Technische Dokumentation

Autor:
IPResearch
Industriestrasse 23
97437 Hassfurt
info@ipresearch.de

Inhaltsverzeichnis

1 Einführung

tricoma ist ein Content Management System (CMS) und Customer Relationship Management-System (CRM) der Zukunft. Es wurde speziell an die Bedürfnisse eines Unternehmens angepasst.

Von der einfachen Webseiten-, Kunden- und Produktverwaltung bis hin zur Echtzeitwebseitenstatistik über das Besucherverhalten ist vieles möglich.

Und sollten Ihnen die Funktionen einmal nicht reichen ist das auch kein Problem, denn gerne entwickeln wir für Sie ein persönliches Modul mit den von Ihnen gewünschten Funktionen.

2 Allgemeines

2.1 Allgemeine Ordnerstruktur von tricoma

Nun möchten wir mal über den strukturiellen Aufbau der Ordner des Systems eingehen.

Pfad	Beschreibung
/	Allgemeine Root Hier liegt die index.php welche die Webseitenverarbeitung erledigt.
/cmssystem	Hauptverzeichnis vom CMS In diesem Verzeichnis liegen die Standarddateien der Administrationsoberfläche von tricoma
/cmssystem/standard/	In diesem Ordner liegen alle Standardklassen und wichtige funktionsdateien zur Funktionalität des Systems. (Ähnlich einem Betriebssystemkern)
/cmssystem/GeneratedItems/	Hier sind alle für die Grafische Oberfläche wichtigen Dateien ausgelagert.
/cmssystem/*	In den Unterordnern von /cmssystem/ liegen die Ordner von den einzelnen Modulen
/templates/	Im Ordner /templates/ liegen alle Webseitentemplates
/GeneratedItems/	Dieser Ordner enthält alle zum Laden der Webseite notwendige Dateien und die Konfigurationsdatei. d. H. Templatemanagement, Modulmanagement In diesem Ordner liegt zudem die config.php in welcher die Datenbankzugangsdaten geändert werden können.

3 Templates

3.1 Erklärung der Templates (/templates/)

Bei Templates handelt es sich grundsätzlich um ein HTML Dokumente welche später in den Quellcode bei der Programmierung geladen werden.
Bei Tricoma wird strikt zwischen Programmiercode und Design getrennt. Das bedeutet für Sie, dass Sie das Design in so genannten Templates festlegen, tricoma diese einliest und dann mit Inhalt an den Benutzer ausgibt.
Somit können Sie sich auf Ihr Layout und Design konzentrieren und kommen dabei in keinster Weise mit Quellcode in Berührung.

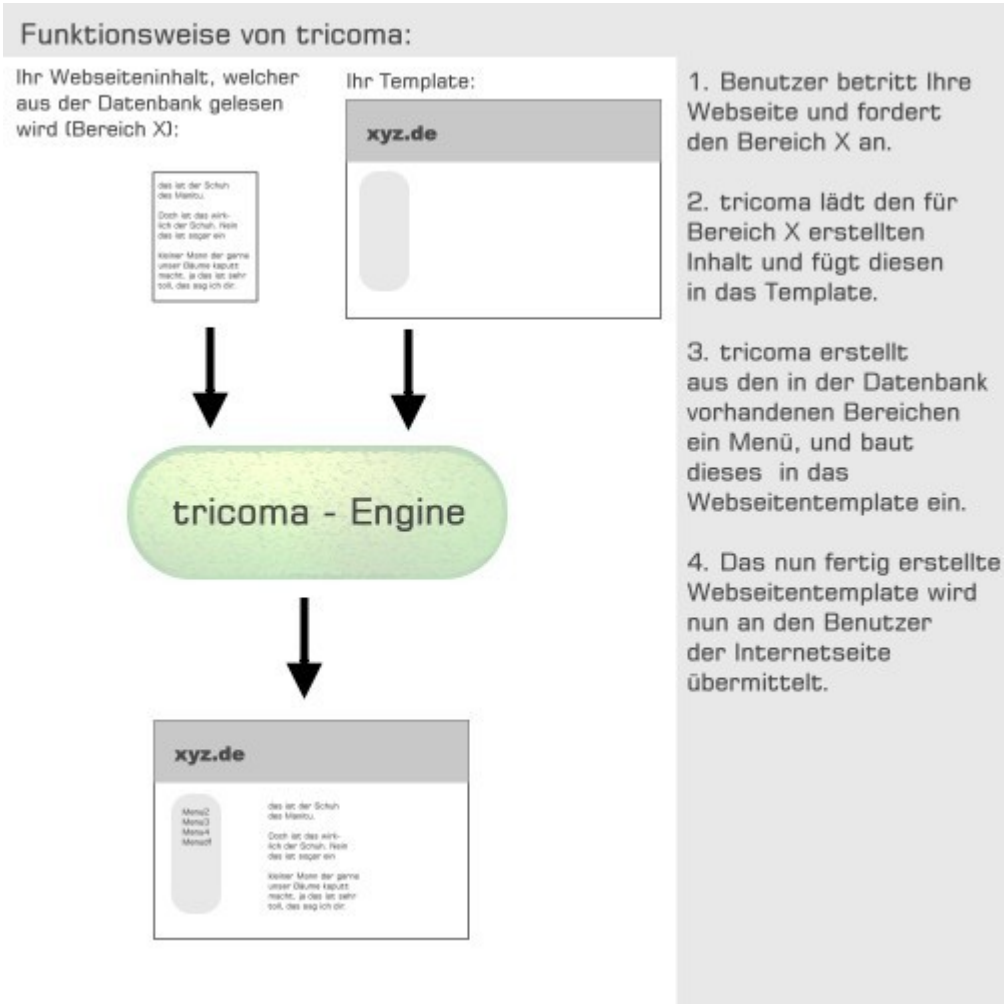


Abbildung 1: Templates

3.2 Templateordner

Templates liegen bei tricoma unter:
/templates/

Sie können in den Templateordner beliebig viele Templates laden. Diese werden durch eigene Ordner unterteilt, z. B. /templates/*templatename*/
Jede Webseite hat einen Templateordner.

Jedes Template bringt somit sein eigenes Webseitendesign und auch individuelle Programmierungen mit. Somit kann auch ein Template z. B. einen eigenen Warenkorb oder spezielle Navigationen beinhalten.

