



From Technologies to Solutions

Building Websites with

Joomla! 1.5 Beta 1

The best-selling Joomla tutorial guide updated for the latest download release

Hagen Graf

[PACKT]
PUBLISHING

Building Websites with Joomla! 1.5 Beta 1

The best-selling Joomla tutorial guide updated for the
latest download release

Hagen Graf



BIRMINGHAM - MUMBAI



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

Building Websites with Joomla! 1.5 Beta 1

Copyright © 2007 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, Packt Publishing, nor its dealers or distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: February 2007

Production Reference: 1160207

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-84719-238-7

www.packtpub.com

Cover Image by www.visionwt.com

Copyright © 2006 by Pearson Education Deutschland GmbH, München.

First published in the German language under the title "Joomla! 1.5" by Addison-Wesley, an imprint of Pearson Education Deutschland GmbH, München.

Credits

Author

Hagen Graf

Project Manager

Patricia Weir

Translator

Wolfgang Spegg

Indexer

Bhushan Pangaonkar

Development Editor

Louay Fatoohi

Proofreader

Chris Smith

Technical Editor

Divya Menon

Production Coordinator

Manjiri Nadkarni

Editorial Manager

Dipali Chittar

Cover Designer

Shantanu Zagade

About the Author

Hagen Graf was born in July 1964. Born and raised in Lower Saxony, Germany, his first contact with a computer was in the late seventies with a Radioshack TRS 80. As a salesperson, he organized his customers' data by programming suitable applications. This gave him a big advantage over other salesmen. With the intention of honing his skills, he joined evening courses in programming and became a programmer. Nowadays he works in his wife's consulting company as a trainer, consultant, and programmer (<http://www.cocoate.com>).

Hagen Graf has published other books in German, about the Apache web server, about security problems in Windows XP, about Mambo, and about Drupal. Since 2001, he has been engaged in a nonprofit e-learning community called "machmit.org e.V.", as well as in several national and international projects. All the projects are related to content management, community building, and harnessing the power of social software like wikis and weblogs. He chose Joomla! CMS because of its simplicity and easy-to-use administration. You can access and comment on his blog (<http://www.bloghouse.org/en/hagen>).

This is the third time a book of mine has been translated from German to English. It isn't easy to organize the translation in another language in a reasonable way, especially on a topic on Open Source Software. One point is that most of the software is developed in international communities basically in the English language. Another point is the speed of the development. Release fast, release often! Today we have Joomla! 1.5 Beta 1 and the development is going on.

It is now time for thanks...

I wish to thank the Joomla community who made this wonderful world wide project possible.

I also wish to thank the Packt Publishing team, especially Louay, Divya, Dipali, Bhushan, Chris, Manjiri, and Patricia. I also wish to thank Alex Kempkens, core member of the Joomla! devteam. I also thank Tom Boháček, Anne-Kathrin Merz, and Andy Miller for their templates.

They all have done an excellent job!



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

Table of Contents

Preface	1
Chapter 1: Terms, Concepts, and Deliberations	7
Content Management System	7
A Quick Glance into History	8
Joomla!—How was it Developed?	11
Structure of a WCMS	13
Front End and Back End	13
Access Rights	13
Content	13
Extensions	13
Components	14
Templates	14
Module	14
Plug-ins	14
Workflow	14
Configuration Settings	15
Joomla! as Real Estate	15
Joomla! Versions	16
Numbering System of Joomla! Versions	16
Roadmap	17
Changes in Detail	18
Internationalization	18
User Plug-ins	18
XML-RPC support	18
Support of Several Databases	18
FTP System	19
Overhaul of the Joomla! Framework	19
Web Accessibility	19
Search Engine Friendliness	19
Google Summer of Code Projects	19

Table of Contents

Joomla! Features	20
Examples of Joomla! Pages	21
Joomla.org	21
Ada Gaffney Shaff, USA	22
Janette Norton, France	23
Urth.tv, USA	24
unric.org, Europe	25
Summary	25
Chapter 2: Installation	27
Client-Server System	27
Accessing a Joomla! Website on the Internet	27
Technical Requirements for Joomla!	28
Necessary Elements for a Joomla! Installation	28
Local Test Environment	29
Windows Operating System	29
Linux Operating System	29
Mac OS X Operating System	29
Production Environment	30
Rented Virtual Server	30
Your Own Server	30
Setting Up the Local Server Environment	31
Windows	31
XAMPP for Windows	32
Linux	35
SUSE > 9.1	35
Debian/Ubuntu	36
Your Own Server at a Provider	36
Joomla! Installation on a Virtual Server on the Net	36
Joomla! Installation	38
Selecting a Directory for Installation	38
An Example	38
Directory	38
Extracting	40
Joomla! Web Installer	41
Step 1: Choose Language	41
Step 2: Pre-Installation Check	42
Step 3: License	43
Step 4: Database Configuration	44
Step 5: FTP Configuration	46
Step 6: Configuration	46
Step 7: Finish	48
Summary	50

Chapter 3: A Tour of Your New Website	51
Front End	51
Menus	53
Top Menu	53
Main Menu	54
Other Menu	54
Content	54
What is Content?	55
First Page/Front Page	56
The Latest Messages/The Most Often Read Messages	57
Advertising	57
Banner Area	58
Functions	58
Login Area	58
Polling	59
Who is Online?	60
Feeds	60
Search Field	60
Decorative Elements	61
Back End	61
Summary	64
Chapter 4: Customizing Joomla!	65
Configuration of Joomla! Administration	65
Help Menu	67
Joomla! Help	67
System Info	68
Site Menu	69
Control Panel	70
User Manager	70
Editing Users	72
Adding New Users	76
Media Manager	77
Creating a Directory	78
Uploading a File	79
Global Configuration	80
Site	81
Users	86
Content	88
Server	91
Logout	96
Menus Menu	96
Menu Manager	97
Main Menu	98

Table of Contents

Trash Manager	106
Creating a New Menu	107
Content Menu	113
Article Manager	114
Creating a New Article	118
Article Trash	127
Section Manager	128
Category Manager	130
Frontpage Manager	132
Article Statistics	133
Editing Content from the Front End	134
Tools	135
Private Messaging	135
Mass Mails	136
Global Check-In	137
Summary	138
Chapter 5: Components and Extensions	139
Components Menu	139
Banners	139
Banner Clients	140
Manage Banners	141
Contacts	145
Contact Manager	145
Category Manager	149
News Feeds	150
Polls	152
Web Links	154
Links	155
Categories	156
Extensions Menu	157
Install/Uninstall	157
Module Manager	158
Site Modules	158
Banner	160
Breadcrumbs	161
Footer	162
Main Menu	162
Login Form	164
Statistics	165
Archive	165
Sections	165
Related Items	165
Wrapper	166

Table of Contents

Feed Display	167
Joomla! Book	167
Polls	168
Who's Online	168
Random Image	168
Advertisement	169
Syndication	169
Newsflash	169
Latest News	170
Popular	170
Search	171
Copy a Module	172
Administrator Modules	173
Logged in Users	173
Popular	174
Recent added Articles	174
Menu Stats	174
Footer	174
Unread Messages	174
Online Users	174
Quick Icons	174
Login	174
Admin Menu	174
User Status	175
Admin Submenu	175
Title	175
Toolbar	175
Cpanel Shell	175
Plug-In Manager	175
Authentication Plug-Ins	176
Content Plug-Ins	178
Editor Plug-Ins	180
Editors-Xtd Plug-Ins	180
Search Plug-Ins	180
System Plug-Ins	180
XML-RPC Plug-Ins	181
Template Manager	181
Site Templates	182
Module Position	182
Administrator Templates	182
Adding a New Template for Your Website	182
Modifying a Template	185
Language Manager	187
A Different Language for the Website and the Administrator	187
More on Extensions	192
Exposé Flash Gallery	194
Installation	195
Integration into the Website	196

Table of Contents

Videos	199
Album Manager in the Administration Section	200
Uninstallation	203
Album Manager as a Stand-Alone Program	203
Integration into the Joomla! Framework	204
Problems with Third-Party Components	205
Updates	205
Security	205
What to Do?	205
Summary	206
Chapter 6: Writing Your Own Joomla! Templates	207
Corporate Identity	207
HTML/XHTML, CSS, XML	208
HTML/XHTML	208
CSS	208
XML	209
Creating Your Own Templates	210
Concept	210
Fixed Size or Variable	210
Structure	211
HTML Conversion	212
Directory Structure of the Template	213
First Trial Run	215
Integration of the Joomla! Module	217
Creating a Template Package	222
Installation with the Joomla! Template Installer	223
Creating Templates with the Dreamweaver Extension	224
Installation	225
Create a New Template File	226
Template Structure	229
With Table	229
Insertion of the Joomla! Modules	232
Live Site	233
Templates and <div> Tags	236
The Left Module Position in Detail	236
Modifying HTML Output without Changing the Core Files	239
Web Accessibility for Joomla!	240
Criteria for Accessible Websites	240
The Reality	241
Is Joomla! 1.5.0 Web Accessible?	241
Is it Possible to make Joomla! Web Accessible?	242
The Technology	242
The People	242

Web Accessible Sites with Joomla!	244
Summary	244
Chapter 7: Your Own Components, Modules, and Plug-Ins	245
Sample Helloworld Component	245
A Home for Helloworld	246
Making Use of the Joomla! Framework	247
A Little more Functionality	248
Separation of HTML Output and Programming Logic	249
An Example Component	250
The MySQL Table	252
The Front End	253
auto.php	254
auto.html.php	255
Integration into the Main Menu	257
The Joomla! Administration	258
The Component Table	258
admin.auto.php	260
auto.class.php	264
admin.auto.html.php	265
toolbar.auto.php	267
toolbar.auto.html.php	268
Test	269
Creating an Installation Package	269
auto.xml	270
Modules	272
Source code	272
mod_auto.xml	273
Installation	274
View on the Website	276
Plug-Ins	276
Source Code	276
View on the Website	279
Summary	280
Chapter 8: A Website with Joomla!	281
Idea	281
Preparations	282
Logo and Appearance	282
Photographs	284
Texts	284
Technical Conversion	284
Local Installation	284
The First Few Articles	286
Masthead	289

Table of Contents

Menu Structure	292
Structure of the Main Menu	292
Structure of the Top Menu	293
Structure of the User Menu	293
Setting Up the Texts and the Menu Links in the Main Menu	293
Contact Link	295
Top Menu	296
Shop	297
The Impressions	299
The User Section	300
The User Menu	300
The Download Section	303
User Details	307
First Results	308
M. Bertrand Learns CSS	309
Formatting Step by Step	312
Changes for Joomla! Version 1.5	316
Installation on the Webserver	316
How to Do the Installation?	317
FTP	317
MySQL	317
Installation of Joomla! at the Provider	317
Importing the Data	318
File and Directory Rights	320
Search Engines	320
Search Engine Friendly URLs	321
Metatags	321
Design	321
Installation of the Template	321
The Template	323
Happy End	324
How Does it Work in Reality?	325
Summary	325
Chapter 9: Bonus Templates	327
NGO	327
Tom Boháček	328
The World Knowledge Template	328
Creative Approach	329
Structural Approach	329
Online Community	329
Anne-Kathrin Merz	330
A Question of Orientation	331
The Header	332
Left, Right, and the Contents	332

Table of Contents

Does this Theme Work with My Website?	333
Business Establishment	334
Andy Miller	334
Summary	335
Appendix A: Online Resources	337
Content for Download	337
Windows	338
Linux	338
Independent of the Operating System	338
Template Structure	339
How do I Exchange a Picture (Logo) in the Template?	341
Joomla! API	341
Forgot the Admin Password	342
Migration from Joomla! 1.0.x to Joomla! 1.5	343
Backing Up Your Data	344
Backing Up the Files	346
The Migration Script	346
New Installation of Joomla! 1.5.0	346
What Happens Now?	347
Manual Method	347
Modifying the Joomla! 1.5 Database Scheme	348
Importing the Tables	348
Security without Global Variables	349
Index	351



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

Introduction

More than a year has passed since the founding of the Joomla! project. And what an exciting year it has been!

The Joomla! team was organized and has built a solid foundation; it has continued developing Joomla! 1.0.0 to version 1.0.12, and now has the largest developmental leap so far to Joomla! 1.5 in sight. The users of the system have had just as exciting a year. Many have upgraded their site to Joomla! and a lot of new users have discovered Joomla!, but there are still a lot who don't know the system. Joomla! is by far the most used open-source Web Content Management System in the world. The development team believes that there are currently about 5,000,000 installations on public web servers. More than 45,000 registered developers are working on 1,100 projects to extend Joomla!. There are more than 450,000 posts from 50,000 users on the forum at www.joomla.org. It is being used all over the world. The bandwidth ranges from very simple homepages to very complex business applications.

In the course of this book I will show what it is that has made Joomla! so successful. I wrote the first Joomla! book, Joomla! 1.0, in Ausleben, a tiny village in Sachsen Anhalt in Germany. We didn't have DSL there, no public WLAN Hotspots, no UMTS, no international corporations, and no city noises. The book you are reading now was created in Fitou, also a small village, but in France and with WLAN hotspots and ADSL. Being online all of the time is slowly becoming a reality. This does serve to make daily life more hectic, but on the other hand it simplifies and enables you to accomplish things that were terribly laborious before, having to keep duplicate data sets and constantly having to verify the data.

Five years ago, it was normal to store emails on your home or office computer. Today, various service providers are offering almost inexhaustible disc space on the Internet for these purposes. In larger companies, terminal servers are becoming more and more influential. The bandwidth of Internet connections is increasing. Unfortunately there is a big difference between urban and rural locations in Germany when it comes to these services.

Introduction

I often work on different computers in different parts of the world; sometimes indoors, sometimes outdoors. The terminal, with which you and I access our information, becomes ever less important. What you really need is a stable, affordable Internet connection over WLAN, UMTS, telephone or satellite, a browser, a screen that can display the information, a keyboard that is as ergonomic as possible, and of course, electricity.

I naturally don't know what exactly you are working with, but a lot of people that I deal with work in similar ways to what I have described above.

Even the experiences with computers resemble each other. One mostly starts with an older PC and a Windows system in school and learns the hard reality of office applications, loss of data, memory problems, crashed hard drives, printer and configuration adventures, from experience. If you haven't developed a passion for these things, tinkering nightly with the operating system, then you are probably just as lucky as I am that your equipment and your applications work, that you are able to access the Internet and your data, and that can do your work in peace.

It is particularly important for everything to work properly if you work from a home office. Your employer saves the costs of your office and you have more flexibility. But software and hardware that doesn't work right can quickly turn this into a nightmare.

Parallel to the changes in the way we work, software is being developed that supports exactly this way of thinking. Browser-based applications such as email services, on-line banking, group calendars, document management systems, communities, dating services, and, of course, online auctions etc. are become ever easier, more sophisticated, and more user friendly.

Web 2.0 is clearly the buzzword. Services like Google Maps are becoming navigation systems, photographs are stored at Flickr, links from del.icio.us and all other services can be integrated in your own website. Mobile telephones are merging with PDAs, MP3 players, cameras, and other terminals. You can even call up websites with mobile telephones, fill out and send forms, send and receive emails, take and send photographs, listen to music, and much more.

The stationary PC is becoming smaller and smaller and laptops ever more ubiquitous. Advancements are taking place primarily with wireless connection technologies, miniaturization, and efficient rechargeable batteries.

In this world, a company, an institution, an association, an organization needs an Internet presence that is also user-friendly and flexible. One that is in tune with the times, one that can be easily modified from a browser and that replaces your briefcase and your address directory, one that can communicate with all kinds of systems and that is easily expanded.

Your website is where you can explain to others what you do, and/or what your company does. It is open 24 hours a day, 7 days a week to maintain your customer relations. Until recently, the production of such a homepage was a difficult thing. You didn't have to be a certified specialist, but you had to persevere and have an interest in the topic to produce an appealing result. You had to create static HTML pages with an HTML editor and subsequently load them onto a server via File Transfer Protocol. To provide even the simplest interactivity such as a guest book or a forum, you had to learn a programming language. Many people, for understandable reasons, were reluctant to take on this hardship and therefore either handed the production of their homepage to a web agency or decided to not even start such a project.

But help is near, because what you now have in your hand, this book, is the travel guide to Joomla!, one of the smartest website administration systems of the world.

The word Joomla! is derived from "Jumla" from Swahili and means "all together".

Joomla! is the software result of a serious disagreement between the Mambo Foundation, which was founded in August 2005 and its development team. Joomla! is the continued development of the successful Mambo system and, like Mambo, is a piece of software that enables simple administration of websites from a web browser.

Joomla!, according to its own description, is a "Cutting Edge Content Management System" and one of the most powerful Open Source Content Management systems in the world. It is used world-wide for anything from simple homepages to complicated corporate websites. It is easy to install, easy to manage, and very reliable.

What This Book Covers

Chapter 1 explains the term "content management" and delves into the structure of CMS and various features of Joomla!. It also lists some example websites based on Joomla!.

Chapter 2 describes the technical requirements of Joomla! and takes a step-by-step approach to installing your Joomla! 1.5 in different environments.

Chapter 3 will take you through a tour of the front end and back end of Joomla! 1.5.

Chapter 4 takes an in-depth look at the administration and configuration options.

Chapter 5 deals with customizing your extensions and components. You will also learn to install a local language file for different users and use a third-party component in your website.

Chapter 6 will help you create your own template. It will also briefly touch up on HTML/XHTML, CSS, and XML.

Introduction

Chapter 7 will help you extend Joomla!'s functionality with your new components, modules, and plug-ins.

Chapter 8 will deal with the most important aspect of this book – building your website.

Chapter 9 will give you three templates as bonus; for a non-governmental organization, an online community website, and a venture.

The *Appendix* will provide you with a list of necessary software packages, and help you in migrating from Joomla! 1.0.x to Joomla! 1.5 beta 1.

What You Need for This Book

You need the Beta 1 version of Joomla! 1.5 that will run in a environment consisting of PHP/Apache/MySQL.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

There are three styles for code. Code words in text are shown as follows: "In the case of the `com_contact` component, this is the `contact.php` file."

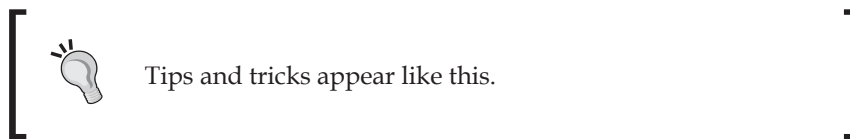
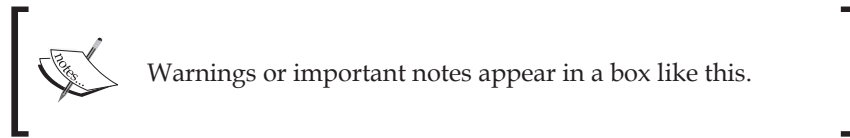
A block of code will be set as follows:

```
<?php
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
echo 'Hello World!';
?>
```

Any command-line input and output is written as follows:

```
-- Table structure for 'jos_auto' table
CREATE TABLE 'jos_auto' (
  'id' INT NOT NULL AUTO_INCREMENT, 'text' TEXT NOT NULL,
  'published' TINYINT(1) NOT NULL, PRIMARY KEY ('id') );
```

New **terms** and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "click on **Extensions** | **Install/Uninstall**, check your component, and click on the **Uninstall** icon".



Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to feedback@packtpub.com, making sure to mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or email suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the Example Code for the Book

Visit <http://www.packtpub.com/support>, and select this book from the list of titles to download any example code or extra resources for this book. The files available for download will then be displayed.



The downloadable files contain instructions on how to use them.

Errata

Although we have taken every care to ensure the accuracy of our contents, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in text or code – we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent versions of this book. If you find any errata, report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **Submit Errata** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata added to the list of existing errata. The existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Questions

You can contact us at questions@packtpub.com if you are having a problem with some aspect of the book, and we will do our best to address it.

1

Terms, Concepts, and Deliberations

Before you can understand how to operate Joomla!, allow me to explain the basic principles that underlie the system.

Content Management System

Content Management System (CMS) contains the terms content and management (administration), that imprecisely refer to a system that administers content. Such a system could be a board and a piece of chalk (menu or school chalkboard), or it could be something like Wikipedia (the free online encyclopedia at <http://www.wikipedia.org>), or an online auction house such as eBay (<http://www.ebay.com/>). In all three of these cases contents are administered, in the last instance by numerous participants. These participants play a major role with content management systems: on the one hand as administrators and on the other hand as users.

But it gets even better. Apart from CMSs there are **Enterprise Resource Planning Systems (ERP)**, administration of corporate data), **Customer Relationship Management Systems (CRM)**, management of customer contacts), **Document Management Systems (DMS)**, administration of documents), **Human Resource Management Systems (HRM)**, administration of staffing), and many others. An operating system such as Windows or Linux also administers content.

It is difficult to define the term CMS because of its encompassing nature and variety of functions. Wikipedia's definition is my favorite:

A content management system (CMS) is a computer software system used to assist its users in the process of content management. A CMS facilitates the organization, control, and publication of a large body of documents and other content, such as images and multimedia resources.

Lately **ECMS** has established itself as the buzzword for **Enterprise Content Management Systems**. The other systems listed above are subsets of ECMS.

Joomla! belongs to the category of **Web Content Management Systems (WCMS)**, since it exclusively administers content on a web server.

Since these terms are still relatively new in the enterprise world, these systems will surely be developed even further. In principle, however, there will always be an integration system that tries to interconnect all of these systems.

In general, the term "content management" is used in connection with web pages that can be maintained by a browser. This doesn't necessarily make the definition any easier.

A Quick Glance into History

While Sun Microsystems maintained in the nineties that "the Network is the computer", Microsoft was not going to rest until a Windows computer sat on every desk.

The computer that Microsoft was concerned with was a mixture of data files and binary executable files. Files with executable binary contents are called **programs**, and were bought and installed by customers to manipulate data. Microsoft Office was the winner in most of the offices around the world.

The computer that Sun was working with was a cheap, dumb terminal with a screen, a keyboard, a mouse, and access to the Internet. The programs and data were not stored on this computer, but somewhere on the net.

The *mine* philosophy governed Microsoft's practices whereas the *our* philosophy was adopted by Sun. The motivation for these philosophies was not for pure humanitarian reasons, but for economic interest. Primarily, Microsoft sold software for PCs to the consumer market, Sun, on the other hand, sold server hardware and programs to the enterprise market.

The Internet, invented in the sixties, spread like an explosion in the mid-nineties. Among other things, **Hyper Text Markup Language (HTML)**, the language used to write web pages, and the development of web servers and web clients (browsers) helped its expansion. The Internet itself was a set of rules that could be understood by different devices and was developed so skillfully that it covered the entire planet in almost no time.

