *robots* - Anweisung für Spider / Robots
- *title* - Titel der Website

```
$var = cp_url_metatags("datei.htm");  
// liest die Meta-Tags von "datei.htm" aus
```

8.10.9. cp_url_port()

Um Ihren Webserver vor unerwünschten Zugriffen zu schützen, sollten Sie stets überwachen, welche Ports geöffnet oder geschlossen sind. Die Funktion "cp_url_port" stellt das hierfür notwendige Hilfsmittel dar. Übermittelt werden neben der URL zum Webserver der Port sowie ein Zeitlimit (in Sekunden)

```
$var = cp_url_port("localhost",80,5);  
// überprüft den Port 80 auf "localhost"
```

8.10.10. cp_url_title()

Mit "cp_url_title" ermitteln Sie den Titel/Namen einer Website, sofern dieser im Quelltext angegeben wurde. Der Funktion wird hierzu die gewünschte URL übergeben. Intern wird auf die Funktionen "cp_url_extend" und "cp_url_format" zurückgegriffen, um die Konformität der URL zu gewährleisten.

```
$var = cp_url_title("de.yahoo.com/");  
// gibt den Titel der Yahoo! Deutschland-Website aus
```

8.11. Verschiedenes

In dieser Funktionssammlung sind verschiedene Bestandteile der Entwickler-API enthalten, welche die Extrahierung von Dateiinformationen wie ID3v2, IPTC oder EXIF usw. ermöglichen. Um die Funktionen in Module einzubinden, muss die Datei "functions/cp.misc.php" per "include" oder "include_once" eingebunden werden.

8.11.1. cp_clean_wysiwyg()

Dieser Funktion kommt in vanillaXML Webshop besondere Bedeutung zu. Mit "cp_clean_wysiwyg" lässt sich ein per Parameter übergebener Text aus dem WYSIWYG-Editor standardkonform aufbereiten. Es wird versucht, alle HTML-Tags in das XHTML-Format zu konvertieren.

```
$var = cp_clean_wysiwyg($string);  
// gibt den neuen Inhalt von "$string" zurück
```

8.11.2. cp_convert_filesize()

Verschiedene Dateigrößen können mit dieser Funktion in ein anderes Format konvertiert werden. Der Funktion wird neben der Dateigröße in Bytes das gewünschte Zielformat (KB, MB, GB, TB, PB, EB) sowie optional die Anzahl der Dezimalstellen übergeben.

```
$var = cp_convert_filesize("204922","KB","2");  
// gibt die Dateigröße in KB mit zwei Dezimalstellen zurück
```

8.11.3. cp_extract_exif()

Die Funktion "cp_extract_exif" extrahiert die EXIF-Informationen in einer JPEG- oder TIFF-Datei und gibt diese als assoziatives Array zurück. EXIF enthält wichtige Angaben zu Grafiken wie die Bittiefe, die Verschlussgeschwindigkeit und das Erstellungsprogramm. Diese Angaben sind nicht immer zwingend für eine Grafik erforderlich und können daher variabel ausfallen.

EXIF-Informationen werden meist von Digitalkameras in die Grafiken eingefügt, damit eine weitergehende Anpassung der Grafiken anhand der Einstellungen bei der Aufnahme vorgenommen werden kann. Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom vanillaXML Webshop-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

- *aperture_fnumber* - Blendeneinstellung
- *aperture_value* - Blenden-Wert in APEX
- *artist* - Autor/Artist
- *brightness_value* - Helligkeitswert der Grafik in APEX
- *color_space* - Farbraum / Bittiefe des Bildes
- *copyright* - Copyright-Informationen
- *date* - Datum der Erstellung
- *date_time* - Datum/Zeit der letzten Änderung
- *date_time_digitized* - Datum/Uhrzeit der Digitalisierung
- *date_time_original* - Datum/Uhrzeit des Originals
- *description* - Beschreibung der Grafik
- *exif_image_length* - Größe der EXIF-Grafik
- *exif_image_width* - Größe der EXIF-Grafik
- *exif_version* - Verwendete Version der EXIF-Spezifikation
- *exposure_bias_value* - Exposure Bias-Wert in APEX
- *exposure_program* - Genutztes Programm für die Datei
- *exposure_time* - Verschlussgeschwindigkeit
- *file_source* - Quelle der Datei (Digitalkamera etc.)
- *flash* - Blitzlicht (0 = kein Blitzlicht, >= 1 = Blitzlicht)
- *flash_pix_version* - Verwendete FlashPix-Version
- *fnumber* - F-number (F-stop)
- *focal_length* - Brennweite in Millimeter
- *focal_plane_resolution_unit* - Abmessungen der Brennebene (1 = keine Einheit, 2 = Inch, 3 = Zentimeter)
- *focal_plane_x_resolution* - Horizontale Auflösung der Brennebene
- *focal_plane_y_resolution* - Vertikale Auflösung der Brennebene
- *height* - Höhe der Grafik
- *is_color* - Farbbild (0 = Nein, 1 = Ja)
- *iso_speed_ratings* - Äquivalent zum CCD Index
- *make* - Hersteller der Digitalkamera
- *metering_mode* - Modus der Entfernungsmessung
- *mime_type* - MIME-Type (z.B. "image/jpeg")
- *model* - Modellbezeichnung der Digitalkamera
- *name* - Name der Datei ohne Pfadangaben
- *orientation* - Ausrichtung (1 = oben links, 2 = oben rechts usw.)
- *resolution_unit* - Abmessungen (1 = keine Einheit, 2 = Inch, 3 = Zentimeter)
- *scene_type* - Typ der Szene (direkt aufgenommen etc.)

- *sensing_method* - Typ des genutzten Sensors der Digitalkamera
- *shutter_speed_value* - Verschlussgeschwindigkeit in APEX
- *size* - Größe in Bytes
- *size_formatted* - Größe in MB umgerechnet
- *software* - Software der Digitalkamera bzw. Bildbearbeitungsprogramm
- *status* - Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden, 3 = falscher Dateityp, 4 = keine EXIF-Unterstützung vorhanden)
- *type* - Dateityp
- *x_resolution* - Breite der Grafik für den Druck (meist in Inch)
- *y_resolution* - Höhe der Grafik für den Druck (meist in Inch)
- *width* - Breite der Grafik
- *ycbcr_positioning* - Positionierung der "YCbCr"-Elemente

Weitere Angaben können je nach Modell der Digitalkamera zurückgegeben werden, sind jedoch nicht standardisiert und daher nicht immer verlässlich.

```
$var = cp_extract_exif("images/regenwald.jpg");
// liefert alle Angaben zu dieser Datei zurück
```

8.11.4. cp_extract_id3()

Die Funktion "cp_extract_id3" extrahiert die ID3- bzw. ID3v2-Informationen in einer MP3-Datei und gibt diese als assoziatives Array zurück. ID3/ID3v2 enthält wichtige Angaben zum Musikstück wie die Dauer, die Abtastrate und Bitrate, MPEG-Version und MPEG-Layer, aber auch zusätzliche Angaben wie Autor, Titel, einen Kommentar usw.

Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom vanillaXML Webshop-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

- *album* - Album, aus dem die Datei stammt
- *artist* - Autor/Artist des Musikstückes
- *bitrate* - Bitrate der Datei in kBit/s
- *comment* - Kommentar zum Musikstück
- *copyright* - Copyright-Informationen (-1 = Nicht geschützt, 1 = Geschützt)
- *emphasis* - Betonung des Musikstückes (keine einheitliche Nutzung)
- *framerate* - Anzahl der Datenblöcke in der Datei
- *genre* - Genre-Nummer
- *genre_name* - Name des Genres
- *length* - Gesamte Spieldauer in Sekunden
- *length_formatted* - Spieldauer in lesbarer Form mit Minuten und Sekunden
- *length_minutes* - Verbleibende Minuten der Spieldauer
- *length_seconds* - Verbleibende Sekunden der Spieldauer
- *mode* - Anzahl der genutzten Kanäle ("Stereo", "Joint-Stereo", "Dual-Channel" oder "Mono")
- *mode_extension* - Nicht genutztes Bit
- *mpeg_layer* - Der genutzte MPEG-Layer (Layer I/II/III)
- *mpeg_version* - MPEG-Version der Datei (MPEG 1/2)
- *name* - Name des Musikstückes ohne Pfadangaben
- *original_home* - Modus (0 = Kopie, 1 = Original)
- *padding_bit* - Dient der Berechnung der Frame-Anzahl
- *private_bit* - Bit zur privaten Verfügung in einer Datei
- *protection_bit* - Schutz-Bit am Anfang einer Datei (0 = Deaktiviert, 1 = Aktiviert)
- *sampling_frequenze* - Abtastrate in Hz
- *size* - Größe in Bytes
- *size_formatted* - Größe in MB umgerechnet
- *status* - Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden, 3 = falscher Dateityp)
- *syncword* - Synchronisierungs-Bits am Anfang einer Datei
- *time* - Erstellungsdatum der Datei
- *title* - Songtitel

```
$var = cp_extract_id3("music/yvonne_cutterfeld_solo.mp3");
// liefert alle Angaben zu dieser Datei zurück
```

8.11.5. cp_extract_iptc()

IPTC ist ein Standard der "International Press Telecommunications Council" und erlaubt die zusätzliche Integration von beschreibenden Informationen in den Kopf-Bereich einer JPEG-Datei. Die Funktion "cp_extract_iptc" extrahiert diese Informationen und gibt diese als assoziatives Array zurück. Diese Angaben sind nicht immer zwingend für eine Grafik erforderlich und können daher variabel ausfallen.