Hierzu gibt es im Templateordner folgende Unterordner:

/templates/*templatename*/style/ **Stylesheets**
Hier liegt Ihre style.css in welcher Sie Grunddesigns wie Schriftart und Farbe ändern können

/templates/*templatename*/js/ **Javascript Dateien**
Standardmäßig wird in diesem Ordner Lightbox mitgeliefert welche für Effekte beim laden von Bildern verantwortlich ist.

/templates/*templatename*/Labels/ **Grafikdateien**
In diesem Unterordner werden alle im Template verwendeten Grafiken ge

speichert (Logos, Hintergründe usw.)

/templates/*templatename*/templatemodule/ **Templatemodule**

Im Templatemodulordner kann man eigene Programmierungen ablegen.

Beispiel:

menu.php Der Quellcode der menu.php wird beim Aufruf der Webseite in die index.html rein geladen und zwar an der Stelle, an der {menu} (Dateiname ohne .php) steht.
So kann man in der menu.php direkt mit PHP Quellcode arbeiten und eigene Menüstrukturen programmieren.

/templates/*templatename*/modext/ **Modulerweiterungen (ModExt)**

Manchmal kommt es vor, dass die Eigenschaften eines Moduls nicht 100%ig auf die Bedürfnisse des Kunden abgestimmt sind. Daher wurden sogenannte Modul Extensions (modext) ins Leben gerufen.
Es lassen sich somit die Eigenschaften in der modulloading.php erweitern und somit eigene Funktionen implementieren.

Beispiel:
Der Kunde möchte dass der Warenkorb einen ganz anderen Aufbau hat als der Standardwarenkorb welcher von tricoma mitgeliefert wird.

Name	Änderungsdatum	Typ	Größe
css	21.11.2008 11:25	Dateiordner	
js	21.11.2008 11:25	Dateiordner	
Labels	03.12.2008 17:22	Dateiordner	
modext	29.12.2008 12:53	Dateiordner	
templatemodule	27.11.2008 18:03	Dateiordner	
benutzergesperrt.html	30.03.2006 13:12	HTML-Dokument	1 KB
fehlerseite.html	18.11.2006 01:28	HTML-Dokument	2 KB
index.html	03.12.2008 17:23	HTML-Dokument	5 KB
index_logout.html	02.08.2007 10:34	HTML-Dokument	1 KB
nichteingeloggt.html	31.12.2006 14:17	HTML-Dokument	2 KB
offline.html	18.04.2006 22:36	HTML-Dokument	2 KB
page.html	30.10.2008 17:34	HTML-Dokument	1 KB
page_druckansicht.html	30.06.2007 10:23	HTML-Dokument	2 KB
seitenposition.html	19.04.2006 00:42	HTML-Dokument	1 KB
sitemap_bereiche.html	12.04.2006 20:24	HTML-Dokument	1 KB
sitemap_hauptbereich...	18.04.2006 11:55	HTML-Dokument	1 KB
sitemap_unterbereiche...	12.04.2006 20:24	HTML-Dokument	1 KB
sitemap_webseite.html	18.04.2006 12:00	HTML-Dokument	1 KB
standard_formmailer.h...	30.12.2006 19:07	HTML-Dokument	2 KB
standard_formmailer_e...	30.12.2006 19:07	HTML-Dokument	1 KB
standard_formmailer_f...	30.12.2006 19:08	HTML-Dokument	1 KB
standard_formmailer_s...	30.12.2006 19:09	HTML-Dokument	1 KB
standard_suchtags.html	29.02.2008 10:27	HTML-Dokument	1 KB
standard_suchtags_tag...	29.02.2008 10:19	HTML-Dokument	1 KB
suche.html	18.04.2006 10:33	HTML-Dokument	1 KB
suche_ergebnisse.html	19.04.2006 00:25	HTML-Dokument	1 KB
suche_keinergebnis.ht...	05.04.2006 15:46	HTML-Dokument	1 KB
suche_zukurzersuchbe...	05.04.2006 15:45	HTML-Dokument	1 KB

Abbildung 2: Beispiel eines Standardordners.

3.3 Modultemplates

Um die Webseite so individuell wie möglich gestalten zu können, muss natürlich auch jedes Modul seine eigenen Templates mitbringen.
Diese befinden sich beim jeweiligen Modul in folgenden Ordner
/cmssystem/*modulname*/templatesystem/

Wenn wir nun eine Webseite mit einem Template haben, in welchem die entsprechenden Modultemplates fehlen, werden diese aus dem Modulordner in das Template bei der ersten Nutzung (Webseitenaufruf mit jeweiligen eingebundenem Modul) kopiert.

Beispiel:
Webseite A bildet einen Onlineshop ab. Es wird allerdings das Standardtemplate zur Nutzung verwendet, dann werden beim ersten Aufruf von Produkten die entsprechenden Templates für die Produktdetailansichten in den Templateordner der Webseite A kopiert.

3.4 Globale Templatevariablen

Bei Tricoma wird strikt zwischen Programmiercode und Design getrennt. Das bedeutet für Sie, das Sie das Design in sogenannten Templates festlegen, tricoma diese einliest und dann mit Inhalt an den Benutzer ausgibt.
Somit können Sie sich auf Ihr Layout und Design konzentrieren und kommen dabei in keinster Weise mit Quellcode in Berührung.

Im System gibt es daher einige verdefinierte Templatevariablen für das Template **index.html**:

{seitenposition}	Position auf der Webseite: z. B. Home >> Willkommen >> Bereich
{webseitenurl}	aktuelle Domain aus z. B. http://demo.tricoma.de
{seitentitel}	aktuellen Webseitentitel aus z. B. tricoma.de - Bereich 1
{meta-keywords}	Keywordausgabe
{meta-autor}	Ausgabe vom Autor
{meta-publisher}	Publisher
{meta-copyright}	Copyright
{meta-pagetopic}	Inhalt der Webseite
{meta-pagetyp}	Webseitentyp
{meta-description}	Webseitenbeschreibung
{page}	Position zum laden der page.html
{ID}	Aktuelle ID

4 Module

4.1 Module (/cmssystem/modulordner/)

Einer der Hauptvorteile von tricoma ist der modulare Aufbau des Gesamtsystems. Das System hat eine Grundbasis uns lässt sich durch eigene Module, Templatemodule oder ModExt's erweitern.