An individual without an email address could no longer be reached, and a company without a website was not only old fashioned, but didn't exist in the eyes of many customers. The whole world swarmed to the Internet within a short time to become a part of it. Movies like *The Matrix* (<http://whatisthematrix.warnerbros.com/>) became huge hits and *1984* (<http://en.wikipedia.org/wiki/1984>), a book by George Orwell, was forgotten.

In 2003 Sun discovered the mobile work place and immediately started using that concept in its own offices. I quote Crawford Beveridge, chief human resources manager and Vice President of People and Places at Sun Microsystems, Inc.:

'Few of us recognize just how much the workplace has changed – and fewer still have done much about it.

Some of the changes are well known: Telecommuting, for instance. There are 2 million more telecommuters today than there were two years ago, and 6 million more are expected by 2004. But even 30 million people working from home is just a small part of the story and a small part of what companies need to gear up for.' (See complete article at <http://www.sun.com/2003-0114/feature/index.html>.)

New net citizens on one hand came from the *mine* world and on the other hand from the *our* world. Those who were used to buying programs bought HTML editors and created Internet pages with them. The others preferred to write their own HTML code with any text editor they had on hand. And the web agency, where one could order a homepage, was born.

Both groups faced the problem that HTML pages were static. To change the content of the page, it had to be modified on a PC and then copied to the server. This was not only awkward and expensive, but also made web presences like eBay or Amazon (<http://amazon.com/>) impossible.

Both groups came up with fixes to more or less solve this problem. The *mine* faction developed fast binary programs, with which one could produce HTML pages and load them via automated procedures onto the server. Interactive elements, such as visitor counters, among others, were built into such pages. The *our* faction discovered Java applets and with them the capability of writing a program that was operated via a browser, which resided centrally on a server. Entire business ideas were based on this solution – like online booking and flight reservation concepts.

Both groups tried to develop market share in different ways. The result was quite a stable market for both, in which passionate battles over the correct operating system (Windows, Linux, or Mac OS X) constantly drove the version numbers higher and higher. Customers got used to the fact that the whole thing wasn't that easy.

Terms, Concepts, and Deliberations

There is always a third option in these situations. In our case, it was, among other things, the emergence of open-source scripting languages like PHP (<http://www.php.net/>). Rasmus Lerdorf had the goal of offering interactive elements on his homepage and with that a new programming language was born. From the outset, PHP was optimized in perfect cooperation with the MySQL database, which also worked on the GNU/GPL platform (<http://www.gnu.org/licenses/gpl.html>).

Fortunately, there was the Linux operating system and the Apache web server that offered the necessary infrastructure on the server. The display medium at the client side was the browser. **LAMP** (Linux, Apache, MySQL, and PHP) soon became synonymous with database-supported, interactive presence on the Internet.

The most diverse systems like forums, communities, online shops, voting pages, and similar things that made it possible to organize contents with the help of a browser were developed in an enthusiastic creative rush.

Soon after the 'difficult' things such as Linux and Apache, 'soft' products were developed. The nineties were nearing their end; the Internet share bubble burst and suddenly the trend was to build unmitigated classical business models with unmitigated classical methods.

Whenever the economy isn't doing well, costs are scrutinized and the possibility of lowering costs is contemplated. There are now, as there were earlier, numerous possibilities. PHP applications were always sold in the millions. We only need to look at the phpBB (<http://www.phpbb.com/>) and phpMyAdmin (<http://www.phpmyadmin.net>) projects as examples – one was developed into the quasi-standard for forum software, the other one into the standard for manipulating MySQL databases via web interfaces.

The source code of the PHP language and the applications became better and better quickly due to the enormous number of users and developers. The more open a project was, the more successful it became. Individual gurus were able to save enterprises immense amounts of money in the shortest time.

Static HTML pages were considered old and expensive and were overhauled. They had to be dynamic! Developers have been working in this environment for a few years now. Linux, Apache, MySQL, and PHP were readily accepted in the industry. The search for professionally usable PHP applications had begun. With this search one looks for:

- A simple installation process
- Easy serviceability of the source code
- Security of the source code

- User-friendliness
- Easy extendibility
- Standardized interfaces to other programs
- Cost
- Independence from the supplier

The special advantage of PHP applications is the independence from hardware and operating system. LAMP also exists as WAMP (Windows, Apache, MySQL, and PHP) for Windows, MAMP (Mac, Apache, MySQL, and PHP) for Apple, and for numerous other platforms. And now Joomla! finally enters the picture.

Joomla!—How was it Developed?

An Australian company, Miro (<http://www.miro.com.au/>), developed a CMS named Mambo in the year 2001. It made this system available as open-source software to test it and to ensure a wider distribution. In the year 2002, the company split its Mambo product into a commercial and an open-source version. The commercial variant was called **Mambo CMS**, the open-source version **Mambo Open Source (MOS)**. By the end of 2004 all parties involved had agreed that MOS can officially be called Mambo and that a successful future for the fastest developing CMS of the moment would be secured.

The advantages of the commercial version were primarily the increased security for companies and the fact that they had Miro, which also supported further development, as a partner.

The open-source version offered the advantage that it was free and that an enormous community of users and developers alike provided continuous enhancements. In addition, it was possible for enterprises to take Mambo as a base and to build their own solutions on top of it.

In order to secure the existence and the continued development of Mambo, there were deliberations on all sides in the course of the year 2005 to establish a foundation for the open-source version of Mambo. On August 10, 2005, it happened: The Mambo Foundation was announced on the Mambo project page. After positive reactions during the first few hours, it quickly became obvious that Miro in Australia had established the foundation and that the developer team had not been included into the plans for the incorporation. Heated discussions erupted in the forums of the community and the developer team wrapped itself in silence for a few days.

On August 17, 2005, a statement was finally published on the OpenSourceMatters, announcing that it would be advised by the neutral Software Freedom Law Center and that it was planning the continued development of Mambo.

Terms, Concepts, and Deliberations


Just like in a shattered marriage, a war of roses quickly developed between the Miro-dominated Mambo Foundation that was all of a sudden without a development team, and the new development team itself, which, of course, needed a new name for the split entity. The parties sometimes called each other names in blogs, forums, and the respective project pages. Meanwhile, development of both projects continued.

On September 1, 2005, the name for the split entity was announced – Joomla!. This time the developer team secured itself the rights for the use of the name and also gave the community the option of changing their existing Mambo domains over to the new name before it was announced publicly. In no time at all, about 8,000 users had registered with the new forum.

The new project needed a logo and thus, on September 7, 2005, a competition was announced to the community. Meanwhile, the Mambo Foundations announced its new development team on September 13, 2005. A number of logo suggestions were published on September 14, 2005, and the new (old) community was asked to vote for the new Joomla! logo. The suggestions and results can, of course, be found online at <http://forum.joomla.org/index.php/topic,5953.0.html>.

Version 1.0 of Joomla! was published on the September 17, 2005. Quickly many of the third-party developers, among them Project Simpleboard, Docman, and many others, also switched from Mambo to Joomla! and strengthened the faith in the new project.

The company VA Software, which also operates the developer site sourceforge.net, decided to sponsor the Joomla! project's server infrastructure. It is hosted at Rochen, UK (<http://www.rochenhost.com/>). Joomla! handles around 1.8 terabytes of traffic from these servers, and that is without any video downloads!


 You can find a detailed summary of the events on the Internet at <http://www.devshed.com/c/a/BrainDump/Joomla-is-the-New-Mambo/>.

In October 2005, Joomla! won two prizes at LinuxWorld show in London: One for the best Linux and/or Open Source project in the year 2005 and the other prize for the core member Brian Teeman for his support to Open Source projects (UK Individual Contribution to Open Source).

One year after its foundation, Joomla! became one of the 50 most important Open Source projects in the world (<http://blogs.zdnet.com/open-source/?p=644>).

Joomla!'s future is more than rosy. The leap from version 1.0.x to 1.5.x is certain to keep all of the participants busy for a while, but it will irrevocably catapult Joomla! into the league of capable business content management systems. The team has cleverly chosen to make a kit of Joomla! that developers can use to create

websites that look nothing like Joomla! and that may even have completely different functions. With the existing base of user community, developers, and installations, Joomla! will cover many areas of that market. Numerous hosting providers are offering it pre-installed and the new version offers backward compatibility to version 1.0.x which should preclude any serious upgrade problems.

Structure of a WCMS

Using Joomla! as an example, I will briefly explain the structure of a WCMS.

Front End and Back End

A **web content management system (WCMS)** consists of a front end and a back end. The front end is the website that the visitors and the logged-on users see. The back end, on the other hand, contains the administration layer of the website for the administrator. Configuration, maintenance, cleaning, creation of statistics, and new content creation are all done in the back end. The back end is at a different URL than the website.

Access Rights

Whenever we talk of management, we talk of the clever administration of existing resources. In a WCMS, user names and group names are assigned to the people involved and these are assigned different access rights. This ranges from a simple registered user through an 'author' and 'editor' up to the 'super administrator', who has full control over the domain. Based on the rights, the website then displays different content, or the user is given the right to work in the back end.

Content

Content can come in all kinds of forms; in the simplest case, it is text. But content can also be a picture, a link, a piece of music, or a combination of all of these. To give an overview of the content, one embeds it in structures, for example, texts in different categories. The categories, of course, are also content that needs to be administered. Newsfeeds have become very popular this past year. The integration and actualization of newsfeed content is becoming more and more important.

Extensions

Components, modules, templates, and plug-ins are all referred to as extensions.

Components

Joomla! has to be expandable and able to grow with the requirements. Extensions that offer additional functionalities and that usually have their own area in Joomla!'s administration are called components. For example, typical components are an online shop, a user manager, a newsletter system, or a forum. Components contain the business logic of their site.

Templates

A template is a kind of visual edit format that is placed on the top of content. A template defines the colors, character fonts, font sizes, background images, spacing, and partitioning of the page, in other words, everything that has to do with the appearance of a page. A template is made up of at least one HTML file for the structure of the page and one CSS file for the design.

Module

A module is a field in the front end that usually displays data from a component. Modules can be displayed at predefined places in the template.

Special modules pertaining to components are used to integrate content in the desired form into templates. For example, a recent news module supplies the headings of the five most recent pieces of news that were placed by the Content Components to the template. Another module reports the number of users that are online at the time.

Plug-ins

A plug-in is a piece of programming code that is appended at certain places in the Joomla! framework in order to change its functionality. Such plug-ins can, for instance, be used within content text. Plug-ins are also used in a comprehensive website search in order to integrate additional components.

Workflow

By workflow one understands a work routine. The bureaucratic set of three (mark, punch, and file) is an example of a workflow. A recipe for baking a cake is a workflow. Since several people usually work with CMS content, well-organized workflows are a genuine help.

In this connection, one sometimes speaks of the work pending that a certain user has. For example, the editor sees a list of posted pieces of news, which he or she has to examine for correctness. After examining them, the editor marks the pieces of news as correct and they appear in the work pending of the publisher. The publisher then decides whether to publish the piece on the front page.

Configuration Settings

Settings that apply to the entire website are specified using the configuration settings. This includes the title text in the browser window, keywords for search engines, switches that permit or forbid logging on to the site or that switch the entire page offline or online, and many other functions.

Joomla! as Real Estate

Joomla! is a kind of construction kit that lets you, once it is installed on the server, create and maintain your website. Joomla! is like a house that you build on a property of your choice and that you can furnish gradually. Thus, to a certain extent, it is real estate.

Stop! I was talking about mobility all the time and now I'm asking you to build real estate? Have no fear, the real estate you build is physically at one place (your server), but is accessible from every place. To make a piece of real estate habitable, you need necessary services such as heating, electricity, and water supply. That is the reason your Joomla! is deposited at as safe a server as possible, where hopefully the electricity will never be cut (we are talking 24/7).

Just like your house, you also have a certain room layout in Joomla!. You have a room for presentations, for cooking and talking, for working, and a completely private one that you only show to good friends. Perhaps you also have a large room that integrates all areas.

It doesn't matter which room layout you decide on, you have to furnish your house, lay a beautiful floor, paper the walls, hang a few pictures, and of course, clean it regularly. The numerous guests leave traces that are not always desirable.

A visitor needs an address in order to find your house. This address has to be familiar to as many people as possible. Since there is no residents' registration office on the Internet, you have to be the one that takes care of the topic "How can I be found?"

Perhaps you also have a garden that surrounds your house and has different entry gates. There is an official entrance portal, a back door, and perhaps another small, weathered garden gate for good friends.

And perhaps you don't like such houses and would rather use trailers, tents, mobile homes, hotels, or maybe prefer community living and are glad to pay rent and don't want to have to think about all of the details. If you apply the last few sentences to your website, then you can already see how important it is to know what you want, who you are, and how you want to look at your community. One cannot *not* communicate! One can, however, be quickly misunderstood.

So plan your website on the Internet properly. Put thought into the texts, into possible interactive elements like a calendar or a forum, and of course, an area that only registered users are allowed to see. Think about prompts that direct and don't patronize users and take a look at how others do it. Talk with the people you want to address through your website and invest your heart and soul into those things that are absolutely crucial for the success of your website.

Joomla! Versions

As with all software, there are different development versions with Joomla!. The Joomla! team published a roadmap (<http://dev.joomla.org/content/view/23/61/>) on September 1, 2005, that started with Joomla! version 1.0.

The first Joomla! version consequently received the number 1.0, in order to not be confused with existing Mambo versions. Version 1.0 is a revised version of the last Mambo version 4.5.2.3. The revisions relate to the new name, known errors, and security patches.

There were ten Joomla! versions released during the course of the last year, which have improved and corrected a lot of small details in the code. If you have followed the development, you have probably noticed that Joomla! has become more and more reliable from version to version.

Numbering System of Joomla! Versions

Joomla! versions abide by the X.Y.Z system:

- **X = major release number:** This is incremented whenever profound changes are made at the source-code level. The version with the higher number sometimes is not compatible with earlier versions.
- **Y = minor release number:** This is incremented whenever significant changes to functionality are made. The higher version number is usually compatible (with minor customizing) with earlier versions.
- **Z = maintenance release number:** This is incremented whenever errors are repaired and security gaps are plugged. An increase of this number indicates only minor changes and very minor new features. These versions are fully compatible with the versions of the same X and Y number.
- **Full release:** This is a change in the X and Y system. With these, alpha and beta test periods are given. The length of the test periods is not fixed and is at the discretion of the development team. Beta versions should be available for testing for at least three weeks in order to give component developers the chance to customize their components.
- **Maintenance release:** This is a release that can be used immediately.

Version 1.5.0 represents the first full release after a year. There are alpha and beta versions – third-party developers are customizing their components; the community is testing the software for any incompatibilities to the prior version. This process has largely been going on unnoticed since the fall of 2005. The concepts for version 1.5 were already quite concrete at that time and in February 2006 the first alpha version was released. The first beta version was released on October 12, 2006. By the time this book is published, we might have a Beta2 or a Release Candidate.

Roadmap

This roadmap can, of course, change at any time, it does, however, represent a good framework for orientation.

Version	Date of Release	Comments
Mambo 4.5.2	17. Feb. 2005	Last stable version of Mambo
Joomla 1.0.x	From Sep. 2005	Transfer of Mambo version 4.5.2.3 Corrections of bugs and security patches Last stable version of Joomla! 1.0.12
Joomla 1.5	Beta 1: 12 October 2006	Internationalization (total changeover to UTF-8) Administration interface capability for every language User plug-ins Database: Support of MySQL- and MySQLi-database servers FTP system, to avoid the PHP safe mode with providers Fundamental changes and overhaul of the structure, the framework, of Joomla! itself and with that preparation for the possibility to create barrier-free websites with Joomla! Separation of programming logic and presentation Improvement of search engine friendliness (SEF) Reworked caching mechanism
Further development	No date given	New access control system New JavaScript framework
Joomla 2.0		Better search engine support Barrier freedom Version control Update mechanism Virtual file system Support of more databases

Joomla! roadmap – Status as of February 2006

Changes in Detail

As can be seen from the table, the 1.5 version is the first true Joomla!. The Joomla! team spent the first year stabilizing the inheritance from Mambo under the Joomla! name and charting their own direction. The changes in Joomla! 1.5 clearly reveal the direction of future developments.

Internationalization

With respect to internationalization, the following features are to be added in Joomla! version 1.5:

- Every piece of static text can be translated into language files. This is in particular relevant for the administration area, which up to now was only available in English.
- Support of scripts that are written from right to left (i.e. Arabic, Hebrew, Farsi, and Urdu).
- Complete changeover to the UTF-8 character set for coding and displaying all characters in Unicode (<http://en.wikipedia.org/wiki/UTF-8>).

User Plug-ins

Mambots are now called plug-ins, and user plug-ins now join content, system, and other plug-ins. Alternative login mechanisms from external programs, among others, can be used with the help of user plug-ins.

XML-RPC support

XML Remote Procedure Call (XML-RPC) is a specification that allows programs on different systems in different environments to communicate with one another. All of the important programming languages are supported and there are libraries that change the code into XML-RPC (<http://en.wikipedia.org/wiki/XML-RPC>). Joomla! also offers such an interface. With it, for instance, it is possible to describe an image from Flickr (<http://www.flickr.com>) or an article with the w.blogger (<http://www.wbloggar.com/>) editor that is being distributed in blogger circles.

Support of Several Databases

Joomla! 1.5 contains an abstraction interface that makes it possible to run Joomla! with various database versions. However, only one of these databases can be used for each particular Joomla! installation. At the moment MySQL 4.x and 5.x are supported. Additional databases will be supported in the future.

FTP System

An FTP interface has been added to avoid problems with file access rights. Therefore installation of new components and other uploads can be handled via PHP upload and via FTP. The restrictive (but reasonable) approach of the service providers in terms of the PHP language has made the installation of extensions and the downloading of files in general more difficult.

Overhaul of the Joomla! Framework

There has been no such thing as a framework in terms of a packaged kit for Joomla! functionality so far. It did, however, become crystal clear last year that the old Mambo source code had to be improved just about everywhere. It became necessary to rewrite and code Joomla!'s functionality cleanly. A framework has to be flexible, scalable, separated from the output, and above all be comprehensible so that a third-party developer can write good components in a reasonable amount of time. A proprietary API (**Application Programming Interface**) is essential for that.

Web Accessibility

Web accessibility is an important topic, which, in Germany, has been a legal obligation since January 1, 2006. The W3C (<http://www.w3.org/>) has written standards for it. Joomla! 1.5 has the tools to comply with these standards (<http://www.w3.org/TR/WAI-WEBCONTENT/full-checklist.html>).

These standards can be achieved by the complete separation of the HTML code that is created from the business logic in the programs that run under it. This statement applies to the front end at the moment. The administration area is also scheduled to become barrier free in later versions.

Search Engine Friendliness

Support for search-engine friendly URLs has been removed from the Joomla! core and swapped into a plug-in. This enables the use of extensions, which was very difficult before.

Google Summer of Code Projects

In 2005, Google (<http://code.google.com/summerofcode05.html>) supported talented students and their ideas for certain Open Source projects with \$ 4,500 each. The results of these projects will be and have been gradually integrated into Joomla!. Five of the six projects were completed (http://www.opensourcematters.org/soc/index.php/Main_Page). At the time of writing as well, there were students programming for Joomla! and being paid by Google (<http://code.google.com/summerofcode.html>).

Terms, Concepts, and Deliberations

These projects include the following (<http://code.google.com/soc/joomla/about.html>):

- Java XML-RPC application: J!Explorer (an administration group of objects written in Java)
- Joomla! Cross Database Support and Node Based Schema (the already discussed database support including the installation of new components)
- Package Management System for Joomla! (this is an automatic update procedure, similar to those used by operating systems and various applications such as Firefox)
- Joomla!/AJAX Integration (AJAX-DHTML integration to use JavaScript as a dynamic part of Joomla!)
- Joomla! Accessibility: Updating for WCAG 2.0 guidelines (this has to do with the Joomla! code that has to be customized to the needs of barrier free websites)
- USER & Access Management (making the presently very static user administration more flexible)

Joomla! Features

The following is a listing of Joomla! features:

- Free source code
- A large and eager community of users and developers
- Simple workflow system
- Publishing system for content
- File manager for uploading and administering files
- Content summaries in RSS format
- Wastepaper basket
- Search-engine-friendly URLs
- Banner management
- Multilingualism for website and administration interface
- Administration interface that is separated from the website
- Macro language for content (Plug-ins)
- Caching mechanism to secure fast page creation with favorite pages
- Simple installation of additional extensions
- Powerful template system (HTML, CSS, PHP)
- Hierarchical user groups
- Simple visitor statistics

- WYSIWYG editor for content
- Simple polling
- Rating system for content

There are numerous free and commercial extensions at <http://extensions.joomla.org/> for:

- Forums
- Image galleries
- Document management systems
- Calendars
- And many more

Examples of Joomla! Pages

In order to get a feeling of what Joomla! pages look like and whether "the" Joomla! page even exists, have a look at a few of the pages.

Joomla.org

At www.joomla.org, for example, you can display the text in various sizes with a click of the mouse:

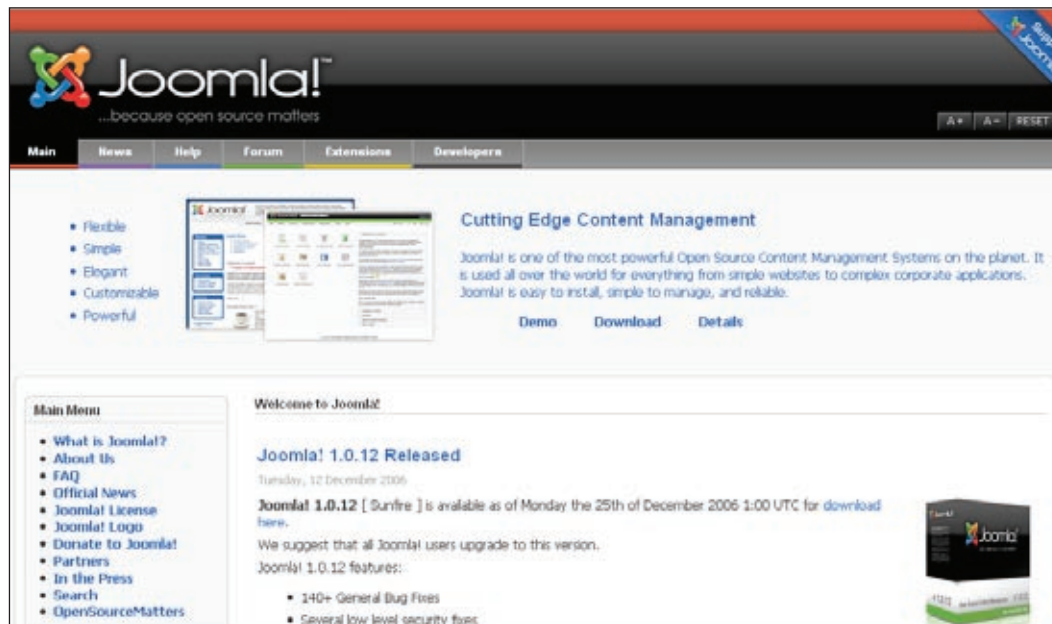


Figure 1.1: <http://www.joomla.org>

Terms, Concepts, and Deliberations

Ada Gaffney Shaff, USA

A cosmetics company with an online shop:

ADA GAFFNEY SHAFF  EXCLUSIVE SKIN CARE & FRAGRANCE PRODUCTS 

HOME | ABOUT | SKIN CARE | FRAGRANCE | CONTACT US |



FEATURED PRODUCTS

Oil-Blotting Hydrating Lotion



\$24.00

Discover the most effective skin care and anti aging products available ... a proven way to look natural and feel confident if you want to cleanse, hydrate and rejuvenate your skin. *The solution to skin renewal and beauty begins with Ada Gaffney Shaff.*

Call us toll-free!
1-877-257-0006

Click here to request a free brochure. 