Damit die Funktion arbeiten kann, wird ihr ein Dateiname samt Pfad ausgehend vom vanillaXML Webshop-Verzeichnis übergeben. Zurückgegeben werden folgende Angaben, sofern diese in einer Datei vorhanden sind:

- *author* - Autor
- *author_description* - Autor der Beschreibung
- *caption* - Bildbeschreibung
- *category* - Kategorie
- *city* - Stadt
- *copyright_notice* - Copyright-Informationen
- *country* - Land
- *creation_date* - Erstellungsdatum
- *credit_byline_title* - Position des Autors
- *headline* - Überschrift
- *height* - Höhe der Grafik
- *keywords* - Stichwörter
- *name* - Name der Grafik ohne Pfadangaben
- *otr* - Aufgeber-Code
- *photo_source* - Quelle
- *priority* - Dringlichkeit
- *size* - Größe in Bytes
- *size_formatted* - Größe in MB umgerechnet
- *source* - Bildrechte
- *spec_instr* - Anweisungen
- *state* - Bundesland
- *status* - Error-Code der Datei (1 = alles OK, 2 = Datei nicht vorhanden)
- *supplemental_category* - Ergänzende Kategorie
- *title* - Bildtitel
- *width* - Breite der Grafik

```
$var = cp_extract_iptc("images/regenwald.jpg");  
// liefert alle Angaben zu dieser Datei zurück
```

8.11.6. cp_get_charset()

Ermitteln Sie den zu einer Sprache gehörigen Zeichensatz. Die Funktion erwartet den Namen einer Sprache im zweistelligen ISO 639-1-Format. Der einzige Rückgabewert ist das verwendete ISO-Format des Zeichensatzes.

```
$var = cp_get_charset("jp");  
// gibt den Zeichensatz für Japanisch zurück ("iso-8859-1")
```

8.11.7. cp_js_alert()

Wenn Sie auf einfachem Wege eine Alarm-Meldung per Javascript ausgeben wollen, dann sollten Sie "cp_js_alert" einsetzen. Per Parameter wird der auszugebende Text übergeben, ein zweiter Parameter gibt an, ob der resultierende Quellcode zurückgegeben ("1") oder direkt ausgeführt ("0") werden soll.

```
cp_js_alert("Hallo Welt!", "0");  
// hier wird direkt "Hallo Welt!" angezeigt
```

8.12. XML

Diese Funktionen ermöglichen die Interaktion mit XML-Inhalte bzw. -Dateien. Diese können eingelesen, zu Arrays verarbeitet und gespeichert werden. Um die Funktionen in Module einzubinden, muss die Date "functions/cp.xml.php" per "include" oder "include_once" eingebunden werden.

8.12.1. add_attribute()

Jeder Knoten einer XML-Datei kann zusätzlich zum eigentlichen Inhalt (d.h. weiteren Knoten oder einem String) weitere Informationen in Form von Attributen transportieren. Für jedes Attribut muss neben dem Namen und Inhalt des Attributes zusätzlich der Pfad zum gewünschten Knoten angegeben werden.

```
$xml->add_attribute("Shop/Produkte/Fussball", "katnummer", "1");  
// fügt dem Knoten das Attribut katnummer="1" hinzu
```

8.12.2. add_content()

Die Methode "add_content" fügt einem Knoten weitere Daten hinzu. Sind bereits Daten vorhanden, so werden die neuen Daten angehängt. Der Funktion der Pfad zum genutzten Knoten und der neue Inhalt übergeben.

```
$xml->add_content("Katalog/Produkt/vanillaXML","Webshop für Amazon");
```

8.12.3. add_node()

Sofern bereits eingelesene XML-Dateien als Array vorliegen, kann man diesen mit "add_node" neue Knoten hinzufügen. Hierzu übergibt man der Method den Pfad zum neuen Knoten und den Namen des Knotens. D.h. soll ein Knoten auf der Ebene "Katalog > Produkt" angelegt werden mit dem Namen "Beschreibung", so wendet man die Funktion folgendermaßen an:

```
$xml->add_node("Katalog/Produkt","Beschreibung");
```

8.12.4. Beispiel 1

Dieses kurze Beispiel soll veranschaulichen, wie die XML-Funktionen von vanillaXML Webshop genutzt werden können. Zuerst wird eine vorhandene XML-Datei eingeladen.

Anschließend werden verschiedene Knoten hinzugefügt, entfernt und mit Inhalten gefüllt. Die bearbeitete Datei wird anschließend als XML-Datei neu gespeichert. Der Haupt-Knoten ist in diesem Fall "channel":

```
<?
// ...
// vanillaXML Webshop-Framework
// ...

include("functions/cp.xml.php");
$xml = new van_xml; $xml->open("text.xml");
$xml->add_node("channel/0/item/$a","options");
$xml->remove_node("channel/0/item/$a/link");
$xml->remove_content("channel/0/item/$a/title");
$xml->set_content("channel/0/item/description",$content);
$xml->save("test.xml","xml");

// ...
// vanillaXML Webshop-Framework
// ...

?>
```

8.12.5. Beispiel 2 (RSS-/RDF-Import)

Um externe Inhalte im RSS-/RDF-Format auf der Website anzuzeigen, muss stets der Pfad zu einer Vorlage angegeben werden. In dieser Vorlage kann eine freie Formatierung der einzelnen Elemente vorgenommen werden. Es muss hier jedoch stets die Auflistung der Einträge mittels {cp:repeat} intakt bleiben:

```
<strong>{cp:xml:title}</strong><br/>
Beschreibung: {cp:xml:description}<br/>
Website: {cp:xml:link}<br/><br/>

{cp:repeat}
- <a href="{cp:xml:link}">{cp:xml:title}</a><br/>
{/cp:repeat}
```

8.12.6. db_input()

"db_input" ermöglicht die Integration von XML-Daten in eine Datenbank. Dies kann z.B. genutzt werden, um mittels "db_output" als XML exportierte Datensätze wieder in die Datenbank einzuspielen. Die Methode erwartet als Parameter den Pfad zur XML-Datei, den Tabellennamen (in diese werden die Datensätze eingefügt) sowie den Pfad zum genutzten Knoten.

Alle notwendigen Umwandlungsprozesse werden automatisch vorgenommen. Zusätzlich wird geprüft, ob die

Felder in der XML-Datei mit denen der Datenbank übereinstimmen (u.a. Reihenfolge, Feldnamen). Wenn dieser Vorgang erfolgreich abgeschlossen wurde, dann wird TRUE zurückgegeben, andernfalls FALSE.

8.12.7. db_output()

Diese Methode exportiert eine Tabelle einer Datenbank in ein XML-Format und schreibt den Inhalt in eine angegebene Datei. Optional kann als dritter Parameter der Name des Wurzelements benannt werden.

```
$xml->db_output("test.xml","information");  
// legt die Tabelle "information" als XML-Datei "test.xml" an
```

8.12.8. open()

Die Methode "open" lädt eine XML-Datei aus einem angegebenen Verzeichnis und wandelt den Inhalt komplett in ein Array um. Dieses Array wird für die spätere Verarbeitung benötigt. Über den optionalen Parameter "cache_time" kann angegeben werden, wie lange eine Zwischenspeicherung im Cache vorgenommen werden soll. Diese Zeit wird in Sekunden festgelegt. Wird der Parameter weggelassen, dann werden 3600 Sekunden als Standard verwendet.

```
$xml->open("test.xml");  
// lädt die Datei "test.xml"
```

8.12.9. remove_attribute()

Attribute zu Knoten, welche per XML-API angelegt wurden, müssen auch wieder entfernt werden können. Für diese Aufgabe gibt es die Methode "remove_attribute". Es muss hier lediglich der Pfad zum Knoten angegeben werden.

```
$xml->remove_attribute("Shop/Produkte/Fussball");  
// entfernt die Attribute des Knotens
```

8.12.10. remove_content()

Ähnlich wie die Methode "remove_node" entfernt auch "remove_content" Bestandteile aus einem Array. In diesem Fall jedoch wird nur der Inhalt eines Knotens gelöscht. Der Knoten selbst bleibt bestehen und kann weiterhin genutzt werden.

```
$xml->remove_content("Katalog/Produkte/Beschreibung");  
// "Beschreibung" mit dem Pfad "Katalog/Produkte" wird geleert.
```

8.12.11. remove_node()

Vorhandene Knoten in einem Array können mit dieser Methode entfernt werden. Hierzu übergibt man der Methode lediglich den Pfad zum Knoten. D.h. soll ein Knoten auf der Ebene "Katalog > Produkt" entfernt werden mit dem Namen "Beschreibung", so wendet man die Funktion folgendermaßen an:

```
$xml->remove_node("Katalog/Produkt/Beschreibung");
```

8.12.12. save()

"safe" sorgt für die Aufbereitung der in einem Array vorliegenden Daten in das gewünschte Zielformat. Aktuell werden XML sowie PHP unterstützt. In der XML-Variante wird das komplette Array umgewandelt, die PHP-Variante stellt eine Abbildung des Array-Inhaltes in einer Datei in lesbarer Form dar.