Allgemein ist jedes Modul standardisiert. Das bedeutet für den Programmierer er muss sich beim anlegen von Dateien an gewisse Dateinamen halten.

Hier eine Erklärung der Standarddateien:
datachange.php Bietet eine Übersicht über alle Einträge (Nachrichtenübersicht)

zeigedaten.php	öffnet einen Eintrag und bietet die Möglichkeit zur Bearbeitung. (z. B. bearbeiten einer Nachricht)
einstellungen.php	Verwaltung von Bereichen und definieren von Einstellungen (z. B. Nachrichtenbereiche)
eintragen.php	Dient zum Erstellen neuer Einträge

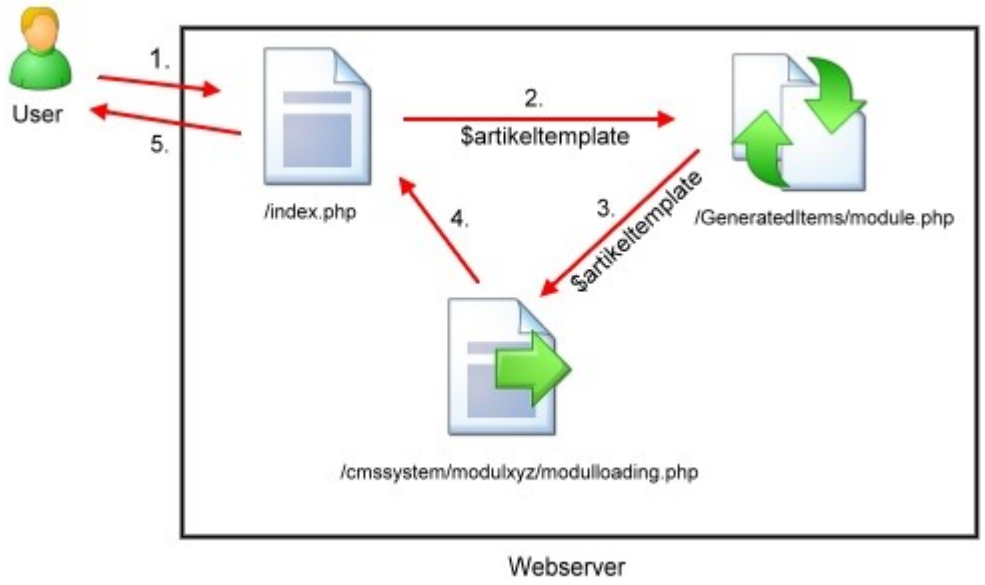
Im Modulordner gibt es reservierte Dateinamen, welche für externe Modulaufrufe belegt sind:

modulloading.php	Dient zum laden der externen Anzeige für die Webseite. Funktionsweise durch Templates
modulauswahl.php	Über diese Datei kann man die Eigenschaften für die Einbindung in die Webseite ausgeben. Wird im Modul Bereiche als "Zusatzmodul" ausgegeben.
modulext_*.php	Durch diese Datei lässt sich das Modul von extern auf der Seite verfügbar machen. z. B. zum Öffnen von Downloads oder Bildern. Ausgabe läuft extern über die /modul.php.
light.php	Dieser Dateiname ist für den Aufruf über das Tricoma!Light System reserviert. (Befindet sich noch in Teststadium)
cronjob.php	Diese Datei wird alle X Minuten durch die /cmssystem/cronjob.php aufgerufen, sofern ein minütlicher Cronjob auf diese Datei gesetzt wurde. Die Einstellung wie oft die Datei aufgerufen werden soll kann unter dem Menüpunkt *Administration --> Erweitert --> Crontab* geändert werden
icon.jpg	Diese Datei ist das Icon für das jeweilige Modul was z. B. im Menü angezeigt wird. (32x32px)
icon.png	Wie icon.jpg nur als PNG Datei mit Transparanz Effekt
icon_gross.jpg	Modulicon in der Größe 128x128px
meinedaten_einstellungen.php	Datei zum vornehmen von benutzerspezifischen Einstellungen. Beispiel: Standardnachrichtenkategorie
kundenauswahl.php	Diese Datei wird benötigt um im Kundenmodul externe Modul einzubinden. Diese sind dann für jeden Benutzer in den Einstellungen des Kundenmoduls mit Prioritäten belegbar
api.php	Dient um einen API Zugriff auf das Modul zu ermöglichen.

4.2 Moduldatei modulloading.php

Die modulloading.php ist die wichtigste Datei um Inhalte aus dem CMS auf der Webseite anzuzeigen.

Funktionsweise
Die Funktionsweise dieser Datei möchten wir kurz an diesem Beispiel zeigen:



1. Der Webseitenuser stellt eine Anfrage an den Webserver
2. Die index.php wird aufgerufen und das Template /templates/$template/page.html wird geladen und in die Variable $artikeltemplate gespeichert. Es wird deweiteren die die /GeneratedItems/module.php includiert und ausgeführt.
3. Die module.php lädt nun die modulloading.php aus den einzelnen Modulen, und überliefert hierbei das $artikeltemplate welches von den einzelnen Modulen verarbeitet werden kann.
4. Das $artikeltemplate wird jetzt wieder an die /index.php zurückgeliefert
5. Ausgabe der Webseite an den User

Globale Variablen

$kundennummer	ID des eingeloggten Benutzers (Nur wenn Kundenmodul installiert ist)
$ID	ID der aufgerufenen Seite
$wechselfarbe1	Erste Wechselfarbe für die Webseite
$wechselfarbe2	Zweite Wechselfarbe für die Webseite
$template	Gibt den Namen des aktuellen Templates zurück
$artikeltemplate	Variable in der die bisherige Ausgabe gespeichert wurde