SECURED BY **GeoTrust**
click to verify
26-Dec-06 09:14 GMT

Figure1.2: <http://www.adagaffneyshaff.com>

Janette Norton, France

A website of various tour guides by Janette Norton:



Janette Norton

Walking in France

Welcome

Home
Cevennes
Dordogne
Haute Savoie Nth
Haute Savoie Sth
Provence
Feedback
Contact me
How to buy

Login Admin
Username

Password

My name is Janette Norton and I have lived in France for more than 35 years, just across the border from Geneva, neatly sandwiched between the Jura (north), the Alps of the Haute-Savoie (south) and the Swiss Alps (east). I developed a passion for walking or trekking while I was a guide in Switzerland in my twenties. More recently I decided to record my walks on dictaphone and then on computer for the benefit of myself and friends in the region. Over 10 years I have published five books of walks in different regions of France.

I have decided to start this website to talk about my books, mainly to add information and give some useful updates or anecdotes. I am especially interested in feedback from walkers who have used my descriptions, so please email me if, for example, my directions were difficult to follow, or paths have been recently re-routed. I know from bitter experience that one may return to the same walk after a few years to find it partially obliterated by a housing estate, modified boundaries of private



Figure1.3: <http://www.janettenorton.info>

Urth.tv, USA

A social platform for citizens of the world!:




Figure1.4: <http://www.urth.tv>

unric.org, Europe

Alex Kempken sent me this example at the last minute. Not only because the United Nations works with Joomla!. Particularly interesting in this case is the administration of 5,000 articles from about 15 authors in 13 languages! Among others in the Greek language with a UTF-8 character set. In addition, this website uses several of their own extensions.



Figure1.5: <http://www.unric.org>

 You can find a listing of Joomla! websites in the SITESHOWCASE of the forum at <http://forum.joomla.org/index.php/board,58.0.html>.

Summary

In this chapter we took a quick look at the history of Joomla! and discussed about Joomla! as a CMS. Then we learned about the features and the changes to take place in Joomla! version 1.5. Lastly we also saw some of the websites powered by Joomla! In the next chapter we will learn about installing Joomla! version 1.5.



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

2

Installation

The installation of Joomla! is a matter of five minutes. If you have installed it before, you can do it in two minutes. You will have the opportunity to test this in Chapter 8. In order to be able to even start the installation, your development environment must include a web server that supports PHP and a database that is supported by Joomla!. Joomla! is one piece of the puzzle in this so-called Client-Server System.

Client-Server System

A Client-Server System is a network structure, in which an unlimited number of work stations (clients) can access services offered by a central server. The server is responsible for the delivery of the services. The client supplies the user interface, which is a web browser in our case.

Accessing a Joomla! Website on the Internet

This is a simplified list of what has to happen in order for a computer to be able to access a Joomla! website on the Internet:

- Establish an Internet connection via a provider.
- Call up the desired web address (URL) from the browser.
- The browser makes contact with the web server.
- The web server sends a query to the PHP-language interpreter that is installed on the server.
- The PHP interpreter requests the necessary data from the database.
- The PHP interpreter creates HTML or XHTML code, depending on the web server.
- The web server delivers the page or the generated code to the client browser.

Installation

- The browser discovers references to CSS and other files (images, flash elements, etc.) in the page just received and requests these separately from the web server.
- While loading all the necessary resources, the browser attempts to parse and to render the page, in other words to take it apart and to display it.

You can see from this course of events that the installation consists of a lot of components that have little direct relation to Joomla!.

Technical Requirements for Joomla!

Joomla! requires:

- An installed and functioning web server, for instance, Apache version 1.13.19 or later or Microsoft IIS.
- PHP scripting language version 4.3 or later, and support for MySQL and Zlib has to be compiled. **Zlib** is a library that enables PHP to read file packages that have been compressed with a ZIP procedure.
- The database system – MySQL from version 3.23.x on or with Unicode character sets MySQL from 4.1.x on.

Necessary Elements for a Joomla! Installation

You need all of the components mentioned above to be able to install a Joomla! system for yourself.

PC, browser, and Internet connection are usually available. There are a number of options for web server, PHP interpreter, and database.

You can:

- Set the system up locally on your PC
- Set the system up on a server in a company's Intranet
- Rent a virtual server from a provider
- Rent or purchase a server from a provider (with root access)

You can also have the web server and database located on computers that are physically separated from each other. You can also, of course, use different brands of web servers, versions of PHP interpreters, versions of MySQL databases. And on top of that, you can install and operate these components on different operating systems.

This freedom in the choice of resources sometimes confuses the layperson, therefore we will discuss a few typical scenarios in this chapter.

Local Test Environment

In the scenario where you are at home or in your office and want to set up a Joomla! website, then you can use any of the following operating systems.

Windows Operating System

For a Windows operating system you can use either of two web servers:

- Windows XP Professional comes with a web server, the Internet Information Server. You still have to install PHP and a database, and then you're ready.
- You use a preconfigured package (*XAMPP*), unpack it on your computer, and everything you need is there.

Linux Operating System

Here it depends on which distribution you have used. All of the distributions allow simple installation (with a click of the mouse) of the Apache, PHP, and MySQL packages. At times, depending on distribution, they may already be preinstalled. So you can:

- Use the programs contained in the distribution
- Use a preconfigured package (*XAMPP*), extract it on your computer, and everything you need is there

Mac OS X Operating System

In Mac OS X operating system, you have a default web server (Apache) in your system that you have to activate, but unfortunately no PHP. There is no official version of PHP for Mac OS X, but there is a PHP Apache module that you can install (<http://www.entropy.ch/software/macosx/welcome.html>). There are also executable versions of MySQL for Mac OS X, which can be installed after the relevant download (<http://dev.mysql.com/downloads/mysql/5.0.html>).

So you can:

- Use the installed Apache web server and install the missing software.
- Use a beta version of the preconfigured *XAMPP* package for Mac OS X. Extract it on your computer and everything you need is there.

Production Environment

You have several options here as well.

Rented Virtual Server

You can rent a web-space package with database, PHP support, and often also your domain name from a provider. In this case you have a functional environment and you can install your Joomla! into it. Consult your provider as to the version choices (PHP, MySQL).

Your Own Server

You rent a server from a provider and install the operating system of your choice. You are the administrator of the system and can work from your PC at home.

Before you venture into the wilderness of the Internet, you should first practice on your local computer. This is an advantage as there are no connection fees, it is very fast, and you can practice at a leisurely pace. You may even have a small local network at home where you can install Joomla! on one computer and access it from another.

To make the necessary downloads a little easier, I have gathered all of the downloads discussed in this book on the Packt website (www.packtpub.com/support) and on the <http://www.cocoate.com/links> website. A list of file packages can be found in the Appendix. These files are for all intents and purposes suitable for local installation, since the examples in this book can be reconstructed that way.

Remember, however, that there are more current versions on the respective project sites on the Internet.



If you install Joomla! in the wild, on a server on the Internet, you should always use the latest stable version. *Never* install a Beta version for an actively used site.

Setting Up the Local Server Environment

To install Joomla! locally, you have to set up the appropriate server environment as described previously.

Windows

Due to the user-friendliness of Windows, over 90 percent of computers work with Windows as operating system. Unfortunately, Apache web server, MySQL database, and PHP are not included with Windows. A practical approach would be to install each of these programs separately, or grab a preconfigured package (this is very practical and it will save your nerves).

Log on to the system in administrator mode. If you don't know you have administrator rights, check your account type: click **Start | Control Panel | User Accounts** and ask your administrator to change your rights if required:

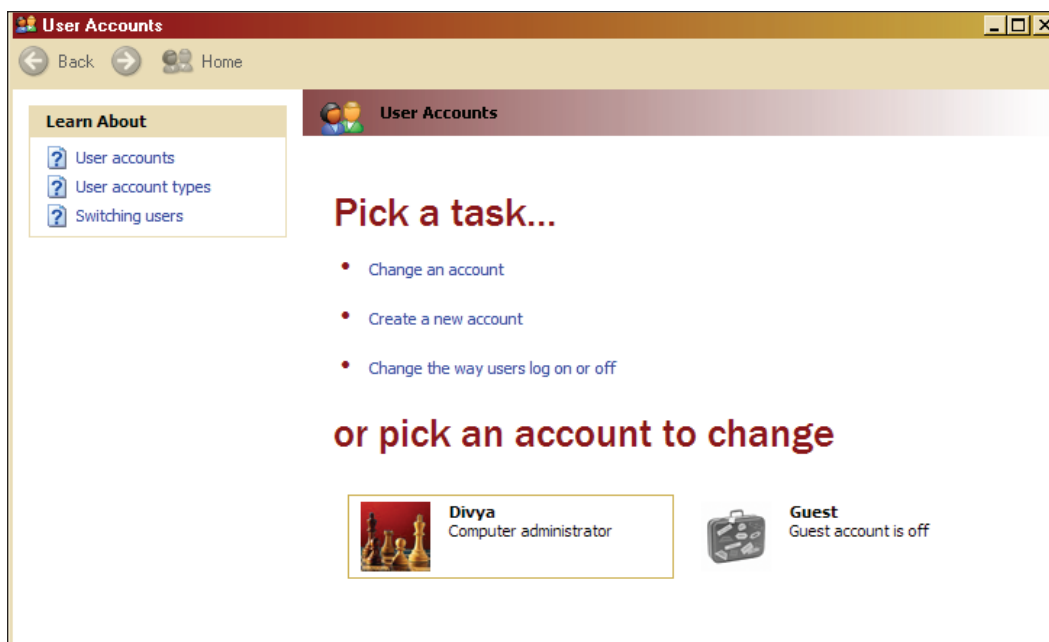


Figure 2.1: User Accounts in WinXP

XAMPP for Windows

XAMPP is a project of Kai 'Oswald' Seidler and Kay Vogelgesang. For several years now, these two have been creating a complete development environment with the ingredients: Apache, MySQL, PHP, Perl, and various extensions. XAMPP can be downloaded from <http://www.apachefriends.org/> as a ZIP archive for various operating systems. This is an immense advantage for people like you and me, who are primarily interested in Joomla! and not so much in how all of it works. The entire installation can also be removed from the computer with one mouse-click without leaving a trace. Kai and Kay have also written a book about XAMPP, *The XAMPP Handbook*, (ISBN 3-8273-2281-2 available only in German).

To download and install XAMPP:

1. Download the `xampplite-win32-1.5.5.zip` file from the website <http://www.apachefriends.org/en/xampp.html> and extract it on the local drive. The directory `xampplite` is created.

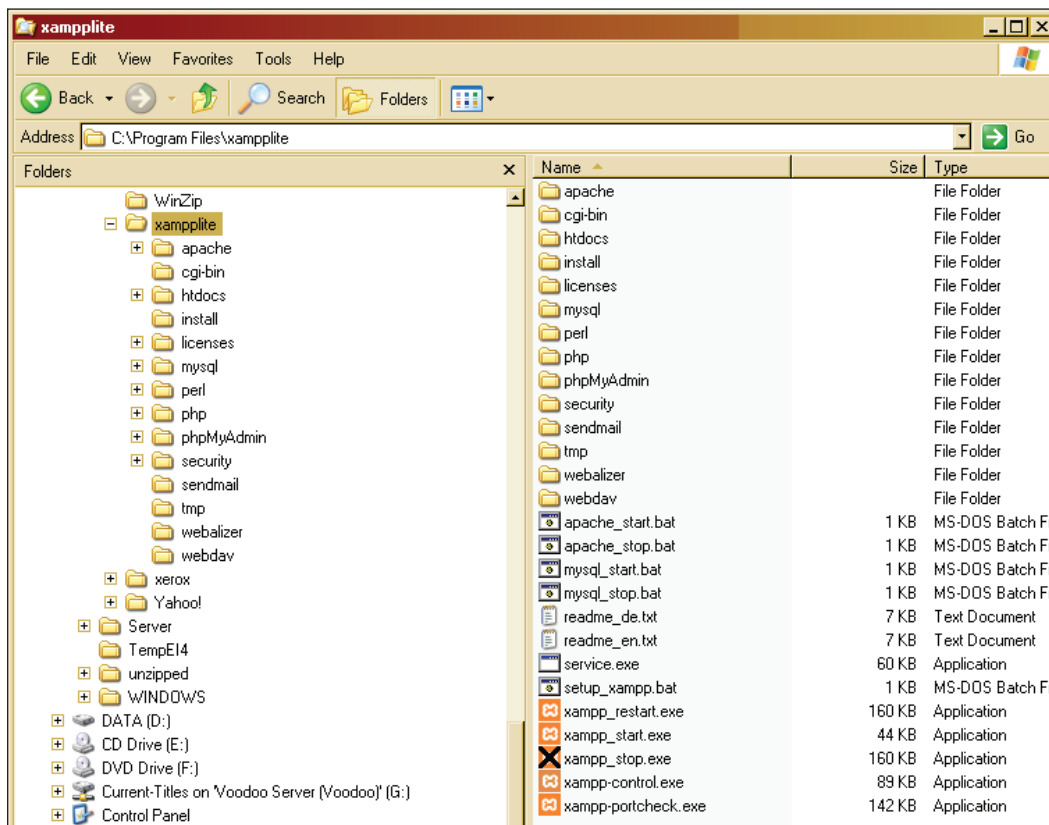


Figure 2.2: The xampplite Directory

2. Open the `setup_xampp.bat` file from the `xampplite` directory. XAMPP makes no entries in the Windows Registry and sets no system variables:

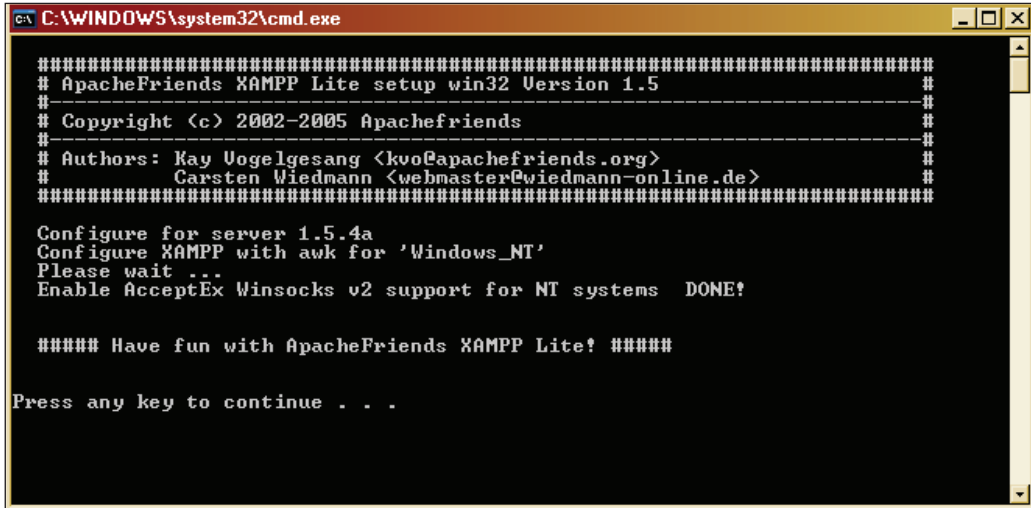



Figure 2.3: Executing `setup_xampp.bat`

3. Now you can start the server. PHP starts automatically as a module. Open `[Drive]:\xampplite\xampp_start.exe`. A command-prompt window opens, which indicates that Apache has started.

 The command window can be minimized, but closing it will terminate the Apache web server and MySQL.

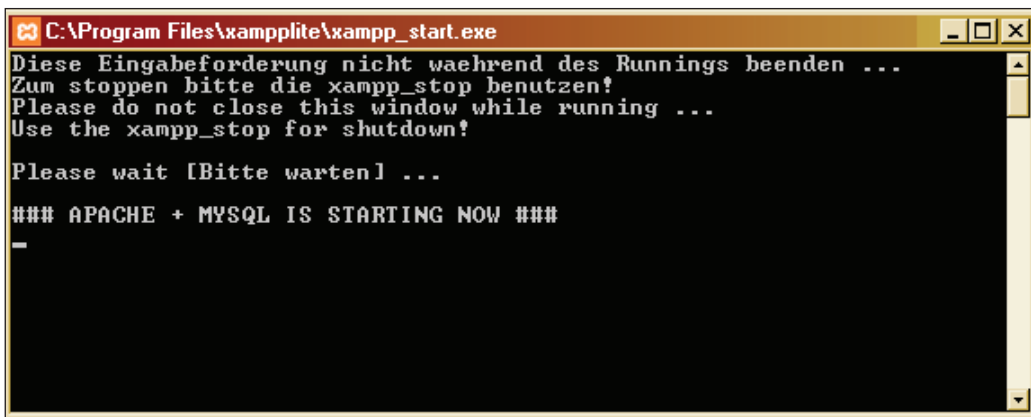


Figure 2.4: Start `xampplite`

Installation

- Open your Internet browser and enter `http://127.0.0.1/` or `http://localhost/`. You should now see the XAMPP start page. Click on the **English** link and the following page should appear:

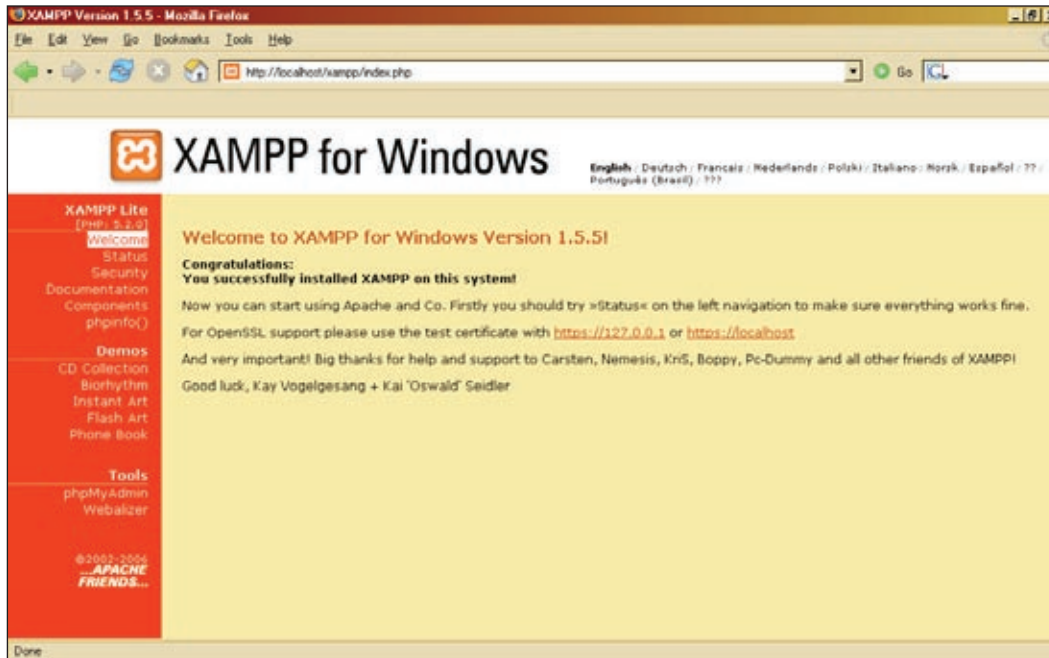


Figure 2.5: Start Page of XAMPP for Windows

The document directory of your website is `[Drive]:\xampplite\htdocs`. This directory contains all of the pages that are accessible by a remote computer on the Internet. Read the included `readme_en.txt` file for additional information about the necessary passwords.

To uninstall the package, close all current servers and simply delete the `xampplite` directory.



If the Apache web server does not start when you start `xampplite`, it could be that a different service already occupies port 80 of your computer. One very popular program that occupies port 80 with its older versions in its default setting and thereby preventing another program from using it, is Skype. You can either change the port number in Skype or start `xampplite` first and Skype after.

Linux

With Linux, everything is usually simpler. Different distributions with different standard configurations are available. Usually our dream team is pre-installed and just needs to be started. A XAMPP version can also be installed for Linux. My opinion, however, is that it makes more sense to grab the original programs. The installation is done by a package manager and is very simple.

SUSE > 9.1

With the help of a configuration program, YaST, you can check whether Apache, MySQL, and PHP are already installed. If that is not the case, select the appropriate packages for installation and let YaST install them.

These are the packages — `apache2`, `apache2-devel`, `apache2-mod_php4`, `mysql`, and `php4-mysql`.

You can find these packages via the YaST interface on your SUSE distribution media or on special package pages on the Internet (<http://www.rpmseek.com/>).