Ein vierter (optionaler) Parameter kann in Form eines Arrays Angaben zur verwendeten Version ("version") und dem Zeichensatz ("encoding") der XML-Datei beinhalten. Weiterhin kann festgelegt werden, ob der generierte Inhalt zurückgeliefert werden soll (in diesem Fall keine Datei angeben), statt ihn in der Datei zu hinterlegen.

```
$xml->save("test.xml","xml",$array,$options);  
// das Array "$array" wird als XML-Datei "test.xml" exportiert
```

```
$xml->save("test.php","php",$array);  
// das Array "$array" wird als PHP-Datei "test.php" exportiert
```

8.12.13. set_content()

Die Methode "set_content" füllt einen Knoten mit beliebigen Daten. Sind bereits Daten vorhanden, so werden diese mit den neuen Daten überschrieben. Der Methode wird ein Pfad zum genutzten Knoten und der neue Inhalt übergeben.

```
$xml->set_content("Katalog/Produkt/Contentpapst","CMS für den Mittelstand");
```

8.12.14. show()

Mit der Methode "show" wird eine verarbeitete Datei im XML-Format zurückgeliefert. Optional kann der einzige Parameter auch auf "0" gesetzt werden. In diesem Fall erfolgt eine direkte Ausgabe an den Browser. Je nach Browser erscheint dann eine Darstellung der XML-Datei in einer Baumstruktur bzw. eine Downloadaufforderung.

```
$xml->show("0");  
// XML-Datei wird an den Browser ausgegeben
```

8.12.15. show_feed()

Externe Inhalte im RDF-/RSS-Format können auch per API direkt in eigene Module integriert werden. Die Methode benötigt Parameter zum Pfad der Datei, zum Pfad der verwendeten Vorlage und zur maximalen Anzahl der Einträge. Ein letzter Parameter bestimmt, ob die Inhalte direkt ausgegeben werden ("0") oder zurückgeliefert werden ("1" bzw. weglassen). Alle Dateien werden automatisch zwischengespeichert. Die Ladezeit wird dadurch erheblich reduziert.

```
$xml->show_feed("slashdot.rdf","themes/$theme/xml.htm",5);  
// gibt die Inhalte der Datei "slashdot.rdf" mit der Vorlage "themes/$theme/xml.htm" aus (maximal 5 Einträge)
```

8.13. Zeichen & Texte

Diese Funktionen dienen der Manipulation von Zeichen & Texten ("strings"), ermöglichen so beispielsweise eine temporäre Verschlüsselung. Um die Funktionen in Module einzubinden, muss die Datei "functions/cp.string.php" per "include" oder "include_once" eingebunden werden.

8.13.1. cp_string_convert()

Die Funktion "cp_string_convert" konvertiert ein eindimensionales Array zu einem String. Die einzelnen Array-Elemente werden dabei anhand eines Trennzeichens - standardmäßig ",", - miteinander verbunden.

```
$var = cp_string_convert($my_array, "-");  
// die Elemente des Arrays "$my_array" werden als String zurückgegeben
```

8.13.2. cp_string_count_words()

Diese Funktion zählt die Anzahl der Wörter in einem übergebenen String und liefert diese zurück. U.a. kann "cp_string_count_words" Anwendung finden in Formularen, in denen die Eingaben durch den Nutzer auf eine bestimmte Wortzahl beschränkt werden sollen.

```
$var = cp_string_count_words($string);  
// liefert die Anzahl der Wörter in "$string"
```

8.13.3. cp_string_decode()

"cp_string_decode" entschlüsselt einen beliebig langen String, welcher vorher mit "cp_string_encode" verschlüsselt wurde. Der Rückgabewert kann dann wieder in eine Variable geschrieben werden.

```
$var = cp_string_decode($string);  
// entschlüsselt die Variable "$string"
```

8.13.4. cp_string_encode()

"cp_string_encode" verschlüsselt einen beliebig langen String, welcher dieser vorher übergeben wurde. Der Rückgabewert kann dann wieder in eine Variable geschrieben werden.

```
$var = cp_string_encode("Hallo");  
// verschlüsselt "Hallo"
```

8.13.5. cp_string_rm_spaces()

Diese Funktion wandelt Sonderzeichen um und entfernt Leerzeichen aus einem übergebenen Text.

```
$var = cp_string_rm_spaces($content);  
// liefert den neuen Inhalt von "$content"
```

8.13.6. cp_string_shorten()

Ähnlich wie die Funktion "cp_word_shorten" verkürzt "cp_string_shorten" einen String. In diesem Fall jedoch nicht die einzelnen Elemente, sondern den gesamten String. Hierzu wird eine gewünschte Länge angegeben, standardmäßig sind dies 50 Zeichen.

```
$var = cp_word_shorten($text,50);  
// verkürzt "$text" auf 50 Zeichen
```

8.13.7. cp_string_summarize()

Die Funktion "cp_string_summarize" verkürzt im Gegensatz zu "cp_string_shorten" einen Text nicht auf eine maximale Anzahl an Zeichen, sondern führt eine Wortlimitierung durch. Neben dem Text wird hierzu die gewünschte Anzahl an Worten (vom Anfang ausgehend) angegeben. Als Standardwert werden 10 Worte verwendet.

```
$var = cp_string_summarize($text,"20 ");  
// der Text wird auf 20 Worte gekürzt und zurückgegeben
```

8.13.8. cp_string_unicode()

"cp_string_unicode" wandelt einen ISO-konformen String in Unicode um. Unicode ist eine Zeichenkodierung, mittels derer sich beliebige Sprachen mit deren jeweiligen Sonderzeichen darstellen lassen. Wollen Sie also sichergehen, dass Texte bei jeder Spracheinstellung gleich aussehen, setzen Sie diese Funktion ein. U.a. lassen sich hiermit auch Email-Adressen vor Spidern schützen, da Unicode für diese nicht lesbar ist.

```
$var = cp_string_unicode($string);  
// liefert die Unicode-Version von "$string" zurück
```

8.14. ZIP

Mit der ZIP-Klasse werden im ZIP-Format gepackte Dateien gelesen und entpackt. Über weitere Methoden können neue ZIP-Dateien mit beliebigen Inhalten und Ordnern erstellt werden. Um diese Methoden in Module einzubinden, muss die Datei "functions/cp.zip.php" per "include" oder "include_once" eingebunden werden.

8.14.1. add_dir()

Auch Verzeichnisse können in der ZIP-Datei erstellt werden. Über die Methode "add_dir" wird der aktueller Datei ein neues Verzeichnis hinzugefügt. Auch Verzeichnisse in der Form "verzeichnis/unterverzeichnis" sind möglich.

```
$zip->add_dir("sport/formel1");  
// erstellt ein Verzeichnis "sport/formel1" in der ZIP-Datei
```

8.14.2. add_file()

"add_file" ist die wichtigste Methode für die Erstellung neuer ZIP-Dateien. Über diese Methode bestimmen Sie

welche Dateien in die Datei eingefügt werden. Als Parameter werden der Pfad zur Datei, sowie der gewünschte Name und das Verzeichnis in der ZIP-Datei angegeben (letztere sind optional). Zwingend erforderlich ist allein die Angabe des Pfades.

```
$zip->add_file("ferrari.jpg", "mschumacher_ferrari.jpg", "sport/formel1");  
// hier wird die Datei "ferrari.jpg" als "mschumacher_ferrari.jpg" im passenden Verzeichnis eingefügt
```

8.14.3. Beispiel 1

In diesem Beispiel wird eine einzelne Datei aus einer ZIP-Datei extrahiert und getrennt von dieser gespeichert. Die ZIP-Datei wird durch "open" zuerst geöffnet und analysiert, "unpack" führt die Extrahierung durch und "close" schließt die ZIP-Datei.

```
<?  
// ...  
// vanillaXML Webshop-Framework  
// ...  
  
include("functions/cp.zip.php");  
// Einbindung der Klasse  
  
$zip = new van_zip;  
$zip->open("sport.zip");  
$zip->unpack("bayern_muenchen.jpg", "bayern_muenchen.jpg", "bilder");  
$zip->close();  
?>
```