Beispielcode einer modulloading.php

```
<?php
$artikeltemplate=anfrage_uebersetzen($artikeltemplate,$template);
function anfrage_uebersetzen($artikel,$template){
        if(is_int(strpos($artikel, '[anfrage]'))==TRUE){
                        $artikel=str_replace("[anfrage]", anfrage_ausgeben($template),
$artikel);
        };
        return $artikel;
};

function anfrage_ausgeben($template){
        echo $GLOBALS['anfrageobj'];
        if(is_numeric($GLOBALS['anfrageobj'])==TRUE){
                $ausgabegesamt=templateeinlesen($template,"anfrage_detailansicht");

                return $ausgabegesamt;
        }else{
                $ausgabegesamt=templateeinlesen($template,"anfrage");

                $tmpl_zeile=templateeinlesen($template,"anfrage_vorschau");
```

```
                $cache='';
                $res = mysql_db_query ($GLOBALS['datenbanknamecms'], 'SELECT * FROM an-
frage') or error_mysql_debugger(mysql_error(),__FILE__,__LINE__);
                while ($row = mysql_fetch_array ($res))
                {
                        $cache.=$tmpl_zeile;
                        $cache=str_replace('{titel}',$row['titel'],$cache);


        $cache=str_replace('{linkobjekt}',linkpfad_erweitert($GLOBALS['ID'],$row['ti-
tel'],TRUE,'&anfrageobj='.$row['ID']),$cache);
                };

                $ausgabegesamt=str_replace('{inhalt}',$cache,$ausgabegesamt);

                return $ausgabegesamt;
        };
};

?>
```

4.3 Moduldatei api.php

Allgemeines

Ab der Version 1.3 unterschützt das System eine API um von extern auf die Daten von Tricoma zugreifen zu können. Die Einstellungen zur API können Sie in Ihrem tricoma unter Administration --> Einstellungen --> API vornehmen.´
Sie können dort die API deaktivieren, und ein Passwort für den Zugriff festlegen.

API anprogrammieren

Möchten Sie die API anprogrammieren, müssen Sie folgende Adresse öffnen:
http://ihre-domain/cmssystem/api.php?modul=modulname&apipasswort=passwort&modulkat=kat

Je nach Modul können bzw. müssen Sie unterschiedliche Parameter übergeben um Daten zu erhalten. Um herauszufinden welche Parameter möglich sind, können Sie bei dem Modul als Parameter modukat=page angeben.
Beispiel für einen Modulaufruf:
http://ihre-domain/cmssystem/api.php?modul=bereiche&modulkat=help

API für ein Modul programmieren

Um eine API für ein Modul zu programmieren müssen Sie eine Datei mit dem Namen "api.-php" in Ihrem Modulordner erstellen. Sie können jetzt über die API auf das Modul zugreifen. Die API inkludiert schon die config.php, allgfunktionen.php, debug.php und erstellt eine Datebankverbindung.

4.4 Moduldatei modulext_*.php

Manchmal kommt es vor, dass man für den Webseitenuser einen Zugriff auf ein Dokument ermöglichen möchte, welches ohne Webseitentemplate ausgegeben werden soll. Oder man möchte eine Datei zum Download anbieten, aber nicht den Dateipfad freigeben.

Man kann daher in seinem Modulordner eine Datei anlegen welche folgenden Dateinamenaufbau haben muss:
modulext_*.php

Um diese Datei von extern zugänglich zu machen, „tunnelt“ man diese über die /modul.php:
http://localhost/modul.php?modul=modulname&modulkat

Der * muss jeweils durch eine Bezeichnung ersetzt werden.

In der modul.php wird vor dem include der modulext_*.php folgender Code ausgeführt:

```
include("GeneratedItems/config.php");
 mysql_connect ("$host","$datenbankbenutzername","$datenbankpasswort");
 include("GeneratedItems/debug.php");
 include("cmssystem/allgfunktionen.php");
 include("GeneratedItems/firewall.php");
 include("GeneratedItems/templates.php");
 include("GeneratedItems/webseiteerkennen.php");
 include("cmssystem/mailer.php");
 include("cmssystem/rubcoder.php");
```

Codebeispiel einer modulext_*.php:

```
$IP=getenv("REMOTE_ADDR");
if(is_numeric($ID)==TRUE){
 $res=mysql_db_query ($GLOBALS['datenbanknamecms'], "SELECT dateityp,name FROM files
where ID='$ID'") or error_mysql_debugger(mysql_error(),__FILE__,__LINE__);
 while ($row = mysql_fetch_array ($res))
 {
      header("Content-type: $row[dateityp]");
      header("Content-Disposition: attachment; filename=$row[name]");
      readfile("cmssystem/downloads/dateien/$row[name]");
 };
};
```

4.5 Manuelle Erstellung eines eigenen Moduls

Module welche offizielle von IPResearch ausgeliefert werden, müssen nicht händisch installiert werden, sondern läuft dies über den Updateserver.
Oftmals ist es nötig sich ein individuales Modul zu programmieren, oder programmieren zu lassen. Wie man sich sein eigenes Modul im tricoma anlegen kann möchten wir in diesem Kapitel kurz beschreiben.

1. Anlegen der Modulordners
Beim anlegen eines neuen Moduls ist die Ordnerstruktur unter im Kapitel 4.1 zu beachten. Jedes Modul bekommt einen eigenen Ordner unter /cmssystem/*. Der Ordnername wird immer klein geschrieben. Wenn wir beispielsweise ein neues Modul mit dem Namen "Testmodul" anlegen dann müssen wir folgenden Ordner anlegen: /cmssystem/testmodul.

2. Einbinden des Moduls in das System über die Datenbank
Jetzt haben wir den Ordner für das Modul angelegt, dieses ist dadurch aber noch nicht im Tricoma CMS verfügbar. Um dieses dort anzeigen zu lassen müssen Einträge in folgenden Tabellen von Tricoma vorgenommen werden:

2.1 tri_menu

Wir müssen in der Tabelle tri_menu einen neuen Eintrag erzeugen, um das Modul für das CMS verfügbar zu machen.

2.2 tri_untermenu

Jetzt ist das Modul im Tricoma installiert, aber es sind noch keine Menüpunkte verfügbar. Dafür gibt es die Tabelle tri_untermenu.

2.3 tri_rechte

In der Tabelle tri_rechte werden alle verfügbaren Benutzerrechte gespeichert. Wichtig ist, dass ein Recht mit dem Namen des Moduls angelegt wird. In diesem Beispiel „testmodul“ (Wichtig: klein geschrieben)

Beschreibungen zu den oben genannten Tabellen sind im Kapitel 5.2 zu finden.

3. Anlegen der Moduldateien
Im Kapitel 4.1 sind die Konventionen zum anlegen von Moduldateien zu finden.
Um das Modul auch auf der Internetseite verfügbar zu machen, müssen wir eine modulloading.php anlegen. Die Funktionsweise der modulloading.php ist im Kapitel 4.2 beschrieben.