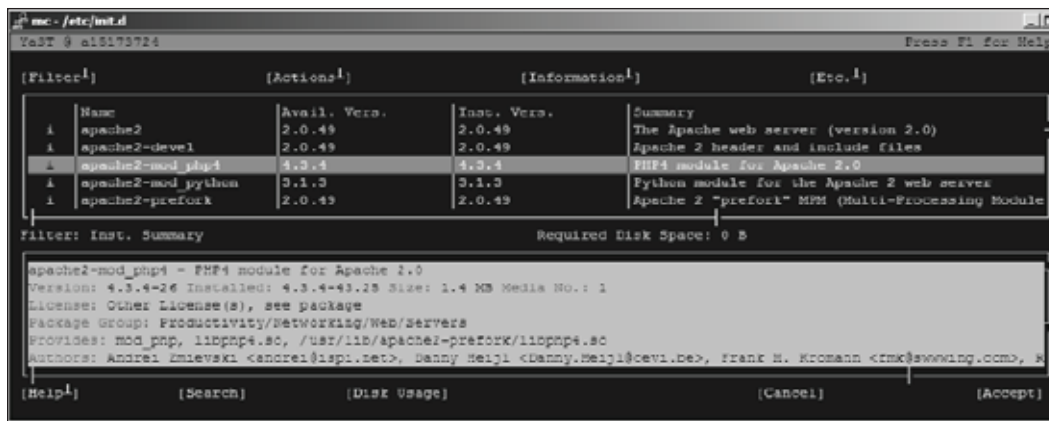


Figure 2.6: YaST Called from a Windows-PC in a Shell

Start the Apache web server with the `/etc/init.d/apache2 start` command. Start the MySQL database server with `/etc/init.d/mysql start` command

You can stop both of the servers with the `stop` command. By typing `help`, you get an overview of all of the parameters.

Debian/Ubuntu

With Debian and Ubuntu, the agent of choice is `apt` – you can install Apache, MySQL, and PHP with the program `apt`.

```
apt-get install [packetname]
```

The following are the packages in detail:

- `apache-common`: Support files for all Apache web servers
- `php4`: A server-side, HTML-embedded scripting language
- `mysql-common`: MySQL database common files (e.g. `/etc/mysql/my.cnf`)
- `mysql-server`: MySQL database server binaries

You can find these packages automatically over the Internet or on the Debian CD/DVD by using `apt`.

Now start Apache with the command `/etc/init.d/apache2 start` and MySQL with `/etc/init.d/mysql start`.

Your Own Server at a Provider

If you have rented a complete server from a provider, then you usually have shell access and free choice of the Linux distribution that you want to use. In addition, the system is preconfigured and contains all necessary file packages and configurations. Usually special administration interfaces, such as SiteBuilder (<http://www.swsoft.com/en/products/sitebuilder/>) or Plesk (<http://www.swsoft.com/en/products/plesk/>), are used for configuring these servers. You can comfortably start, stop, and configure your server and the Apache and MySQL services from a browser interface with this tool.

Joomla! Installation on a Virtual Server on the Net

This topic is very complex, since there is an unmanageable number of providers and an even more unmanageable combination of installed Apache, PHP, and MySQL versions and Webspaces administration tools such as Visas and Plesk.

These are the sticking points:

- A PHP safe mode, possibly activated in the `php.ini`
- Prohibited conversion of URLs with Apache because of the non-activation of the so called *rewrite engine*
- Directory rights in Linux that are set differently than in Windows

In principle the simplest approach that actually always works is the following:

1. Load the Joomla! 1.5 beta .zip file onto your local PC and unpack it in a temporary directory.
2. Load the just unpacked files by means of FTP onto your rented server. The files must be installed in the publicly accessible directory. These directories are usually named `htdocs`, `public_html`, or simply `html`. You can specify a subdirectory within the directory into which you install your Joomla!. Many web hosts allow you to link your rented domain name to a directory. This name is necessary to call your website from a browser.
3. You have to find out what your database is called. Usually one or several databases are included in your web-hosting package. Sometimes the user name, database name, and password are fixed; sometimes you have to set them up. There is usually a browser-based configuration interface at your disposal. You can see an example of such an interface in the following figure. You will need these access data for Joomla!'s web installer.

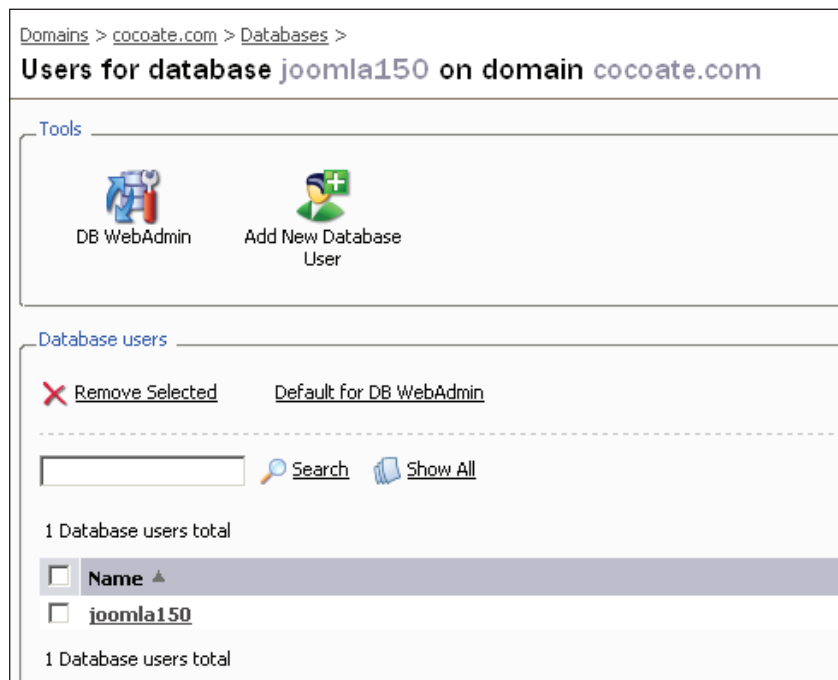


Figure 2.7: Plesk's Web Configuration Tool

You can get going after you have loaded these data onto your server and are in possession of your access data.

Joomla! Installation

To install Joomla!, you need the source code. Download the `Joomla_1.5.0-Beta-Full_Package.zip` package and save it on your system (see the Appendix for the download options).

Selecting a Directory for Installation

You have to decide whether Joomla! needs to be installed directly into a document directory or a subdirectory. This is important, since many users prefer a short URL to their homepage.

An Example

If Joomla! is unzipped directly in `/htdocs`, the web page starts when the domain name `http://www.myhomepage.com` is accessed from its local computer `http://localhost/` and/or from the server on the Internet. If subdirectories are created under `/htdocs/`, for example, `/htdocs/Joomla150/` and we unzip the package there, we have to enter `http://localhost/Joomla150/` in the browser. This isn't a problem locally, but doesn't look good on a production Internet page.

Some HTML files and subdirectories, however, are already in `/htdocs` in the local XAMPP Lite environment under Windows, which, for example, displays the start page of XAMPP Lite. In a local Linux environment, a starting page dependent on the distribution and the web server settings is also displayed.

Directory

I recommend that you create a subdirectory under the document directory named `Joomla150` in Windows by using Windows Explorer. (With Linux, use the Shell, KDE Konqueror, or Midnight Commander.)

```
[home]/htdocs/Joomla150/
```

The directory tree in Windows Explorer should look like this:

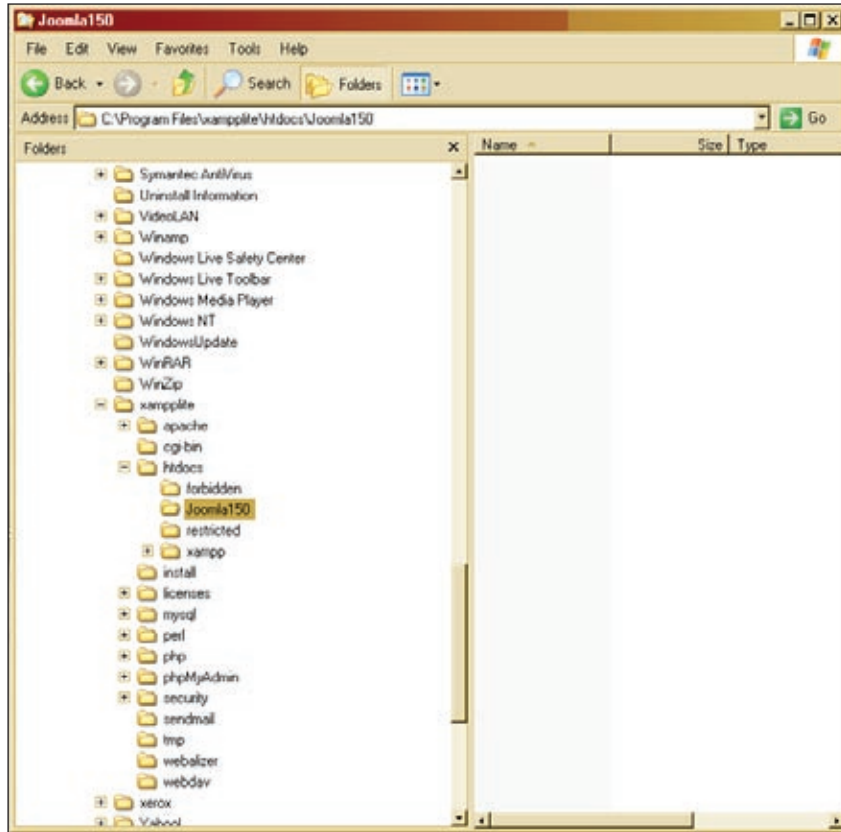


Figure 2.8: Joomla! Directory

An empty index appears in the XAMPP Lite version when the URL `http://localhost/Joomla150` is entered in the browser:



Figure 2.9: Display of the Apache Directory

Installation

With Linux or with another configuration it can happen that you get a message saying that you don't have access to this directory. This depends on the configuration of the web server. For security reasons, the automatic directory display is often deactivated in Apache's configuration. A potential hacker could draw many interesting conclusions about the directory structure and the files on your homepage and target your computer for an attack.

For security reasons, you are usually not allowed to access the appropriate configuration file of the Apache web server. Should you be able to, you should leave the content directories deactivated and/or only activated for those directories that contain files for downloading.

Extracting

Now you can finally extract the package file Joomla_1.5.0-Beta-Full_Package.zip into the designated directory. The file package is a so-called compressed tarball. In Windows XP, this package can be directly extracted with the file explorer. In all other versions of Windows a separate extraction program is required, for example, the shareware program Filzip.

In Linux, use the following command from a shell:

```
$ tar -zxvf Joomla_1.5.0-Beta.tar.gz
```

After unpacking, the following directories and files will be seen in Windows Explorer:

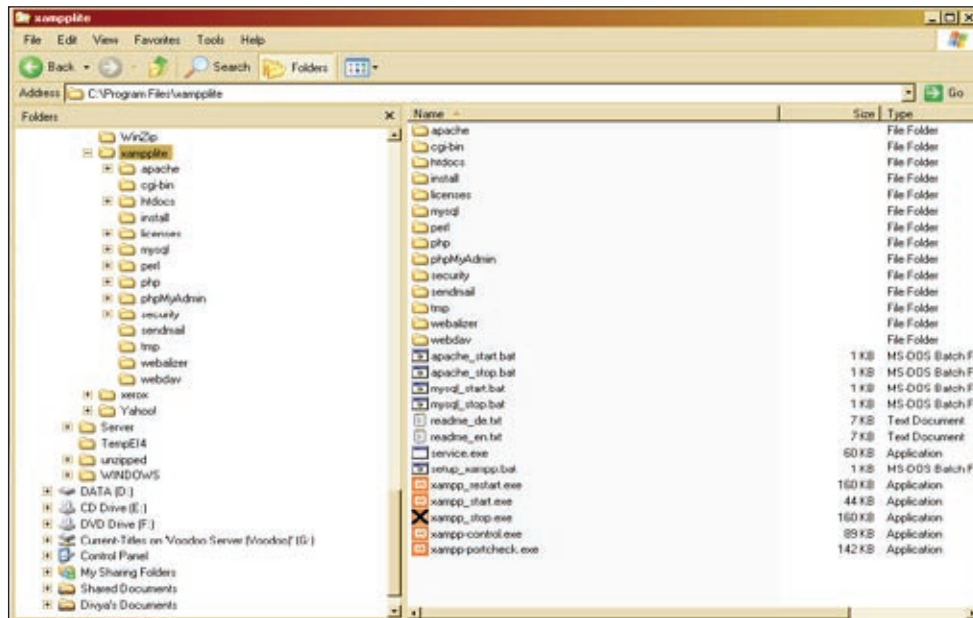


Figure 2.10: Joomla! Source Code Files

This structure is the same on all operating systems; only the presentation differs. The following figure shows a presentation in an FTP client where the local PC is in the left window and the remote web server in the right one:

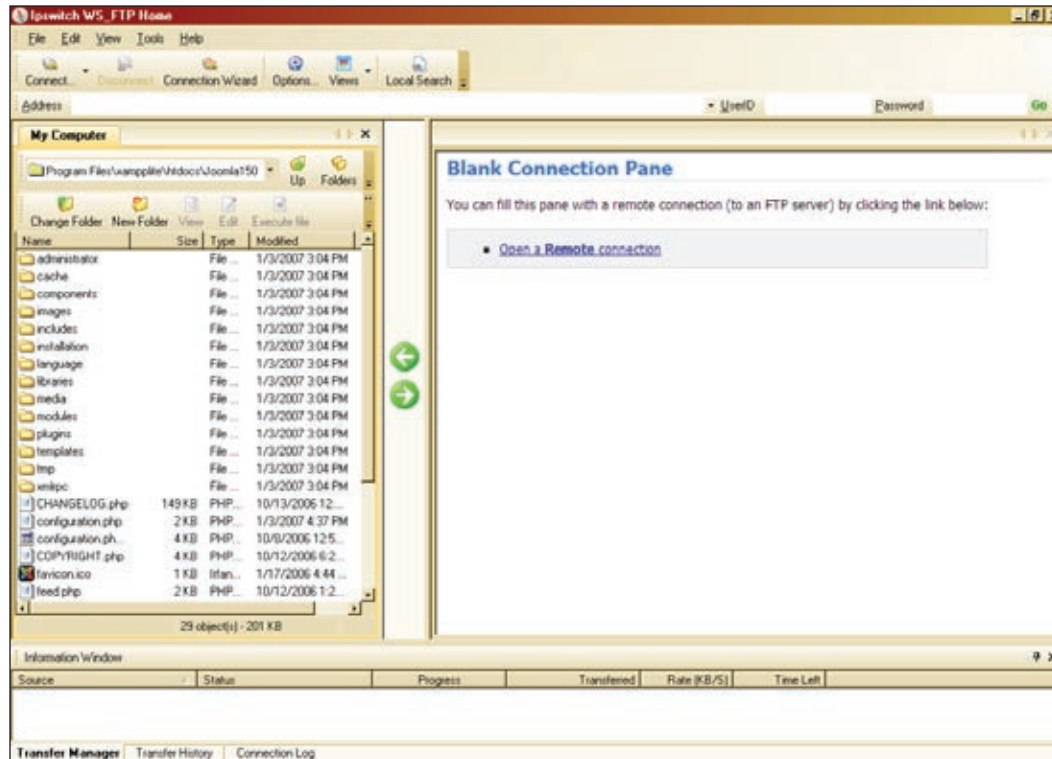


Figure 2.11: Joomla! Files in the FTP Client WS_FTP

Joomla! Web Installer

From now on, everything is going to go lightning fast because the Joomla! web installer will be taking over command. Go to the URL <http://localhost/Joomla150/>.

Step 1: Choose Language

Choose Language is the first of seven installation steps. Select the desired language and click on the **Next** button.

Installation

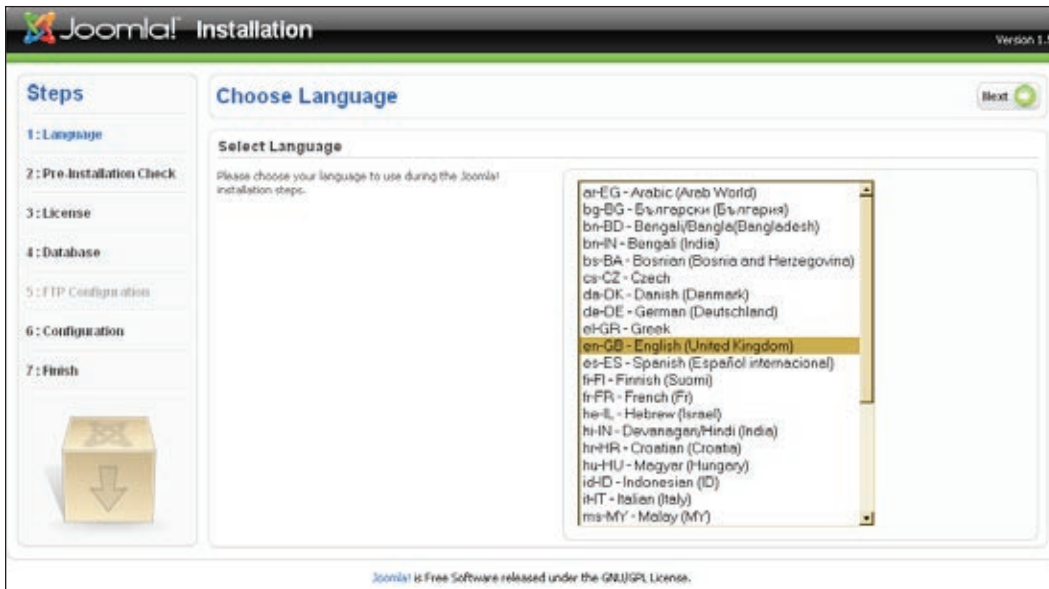


Figure 2.12: Installer: Language Selection

Step 2: Pre-Installation Check

Next, you will see the **Pre-Installation Check**. This survey should help you determine whether your server environment is suited for a Joomla! installation.

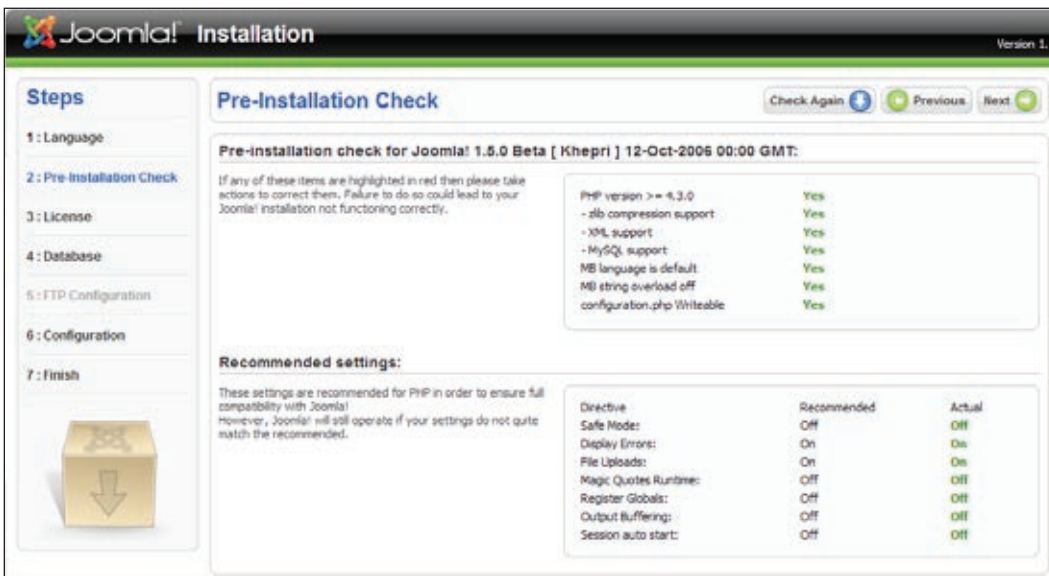


Figure 2.13: Installer: Installation Survey

A lot of green test results is a good sign. Depending on your configuration there can be differences here.

The web installer takes the configuration settings of the web server (in our case Apache), PHP, and the operating system into consideration. On Unix-based systems (Linux, Mac OS X) attention should be given to writing rights. This is in particular important for the `configuration.php` file. At the end of the installation, this file gets created with its individual values. If the installer does not have write rights, Joomla! cannot create this file and the installation will fail. If this happens, try to configure the rights appropriately and click on **Check Again**. If you are working with the XAMPP Lite solution under Windows, your screen will look like the previous screenshot.

Click on **Next** and you are in the licensing step.

Step 3: License

Every piece of software is licensed under certain conditions. Joomla! uses the GNU/GPL licence:

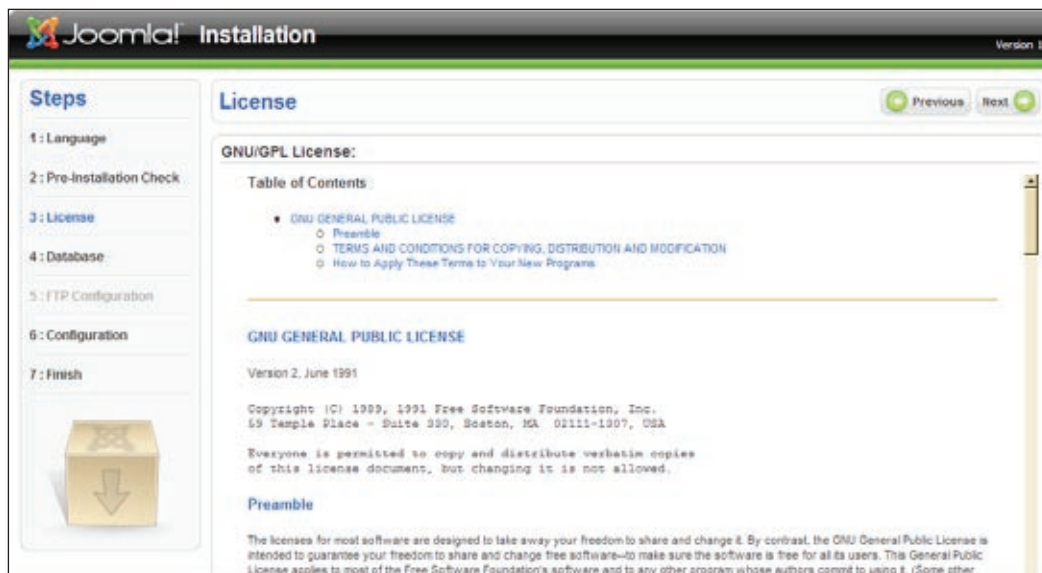


Figure 2.14: Installer: License

After you read this license click on **Next**.

Step 4: Database Configuration

In the fourth step, **Database Configuration**, your database parameters are queried. You can set up as many databases as required in an XAMPP Lite server environment and you have a MySQL user set up with the name `root` (without a password). The user `root` is the MySQL administrator and can do everything in the MySQL system. An installation without a password is a significant security risk. In the beginning, locally, in order to get the system up and running as quickly as possible, this is not a problem. But when you do the security check, you should absolutely provide your XAMPP Lite installation with passwords (`http://localhost/security/`).