8.14.4. Beispiel 2

Jetzt wird eine neue ZIP-Datei erstellt. In dieser Datei sollen Test-Berichte gespeichert werden. Eine Datei "readme.txt" im Hauptverzeichnis gibt allgemeine Hinweise. Die Testberichte selbst sind jeweils thematisch geordnet in Unterordnern abgelegt. Zuletzt wird die Datei als "testberichte.zip" gespeichert.

```
<?  
// ...  
// vanillaXML Webshop-Framework  
// ...  
  
include("functions/cp.zip.php");  
// Einbindung der Klasse  
  
$zip = new van_zip;  
$zip->add_dir("musik");  
$zip->add_dir("sport");  
$zip->add_file("cdplayer.doc", "test.cdplayer.doc", "musik");  
$zip->add_file("fussball.doc", "test.fussball.doc", "sport");  
$zip->add_file("readme.txt");  
$zip->pack("testberichte.zip");  
?>
```

8.14.5. close()

"close" schließt eine geöffnete Datei. Diese Methode muss nur aufgerufen werden, wenn eine vorhandene ZIP-Datei genutzt wird. Sie ist aufzurufen, sobald alle gewünschten Dateien entpackt wurden.

```
$zip->close();  
// schließt die aktuelle Datei
```

8.14.6. download()

Die Methode Download führt einen direkten Download der zusammengestellten ZIP-Datei durch. Die Methode wird nur intern aufgerufen und benötigt als Parameter den Dateinamen sowie den Inhalt der ZIP-Datei. Je nach

Browser werden unterschiedliche Einstellungen vorgenommen. Im Ergebnis sieht der Benutzer stets ein neues Fenster mit einer Download-Aufforderung.

```
$zip->download($file,$content);  
// Download von "$content" als "$file"
```

8.14.7. get_dos_time()

Diese Methode verwandelt eine Datumsangabe im UNIX-Format in das DOS-Format. Aufgrund der Spezifikationen des ZIP-Formates wird eine vorherige Umwandlung notwendig. Jedoch wird diese Methode allein intern aufgerufen und muss nicht manuell ausgeführt werden.

8.14.8. open()

Mittels "open" wird eine vorhandene ZIP-Datei gelesen. Die enthaltenen Informationen werden überprüft und strukturiert in Form eines Arrays für die weitere Verarbeitung (z.B. "unpack") zur Verfügung gestellt.

```
$zip->open("sport.zip");  
// öffnet die Datei "sport.zip"
```

8.14.9. pack()

Die Methode "pack" dient der Erstellung einer neuen ZIP-Datei. Vorher müssen mittels "add_dir" neue Ordner (optional) und mittels "add_file" neue Dateien (notwendig) hinzugefügt worden sein. "pack" erhält als Parameter den Dateinamen, eine Angabe, ob eine bereits vorhandene Datei überschrieben werden soll (standardmäßig "1") und ob ein automatischer Download durchgeführt werden soll (standardmäßig "0"). Wenn kein Download gewünscht ist, wird die Datei am angegebenen Ort gespeichert.

```
$zip->pack("sport.zip","1");  
// speichert und überschreibt die Datei "sport.zip"
```

```
$zip->pack("sport.zip","0","1");  
// lädt die erstellte ZIP-Datei direkt herunter
```

8.14.10. show_content()

"show_content" liefert im Gegensatz zu "show_info" ein Array mit Informationen über die einzelnen Dateien in einer ZIP-Datei zurück. Voraussetzung ist, dass die ZIP-Datei vorher mittels "open" geöffnet und analysiert wurde. Die Informationen beinhalten u.a. Größe, Erstellungsdatum und Typ der Datei sowie einen optionalen Kommentar. "show_content" dient der externen Weiterverarbeitung der Dateinformationen. Über einen optionalen Parameter können gezielt Informationen über eine Datei herausgegriffen werden.

```
$zip->show_content();  
// liefert Informationen über alle enthaltenen Dateien
```

```
$zip->show_content("readme.txt");  
// liefert Informationen über die Datei "readme.txt"
```

8.14.11. show_info()

"show_info" liefert ein Array mit Informationen über die Struktur der ZIP-Datei zurück. Voraussetzung ist, dass die ZIP-Datei vorher mittels "open" geöffnet und analysiert wurde. Die Informationen beinhalten u.a. die Anzahl der enthaltenen Dateien sowie die Größe und einen optionalen Kommentar. "show_info" wird intern genutzt, die Informationen sind jedoch auch für die externe Weiterverarbeitung abrufbar.

```
$zip->show_info();  
// liefert Informationen über die aktuelle Datei
```

8.14.12. Statusmeldungen

Je nach genutzter Methode werden unterschiedliche Statusmeldungen über Erfolg/Misserfolg einer Aktion zurückgegeben. Im Folgenden finden Sie eine Liste der häufigsten Meldungen:

- 0 = Leeres Ergebnis, Information/Datei ist nicht vorhanden.
- 1 = Vorgang erfolgreich abgeschlossen.
- 2 = Die Zielfeile konnte nicht geschrieben werden.
- 3 = Es wurde keine Zielfeile angegeben. (Pack)
- 4 = Cache-Datei konnte nicht geöffnet werden.
- 5 = Datei konnte nicht im Archiv gefunden werden. (Unpack)
- 6 = Benötigte Erweiterung "zlib" und die Funktion "gzopen" konnten nicht geladen werden. (Unpack)
- 7 = Die Zielfeile ist bereits vorhanden.
- 8 = Es wurden bisher keine Dateien hinzugefügt.

8.14.13. unpack()

"unpack" entpackt eine Datei aus einer ZIP-Datei und speichert diese unter einem neuen Namen im angegebenen Verzeichnis. Hierzu werden als Parameter der ursprüngliche sowie der neue Name der Datei und ein Verzeichnis benötigt. Im Erfolgs- oder Misserfolgsfall werden Statusmeldungen zurückgegeben (siehe "Statusmeldungen").

```
$zip->unpack("alte_datei.txt", "neue_datei.txt", "pfad/verzeichnis");  
// entpackt die Datei "alte_datei.txt"
```

8.15. FTP - File Transfer Protocol

Mit den Methoden dieser Klasse können Sie per PHP mit FTP-Servern interagieren, Datei verschieben, hochladen und entfernen usw. Auch das setzen von Zugriffsrechten ist selbstverständlich möglich. Um diese Methoden in Module einzubinden, muss die Datei "functions/cp.ftp.php" per "include" oder "include_once" eingebunden werden.

8.15.1. Beispiel 1:

Mit diesem Beispiel soll der Funktionsumfang der Klasse demonstriert werden. Hier wird ein FTP-Server geöffnet und anschließend wird der Inhalt eines Unterordners als Array ausgegeben.

```
<?  
// ...  
// vanillaXML Webshop-Framework  
// ...  
  
include("functions/cp.ftp.php");  
// Einbindung der Klasse  
  
$ftp = new van_ftp();  
  
$ftp->open("ftp.microsoft.com");  
  
$ftp->change_dir("Products");  
print_r($ftp->show());  
$ftp->close();  
?>
```

8.15.2. change_dir()

Mittels der Methode "change_dir" wird ein Verzeichniswechsel auf einem FTP-Server initiiert. Dazu benötigt die Methode die Angabe des gewünschten Ordners zu dem gewechselt werden soll. Rückgabewerte sind je nach Ergebnis true (1) oder false (0):

```
$status = $ftp->change_dir("filme/komoedien");  
// wechselt in das Verzeichnis "filme/komoedien"
```

8.15.3. chmod()

Wichtig für FTP-Server, die von mehreren Personen genutzt werden ist die Sicherung von Dateien und Ordnern mittels CHMOD. Über den CHMOD kann detailliert beschrieben werden, welche Gruppen welche Rechte für eine Datei / einen Ordner haben. Für das Setzen des CHMOD steht die Methode "chmod" zur Verfügung. Diese

erhält als Parameter den Pfad zur Datei und den neuen CHMOD (3-stellig). Rückgabewerte sind je nach Ergebnis true (1) oder false (0):

```
$ftp->chmod("grafiken_2004.cdr", "777");  
// gibt die Datei für alle Gruppen frei
```

8.15.4. close()

Über die Methode "close" wird lediglich die über "open" aufgebaute Verbindung zu einem FTP-Server wieder unterbrochen:

```
$ftp->close();
```

8.15.5. create_dir()

Die Methode "create_dir" dient - wie könnte es anders sein - der Erstellung eines neuen Verzeichnisses. Hier wird ausgehend vom aktuellen Verzeichnis ein neues Verzeichnis mit dem als einzigen Parameter übergebenen Namen erstellt. Wenn bspw. ein Verzeichnis unterhalb des Startverzeichnisses erstellt werden soll, dann muss vorher mittels "change_dir" dorthin gewechselt werden. Standardmäßig wird versucht, dem neuen Verzeichnis den CHMOD 777 zu geben. Rückgabewerte sind, je nachdem ob das Verzeichnis erstellt werden konnte, true (1) oder false (0):

```
$ftp->create_dir("filme/komoedien");  
// erstellt das Verzeichnis "filme/komoedien"
```