5 Datenbankstruktur

5.1 Vorabinformation

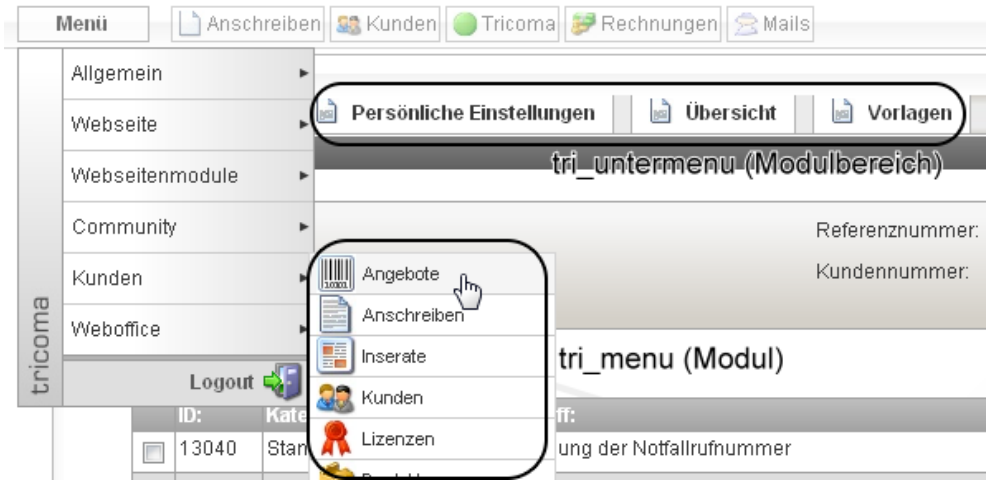
Bei der Entwicklung von tricoma war es uns sehr wichtig eine einfache und klare Datenbankstruktur zu haben.
Daher haben wir uns einen eindeutigen Syntax überlegt:

- Standardtabelle von tricoma Grundsystem bzw. der Grundengine fangen mit „**tri_**“ an.
- Tabellen eines tricoma Moduls fangen folgendermaßen an: „**modulname_**“

Folgende Konvetionen gibt es für Spaltennamen:

ID	In dieser Spalte wird der Primärschlüssel einer spalte abgelegt. Beispiel: *9993*
titel	Diese Spalte wird für z. B. Produktnamen, Betreffs oder Hauptbezeichnungen verwendet. Beispiel: *Wie wird man Millionär?*
artikel	Hier wird der Beschreibungstext oder ähnliches abgelegt. Beispiel: *Millionär wird man, indem…*
edit	In dieser Spalte wird der Name des Autor des aktuellen Datensatzes abgelegt. Beispiel: *Dagobert Gates*
datum	Diese Spalte beinhaltet das Datum an dem der Datensatz angelegt wurde. Beispiel: 2009-02-27 – 12:26:26
onlinefreigabe	Zeigt an, ob der Datensatz extern Verfügbar sein darf, oder nicht.
papierkorb	Zeigt an, ob der Datensatz im Papierkorb ist
papierkorb_edit	Benutzername des Autors, welcher den Datensatz in den Papierkorb verschoben hat. Beispiel: Dagobert Gates
papierkorb_datum	Datum, wann der Datensatz in den Papierkorb verschoben wurde. Beispiel: 2009-02-28 – 18:22:11

5.2 Tricoma Tabellen (tri_*)



5.2.1 Tabelle tri_menu

In der Tabelle tri_menu werden alle installierten Module eingetragen

ID	name	bereich	anzeigen	version	loeschbar	modulloading
statistik	Statistik	2	1	1.0	1	3

ID	Modulname klein geschrieben. Muss genauso wie der Ordner heißen.
name	Modulname wie er später im Tricoma für den Benutzer angezeigt wird.
bereich	Bereich im Menü von Tricoma indem er angezeigt wird: 1= Allgemein, 2 = Webseite, 3 = Webseitenmodule, 4 = Kunden, 5 = Community, 6 = Weboffice
anzeigen	Soll das Menü im Tricoma angezeigt werden? 1 = Ja, 2 = Nein
version	Versionsnummer
loeschbar	Darf das Modul über die Modulverwaltung gelöscht werden? 1 = Ja, 2 = Nein
modulloading	In welcher Priorität soll später die modulloding.php geladen werden. Weiteres siehe Schritt 3.

5.2.2 Tabelle tri_untermenu

Die Tabelle tri_untermenu hält alle Menüpunkte aus dem CMS bereit.

ID	berechtigung	name	link
statistik_online	statistik	Aktuelle Benutzer	statistik/useronline.php

ID	Modulname klein + "_" + erweiterter Name
berechtigung	Welches Recht aus der Tabelle tri_rechte wird benötigt?
name	Name des Links
link	Link zur Datei. (modul/datei.php)

5.2.3 Tabelle tri_rechte

In der Tabelle tri_rechte werden alle verfügbaren Benutzerrechte gespeichert.

recht	beschreibung
statistik	Gewaehrt Zugriff auf Statistiken ueber die Webseit...

recht	Modulname klein (+ "_" + erweiterter Rechtname z. B. statistik_benutzerstatistik)
beschreibung	Rechtebeschreibung

6 Funktionsbibliotheken

6.1 Hauptbibliothek: /cmssystem/allfunktionen.php

6.1.1 Sessions

tri_session_ermitteln()

Rückgabe: String (Sessionid)
Durch Aufruf der Funktion wir die Sessionid des aktuellen Benutzers zurückgeliefert. Hat der Benutzer noch keine Session wird eine angelegt.

tri_session_auslesen($sessionid)

Parameter1: String (Sessionid)
Rückgabe: Array (eindimensional)
Liefert ein Array mit dem Inhalt der Session zurück

tri_session_cleanup()

Rückgabe TRUE
Löscht alte abgelaufene Session

tri_session_speichern($sessionid,$inhalt)

Parameter1 String (Sessionid)
Parameter2 Array (eindimensional)
Rückgabe TRUE
Speichert das Array $inhalt in die Session. Der vorherige Inhalt der Session wird dadurch gelöscht. Es wird daher empfohlen erst über die Funktion tri_session_auslesen() den Inhalt auszulesen, diesen abzuändern und dann zu Speichern.