Figure 2.15: Installer: Database Configuration

Enter the following parameters in a local XAMPP Lite installation:

Host Name: localhost

User Name: root

Password: [leave this empty – but be aware of the security risk!!]

In the following figure you will see a button **Get Privileges** and another button **Get Collations**. After you click the first button, Joomla! checks whether the database user has the necessary rights and the second button establishes what UTF-8 support is possible with your database. This button only appears in version 4.1.x of MySQL and subsequent versions.

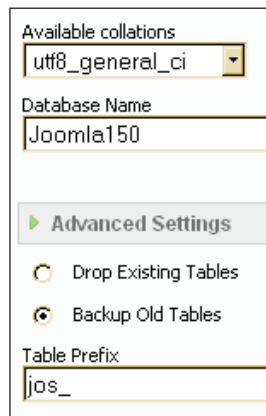


Figure 2.16: Installer: Database Configuration II

Now select a **Database Name**. An unlimited number of databases can be set up in a local XAMPP Lite environment since the user `root` has the right to do that. In an active environment with a provider, you presumably will have a set allocation of databases and the access data to the databases are usually predetermined. Here `Joomla150` is the database name used.


By clicking the green triangle in front of **Advanced Settings**, you can activate additional options. You can select whether the tables of an existing Joomla! installation should be deleted or be provided with the prefix `bak_`.

The MySQL **Table Prefix** is very practical. The web installer writes the text that you enter into the respective field in front of every table that is created. The web installer recommends `jos_` as the default

There is a simple reason for this. Sometimes you only get one MySQL database from an Internet provider. If you have to operate two Joomla! pages in this case, there would be a problem, since you cannot differentiate one table from the other. By means of a **Table Name Prefix**, it is possible to keep apart the tables of different Joomla! installations (`jos_smith_` or `jos_jones_`). In this case you should accept the default `jos_`. This prefix is also used to mark secured data (`bak_`).

Step 5: FTP Configuration


In order to stop problems with access rights and a possibly activated PHP Safe Mode before they start, in Joomla! 1.5 you have the option of using FTP functions for uploading and handling of files. This is not necessary in the local XAMPP Lite installation. But if you install Joomla! on the virtual web server of a provider, enter the FTP data here that your provider has given to you.

 In windows, FTP server is deactivated, but in Linux host it is activated.

Step 6: Configuration

This main step of the configuration is divided into three parts.

The first part of the configuration has to do with the name of your website. This name appears in the header of the browser window when someone accesses your website. This name is also used in other places, for example, with confirmation emails to registered users.



Site name	Joomla! 1.5.0
-----------	---------------

Figure 2.17: Installer: Configuration – Name of the Website

Select a meaningful name. For our example page, we have chosen the name Joomla! 1.5.0.

In the second part, you are asked to enter the administrator email address and the administrator password. A password is suggested in the left area; you can accept it or create your own – chose a similarly complicated one when installing on the Internet. You can use something simpler for a local installation. Best write the password on a piece of paper.



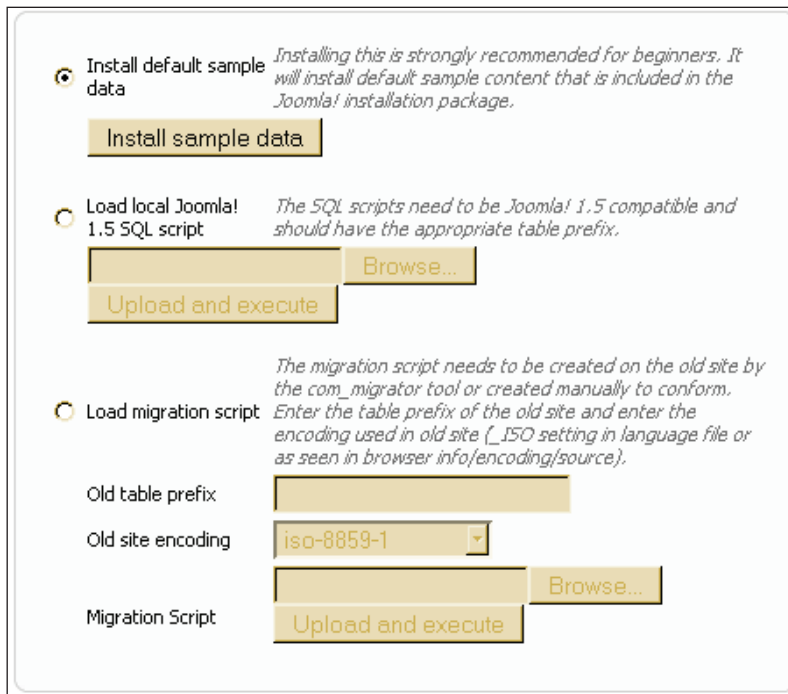
Your E-mail	divyam@packtpub.com
Admin password	*****
Confirm admin password	*****

Figure 2.18: Installer: Configuration – E-Mail, Password

In the third part, you specify the data that your Joomla! installation is to contain.

- **Install default sample data:** The data is the most important part of your installation. Joomla! allows you to install sample data. You can get a feeling for the program that way and experiment without any worries.
- **Load local Joomla! 1.5 SQL script:** You may have created a local Joomla! website and now may want exactly this data in an online database. In this case you can specify the `.sql` file that holds your Joomla! data, which you have exported from the local version. Make sure that this file has the exact Joomla! 1.5 database schema and is UTF-8 coded. More about this topic is covered in the Appendix.
- **Load Migration Script:** This option ports a Joomla! 1.0.x installation into a Joomla! 1.5 version. You will find more about this conversion in the Appendix.

We will take a look at the first option, **Install default sample data**, during the course of the book.



The screenshot shows the Joomla! installer configuration screen for data selection. It features three radio button options:

- Install default sample data:** This option is selected. A description reads: "Installing this is strongly recommended for beginners. It will install default sample content that is included in the Joomla! installation package." Below this is a button labeled "Install sample data".
- Load local Joomla! 1.5 SQL script:** This option is unselected. A description reads: "The SQL scripts need to be Joomla! 1.5 compatible and should have the appropriate table prefix." Below this is a text input field, a "Browse..." button, and an "Upload and execute" button.
- Load migration script:** This option is unselected. A description reads: "The migration script needs to be created on the old site by the com_migrator tool or created manually to conform. Enter the table prefix of the old site and enter the encoding used in old site (_ISO setting in language file or as seen in browser info/encoding/source)." Below this are fields for "Old table prefix" (text input), "Old site encoding" (dropdown menu showing "iso-8859-1"), "Migration Script" (text input with a "Browse..." button), and an "Upload and execute" button.

Figure 2.19: Installer: Configuration – Data

Installation

Click on the **Install default sample data** button, and the installer will load the data into your database and change the display.

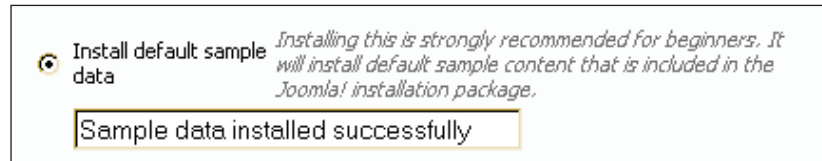


Figure 2.20: Installer: Configuration (Sample Data)

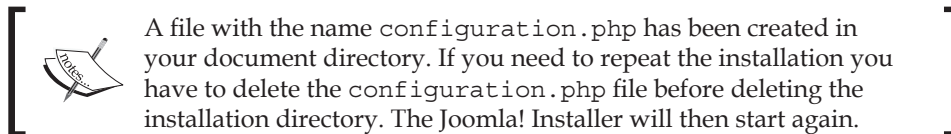
You should accept both suggested options with the file and directory rights. Joomla! automatically sets access rights for those parts of the system where files are uploaded and program parts are installed.

Now click on the **Next** button and your data are stored.

Step 7: Finish

The seventh and final step congratulates you on a successful installation—I congratulate you as well. There is a notice in bold text that prompts you to delete the installation directory. Take good heed of this notice, because your Joomla! website will not run if you don't delete the directory. You have installed your Joomla! and can now start to configure your website and enter content.

Readers who would like to follow a concrete example of how to build a website from scratch can go to Chapter 8



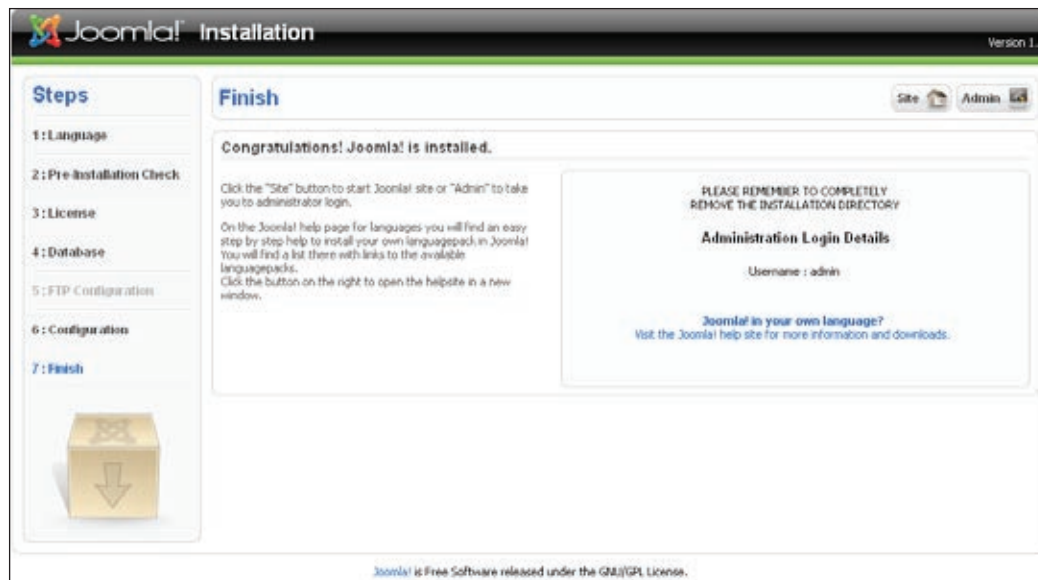



Figure 2.21: Installer: Completion

In addition, your login data will be displayed.

 But make sure that Joomla! gives you a new password if you have go back from Step 7, for instance if you want to change a setting or if the installation did not work properly. If you have lost or forgotten the administrator password, there is a solution in the Appendix.

The installation is now complete and you have a choice between the buttons **Site** (to view your homepage) and **Admin** (administration interface).

To take a look at your newly created homepage, click on **Site**. If you haven't deleted the installation directory as of yet, you will get a friendly reminder to delete it and to check out your page after you've done that.

Installation



Figure 2.22: Your Homepage directly after Installation

Summary

The result is pretty impressive. Look it over at your own pace, click on a few options and try to orient yourself. Lots of Joomla!'s functionalities are used on this homepage, which is loaded with sample data. We will take a good look at these in later chapters.

3

A Tour of Your New Website

Now that you have installed your homepage and carefully explored it, we can take a look at the result together. At first glance, these pages look a bit confusing. In principle, they are divided into a **front end** (your actual website) and a **back end** (the administration view of your website). Customers and web surfers see the front end; the back end is only accessible by co-workers and/or administrators.

Front End

You can see right away that the sample website illustrates a lot of the functions that Joomla! has to offer, giving you a good overview of Joomla!'s capability.

In order to get a better overview, I have marked and labeled the different areas in the figure overleaf. The art of design now consists of recognizing the elements that are important for your website, omitting the unimportant ones, and presenting them to the user in a logical, easy-to-understand, and attractive format. The result is always a compromise between functionality and organization.

A Tour of Your New Website

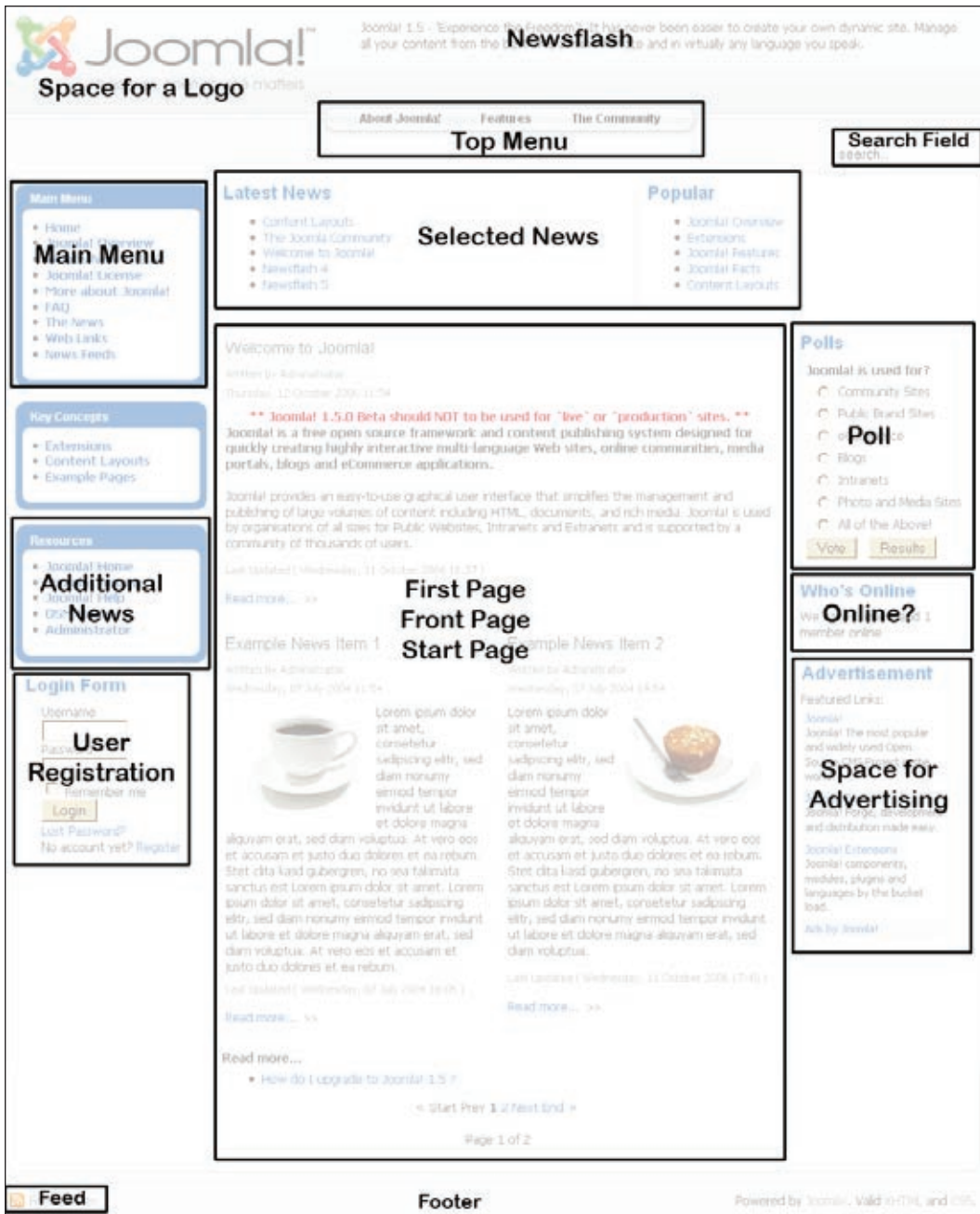


Figure 3.1: Sample Website

From the configuration, this structure reminds one of a portal like MSN or Yahoo!. On the left and the right there are boxes with clearly defined contents and in the centre is the news.

A so-called template determines the layout of the page. **Templates** are exchangeable and modifiable, meaning that the same content can be displayed in different layouts. Every daily newspaper would envy you for this functionality.

Let's go over the example layout a bit more closely. There are five different categories of areas on the page:

- Menus
- Content
- Advertising
- Additional functions
- Decorative elements

Menus

Menus are there to make navigation on the page as easy as possible for the user. There are different menus for different tasks. Joomla! has predefined four kinds of menus in the sample data. You can add as many additional menus as you want or, naturally, also disable them. The fourth menu, by the way, is not shown in our example. The registered user sees the so-called user menu after he or she has logged on.

Top Menu

The **top menu** is high up on the page. It gives the user quick access to the most important content of the page. Such a menu often displays products, contacts, and company information.



Figure 3.2: Top Menu

Main Menu

The **main menu** is the central navigation area of the page. There should always be a link to return to the first page. This menu should appear in exactly the same position on every page of the website. The main menu is an important orientation place for the user.



Figure 3.3: Main Menu

Other Menu

An additional menu, the **other menu** can pop up in all kinds of places (module positions):



Figure 3.4: Other Menu

Depending on the content and context of the page, it can make sense to offer additional menu options, such as a shop component.

Content

Finally the content that we want to manage is here!

What is Content?

Content can be a message, an editorial article, or a static page with explanations. Content can be also a dynamic link directory (<http://www.google.com/>), a shop (<http://dell.com/>), or a flea market (<http://www.ebay.com/>).

Content can also be something completely dynamic and open to everybody. The free encyclopedia Wikipedia, for example, uses a content administration system that allows everyone to change the content. This special form of content administration is called a **wiki** (<http://en.wikipedia.org/wiki/Wiki>). Everyone can change and even delete content. So far it is working amazingly well, despite the occasional argument about structure and content of articles.

The opposite of a wiki is static content, which, once written, is valid for a long time. For example, take this book. It will become outdated regarding the version numbers of the software discussed, although it has the advantage of explaining and illuminating the topic comprehensively in that connection. I produce content in a certain format and therefore operate a type of content management. Folders, flyers, stickers, business reports, and manuals are also usually of static nature. Created for a certain event, they become outdated or simply wrong after some time.

Many older web pages consist exclusively of static elements. On the Internet, however, the clock ticks a little faster. That which is complacently tolerated with books, folders, and other printed materials (after all, I can also read the book at the beach and in the subway), is regarded to be a serious shortcoming by visitors to your website. Nothing is worse for the image of your company than a four-year old static website with a button announcing "Powered by...", which refers to hopelessly outdated software.

The presentational possibilities of content are inexhaustible. They depend on the available terminal, bandwidth, and many other things that are in turn dependent on the user. The user of the message plays an increasingly important role – Who are your actual target group?

A platitude says: *Content is King!*

It depends on the content; every web agency would now probably crack a smile and get on with the daily job of creating the next website. Millions of dollars in advertising budgets for products such as frozen spinach or beer are proof of the fact that a successful communication also works without unique content. The statement that content is crucial, is, however, fundamentally correct. If you have nothing to say or nothing to offer, nobody will listen to you of his or her own free will. Since you probably don't have a million-dollar advertising budget, you also can't seduce people to read your content. No matter how beautiful websites without content may look, or how many terminals may display them, nobody will voluntarily visit them.



Plan the content of your site carefully! Everyone in the world can read it and use it against you in cases of doubt. You could become the victim of a litigation lawyer, disappointed colleagues, or other unpleasantries.

On the other hand, this has an unbeatable advantage: Everyone in the world can read your content, add his or her opinion to it, and contact you. What an opportunity!

You should be aware of both of these possibilities when you conceptualize content.

First Page/Front Page

Content is announced on the first page of the website, as illustrated in the following figure.

Welcome to Joomla!

Written by Administrator
Thursday, 12 October 2006 11:54

**** Joomla! 1.5.0 Beta should NOT be used for `live` or `production` sites. ****

Joomla! is a free open source framework and content publishing system designed for quickly creating highly interactive multi-language Web sites, online communities, media portals, blogs and eCommerce applications.

Joomla! provides an easy-to-use graphical user interface that simplifies the management and publishing of large volumes of content including HTML, documents, and rich media. Joomla! is used by organisations of all sizes for Public Websites, Intranets and Extranets and is supported by a community of thousands of users.

Last Updated (Wednesday, 11 October 2006 16:37)

[Read more...](#) >>



<p>Example News Item 1</p> <p>Written by Administrator Wednesday, 07 July 2004 11:54</p>  <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam</p>	<p>Example News Item 2</p> <p>Written by Administrator Wednesday, 07 July 2004 19:54</p>  <p>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam</p>
--	--

Figure 3.5: Frontpage

Content has an author, a date of preparation, a heading, a hook, and perhaps a picture. The hook is to make the visitor curious and to get him or her to click on a **Read more...** link in order to read the entire message.

The Latest Messages/The Most Often Read Messages

The message can be displayed in different formats. People are usually interested in the *newest* and in what *others* are reading. Because of that, our example layout has an appropriate area within which the last five messages are always displayed, and another area where the most read messages are announced. This second area is possible because Joomla! logs each hit on a message in the database and tracks the number of times it has been accessed.



Figure 3.6: Latest Messages/The Most Often Read Messages

Advertising

When your site becomes popular and if the content is right, you can sell advertising space. Advertising space usually means banner links. Banners are small graphics (in .gif, .jpg, .png, or .swf format) that induce the visitor to leave your website via a single click on the banner. If you really want that, look for a space in your layout and consider using it for advertising. (You may not be able to see this on your front end if this module is not applied to all your pages; you will learn more about this in Chapter 5.)

Banner Area

The banner area can administer text links and graphical banners. Here is an example of the prevalent banner size of 468 * 60 pixels:



Figure 3.7: Text Banner



Figure 3.8: Graphical Banner

Functions

Functions are elements needed to make interactivity possible. In Joomla! these functions are called **modules**. A module is something that takes up space on the web page and fulfils a certain function.

Login Area

A **login module** is important if you want to separate your website into a public and a protected area. The visitor then has to have a way of registering and of logging in. Perhaps he or she may even occasionally forget his or her password. The login module should be able to take all situations into consideration as illustrated in the following figure:

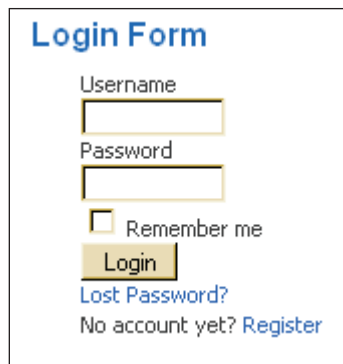


Figure 3.9: Login Module

Polling

Since our content is designed for certain target groups, we should now and then ask the group that actually surfs our site for their opinion. This is the simplest way of getting utilizable opinions about your site.

Joomla! has integrated a survey component, whose display module is on the sample page as illustrated in the following figure:

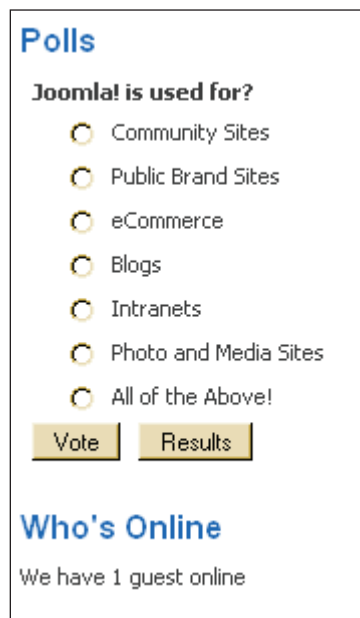


Figure 3.10: Polling Module

Who is Online?