8.15.6. delete_dir()

Mit der Methode "delete_dir" lassen sich schnell und einfach komplette Ordner mit Unterordnern und Dateien rekursiv löschen. Es muss dazu lediglich die Methode mit dem Namen des Verzeichnisses aufgerufen werden, in dem gestartet werden soll. Dieses Verzeichnis wird zum Schluss ebenfalls gelöscht.

```
$ftp->delete_dir("backup/2003/");  
// löscht das komplette Verzeichnis "backup/2003"
```

8.15.7. delete_file()

Im Gegensatz zu "delete_dir" wird mit der Methode "delete_file" nur eine einzelne Datei gelöscht. Rückgabewerte sind je nach Ergebnis (Berechtigung vorhanden ja/nein) true (1) oder false (0):

```
$ftp->delete_file("backup/2003/vorlagen.zip");  
// löscht die Datei "backup/2003/vorlagen.zip" vom Server
```

8.15.8. download_file()

Ein Download einer Datei vom FTP-Server kann über die Methode "download_file" durchgeführt werden. Notwendig sind hier als Parameter (in dieser Reihenfolge) der Name für die lokale Datei, der Name der Datei auf dem Server und der Pfad zur Datei ausgehend vom aktuellen Verzeichnis. Mögliche Rückgabewerte sind "1" (Download erfolgreich), "-1" (es konnte nicht in den Ordner gewechselt werden), "-2" (Datei kann nicht heruntergeladen werden) und "-3" (Datei kann nicht geöffnet werden).

```
$ftp->download_file("lokal/download.gif", "screenshot_kunde.gif", "kunden03");  
// lädt die Datei "screenshot_kunde.gif" als "lokal/download.gif" herunter
```

8.15.9. file_exists()

Ob eine Datei existiert, evaluieren Sie über die Methode "file_exists". Diese benötigt als Parameter den Dateinamen und den Pfad zum Verzeichnis ausgehend vom aktuellen Verzeichnis. Rückgabewerte sind, je nachdem ob die Datei vorhanden ist, true (1) oder false (0):

```
$status = $ftp->file_exists("ibiza2003.jpg", "urlaufsfotos");  
// überprüft, ob die Datei vorhanden ist
```

8.15.10. is_connected()

Mittels "is_connected" kann überprüft werden, ob eine vorher aufgebaute Verbindung zu einem FTP-Server noch offen ist. Rückgabewerte sind je nach Ergebnis true (1) oder false (0):

```
$status = $ftp->is_connected();  
// überprüft die aktuelle Verbindung
```

8.15.11. open()

Durch die Methode "open" wird eine Verbindung zu einem FTP-Server hergestellt. Hierzu ist als Parameter vor allem der Host des FTP-Servers erforderlich. Da einige FTP-Server eine Authentifizierung erfordern oder unter einem anderen Port laufen, können als weitere Parameter optional Benutzername, Passwort und Port übergeben werden (in dieser Reihenfolge). Werden diese Parameter nicht übergeben, dann verwendet "open" Standardwerte (Port 21 und Benutzername "anonymous").

Sobald die Methode alle notwendigen Werte erhalten hat, wird versucht eine Verbindung aufzubauen. Rückgabewerte sind je nach Ergebnis true (1) oder false (0):

```
$ftp->open("ftp.microsoft.com");  
// stellt die Verbindung zu "ftp.microsoft.com" her
```

8.15.12. rename()

Die Methode "rename" ermöglicht die Umbenennung von Dateien und Ordnern. Hierfür muss man sich bereits im richtigen Verzeichnis befinden. Der Methode werden dann als Parameter zuerst der alte und dann der neue Name der Datei / des Verzeichnisses übergeben. Rückgabewerte sind je nach Ergebnis true (1) oder false (0):

```
$ftp->rename("ibiza2003.jpg", "newzealand2003.jpg");  
// gibt Datei wird in "newzealand2003.jpg" umbenannt
```

8.15.13. show()

Um die Methode "show" zu nutzen, muss vorher bereits eine Verbindung mittels "open" hergestellt wurden sein. Dann kann "show" ein Pfad übergeben werden. Zurückgeliefert wird im Erfolgsfall ein Array mit allen Dateien und Ordnern in dieser Ebene (jeweils pro Eintrag - sofern vorhanden - Name, Größe, letzte Änderung und Typ). Wenn in den Ordner nicht gewechselt werden kann (z.B. wenn die Rechte dafür nicht ausreichend sind), wird von der Methode false (0) zurückgegeben. Wenn der Ordner hingegen leer ist, ist der Rückgabewert -1.

```
$status = $ftp->show("fussball/bilder");  
// liefert Array zu "fussball/bilder"
```

8.15.14. upload_file()

Die Methode "upload_file" geht genau den entgegengesetzten Weg zu "download_file". Mit ihr ist es möglich eine Datei auf den Server zu verschieben. Die Parameter und Rückgabewerte bleiben die gleichen wie bei "download_file".

```
$ftp->upload_file("lokal/download.gif", "screenshot_kunde.gif", "kunden03");  
// lädt die Datei "lokal/download.gif" als "screenshot_kunde.gif" hoch
```

9. FAQ - Häufig gestellte Fragen

Hier haben wir Antworten auf Fragen zusammengetragen, welche uns von unseren Kunden häufig gestellt werden. Diese Fragen können sowohl technischer, als auch gestalterischer oder inhaltlicher Natur sein. Sollten Sie eine weitere Frage zu diesem System haben, so kontaktieren Sie uns über den Kundenservice. Mit jedem Update des Systems wird die FAQ dann um weitere hilfreiche Antworten für die wichtigsten Fragen ergänzt.

9.1.0. Es treten beim Absenden von Formularen Fehler auf. Es wird immer ein Download-Fenster geöffnet!

vanillaXML Webshop nutzt für die schnellere Auslieferung der Seiteninhalte eine serverseitige Komprimierung /

Verschlüsselung auf der Basis des GZIP-Algorithmus. Es wird vom System automatisch geprüft, ob dieser von Ihrem Webserver / Browser unterstützt wird. Je nachdem ob dies der Fall ist, wird die GZIP-Komprimierung aktiviert oder nicht. Allerdings ist in einigen Fällen eine automatische Überprüfung von Webserver und (seltener) Browser nicht möglich.

U.a. zeigt der Microsoft Internet Explorer bei einigen Versionen > 4.0 teilweise ein fehlerhaftes Verhalten. Dieser fordert Inhalte per HTTP 1.0 an. Die Entschlüsselung von GZIP-Inhalten koppelt er jedoch an HTTP 1.1. Erhält der Internet Explorer dann GZIP-Inhalte als HTTP 1.0 (wie dies standardmäßig vom Webserver geliefert wird), ignoriert er den angegebenen "Content-Type" (GZIP). Da ein passender "Content-Type" nicht feststellbar zu sein scheint, erklärt dieser Browser die GZIP-Inhalte als Download und zeigt diese nicht direkt im Browserfenster an. Alle anderen Browser reagieren hier wie in den Standards vorgegeben.

Diesem Verhalten des Internet Explorers kann manuell Abhilfe geschaffen werden, indem in den Einstellungen des Browsers unter "Extras > Internetoptionen > Erweitert" die Optionen "HTTP 1.1 verwenden" und "HTTP 1.1 über Proxyverbindungen verwenden" aktiviert werden.

Um die Komprimierung allgemein zu deaktivieren, dann befolgen Sie bitte den folgenden Weg: Öffnen Sie die Datei "config.php" über die Dateiverwaltung (bzw. lokal in einem beliebigen Texteditor) und setzen Sie die Zahl hinter "\$block_local_encoding" auf "1". Dies erhöht zwar die Ladezeiten ein wenig, Sie können jedoch jetzt Formulare usw. im vollen Umfang nutzen.

9.1.1. Ich habe die Verschlüsselung per SHA1-Algorithmus aktiviert, jetzt treten Probleme auf!

Wenn Sie die Verschlüsselung vom MD5- auf den SHA1-Algorithmus umgestellt haben und es zu Problemen kommt (bspw. da die Passwörter der einzelnen Benutzer noch in MD5-Verschlüsselung in der Datenbank liegen), müssen Sie dies manuell rückgängig machen. Öffnen Sie hierzu per FTP die Datei "config.db.php" im Hauptverzeichnis von vanillaXML Webshop. Dort finden Sie diese Zeile:

```
$configuration["main"]["sha1_support"] = "1";
```

Deaktivieren Sie SHA1, indem Sie diese abändern auf:

```
$configuration["main"]["sha1_support"] = "0";
```

Anschließend loggen Sie sich wieder als Administrator ein und setzen Sie die SHA1-Verschlüsselung über "Einstellungen > Allgemeine Einstellungen" wieder zurück.