6.1.2 Zeitfunktionen

monatsausgabe($monat)

Parameter1: Integer (Monatsnummer z. B. 02 für Februar, 11 für November)
Rückgabe: String (Monatsausgabe auf Deutsch z. B. Februar)
Gibt anhand zweistelligen Zahl des Monats die Monatsbezeichnung zurück

datumwandeln_deutsch($datum)

Parameter1: Datum (YYYY-MM-DD)
Rückgabe: Datum (DD. MM. YYYY)
Wandelt ein Datum von ISO Norm auf den alten deutschen Standard um.

datumuhrzeitwandeln_deutsch($datum)

Parameter1: Datum (YYYY-MM-DD - HH:MM)
Rückgabe: Datum (DD. MM. YYYY - HH:MM)
Wandelt ein Datum von ISO Norm auf den alten deutschen Standard um.

datumnachtimestamp($datum)

Parameter1: Datum (YYYY-MM-DD)
Rückgabe: timestamp
Wandelt ein Datum von ISO Norm auf ein Timestamp um.

6.1.3 Strings

umlaute_anpassen($string)

Parameter1: String
Rückgabe: String (gesäuberte Umlaute)
Nimmt den String $string und repariert fehlerhafte Umlaute

prozentausgabe($wert)

Parameter1: Double (Zahl mit . als Kommstelle)
Rückgabe: String (Farbliche Kennzeichnung ob Positiv oder Negativ)
Erstellt aus dem $wert eine farbliche Ausgabe mit HTML und liefert diese zurück

textkuerzen($string, $laenge)

Parameter1: String (Text)
Parameter2: Integer (Länge)
Rückgabe: String (gekürzter String)
Kürzt den String $string auf die Länge $laenge. Wenn der $string kleiner als Laenge ist bleibt er unbearbeitet. Ist er länger wir der gekürzt und „…“ angehängt.

stringteilen($string, $trennzeichen)

Parameter1: String (String der durchsucht werden soll)
Parameter2: String (Trennzeichen nach dem gesucht werden soll)
Rückgabe Stringarray
Beschreibung siehe: http://php.net/explode

sichere_variable($unsicherevariable)

Parameter1: String (unsichere Variable)
Rückgabewert: String (sichere Variable)

Funktion um aus einer unsicheren Variable eine sichere zu machen. Alle „ und , werden gesichert um sichere SQL Anweisungen durchführen zu können.

checkmail($mail)

Parameter1: String (E-mail)
Rückgabe: Boolean (TRUE wenn gültig, FALSE wenn nicht)
Prüft ob die E-mail Adresse gültig ist.

firefox_html_anpassen($artikel)

Parameter1: String (Text)
Rückgabe: String (Text)
Nimmt die Variable Artikel und macht den Quellcode Konform für den Firefox Browser

perso_check($id)

Parameter1: String ($_POST['serial'] . 'D ' . $_POST['birthday'] . ' ' . $_POST['expiration'] . ' ' . $_POST['all'])
Rückgabe: Boolean
Prüft ob die Personalausweisnummer gültig ist

handynummercheck($nummer)

Parameter1: String (Handynummer)
Rückgabe: Boolean
Prüft ob die $nummer eine Handynummer ist

zahlstellen($x,$y)

Parameter1: Int
Parameter2: Int
Rückgabe: Int
Nimmt die Zahl $x und macht daraus eine Zahl mit $y Stellen.

url_text_umwandeln($string)

Parameter1: String
Rückgabe: String (gesäuberte Umlaute)

Nimmt den String $string und ersetzt Textlinks (http://) durch richtige Links

6.1.4 Dateimanagement

dateitypicon($endung,$typ)

Parameter1: String (Dateiendung z. B. html, jpg, bmp)
Parameter2: Integer (1 = Bild für die Datei als STRING,
2 = Dateityp als String
3 = Mit Editor bearbeitbar prüfen
Rückgabe: 1 = String z. B. img src="pfadzumbild"
2 = String z. B. HTML Template
3 = Boolean (TRUE bei mit Editor bearbeitbar, FALSE wenn nicht)
Funktion um einen Dateityp herauszufinden und hierzu passende Ausgaben zu erzeugen.
Um einen Ordner auszugeben muss als Endung „ordner“ verwendet werden.

datei_loeschen($dateiname)

Parameter1: Pfad zur Datei
Rückgabe: Boolean (TRUE wenn Datei gelöscht, FALSE wenn nicht)
Löscht die Datei $dateiname vom Server nach vorheriger Prüfung ob diese vorhanden ist. Vorteil zu unlink ist das keine Fehlermeldung bei nichtexistieren der Datei ausgegeben wird.

dateigroesse($groesse)

Parameter1: Integer (Dateigröße in Byte)
Rückgabe: String (z. B. 1.4 MB)
Über diese Funktion lässt sich die Dateigröße einer Datei in aufgewerteter Form ausgeben.

verzeichnisgroesse($pfad)

Parameter1: String ($pfad zum Ordner)
Rückgabe: Integer (Ordnergröße in Byte)
Liefert die Gesamtgröße eines Ordners und seiner Unterordner zurück.

datei_einlesen($pfad)

Parameter1: String (Pfad zur Datei)
Rückgabe: Inhalt der Datei, oder FALSE wenn die Datei nicht existiert
Liest den Inhalt einer Datei aus.

6.1.5 Tricoma

trieinstellungauslesen($modul,$benutzer,$typ)

Parameter1: String (Modul in welchem die Funktion verwendet wird)
Parameter2: String (Benutzername des aktuellen Benutzers, oder system wenn kein Benutzer vorhanden oder die Einstellung Global ist)
Parameter3: String (Typ zur Identifizierung)
Rückgabe: String
In Tricoma gibt es eine Registry die es ermöglicht Werte aus der Datenbank auszulesen.
Beispiel: trieinstellungauslesen(‚bereiche', ‚Administrator','letzterbereich')

trieinstellungsetzen($modul,$benutzer,$typ,$wert)

Parameter1: String (Modul in welchem die Funktion verwendet wird)
Parameter2: String (Benutzername des aktuellen Benutzers, oder system wenn kein Benutzer vorhanden oder die Einstellung Global ist)
Parameter3: String (Typ zur Identifizierung)
Rückgabe: TRUE
In Tricoma gibt es eine Registry die es ermöglicht Werte in die Datenbank zu Speichern. Empfohlene Anwendung ist die Speicherung von z. B. letzter Kategorieauswahl des Benutzers, letzter Login des Benutzers, beliebige Selektionsmöglichkeiten, Einstellungen in einem Modul.
Beispiel: trieinstellungauslesen(‚bereiche', ‚Administrator','letzterbereich',$ID)

idnachpfadumwandeln($ID)