This module is about communication and community. After the user has been able to see which messages are recent and particularly popular, naturally he or she would like to know who is navigating the site right now. In this case a distinction is made between guests and logged-in users as illustrated in the following figure:



Figure 3.11: Who is Online



Deliberate carefully about whether you want to offer such features on your website. If you claim in your content that you are the largest website in the U.S. and only one guest hangs out in your site, this will hurt your credibility. But if you do indeed constantly have 10-20 visitors and logged-in users, this is a good way to demonstrate dynamics.

Feeds

News feeds are becoming more and more popular. These are standardized collections of content (to some extent the content of your site), which can be processed further without the dead weight of the rest of the website. The Syndication module, as seen in the following figure, offers the website's news feed. You will learn more about this technology in Chapter 5.

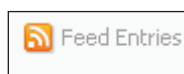


Figure 3.12: Syndication Module

Search Field

Similar to the **Back** button, the **Search** field also contributes greatly to the user friendliness of a website. Many sites have search fields. Often, however, the programs behind the search field don't scan the entire content of the website. With Joomla!, however, all pages are definitely scanned.

You can type a search term and press *Enter* on the keyboard. The result is a hit list, with the desired term visually emphasized.



Figure 3.13: Search Field

Decorative Elements

After so many functions, modules, and contents, the issue of design, corporate identity, and the *look* and *feel* of the website pops up.

A template represents the layout of the page and is laid on top of the content like a screen. Since it is hard to argue about taste or beauty, you have the option of providing different templates for the same content. For example, the look of your website in winter could be different than that in summer, or it could have a soccer look during the World Series (Chapter 6 covers the creation of your own templates).

In principle, a template consists of one or more logos, a certain color combination, selected character fonts and sizes, and as clever an arrangement of the available content as possible. The example template has the following logo:



Figure 3.14: Logo

After this tour and from your own experience from exploring the sample data, I am sure you can understand that the administration of contents can be a very demanding task. Above all, it is important to not lose sight of the overview.

Back End

The administration of the website takes place in the back end with the name **Joomla! Administration**. You can reach the **Joomla! Administration** under the URL [Domain name]/administrator/.

A Tour of Your New Website

If you are working with a local installation, the URL is:

`http://localhost/joomla150/administrator/`

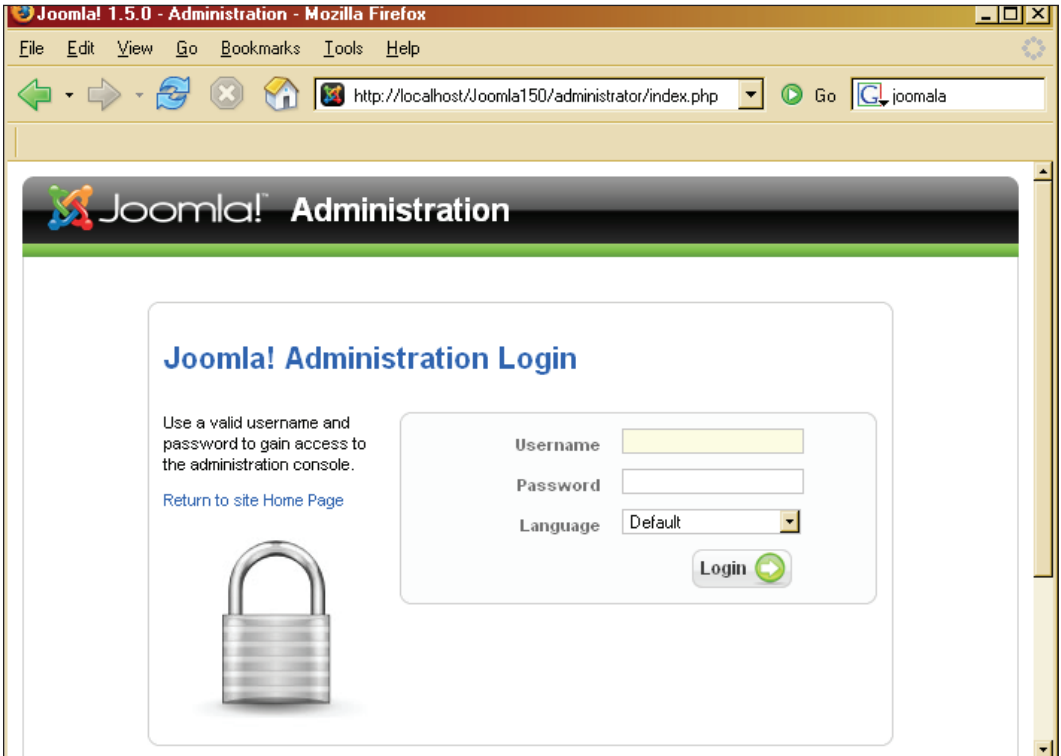


Figure 3.15: Joomla! Administration Login

Log in with your ID (**admin**); you specified the user data yourself in the web installer during installation.

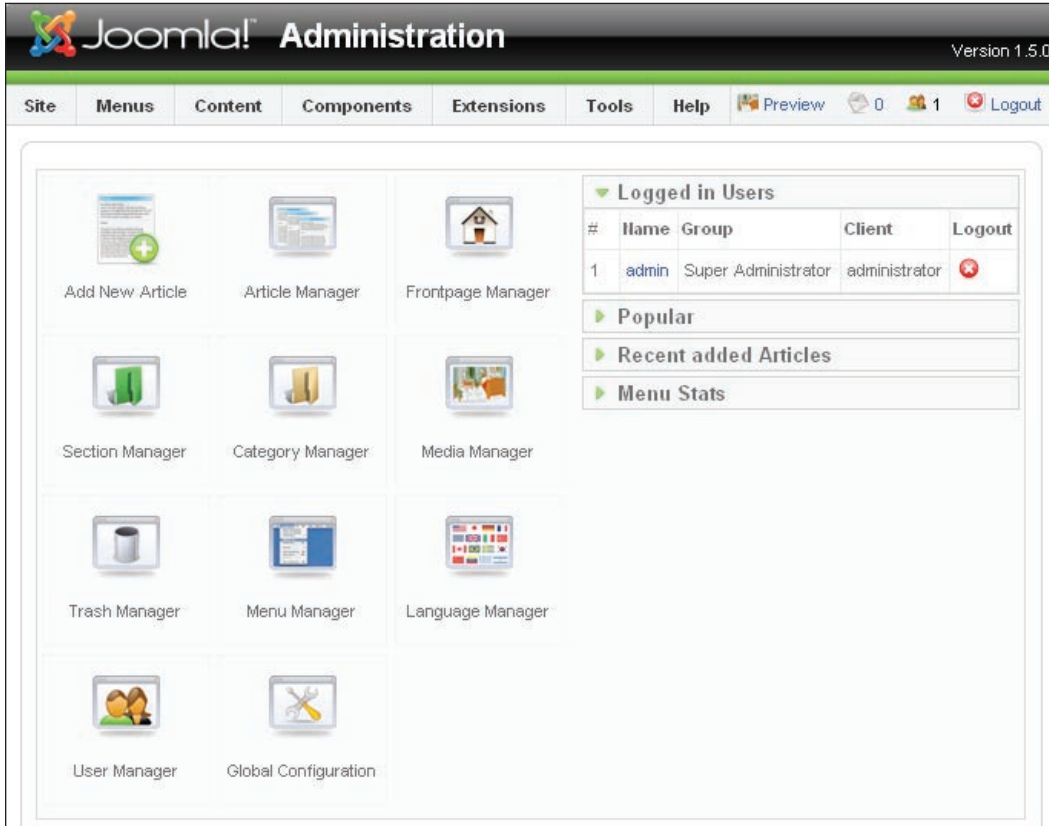


Figure 3.16: Joomla! Administration

You will see an interface with menus, icons, and tabs, identical to the graphic interface of your operating system.



For security purposes, you should protect the directory [path to Joomla!]/administrator/ with a .htaccess file. Because of the widespread distribution of Joomla!, the first successful hacker attempt at the administration is expected soon. You can find a useful tutorial for creating such a file at:
<http://en.wikipedia.org/wiki/Htaccess>
You do not need this protection for a local installation.

Summary

This chapter helped us get a feel of a Joomla! website. We are now familiar with the front end and back end of Joomla!. In the next chapter will learn about customizing Joomla! according to our requirements.

4

Customizing Joomla!

Customizing means adapting a standard program to the needs of the user. In our case, you are the user and the standard program is Joomla!, or to be more correct, the front end and the back end of your Joomla! installation.

In the Joomla! administration (refer to Chapter 3), you can customize your site, make changes, and fill it with content. In this chapter we will customize the Joomla! Site, Menus, Content, Tools, and Help using various options available in the administration; customizing the Components and Extensions will be covered in the next chapter.

Configuration of Joomla! Administration

Joomla! offers nearly the same comfort level as any program with a graphical user interface, such as Windows, KDE, Gnome, or Aqua (Mac OS X). This is not self-evident for web pages and is made possible by the generous use of JavaScript and the integration of AJAX elements.

JavaScript is executed locally on your computer and can be deactivated in the browser. If this is done, you can no longer work with Joomla! administration. Browsers, however, have been able to deal with JavaScript quite well for several years now and there aren't any serious security concerns any more. For this reason, you should enable JavaScript. In this context, I would highly recommend the open-source Internet browser Mozilla Firefox. It is more secure and easier to use than Internet Explorer.

Customizing Joomla!

Joomla! administration, just like your site, consists of different elements. The menus are in the **top menu** bar and on the right-hand side are three elements (as shown in figure 4.1):

- A link to your website
- A note on whether you have received any messages
- Another note on how many users are logged on to your site at the moment

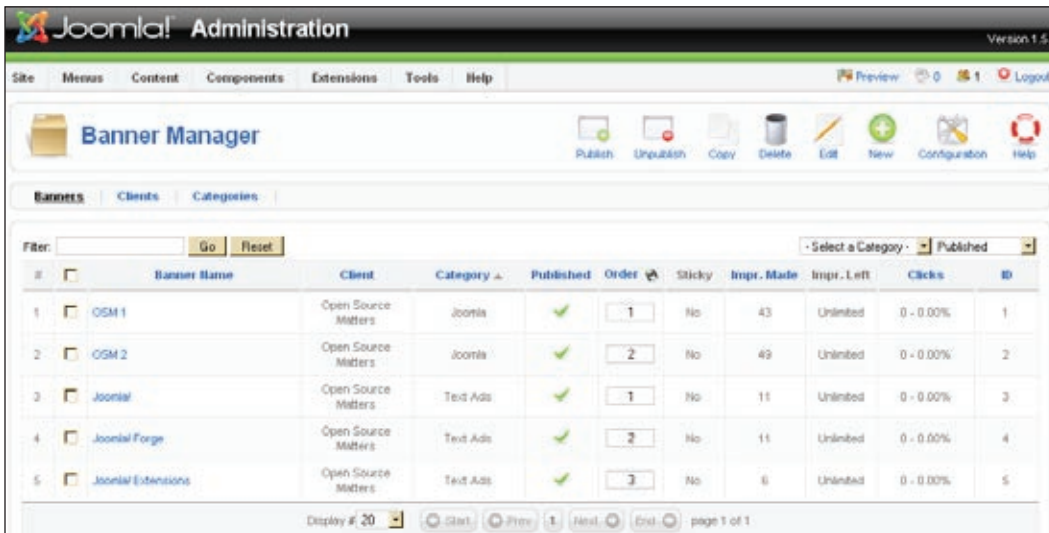


Figure 4.1: Administration Area

Below the top menu bar is the **tool bar**. In the left area of the tool bar, the name of the current edit area (here **Banner Manager**) and its icon are displayed. To the right of it, depending on the manager, there are various disabled icons. If you slide your mouse cursor over one of these icons, it lights up. You can click the left mouse button and implement the appropriate function.

The following table shows the most popular icons and their functions:

Icons	Functions
Publish	The selected element is published.
Unpublish	The selected element is hidden from view.
Archive	The chosen element is moved to the archive.
New	The creation of a new element (link, contact, and message) is started.
Edit	The selected element is loaded into the edit module.
Delete	The selected element is deleted.

Icons	Functions
Trash	The selected element is put in the trash container.
Restore	The selected element is retrieved from the trash container.
Move	The selected element is moved to another section or category.
Copy	The selected element is copied to another section or category.
Save	The selected element is saved and the dialog is closed.
Apply	The changes are saved and the dialog remains open.
Cancel	The edit is ended without any changes being saved.
Preview	The chosen element is shown in a preview window.
Upload	The chosen file is uploaded to the server.
Help	Joomla!'s online help is opened.
Default	The selected element becomes the default.

Table 4.1: Elements of the Toolbar

Below the toolbar there are often additional tabs of the current manager. Figure 4.1 displays the banner components. You will find the tabs **Banners**, **Clients**, and **Categories** there. These are also called submenus or subcategories. Depending on the context, system messages can also appear here.

Below the tabs are **Filter** elements. These can filter the information by different criteria depending on the displayed list.

Below the filter elements is the actual content. This is mostly a list of elements that can be edited after you check the appropriate checkbox, click on the **Edit** icon, or directly click on its name.

Below the content is the navigation bar. You can change the number of listed elements here and browse through the pages.

Help Menu

Let's first take a look at the **Help** menu, which is at the very end. It has two menu entries: **Joomla! Help** and **System Info**.

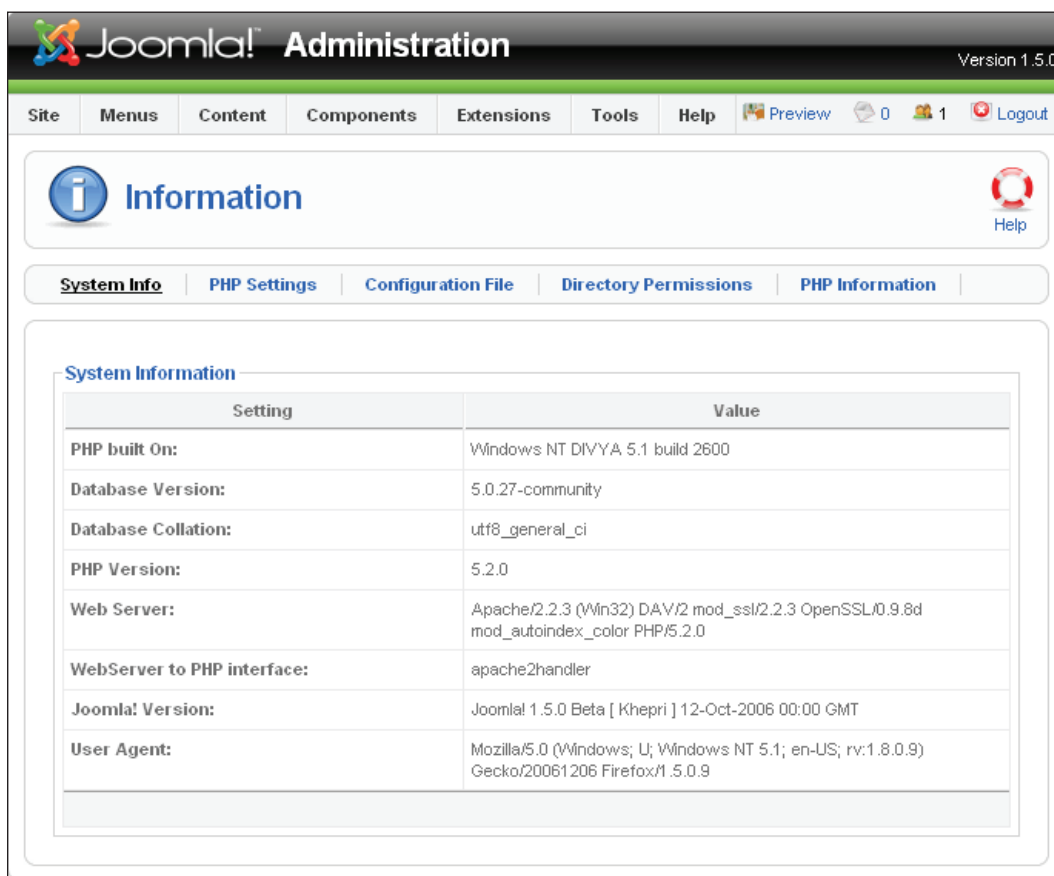
Joomla! Help

This feature is currently unavailable for Beta 1. This should be re-implemented in Beta 2 version.

System Info

You can find the system information in the **Help | System Info** menu item. It is divided into five tabs:

- **System Info**
- **PHP Settings**
- **Configuration File**
- **Directory Permission**
- **PHP Information**



The screenshot shows the Joomla! Administration interface. At the top, it says "Joomla! Administration" and "Version 1.5.0". Below this is a navigation menu with items: Site, Menus, Content, Components, Extensions, Tools, Help, Preview, 0, 1, and Logout. The main content area is titled "Information" and has a "Help" button. Below the title is a tabbed interface with five tabs: **System Info**, PHP Settings, Configuration File, Directory Permissions, and PHP Information. The "System Info" tab is active, showing a table of system information.

Setting	Value
PHP built On:	Windows NT DIVYA 5.1 build 2600
Database Version:	5.0.27-community
Database Collation:	utf8_general_ci
PHP Version:	5.2.0
Web Server:	Apache/2.2.3 (Win32) DAV/2 mod_ssl/2.2.3 OpenSSL/0.9.8d mod_autoindex_color PHP/5.2.0
WebServer to PHP interface:	apache2handler
Joomla! Version:	Joomla! 1.5.0 Beta [Khepri] 12-Oct-2006 00:00 GMT
User Agent:	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.9) Gecko/20061206 Firefox/1.5.0.9

Figure 4.2: Help System Information

The **System Info** tab, for instance, displays a summary of the data of the operating system and the server environment that is most important to Joomla!. The previous screenshot shows the versions of Windows and other software – MySQL database and Apache web server – being used (for the installation procedure refer to Chapter 2).

The **PHP Settings** tab displays all of the important information from the `php.ini` file. This file, for example, determines whether the PHP safe-mode is enabled or disabled. With most rented servers you cannot make any changes to the `php.ini` file.

The **Configuration File** displays the values of the variables in `configuration.php` file. This file is created during the installation. For security reasons, the databases are displayed masked.

The **Directory Permissions** tab displays the rights of your subdirectories. In order to run an error-free Joomla!, all directories must be writable (not write-protected). You can change the directory rights in your FTP program with the `chmod` command.

The **PHP Information** tab displays the results of the `phpinfo()` function. This has to do with the complete configuration of the PHP interpreter.

Site Menu

There are settings that apply to all individual pages and to your server. All of these settings are summarized in the **Site** menu.

There are five menu items in this:

- **Control Panel**
- **User Manager**
- **Media Manager**
- **Configuration**
- **Logout**

Control Panel

The first menu item, **Control Panel**, takes you back to the start page of the administration section:

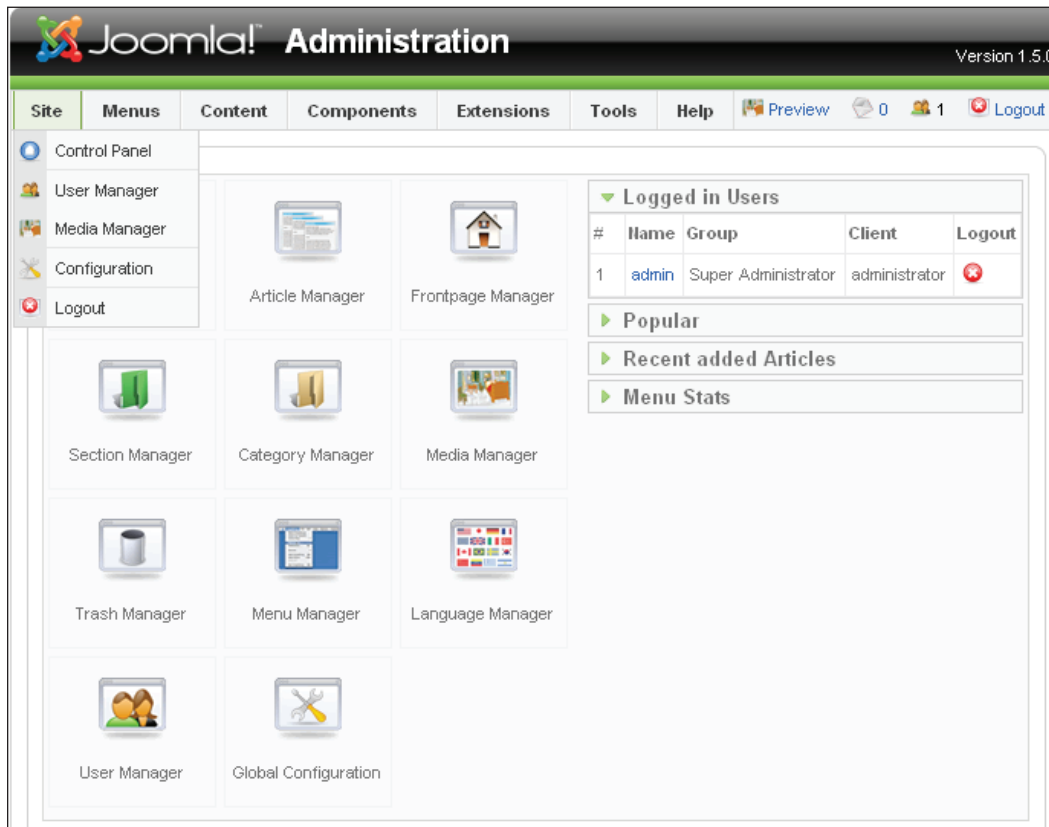


Figure 4.3: Control Panel

You can use the control panel to switch to various areas of administration either by selecting a menu item or by clicking on the displayed icons. Once you have some experience with administration, the fast access icons are the way to go.

User Manager

Users have a special role on your Joomla! site. You can set up as many users as you want and, depending on their rights, they can create their own content and/or view content that has been created for particular users.

At the moment, you are the only user that the Joomla! administration recognizes. Your name is `admin`—you can work on everything in your Joomla! installation. If you allow user registration on your site (see the *Configuration* section), presumably there will be a lot more users very shortly. In the user work area, you can change, delete, block, and assign different rights to users, and you can, of course, also set up new user accounts.