9.1.2. Ich habe mein Passwort verloren, wie kann ich es zurückerhalten?

In diesem Fall gibt es zwei Möglichkeiten. Da das Passwort verschlüsselt ist, kann ihr originales Passwort nicht mehr zurückgewonnen werden. Der Mitglieder-Bereich bietet Ihnen jedoch eine Passwort-Recovery-Funktion an (bei Klick auf "Passwort vergessen"), welche Ihnen ein neues Passwort zuteilt. Geben Sie hierfür Ihren Benutzernamen und die Email-Adresse an, mit welcher Sie sich registriert haben. Ein neues Passwort wird Ihnen per Email mitgeteilt. Alternativ dazu können Sie - sofern Sie vollen Datenbankzugriff besitzen - mit einem entsprechenden Tool (z.B. phpMyAdmin) in der Datenbanktabelle "*" _user" ("*" ist durch Prefix und Sprache zu ersetzen) Ihren Benutzer-Eintrag ändern. Suchen Sie Ihren Benutzernamen und ändern Sie das Passwort auf "21232f297a57a5a743894a0e4a801fc3".

Loggen Sie sich danach im Mitgliederbereich mit Ihrem Benutzernamen und dem Passwort "admin" ein. Bitte ändern Sie nach Möglichkeit dieses Passwort dann sofort in Ihren Einstellungen.

9.1.3. Ist die Installation von Software / Plugins auf dem lokalen Rechner notwendig?

Dies ist nicht notwendig. vanillaXML Webshop wird allein über den Browser bedient, wofür Sie eine möglichst aktuelle Version verwenden sollten. Plugins sind nur notwendig, wenn die zu erstellende Website solche voraussetzt (z.B. das Flash-Plugin von Macromedia). Sie entscheiden also selbst, wie Sie die Vorlagen gestalten und mit welchen Technologien Sie diese ausstatten. Von der such- und behindertengerechten Portalseite bis hin zu einer aufwändigen, mit Flash angereicherten Unternehmenspräsentation ist alles möglich.

9.1.4. Ist die Verwaltung großer Dateien (> 1 MB) möglich?

vanillaXML Webshop greift bei der Dateiverwaltung auf die von Ihrem Webhoster gemachten Einstellungen für die Datenbank bzw. Ihren Webspace zurück. Allgemein wird der Upload ab einer bestimmten Dateigröße (ca. 4

MB) begrenzt. Auch Datenbanken sind in ihrer Größe oft limitiert (jedoch meist erst ab mehreren Hundert MB). Informieren Sie sich bei Ihrem Webhoster, wie dessen Einschränkungen aussehen. vanillaXML Webshop wird sich daran orientieren.

9.1.5. Ist ein Einsatz im Intranet oder Extranet möglich?

vanillaXML Webshop kann auch im Intranet oder Extranet von Unternehmen oder Organisationen eingerichtet werden. Wichtig ist jedoch, dass auch hier die allgemeinen Systemvoraussetzungen erfüllt werden. Informieren Sie sich auch über zusätzlich angebotene Module aus den Bereichen HR und CRM, welche sich für Installationen im Intranet besonders gut eignen.

9.1.6. Kann ich meine Seite gezielt für Suchmaschinen optimieren?

Ja, vanillaXML Webshop bietet Ihnen standardmäßig eine Reihe von Mitteln zur Optimierung der Seite für Suchmaschinen an. Die wichtigsten seien hier kurz genannt:

- zuallererst sollten Sie Ihre Vorlagen auf Validität überprüfen lassen durch den Validator des W3C, denn korrekt geschriebene Seiten können von Suchmaschinen leichter erfasst und verarbeitet werden
- für jeden Artikel sollten Sie die Metatags (unter "Artikel bearbeiten") möglichst präzise ausfüllen, auch die allgemeinen Metatags (unter "Einstellungen > Metatags") sind bedeutsam (vor allem Titel und Schlüsselwörter)
- wenn Ihr Server "mod_rewrite" unterstützt, sollten Sie die suchmaschinenfreundlichen URL's unter "Einstellungen > Allgemeine Einstellungen aktivieren"

Da das Aufgabengebiet der Suchmaschinenoptimierung mittlerweile sehr umfangreich geworden ist, können wir Ihnen hier nur einige Tipps geben. Doch wenn Sie allein diese Tipps einhalten, wird Ihre Seite erheblich suchmaschinenfreundlicher werden und kann somit - einige Zeit nachdem die Suchmaschinen-Spider Sie besucht haben - in den Suchmaschinen besser gefunden werden.

Tipp: Testen Sie mit dem "Poodle Predictor", wie Ihre Seite später in Google erscheinen wird!

Die sandoba.de medien agentur bietet selbst Suchmaschinenoptimierungen für Seiten an, die mit vanillaXML Webshop oder auch statisch umgesetzt worden. Diese Dienstleistungen umfassen u.a. die Optimierung für bestimmte Schlüsselwörter, Anpassung des Quellcodes der Vorlagen, Erfolgskontrolle usw. Sprechen Sie mit uns!

9.1.7. Können {cp:tags} in PHP-Dateien verarbeitet werden?

Standardmäßig werden {cp:tags} die in der Ausgabe einer über {cp:include} eingebundenen PHP-Datei enthalten sind aus Performancegründen nicht verarbeitet. Dies können Sie jedoch leicht manuell aktivieren. Öffnen Sie hierzu die Datei "functions.php" und fügen Sie in der Funktion "cp_include" jeweils vor "RETURN \$string;" folgenden Quellcode ein:

```
include_once("functions/cp.template.php");  
$string = cp_template_output(cp_template_process($string), "1", "0");
```

Dann wird vor der endgültigen Ausgabe der gelieferte Inhalt überprüft, alle {cp:tags} werden verarbeitet und deren Rückgaben werden wie gewohnt anstelle des {cp:tags} integriert.

9.1.8. Nach dem Ändern von Einstellungen werden diese nicht aktualisiert. Warum?

Dies deutet darauf hin, dass der CHMOD bei der Installation nicht korrekt gesetzt wurde. Insbesondere werden für den Hauptordner und die Datei "config.db.php" ein CHMOD von 775 oder 777 (je nach Webspace) benötigt. In der Datei "config.db.php" werden alle Einstellungen des Systems zur Verringerung der Datenbankabfragen nach Änderungen im Administrationsbereich zwischengespeichert.

Ist ein Speichern jedoch nicht möglich, wird zwar die Datenbank aktualisiert, in den Formularen etc. erscheinen jedoch noch die alten Angaben. Ändern Sie daher den CHMOD für die Datei "config.db.php".

9.1.9. Sind fertige Vorlagen-Sets verfügbar?

Jeder Lizenz von vanillaXML Webshop liegt mindestens ein fertiges Vorlagen-Set bei. Diese dienen als Demonstration der Funktionen von vanillaXML Webshop und zum besseren Verständnis der Arbeitsweise. Weitere Vorlagen-Sets können Sie bei uns oder kooperierenden Agenturen bestellen. Die Vorlagen-Sets sind stets in einem Unterordner des Ordners "themes" ausgehend vom vanillaXML Webshop-Hauptordner gespeichert.

9.1.10. Sind zusätzliche Schulungen erforderlich?

Ob ein Schulungsbedarf besteht, ist unterschiedlich von der Art der Nutzung von vanillaXML Webshop. Redakteure arbeiten komplett innerhalb der Benutzeroberfläche im Browser und können vanillaXML Webshop innerhalb nur weniger Tage verstehen und bedienen. Für Entwickler gibt es zusätzliche Support-Möglichkeiten wie die integrierte Online-Hilfe, Support per Email oder im Forum sowie unsere Artikel in der [Knowledge Base](#).

9.1.11. Warum werden nicht alle Inhalte nach dem Bearbeiten aktualisiert?

Wenn nach dem Bearbeiten von Inhalten (bspw. einem Artikel) diese nicht sofort auf der Website aktualisiert werden, so kann dies an den Einstellungen des Caches von vanillaXML Webshop liegen. Soll der jeweilige Inhalte dennoch sofort und damit vor Ablauf der Cache-Zeit in der neuesten Fassung online angezeigt werden, so haben Sie unter "Einstellungen > Cache" die Möglichkeit den Cache zurückzusetzen.

9.1.12. Welche Aufgabe hat die Datei ".htaccess"?

Die Datei ".htaccess" im Hauptordner beinhaltet einige Einstellungen, die den Webserver in seinem Verhalten beeinflussen. Eine wichtige Rolle spielt diese Datei bei den "Suchmaschinenfreundlichen URL's" (mittels "mod_rewrite"). Weiterhin wird der Zugriff auf Dateien mit den Endungen "cfg", "class", "inc", "mod", "lng", "tmp" und "tpl" eingeschränkt und als allgemeine Fehlerseite wird schließlich noch die "index.php" festgelegt.

9.1.13. Welche Limitierungen bei der Erstellung des Layouts gibt es?

Dies ist einfach zu beantworten - Keine! Sie sind vollkommen frei bei der Erstellung des Layouts und nicht an bestimmte Schemata gebunden. Das Layout einer Website wird aus verschiedenen Vorlagen zusammengesetzt. Diese Vorlagen sind reine HTML-Dateien, erweitert um die {cp:tags}. Somit können Sie wie gewohnt Ihre Website erstellen und dabei alle aus HTML bekannten Funktionen nutzen.