Parameter1: Integer (ID aus der Tabelle bereich)
Rückgabe: String (Kompletter Linkpfad auf den Bereich $ID)
Funktion um im tricoma CMS einen Bereich aus der Tabelle „bereich" verlinkt auszugeben.

benutzervorhanden($benutzer)

Parameter1: String (Benutzername eines Benutzers aus Tricoma)
Rückgabe: Boolean (TRUE wenn Benutzer vorhanden, FALSE wenn nicht)
Prüft in der Tricomabenutzertabelle ob der Benutzer $benutzer vorhanden ist.

modulvorhanden($modul)

Parameter1: String (Modulname)
Rückgabe: Boolean (TRUE wenn Modul vorhanden, FALSE wenn nicht)
Prüft ob das Modul $modul installiert ist

modul_recht_ausgeben($recht)

Parameter1: String (Benutzerrecht)
Rückgabe String (Modulname)
Gibt das passende Modul zum Recht aus oder liefert NULL zurück wenn kein Modul vorhanden ist

linkpfad($ID)

Parameter1: Integer (Primärschlussel aus der Tabelle bereich)
Rückgabe: String (Link-Pfad zur Verlinkung)
Generiert einen Webseitenlink auf die ID $ID. Der Dateiname bei einer HTML Webseite ist immer Seitentitel
Beispiel für Rückgabe:
Bei HTML Webseite: Verlinkung-1.html?t=1
Bei dynamischer Webseite: index.php?ID=1

linkpfad_erweitert($ID,$linkname)

Parameter1: Integer (Primärschlussel aus der Tabelle bereich)
Rückgabe: String (Link-Pfad zur Verlinkung)
Generiert einen Webseitenlink auf die ID $ID. Im Gegensatz zu der Funktion linkpfad kann man über den Parameter $linkname den Dateinamen selbst bestimmen.
Beispiel für Rückgabe:
Bei HTML Webseite: $linkname-1.html?t=1
Bei dynamischer Webseite: index.php?ID=1

6.1.6 Ausgabe

helpdesk($titel,$beschreibung)

Parameter1: String (Titel im Helpdesk)
Parameter2: String (Beschreibung im Helpdesk)
Rückgabe: String (Verlinkung auf das Helpdesk)
Gibt einen Link zum Helpdesk aus in welchem die Parameter $titel, $beschreibung angezeigt werden.

hinweisausgabe($titel,$beschreibung)

Parameter1: String (Titel)
Parameter2: String (Beschreibung)
Rückgabe: String (HTML Ausgabe)
Gibt einen grafischen Hinweis in HTML aus.

fehlerausgabe($titel,$beschreibung)

Parameter1: String (Titel)
Parameter2: String (Beschreibung)

Rückgabe: String (HTML Ausgabe)
Gibt eine grafische Fehlermeldung in HTML aus.

6.1.7 Sonstige

tri_geo_entfernung($plz1,$plz2)

Parameter1: Integer (Deutsche PLZ mit 6 Stellen)
Parameter2: Integer (Deutsche PLZ mit 6 Stellen)
Rückgabe: Integer (Entfernung zwischen den PLZ)

Ermittelt die Luftlinie zwischen den Postleitzahlen. Tabelle tri_geo muss vorhanden sein. Wenn keine Ermittlung möglich ist wird 999 als Rückgabewert zurückgeliefert.

statistikausgabe($sql,$farbe,$link,$titel,$typ)

Parameter1: String (SQL Abfrage für die Statistik)
Parameter2: String (Farbcode)
Parameter3: String (Link der aktuellen Datei)
Parameter4: String (Titel)
Parameter5: Integer (1=Monatstatistik, 2=Jahresstatistik)
Rückgabe: String (Statistikausgabe)

Gibt eine Statistik zu dem übermittelten SQL Befehl aus. z. B.
statistikausgabe("SELECT count(ID) as counter FROM kunden where registrierung_datum like '%monat%'",'#343434','statistik.php?auswahl=allgemein','Monatswahl',1);
oder
statistikausgabe("SELECT count(ID) as counter FROM kunden where registrierung_datum like '%jahr%'",'#343434','statistik.php?auswahl=allgemein','Jahreswahl',2);

seitenumschaltung($sql,$farbe,$link,$titel,$typ)

Parameter1: String (SQL Abfrage der aktuellen Selektion)
Parameter2: String (Link der aktuellen Datei)
Parameter3: Integer (1=HTML Code für Seitenumschaltung, 2=Limit für SQL)

Gibt eine Seitenumschaltung zur aktuellen SQL Abfrage aus.
z. B. seitenumschaltung($sql,'datachange.php',1)

6.2 Grafikbibliothek (Class): /cmssystem/standard/ class.tri_bildbearbeitung.php

6.2.1 Grundsätzliches

Die Grafikbibliothek wurde speziell für die tricoma Engine entwickelt um Bilder nachbearbeiten zu können. Die Klasse bietet einem viele verschiedene Funktionen wie z. B. Schatten, Thumpnail, Verkleinern, Drehen uvm.

6.2.2 Bild verkleinern

Mittels einer Thumpnail Funktion lassen sich Bilder verkleinern.

Beispielquellcode

```
include("../standard/class.tri_bildbearbeitung.php");
$meinbild=new bildbearbeitung;
$meinbild->quelldatei=$quelldatei;
$meinbild->zieldatei=$zieldatei;
$meinbild->tump_breite='150';
$meinbild->tump_hoehe='100';
```

```
$meinbild->maxbreite='150';
$meinbild->maxhoehe='100';
$meinbild->tump_zent=TRUE;
$meinbild->laden(); // Lädt die Quelldatei
$meinbild->tumpnail(); //Erzeug das Thumpnail
$meinbild->speichern();// Speichern
```

6.2.3 Bild drehen

Auch die Drehung von Bildern ist mit der Klasse möglich.