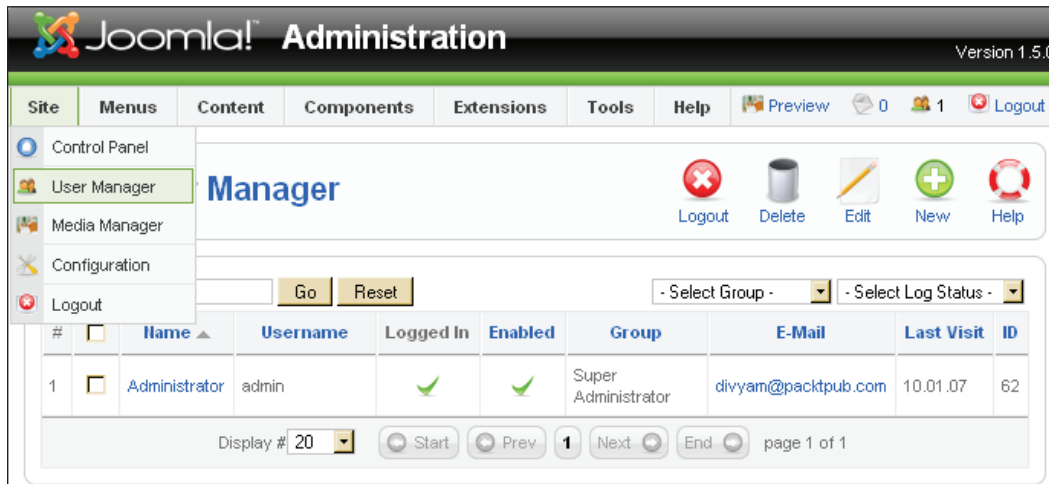


Figure 4.4: User Manager

In the overview list you can see the most important information about the user: The real name of the user (**Administrator**), the username (**admin**), if the user is logged in at the moment (marked by a check mark), if the user is activated, the groups he or she belongs to (**Super Administrator**), his or her email address, and the date of his or her last access to your site. This refers to logins to the front end or back end. You can also check the ID number that this user has in the database (**62**).

From the filter area you can filter the display by user groups and login status. The search field on the left allows you to search for user accounts by entering partial names.

In the toolbar in the user area you can select from the dialogs **Logout**, **Delete**, **Edit**, **New**, and **Help**. Some actions can be applied to several users at once, for instance **Delete**. You just have to check the boxes on the left. The **Logout** function is useful if you want to eject a logged in user. This can act as a great emergency brake. The **Delete** icon permits you to delete one or several user accounts by marking the checkboxes.

Editing Users

This allows you to edit an individual user. Let's take the administrator account as an example. Click on the **Administrator** link or mark the checkbox and click on the **Edit** icon.

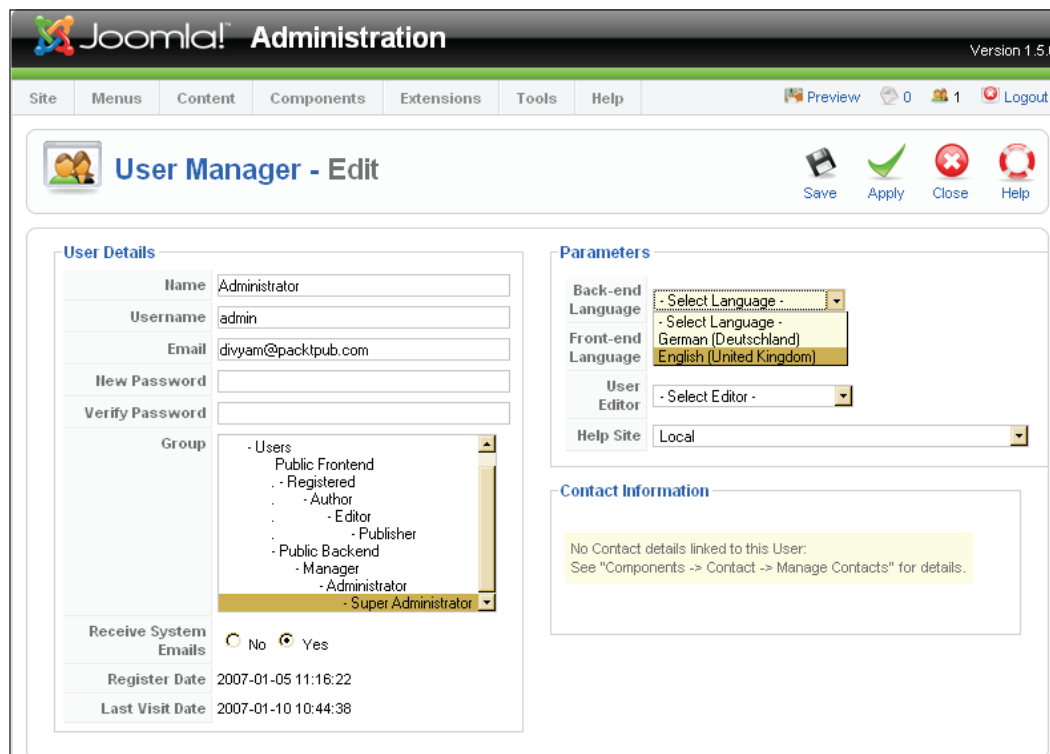


Figure 4.5: User Manager | Edit

From the above screenshot you can see that there are three fields:

- **User Details**
- **Parameters**
- **Contact Information**

User Details has the following fields:

Name: The real name of the user (here **Administrator**).

Username: Usernames are for identification during the login process and are independent of the real name. A username should be short and easy to remember. Hagen Graf, for instance, could become hgraf as username. Joomla! allows spaces,

special characters, and all valid Unicode characters in user names. But keep in mind that you might be sitting at a keyboard that doesn't support these special characters: then how are you going to log in?

Email: The user's valid email address.

New Password and **Verify Password:** The user password has to be entered twice for verification. If you leave this blank, your old password is preserved. The password has to be *at least* six characters long.

Group: Every user is affiliated with a group with certain rights. The group affiliation is divided into two large areas. Users that are only allowed to visit your website (**Public Frontend**) and users that can log on to the Joomla! administration (**Public Backend**). All content in Joomla! can be allotted to these groups:

Group	Rights
Registered	A registered user can log in and see some part of the site that the visitor cannot see.
Author	The author can do everything that a registered user can. An author can also write information and modify his or her information. There is generally a link from the user menu for this.
Editor	The editor can do everything that an author can. An editor can also write and change any information that appears in the front end.
Publisher	The publisher can do everything that an editor can. A publisher can also write information and change any information that appears in the front end. In addition, a publisher can decide whether information is to be published or not.

Table 4.2: Front-End User Group

Group	Rights
Manager	A manager can create content and can see various information about the system. He or she is not allowed to: Administer users Install modules and components Upgrade a user to super administrator or modify a super administrator Work on the menu option Site Configuration Send a mass mailing to all users Change and/or install templates and language files

Customizing Joomla!

Group	Rights
Administrator	An administrator is not allowed to: Upgrade a user to super administrator or modify a super administrator Work on the menu option Site Configuration Send a mass mailing to all users Change and/or install templates and language files
Super Administrator	A super administrator can execute all functions in Joomla! administration. Only a super administrator can create another super administrator.

Table 4.3: Back-End User Groups

In the **User Details** field you can furthermore determine whether the user should receive system emails or not. You can also see the date the user registered and the date of his or her last successful login procedure.



SPECIAL USERS: During the course of this book we will come across issuing rights to *Special Users*. A special user is any user that has more rights than an author. At the moment it is not possible to create your own user groups in Joomla!. The group special users is helpful in limiting content elements to this group. That can be very helpful if, for example, one wants to offer links to internal help files only to these special users.

Modification and expansion of these groups was originally earmarked for Joomla! version 1.5.0 but is now on the roadmap for the next version, and a Google Summer of Code Project team is working on it. This feature may be implemented by the time you read this book.

The **Parameters** field contains switches for the language, the editor being used, and the help site. Since it is possible to operate the front end and the back end in Joomla! 1.5 in different languages, it is possible that you could have mixed language administrator accounts. In the **Back-end Language** and the **Front-end Language** fields, you can select the valid languages for this user account from the available languages.

You can do the same with **User Editor**. So that you can edit texts conveniently, Joomla! 1.5 is delivered with the TinyMCE 2.0 Editor:

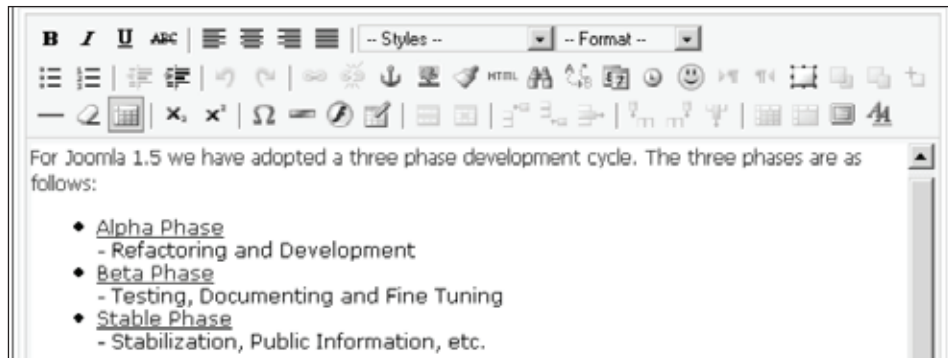


Figure 4.6 : TinyMCE Editor

In addition to TinyMCE, you can also use the Freeware Editor XStandard-lite. This is a very capable WYSIWYG editor that outputs valid XHTML code and offers extensive options for integration. The actual editor, however, is not bundled with Joomla! but has to be downloaded either from the company's website (<http://www.xstandard.com/>) or from the Packt website (<http://www.packtpub.com/support>). Firefox will notice the conversion and if necessary will reinstall the editor as a plug-in. After the installation of the editor and the selection of the user account, it is then shown in the appropriate edit fields:

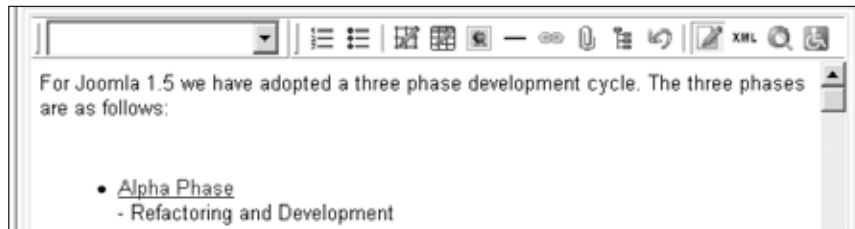


Figure 4.7: XStandard XHTML Editor



WYSIWYG is the acronym for **What you see is what you get** or (what some say is much closer to reality – *This is what you wanted, now you have it!*). This term comes from the early days of graphical user interfaces, when it first became possible to see text on the screen the way it would look when it was printed. Today it is more about how it will look on your website.

Normally on the Internet, you fill out forms without any formatting options. The formatting is done via HTML tags or program-specific mnemonics. A WYSIWYG editor is more convenient since you just have to click on the appropriate icons, similar to editing text. The editor is automatically enabled in the text fields that need formatting. It works with all popular current browsers.

Customizing Joomla!

You can also select an individual **Help Site** from a list in the parameter area. Because of Joomla!'s great popularity, there are now numerous local help servers.

Finally, the **Contact Information** section also provides information about available contact forms that are linked with this user. Joomla! contains a component that lets you create individual contact forms (see Chapter 5 or **Components | Contact | Manage Contacts** for further details).

You can save your changes and exit the dialog by clicking on the **Save** icon. Clicking on the **Apply** icon saves your changes and leaves you in the edit template.

Adding New Users

Now that you have made a few changes, set up a user account with your own name. Give the user an affiliation to the group of registered users and assign the user whatever parameters you want. My user's name is **Hagen Graf**, his user name is **hgraf**, he is not logged in at the moment (but already activated), and is in the user group called **Registered** and has the **ID 63**.

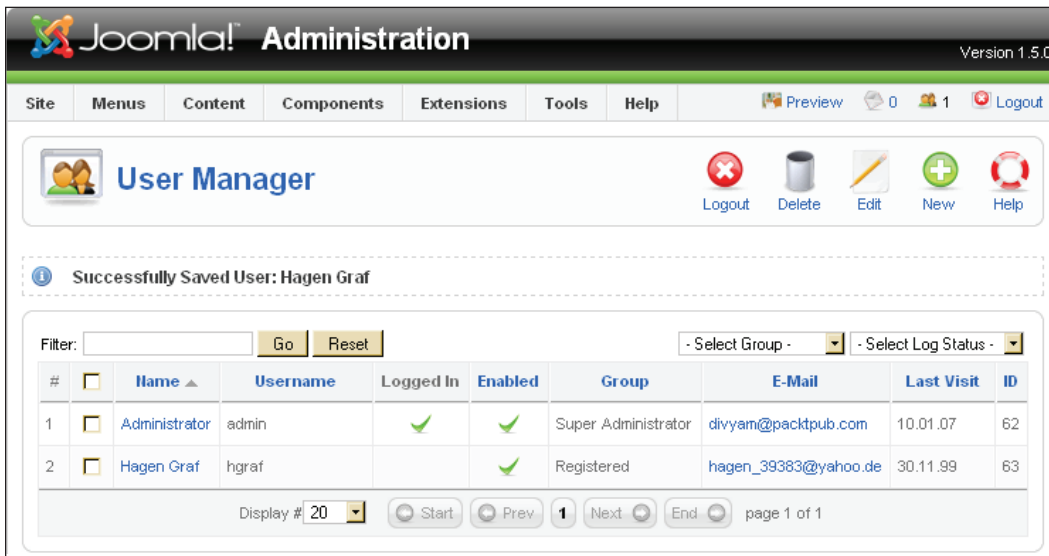


Figure 4.8: User | New

The new user can now log in to the front end using his or her username and password. As administrator you have the choice of setting the user up for the first time or allowing the user to register himself or herself. We will come back to this topic when we discuss Configuration (see the *Global Configuration* section).

Media Manager

You can envision the **Media Manager** work area like a file explorer or like an FTP program in your operating system. By default you can upload files with `jpg`, `gif`, and `png` suffixes and administer them in various directories. This manager is especially useful when you don't have FTP access even though you have administration rights.



Figure 4.9:Site | Media Manager

There are two tabs in the media area: **Thumbnail View**, which displays preview icons of the files, and the **Detail View** tab, which displays the name and size of the files in a list.

Customizing Joomla!

Besides these tabs, there is a separate area for folders and for files. If you click on a folder on the left, you will see the files contained in that folder on the right.

All of the media that Joomla! uses by default, in particular the smilies and graphics in the `M_images` folder, are administered by the media manager.

The following functions are available when it comes to **Media Manager**:

- Creating and deleting directories (deletions can only be done from the **Detail View**).
- Uploading and deleting a single or several files.

Creating a Directory

In the files area you can see the path to its current directory. After it is an input box and a **Create Folder** button. Enter whatever directory name you want in the input box (here it is `france`) and click on the button. The directory is created and displayed with the directory icon:

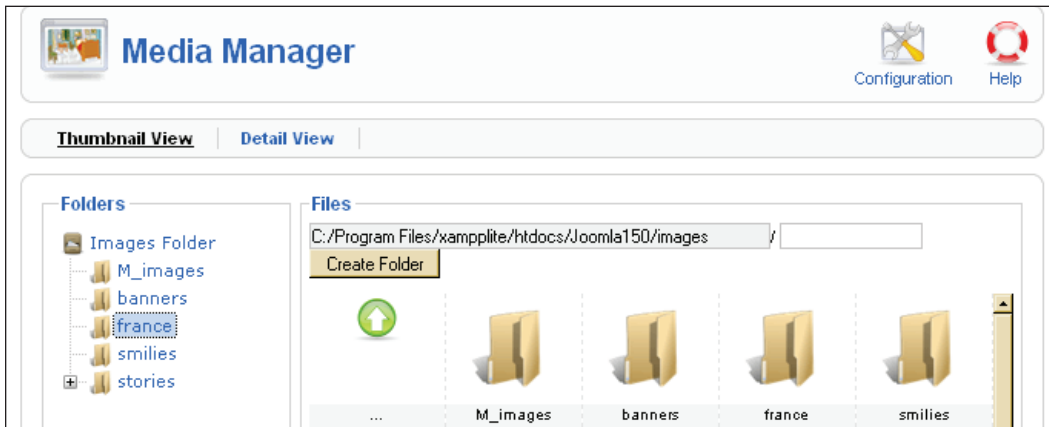


Figure 4.10: Site | Media Manager | Create Folder

Uploading a File

In the lower part of the **Media Manager** window there is an area with the name **Upload File**. By default, you see an input box for a file here. By clicking on the button **+ Add file**, another input box is created:

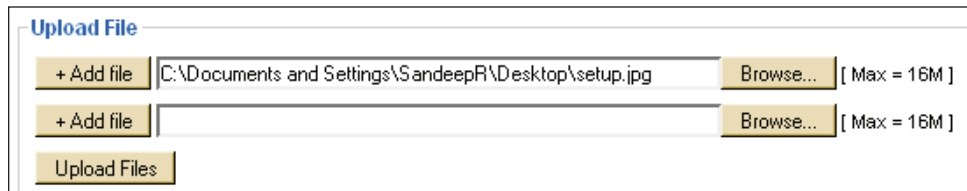


Figure 4.11: Site | Media Manager | Upload File

After clicking on the **Upload Files** button, the files are loaded into the new directory and displayed in the overview. We will integrate them later into various content.

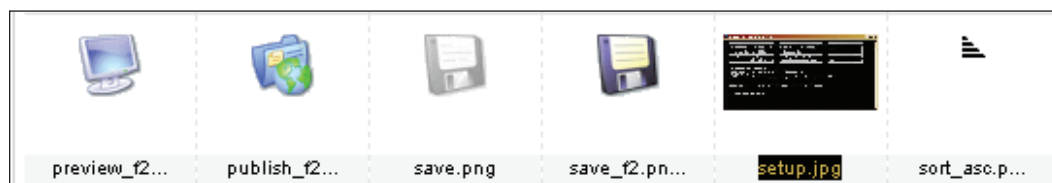


Figure 4.12: Site | Media | Upload File II

Keep in mind that it is not a good idea to display 5 MB images from a digital camera on your website in that resolution. Images should at most be 50 KB. There are people who do not have high-speed access to the Internet!

Keep these ground-rule download times in mind for a 100 KB file (about the size of a portal's web page with pictures). Even if you only have one picture on that page that is 300 KB large, you will lose all visitors that have ISDN and normal telephone line access:

Type of connection	Download time for 100 KB of data
DSL	Depending on the DSL provider, under one second!
ISDN	About 15 seconds
Modem (56 k)	About 25 seconds

Table 4.4: Duration for a Download of 100 KB

Configuration

The type of files and their maximum size (up to the maximum size predetermined by PHP—see information box below), can be set by clicking on the **Configuration** icon in the toolbar. It looks very impressive, as the background dims and a form appears in which you can extend the permitted file suffixes and change the maximum file sizes:

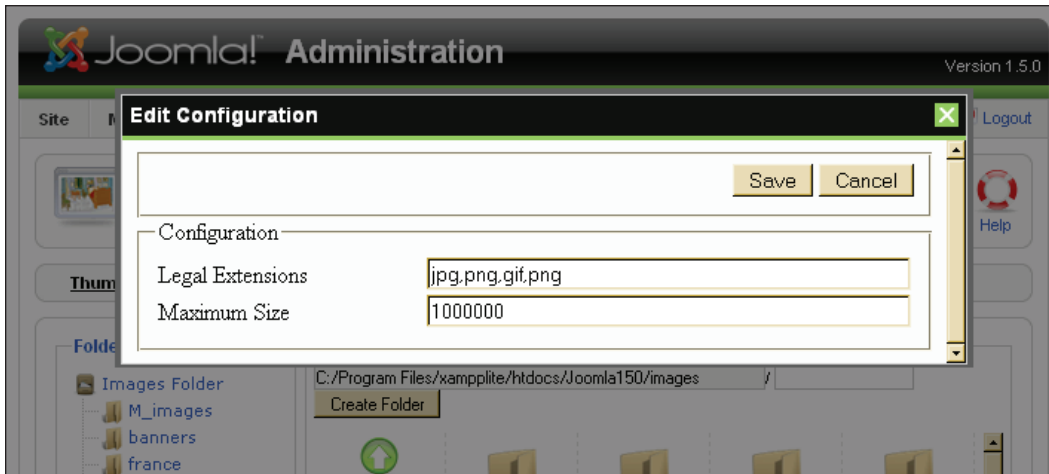



Figure 4.13: Site | Media Manager | Configuration

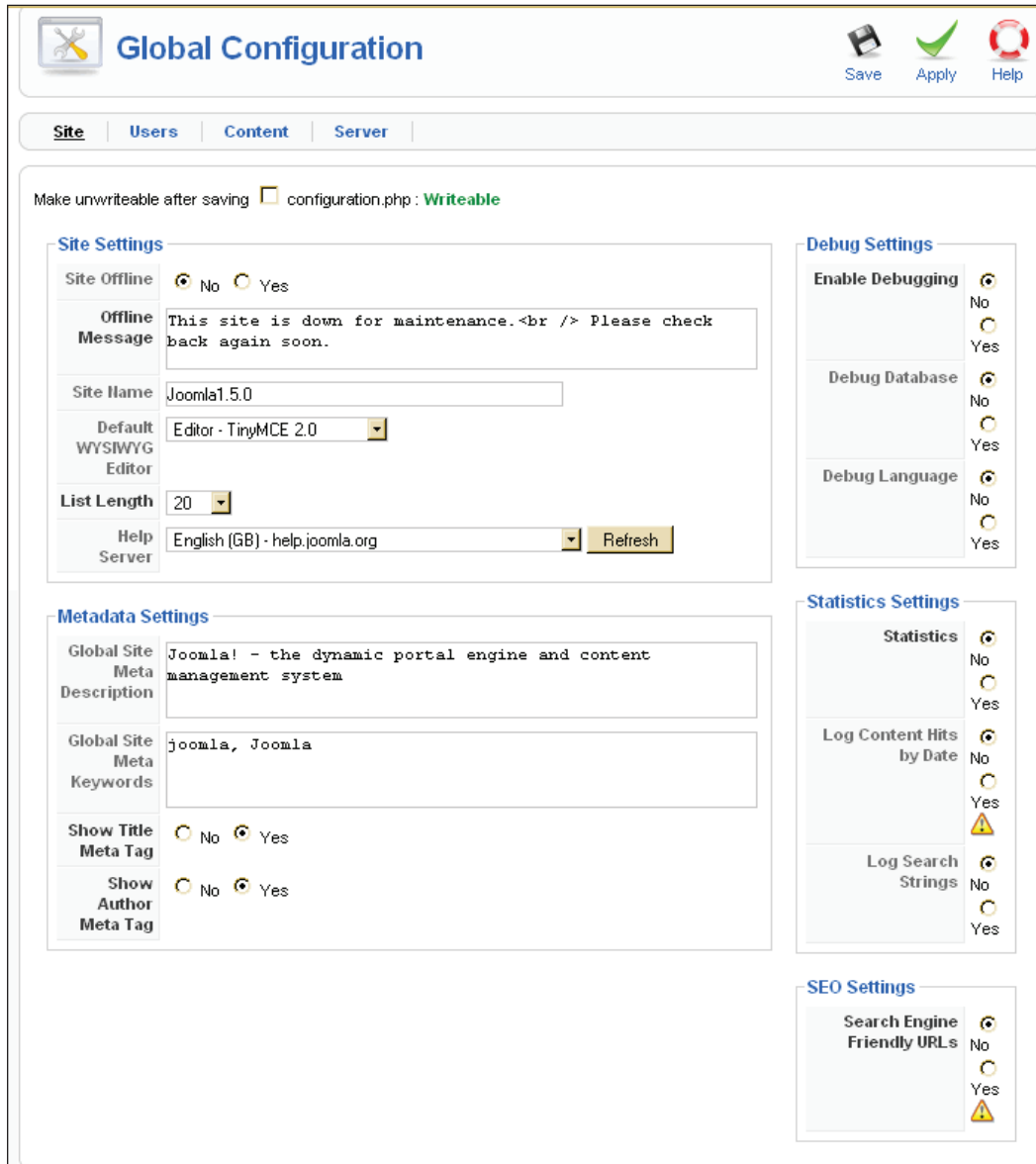
 The maximum upload size per file is dependent on the PHP configuration of your provider. In my case that is 8 MB. Files larger than that have to be uploaded with FTP or the value entered in the `php.ini` configuration file has to be appropriately modified

Global Configuration

The **Global Configuration** work area administers the values of the variables in the `configuration.php` file. Some essential information, like the access code for the database server, but also some less important parameters such as the predetermined length of lists that are displayed, are stored in this file. The work area itself is divided into four tabs: **Site**, **Users**, **Content**, **Server**.