Beim Aufbau Ihrer Seite haben Sie im wesentlichen folgende Möglichkeiten:

- wenn Sie viele unterschiedliche Layouts verwenden, legen Sie einzelne HTML-Vorlagen mit den gewünschten {cp:tags} an und rufen Sie diese über "index.php?static=[PFAD ZUR DATEI]" auf
- Sie gehen vom Standard-Vorlagenset aus und erstellen auf dieser Basis mit den mindestens benötigten Vorlagen Ihr eigenes Layout, welches jederzeit erweitert werden kann
- Sie nutzen XSLT-Stylesheets auf der Basis der Webservices zur Darstellung

Den gängigen Weg stellt dabei die zweite Möglichkeit dar. Die beiden anderen sind nur zu empfehlen, wenn Sie bereits über umfangreichere Erfahrungen mit vanillaXML Webshop verfügen und besonders differenzierte Seiten gestalten müssen.

9.1.14. Wie gehe ich beim Update auf eine neue vanillaXML Webshop-Version vor?

Jedem Update von vanillaXML Webshop liegt eine ausführliche Anleitung mit Anweisungen bei, die den Installationsprozess genau beschrieben. Abhängig vom Umfang des Updates kann es sich hierbei um Datenbankänderungen, Änderungen an Dateien oder Ordnerstrukturen handeln. Sofern dies möglich ist, wird mit dem Update auch eine Installationsdatei mitgeliefert, welche einen Großteil der Änderungen automatisch durchführt. Benötigen Sie Hilfe bei der Installation eines Updates, setzen Sie sich bitte mit dem Support in Kontakt.

9.1.15. Wie kann ich die Fehlerseiten ändern?

vanillaXML Webshop wechselt bei einer nicht vorhandenen Datei automatisch zurück auf die Datei "index.php", also die Startseite der Website. Dieser Wechsel wird vollzogen bei folgenden Fehlercodes:

- *403 Forbidden* - der Server verweigert die Ausführung
- *404 Not Found* - der Server kann die Methode nicht ausführen, weil die Seite nicht gefunden wurde
- *500 Internal Server Error* - beim Server gab es einen internen Fehler

In der Datei ".htaccess" im Hauptverzeichnis des Systems wird für jeden dieser Fehlercodes die entsprechende Alternativ-Seite festgelegt. Dort können Sie beispielsweise die URL zu einem Artikel integrieren, in welchem Sie dem Besucher einen Hinweis auf die nicht vorhandene Seite geben und zugleich eine Sitemap oder Suche anbieten.

9.1.16. Wie sieht es mit der Sicherheit aus?

Alle sensitiven Daten der Mitglieder, Redakteure und Administratoren sind durch Benutzername und Passwort gesichert. Das Passwort wird zusätzlich mittels des MD5-Algorithmus verschlüsselt und liegt auch in der Datenbank nicht in entschlüsselter Form vor. Eine für jeden Benutzer spezifisch erzeugte Session erlaubt den Schutz von Daten. Der MD5-Algorithmus ermöglicht zudem höchste Sicherheit, bisher ist eine Entschlüsselung nicht möglich. Alternativ können Sie zur Verschlüsselung auch den SHA1-Algorithmus verwenden (weniger bekannt, Umstellung unter "Einstellungen > Allgemeine Einstellungen"). Für vergessene Passwörter bietet vanillaXML Webshop eine Passwort-Recovery-Funktion per Email an. Hierfür wird ein neues Passwort generiert.

9.1.17. Wie werden eigene Vorlagen-Sets erstellt?

Die Anpassung des Layouts geht in wenigen Schritten. Kopieren Sie einen der vorhandenen Vorlagen-Ordner und passen Sie die Dateien entweder mit dem integrierten WYSIWYG-Editor oder einem externen Programm an. Dazu gehören auch die Grafiken im Unterordner "images". In die HTML-Vorlagen sind die im Handbuch beschriebenen {cp:tags} einzubetten. An diesen Stellen wird später der Inhalt eingesetzt. Beim Aufbau des Layouts sind Sie vollkommen frei.

Wenn Sie mit der Anpassung des Layouts fertig sind oder Sie bereits vorab einen Blick auf die Seiten werfen wollen, dann können Sie im Modul "Vorlagen" das aktuelle Vorlagenset festlegen.

9.2. Installation / Systemanforderungen

9.2.1. Bei der Installation erscheint "Warning: session_start()...", warum?

Wenn bei Ihnen beim Aufruf der Installationsdatei "install.php" folgende Fehlermeldung erscheint, dann müssen Sie vor der Installation die Zugriffsrechte für einzelne Ordner (CHMOD) per FTP setzen:

Warning: session_start(): open(/sessions/sess_di6fdsaf, O_RDWR) failed: Permission denied (13) in /is/htdocs/..

Warning: session_start(): Cannot send session cookie - headers already sent out...

Dies ist notwendig, damit vanillaXML Webshop temporäre Dateien im Session- und Cache-Verzeichnis anlegen kann und Sie die Dateiverwaltung im vollen Umfang nutzen können. Wie Sie den CHMOD ändern und welche Ordner und Dateien davon genau betroffen sind, erfahren Sie in der Installationsanleitung.

9.2.2. Bei der Installation funktioniert mein "Hostname" nicht!

Einige Webhoster nutzen für FTP und MySQL verschiedene Hostnamen / Server. Fragen Sie bei Ihrem Webhoster explizit nach dem Hostnamen für die MySQL-Datenbank für Ihre Domain. Da viele Webhoster den Kunden direkt nach der Freischaltung die Zugangsdaten für die MySQL-Datenbank per Email übermitteln, sollten Sie auch dort die benötigten Angaben finden. Alternativ dazu können Sie meist als Hostnamen "localhost" nutzen, wenn die Datenbank auf dem gleichen Server wie Ihr Webspace liegt.

9.2.3. Die Installation wird unterbrochen / ist nicht möglich. Was ist zu tun?

Für jeden Prozess der auf einem Webserver läuft ist allgemein eine maximale Zeit festgelegt, innerhalb der

dieser Prozess abgeschlossen werden muss. Bei der Installation ist dies z.B. das Einrichten der benötigten Datenbankstruktur mit zahlreichen Inhalten wie der Anleitung etc.

Allerdings gibt es bei einigen Webhostern sehr restriktive Zeiteinschränkungen, die es unter Umständen unmöglich machen, die Installation komplett durchzuführen. Dies äussert sich dann in entsprechenden Fehlermeldungen bei der Installation.

Wenn dies bei Ihnen der Fall sein sollte, gehen Sie bitte folgendermaßen vor:

- öffnen Sie die Datei "admin_install.sql" aus dem Hauptverzeichnis in einem beliebigen Text-Editor
- ersetzen Sie dann per "Suchen-und-Ersetzen"-Dialog überall "van_" durch den bei der Installation gewählten Präfix sowie "_" (d.h. bei "cp" als Präfix wird "cp_" eingesetzt)
- anschließend laden Sie über ein Tool wie phpMyAdmin jeweils ca. 1000 Zeilen aus dieser Datei in die Datenbank
- zuletzt müssen Sie noch die Datei "admin_install.sql" leeren

Nachdem nun die Datenbank korrekt mit den Tabellen und Inhalten gefüllt ist, kann die Installation normal fortgesetzt werden, indem Sie einen Administrator für das System einrichten (siehe auch Installationsanleitung).

9.2.4. Ich hoste bei 1&1 / Alturo / Schlund & Partner. Wie kann ich das System nutzen?

Um das System bei 1&1, Alturo oder Schlund & Partner (alles Unternehmen von United Internet) zu nutzen, beachten Sie bitte die folgenden Punkte. Diese passen das System-Verhalten an die spezifischen Erfordernisse der Server an:

1. In der Datei "config.php" muss die Variable \$block_local_encoding auf "1" gestellt werden (siehe auch in der FAQ "Es treten beim Absenden von Formularen Fehler auf").
2. In der Datei "functions.php" in Zeile 300 und 302 die beiden header-Aufrufe auskommentieren ("// ...").
3. In der Datei "index.php" den header-Aufruf in Zeile 35 auskommentieren.

9.2.5. Kann vanillaXML Webshop auf Webspaces mit aktiviertem "safe_mode" genutzt werden?

Dies ist leider nicht möglich. Die Einstellung "safe_mode" in der PHP-Konfiguration unterbindet den Einsatz verschiedenster Funktionen und schränkt vor allem die Möglichkeiten zur Arbeit mit Dateien (Anlegen, Löschen, Upload neuer Dateien). Somit wäre die Leistungsfähigkeit von vanillaXML Webshop nicht voll nutzbar. "safe_mode" muss also auf Ihrem Webspaces deaktiviert sein.