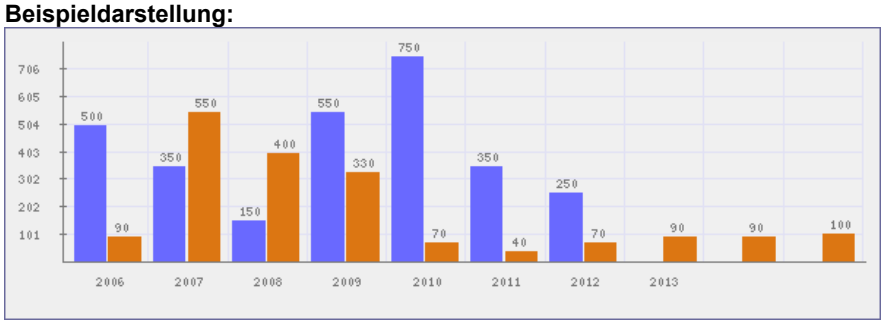
Beispielquellcode

```
$meinbild=new bildbearbeitung;
$meinbild->quelldatei=$quelldatei;
$meinbild->zieldatei=$zieldatei;
$meinbild->laden();    // Lädt die Quelldatei
$meinbild->drehen(1); // Der Parameter 1 sind die Anzahl der Drehungen um 90 Grad
$meinbild->ausgabe(); // Speichern des gedrehten Bildes
```

6.2.4 Balkendiagramm

Mit dem Balkendiagramm lassen sich positive Werte darstellen.

Beispieldarstellung:



Beispielquellcode

```
include("gfx.php");

header("Content-type: image/png");

$farbe['hintergrundfarbe'] = 239239239;
$farbe['hintergrundfarbefeld'] = 240240240;
$farbe['linienfarbe'] = 120120120;
$farbe['trennlinienfarbe']=225225245;
$farbe['randfarbe']=105105155;
$abstand['links']=40;
$abstand['rechts']=10;
$abstand['oben']=10;
$abstand['unten']=40;
$datenfarbe['0']=105105255;
$datenfarbe['1']=220118018;
$datenfarbe['2']=105105255;
$datenfarbe['3']=105205106;
$datenfarbe['4']=125105106;

$anzahlbeschriftungeny=7;

$daten['0']['0']['wert']=500;
$daten['0']['1']['wert']=350;
$daten['0']['2']['wert']=150;
$daten['0']['3']['wert']=550;
$daten['0']['4']['wert']=750;
```

```
$daten['0']['5']['wert']=350;
$daten['0']['6']['wert']=250;
$daten['0']['0']['beschriftung']="   2006";
$daten['0']['1']['beschriftung']="   2007";
$daten['0']['2']['beschriftung']="   2008";
$daten['0']['3']['beschriftung']="   2009";
$daten['0']['4']['beschriftung']="   2010";
$daten['0']['5']['beschriftung']="   2011";
$daten['0']['6']['beschriftung']="   2012";
$daten['0']['7']['beschriftung']="   2013";

$daten['1']['0']['wert']=90;
//$daten['1']['0']['beschriftung']="   2007";
$daten['1']['1']['wert']=550;
$daten['1']['2']['wert']=400;
$daten['1']['3']['wert']=330;
$daten['1']['4']['wert']=70;
$daten['1']['5']['wert']=40;
$daten['1']['6']['wert']=70;
$daten['1']['7']['wert']=90;
$daten['1']['8']['wert']=90;
$daten['1']['9']['wert']=100;

$grafik=tricoma_balkendiagramm(600,200,$abstand,$farbe,$daten,$datenfarbe,
$anzahlbeschriftungeny);

imagepng($grafik);
imagedestroy($grafik);
```

6.2.5 Quartiler Rang

Quartile Ränge sind nützlich um Bewertungen darzustellen.

Beispielquellcode

```
include("cmssystem/gfx.php");
if(is_numeric($gesamt) and is_numeric($position)){
 header("Content-type: image/png");

 $farbe['hintergrundfarbe'] = 249249249;
 $farbe['linienfarbe']=105105105;
 $farbe['farbe1']=100193015;
 $farbe['farbe2']=191191001;
 $farbe['farbe3']=215137001;
 $farbe['farbe4']=155005005;


 $grafik=tricoma_quartildiagramm(60,13,$farbe,$position,$gesamt);

 imagepng($grafik);
 imagedestroy($grafik);
}else{
 echo "Fehlerhafte Parameter";
};
```

6.2.6 Ränge

Das Diagramm ist sehr hilfreich um einen Rang darzustellen.

Beispielquellcode

```
include("cmssystem/gfx.php");
if(is_numeric($gesamt) and is_numeric($position)){
 header("Content-type: image/png");

 $farbe['hintergrundfarbe'] = 249249249;
 $farbe['linienfarbe']=005005005;
 $farbe['linienfarbe_schatten']=172172172;
 $farbe['beschriftung']=105005005;

 $grafik=tricoma_rangdiagramm(80,20,$farbe,$position,$gesamt);

 imagepng($grafik);
 imagedestroy($grafik);
}else{
echo "Fehlerhafte Parameter";
};
```

6.3 Mailversand (Funktion): /cmssystem/mailer.php

Ein wichtiges Feature unter tricoma ist der Mailversand über den Mailspooler. Emails werden hier nicht direkt versendet, sondern vor dem Versand in der MySQL Datenbank abgelegt.

Die Mails aus der Datenbank, werden dann über den Standarcronjob /cmssystem/standard/cronjob.php versendet.

Der Vorteil hierbei ist, dass Mails welche nicht versendet werden können nicht verloren gehen.

Für den Mailversand gibt es eine entscheidende Funktion, nämlich die trimailer_easy welche in der Funktionsdatei /cmssystem/mailer.php zu finden ist.

Beispielquellcode:

```
$return=trimaler_easy(
        $empfaengermail,
        $betreff,
        $nachrichteninhalt,
        $absendermail,
        $absender_name,
        $priodermail,
        $pfadzurdatei1,
        $namederdatei1
);
```

6.4 Datenexport (Class): /cmssystem/standard/class.tri_export.php

Die Datei class.tri_export.php ist eine Klasse um Exports zu erzeugen.

6.5 Debugger (Function): /GeneratedItems/debug.php

Die Datei /GeneratedItems/debug.php sollte immer includiert werden, da Sie Funktionen zum Debuggen enthält.

6.6 Templateverarbeitung (Function): /GeneratedItems/templates.php

Die Datei /GeneratedItems/templates.php enhält Funktionen zum verarbeiten von Templates.