Einige Webhoster deaktivieren den "safe_mode" auf Wunsch für den Webspaces (da der "safe_mode" jeweils lokal und global eingestellt werden kann). Alternativ halten wir eine Liste mit getesteten Webhostern ("[Webhoster im Test](#)") für Sie bereit.

9.2.6. Was muss ich beim Umzug auf einen anderen Server beachten?

Wenn Sie auf einen anderen Server umziehen wollen, lesen Sie bitte den Punkt "Wie wird bei einer Deinstallation vorgegangen?" um zu erfahren, wie Sie ein lokales Backup der Datenbank erstellen und anschließend die Datenbank löschen. Bitte beachten Sie, dass Sie Dateien die Sie selbst hinzugefügt haben (wie z.B. Grafiken, PDF-Dateien etc.) vorher per FTP sichern müssen.

Nachdem vanillaXML Webshop dann deinstalliert ist, wenden Sie sich per Email an uns und lassen Ihre Registrierung wieder für die Installation freischalten. Auf dem neuen Server nehmen Sie dann die Installation erneut vor, die Datei "admin_install.sql" tauschen Sie nun jedoch mit der Backup-Datei aus. Anschließend laden Sie noch die gesicherten Dateien an die vorherigen Positionen im Dateisystem und schon ist der Umzug abgeschlossen.

9.2.7. Welche Bedeutung hat die PHP-Einstellung "open_basedir"?

Die PHP-Einstellung "open_basedir" beschränkt den Zugriff von PHP-Dateien auf die darunterliegenden Ordnern und Dateien. Bei einem Zugriff - wie z.B. dem Öffnen einer Datei - wird überprüft, wo diese liegt und ob entsprechende Berechtigungen vorhanden sind. Ist dies nicht der Fall, wird eine Fehlermeldung wie diese zurückgegeben:

Warning: open_basedir restriction in effect. File is in wrong directory in /htdocs/files/script.php on line 5

vanillaXML Webshop ist so konfiguriert, dass möglichst wenig Operationen durchgeführt werden müssen, die ein deaktiviertes "open_basedir" erfordern. Jedoch lässt sich dies nicht immer vermeiden, wenn ein Cache oder

die Überprüfung externer URL's funktionsfähig sein sollen.

Um "open_basedir" manuell zu deaktivieren, müssen Sie die Datei "php.ini" öffnen (wo diese zu finden ist, ist abhängig vom Betriebssystem). Dort suchen Sie nach "open_basedir" und setzen dies auf "0" oder "off".

9.2.8. Welches Betriebssystem wird von vanillaXML Webshop genutzt?

Da für vanillaXML Webshop keine zusätzliche Software auf dem Server oder einem Client installiert werden muss, ist der Betrieb von vanillaXML Webshop prinzipiell auf jedem Betriebssystem möglich, welches als Webserver Apache oder IIS unterstützt (z.B. Linux, Unix, Windows).

9.3. Lizenzbedingungen

9.3.1. Darf vanillaXML Webshop zur Entwicklung auf einer anderen Seite installiert werden?

Wenn Sie vanillaXML Webshop vor der Fertigstellung Ihres Projektes zuerst auf einer anderen Seite (d.h. unter einer anderen URL) installieren, so müssen Sie sich einmalig registrieren. Sobald die Entwicklung abgeschlossen ist und vanillaXML Webshop auf die richtige Seite geladen werden soll, teilen Sie uns dies bitte per Email mit (über den Kundenservice). Die vanillaXML Webshop-Registrierung wird dann geändert. Die vorherige vanillaXML Webshop-Installation muss dann natürlich gelöscht werden.

9.3.2. Können verschiedene Websites mit einer vanillaXML Webshop-Lizenz administriert werden?

Dies ist abhängig von Ihrer Lizenz. Besitzen Sie die Lizenz für die Grundversion von vanillaXML Webshop ist die Verwaltung von verschiedenen Websites nicht möglich. Sie können nur eine Website administrieren, welche jedoch über verschiedene Domains erreichbar sein darf. Für Agenturen und Internet Service Provider können alternative Lösungen bereitgestellt werden, welche die zentrale Administration eine größeren Anzahl von Websites erlauben. Beachten Sie in diesem Zusammenhang bitte auch die Systemanforderungen.

9.4. Vorlagen / Entwicklung

9.4.1. Kann ich meine PHP-Skripte weiterverwenden?

Ja, dies ist möglich. Sie haben verschiedene Möglichkeiten in vanillaXML Webshop, Ihre Skripte weiterzuverwenden. U.a. können Sie diese über einen Link aus vanillaXML Webshop aufrufen, diese in eine Seite inkludieren ("include", einbinden) lassen über den Tag {cp:include} oder diese direkt in einen Artikel oder eine Vorlage integrieren. Wenn Sie PHP-Skripte in Artikeln oder in Vorlagen einbinden, welche für einen Container genutzt werden, wird zusätzlich (optional) die Ausgabe des Skripts zwischengespeichert. Dies führt zu geringeren Ladezeiten.

9.4.2. Können Sie mir eine Entwicklungsumgebung empfehlen?

Wenn Sie vanillaXML Webshop um eigene Module / Funktionen erweitern wollen, dann können Sie neue PHP-Dateien prinzipiell in jedem einfachen Texteditor (Notepad, Word) verfassen. Dennoch gibt es speziell für den PHP-Bereich leistungsstarke Editoren, die den Benutzer bei der Arbeit durch Funktionsübersichten, eine farbige Quellcode-Darstellung usw. unterstützen.

Einige besonders empfehlenswerte Editoren sind:

- TextPad - www.textpad.com
- PHP Eclipse - phpeclipse.sourceforge.net
- PHP Edit - www.phpedit.net
- PHP Coder - www.phpide.de

Die hier genannten Entwicklungsumgebungen sind zum Teil kostenlos. Einige sind als Shareware erhältlich und müssen nach der Testphase registriert werden, jedoch ist ein Kauf sehr zu empfehlen (aufgrund der teils erheblichen Zeitersparnis).

9.4.3. Variablenübergabe und -verarbeitung

Durch die Möglichkeit, PHP und die {cp:tags} in beliebigen Bereichen einzeln oder auch in Kombination zu verwenden, entstehen ganz neue Wege der Erstellung von neuen Modulen und Interaktionen.

Um beispielsweise bestimmte Inhalte in Abhängigkeit eines Parameters anzuzeigen (d.h. Parameter wie z.B. "action"), wird dieser beim Aufruf der URL an diese angehängt:

Statt `*/index.php?id=300` z.B. `*/index.php?id=300&full_content=yes`

Ob dieser Parameter gesetzt ist, können Sie in dem Artikel - bzw. sonst in der Vorlage - mit dem {cp:tag} "{cp:if:full_content}" überprüfen oder Sie setzen PHP ein und schreiben folgendes:

```
<?php
GLOBAL $full_content;
if ($full_content == "yes") {

// hier der Inhalt, welcher angezeigt werden soll

}
?>
```

Ein Beispiel für eine sinnvolle Verwendung der Variablenübergabe sehen Sie in unserem Lexikon unter <http://www.sandoba.de/article-135-computer-lexikon.html>. Dort ruft ein Artikel (quasi der "Rahmen" des Lexikons) in Abhängigkeit von den übergebenen Variablen die passenden Artikel der Kategorie "Lexikon" auf.

9.4.4. Warum können Dateien / Vorlagen nicht bearbeitet werden?

vanillaXML Webshop ist bei der Bearbeitung von Dateien immer auf ausreichende Zugriffsrechte angewiesen, d.h. es müssen für die jeweiligen Dateien / Ordner Lese- und Schreibrechte vorhanden sein. Sind diese Rechte nicht verfügbar, dann versucht vanillaXML Webshop diese automatisch zu setzen. Je nach genutztem Webspaces-Anbieter ist dies allerdings nicht immer möglich. In diesem Fall müssen Sie dann per FTP den CHMOD auf die Dateien / Ordner manuell ändern (auf 777 bzw. 775). Dann kann die Bearbeitung über vanillaXML Webshop erfolgen.

9.4.5. Was ist mit Änderungen von vanillaXML Webshop-Dateien?

Sie haben generell die Möglichkeit, vanillaXML Webshop-Dateien an Ihre Bedürfnisse anzupassen. Da vanillaXML Webshop in PHP geschrieben ist, liegt der Quellcode für Veränderungen offen. Änderungen an diesen Dateien unterliegen jedoch den Lizenzbestimmungen. Es kann keine Gewährleistung auf die Funktionalität des Systems übernommen werden, wenn vanillaXML Webshop-Dateien von Ihnen bearbeitet werden. Änderungen an Funktionen oder bspw. die Erstellung von neuen Modulen nach Ihren Wünschen können auch direkt durch die sandoba.de medien agentur vorgenommen werden. Sie profitieren doppelt: durch das Wissen der Entwickler und die gesparte Entwicklungszeit. Kontaktieren Sie uns und wir erstellen Ihnen ein unverbindliches Angebot.