Site

The site area in turn is divided into five visual areas: **Site Settings**, **Metadata Settings**, **Debug Settings**, **Statistics Settings**, and **SEO Settings**:



The screenshot shows the Joomla! Global Configuration interface, specifically the 'Site' tab. At the top, there are icons for 'Save', 'Apply', and 'Help'. Below the navigation tabs (Site, Users, Content, Server), a status message indicates that the configuration.php file is 'Writeable'. The main content area is divided into five sections:

- Site Settings:** Includes 'Site Offline' (radio buttons for No/Yes), 'Offline Message' (text area), 'Site Name' (text input), 'Default WYSIWYG Editor' (dropdown menu), 'List Length' (dropdown menu), and 'Help Server' (dropdown menu with a 'Refresh' button).
- Metadata Settings:** Includes 'Global Site Meta Description' (text area), 'Global Site Meta Keywords' (text area), 'Show Title Meta Tag' (radio buttons), and 'Show Author Meta Tag' (radio buttons).
- Debug Settings:** Includes 'Enable Debugging', 'Debug Database', and 'Debug Language', each with radio buttons for No/Yes.
- Statistics Settings:** Includes 'Statistics', 'Log Content Hits by Date', and 'Log Search Strings', each with radio buttons for No/Yes and a warning icon.
- SEO Settings:** Includes 'Search Engine Friendly URLs' with radio buttons for No/Yes and a warning icon.

Figure 4.14: Site | Configuration | Site

We will discuss each of these settings in detail.

Site Settings

Within Site Settings you will come across various options:

- **Site Offline:** If you want to make changes to your website and inform your visitors of this, then you can click on the **Yes** radio button in this area.
- **Offline Message:** The text that you enter here is displayed in the top area of your website when it is "offline". Try it: Set **Site Offline** to **Yes**, click on the **Apply** icon and go to your website. The website can no longer be found. Users who have rights to the administration area can still log in.

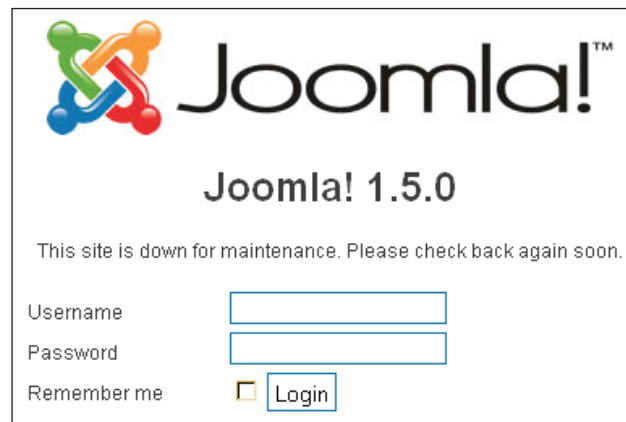


Figure 4.15: Offline Site

- **Site Name:** The name of the site that you entered during the installation. It shows up, for instance, in the header of the browser, as the sender of system emails, on the news feed, and in various other places that have something to do with easy recognition of your website.
- **Default WYSIWYG Editor:** By default, Joomla! currently uses the TinyMCE editor (<http://tinymce.moxiecode.com/>). You can also use XStandard lite 1.7.
- **List Length:** Lists such as news and links will surface all over the place on your website. With this you set the default number of items that a list should have.
- **Help Server:** Here you define the default help server. You can use a separate help server for every individual user account (refer to the *Editing Users* section).



It is also possible to activate a local help server. This local help server must have the help texts in the desired language.

Metadata Settings

Metadata are data about data, for instance a description of a website. Metadata are important for search engines and are written in XHTML. Exactly how important metadata are in the rating by search engines is debatable. However, metadata represent a good way to describe your website in short and precise words. If you check out the HTML source code of a Joomla! site, you will see the following meta tags in the top area.

```
<meta name="generator" content="Joomla! 1.5" />
<meta name="description" content="Joomla!
- the dynamic portal engine and content management system" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, Joomla" />
```

Separate tags with the description and keywords attributes can be used.

- **Global Site Meta Description:** This description of site content is often displayed by search engines. You should pay special attention to this tag, since the user who has entered a search term often uses this information to make a decision of whether to visit a site or not. You can add further information about every individual page to the global description.
- **Global Site Meta Keywords:** Keywords are very important for a document. You should list the most important concepts of your website with them. Some search engines index keywords in particular. The individual words are separated by commas; several words can be entered between two commas with normal spaces in between. The keywords should be limited to a maximum of 1000 characters; more than that will not be read by the search engines. Note that the use of fewer keywords helps each individual word get a higher priority in the search engine. Deliberate carefully about which are the most often used keywords and which are likely to be searched for most. You can add more keywords to the global keywords for every individual page of your website.
- **Show Title Meta Tag:** With individual content pages, the content title is blended in as a meta tag, for instance:
<meta name="title" content="welcome to Joomla!" />
- **Show Author Meta Tag:** With individual content pages, the author's name is blended in as a meta tag, for instance:
<meta name="author" content="Administrator" />

Debug Settings

Following are the options present for **Debug Settings**:

- **Enable Debugging:** This lets you shift the website into a debug mode.
- **Debug Database:** After activating this function, the database queries of the site are displayed below the website. In order to generate a single Joomla! page, 16 database queries are required in this case.
- **Debug Language:** The debug messages refer to language strings.


 Programmers talk about debugging when they are looking for errors. This term has evolved over the years. A bug is a beetle, and 50 years ago it was indeed the beetles that got comfortable next to the warm vacuum tubes of a computer and thereby caused shorts now and then. *Debugging* (removing the beetles) was taken literally in those days. Today these bugs are errors in software, for instance database queries that don't work.



Figure 4.16: Debug Mode with Database Queries

So that you don't forget you are in debug mode, after you enable **Debug Language** thick black marks are positioned before and after all of the identifiers:

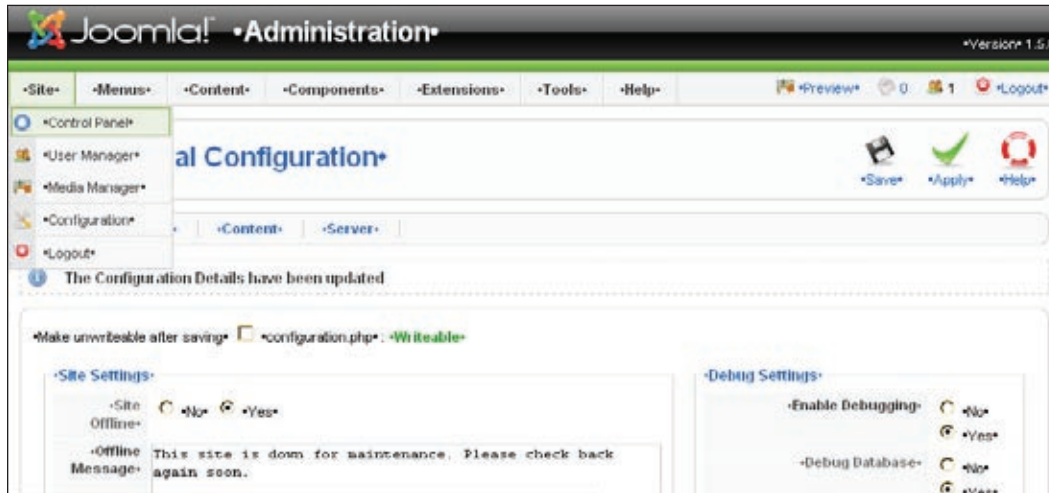


Figure 4.17: Markings in Debug Mode

Statistics Settings

You can enable and disable the statistics for your website from this area:

- **Statistics:** Here you determine whether you want to collect statistics or not.
- **Log Content Hits by Date:** Here you can enable content statistics by date. Each and every page access is logged in the `jos_core_log_items` table. Unfortunately, there is still no analysis program for this data at this time. Also, you should remember that very large data sets are produced within a very short time.
- **Log Search Strings:** This switch can give very interesting results. It collects words that visitors to your site enter into the search field. Google publishes its search strings under the keyword *Zeitgeist* (<http://www.google.com/press/intl-zeitgeist.html>). You can use this information when you allocate your keywords.

SEO (Search Engine Optimization) Settings

This option is about search engine-friendly URLs. Normally a Joomla! URL looks like this:

```
http://www.example.com/Joomla150/index.php?option=com_
contact&Itemid=3
```

Customizing Joomla!

This type of URL is not normally stored by search engines, since the search engine assumes that the content is constructed dynamically and will probably change soon.

- **Search Engine Friendly URLs:** With this switch you can make a search engine-friendly URL from a dynamic URL. If you set the switch to **Yes**, links look something like this: `http://localhost/component/option,com_contact/Itemid,47/`

The principle is based on a feature of the Apache web server. With its rewrite engine it can manipulate URLs at will. Besides the switch, you also have to rename the file `htaccess.txt` in the Joomla! directory to `.htaccess`.

With Windows, such a renaming is only possible with certain programs, for example, with the **Ultraedit** editor (`http://www.ultraedit.com/`) or in the command line with the command `rename`. With Linux, the renaming function is problem-free; the file, however, is subsequently no longer displayed in its FTP client (depending on the server configuration). In addition, the provider may not permit `.htaccess` files, since they represent a security risk for the web server.

Users

In this area you will find global settings for user accounts:

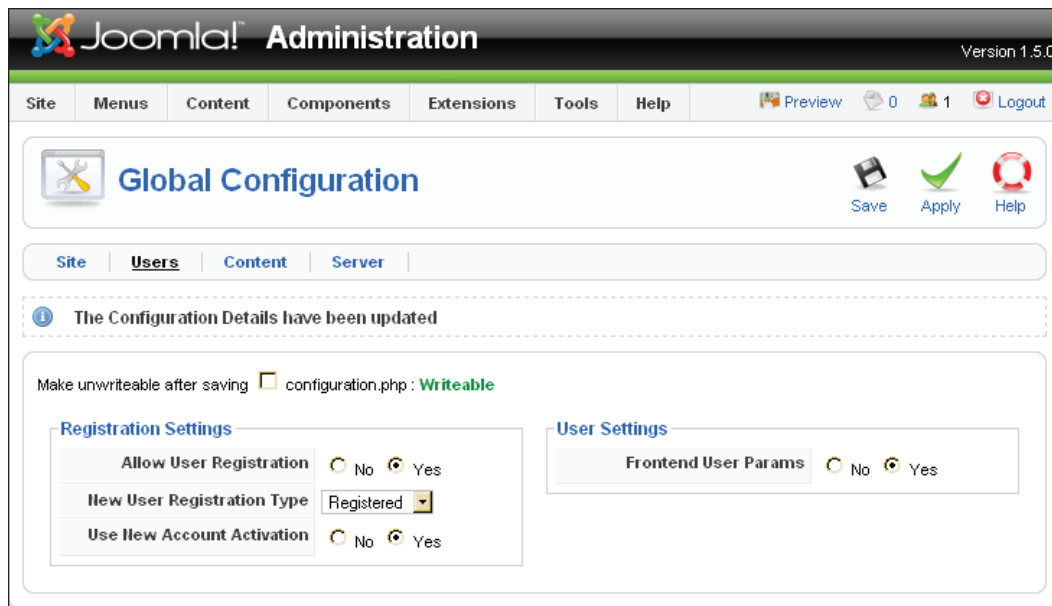
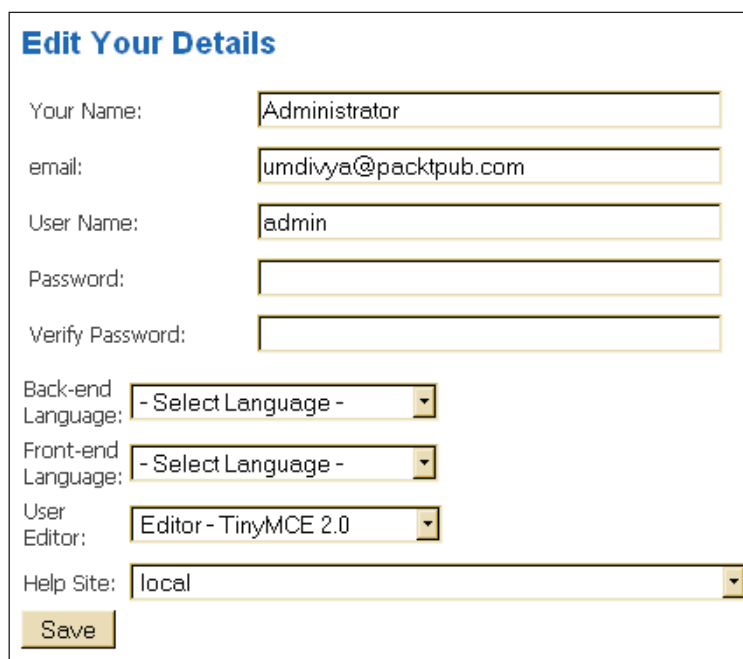


Figure 4.18: Site | Configuration | Users

- **Allow User Registration:** Here you can select whether you want to permit users to do their own registration or not. If you operate a company site, you could set up user accounts for your coworkers, and forbid them to create their own account. With a community site, on the other hand, it is desirable for users to log themselves in.
- **New User Registration Type:** Here you can define which user group the newly registered user will be affiliated with. You learned about the importance of user groups in Tables 4.2 and 4.3.
- **Use New Account Activation:** In order to protect yourself from automated programs that can create 20,000 user accounts on your site, you can ask for separate activation. The user gets an automatic email sent to the address entered by him or her. There is a link in this email that activates the account. After activation he or she can log in normally.
- **Frontend User Params:** With this you define whether the users can affect their own settings. The users can log in and then in **User Menu** click on **Your Details:**



Edit Your Details

Your Name:

email:

User Name:

Password:

Verify Password:

Back-end Language:

Front-end Language:

User Editor:

Help Site:

Figure 4.19: Front-End Users Creating their Own Account Content

Content

In this area you will find global settings that affect all content elements. When a content element is created, these values are used as default values but these can be changed here:

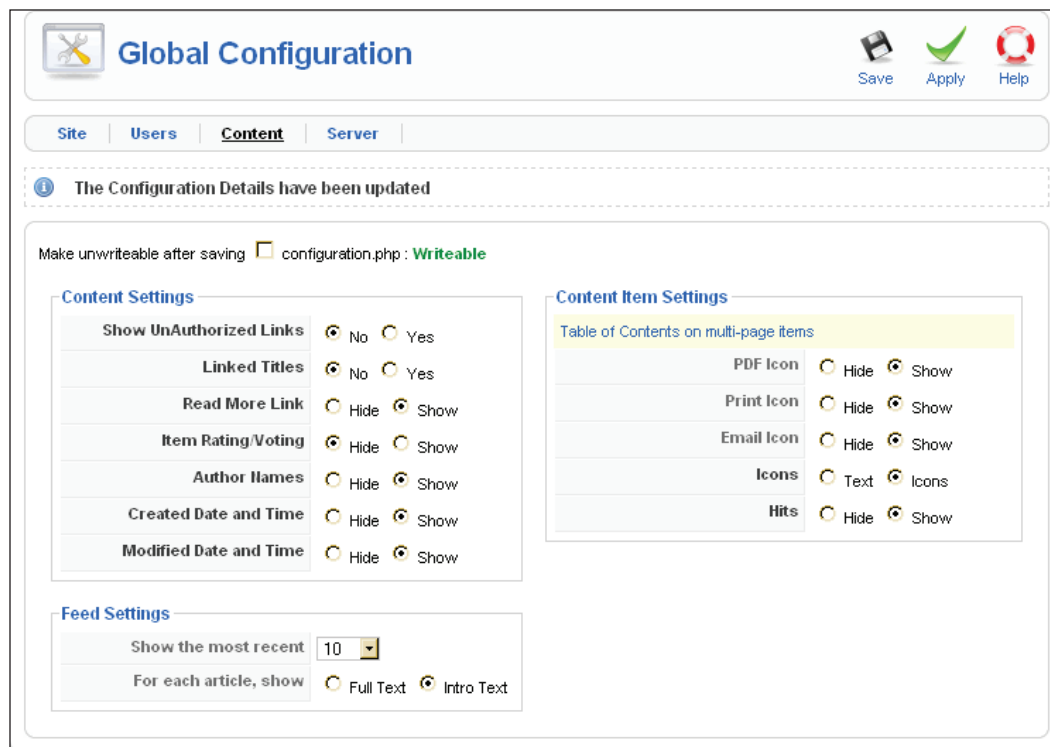


Figure 4.20: Site | Configuration | Content

Content Settings

Following are the options presented in the **Content Settings** area:

- **Show Unauthorized Links:** You have the ability to display individual pages only to registered users. It is possible that these pages get linked from a public connection. If you choose **Yes**, these links are displayed. When a visitor who has not logged in clicks on such a link, a message appears announcing that this is a protected area that requires registration.
- **Linked Titles:** Here you can set the title of a content element to be shown as a link. This link then refers to the same target as the **Read More Link**.

- **Read More Link:** A lot of content consists of a hook (intro text) and the actual text. Here you decide whether you want to have a **Read More** link under the hook that points to the complete text.
- **Item Rating/Voting:** This lets your visitors evaluate the contents of the site. If you click **Yes**, an evaluation component is displayed above the item.



Figure 4.21: Rating Function

- **Author Names:** If you want the name of the author of the content to be displayed, you select **Yes**. The text **Written by [author]** is displayed above the article.
- **Created Date and Time:** If you want the date and time of creation of content to be displayed, then you select **Yes**. Text similar to **Friday, 13 October 2006** (as seen in the previous screenshot) is displayed above the article.
- **Modified Date and Time:** Should the date and time of modification of content be displayed? If you select **Yes**, **Last Updated (Saturday, September 30, 2006)** is displayed under the text.

Customizing Joomla!

Feed Settings

These are the options relevant to **Feed Settings**:

- **Show the most recent:** Joomla! automatically displays news on the frontpage as a feed. Here you can determine how many pieces of news should be shown in the feed.
- **For each article, show:** This will enable you to show the entire text or only the intro text.

Content Item Settings

These contain:

- **PDF Icon:** Should a PDF icon be displayed above the content in your front page? After clicking this icon, your content is prepared as a PDF file! In order to view the PDF file, you need the free Acrobat Reader.



Figure 4.22: PDF-Display of Content

- **Print Icon:** Here you can determine whether a print icon is to be displayed above the content. After clicking this icon, the content is prepared for printing.
- **Email Icon:** If you want you can also display the email icon above the content. After clicking this icon, a form is displayed that allows you to send a link with reference to this content to anybody.

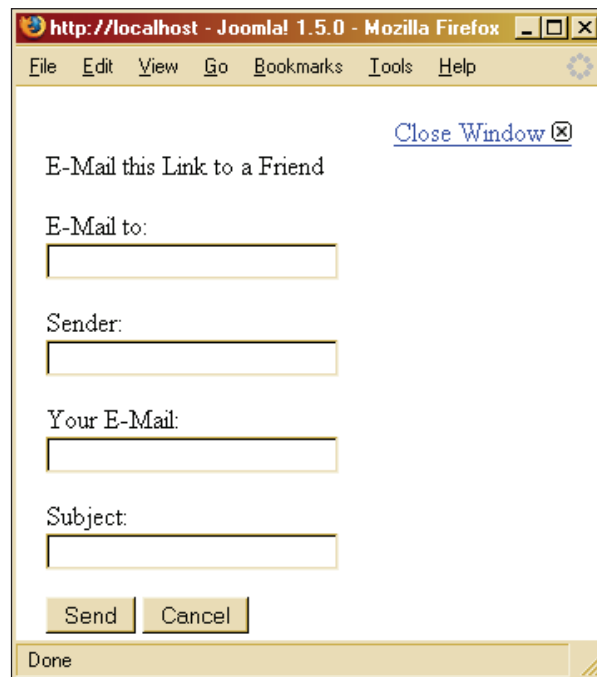


Figure 4.23: Email Link to an Interesting Article

- **Icons:** Here you decide whether to show PDF, Print, and EMail as icons or as links.
- **Hits:** Here you decide whether the number of hits on content in the front end should be shown.

Server

In this tab you configure the technical information of your server environment.

Server Settings

The following screenshot illustrates the **Server Settings** area:

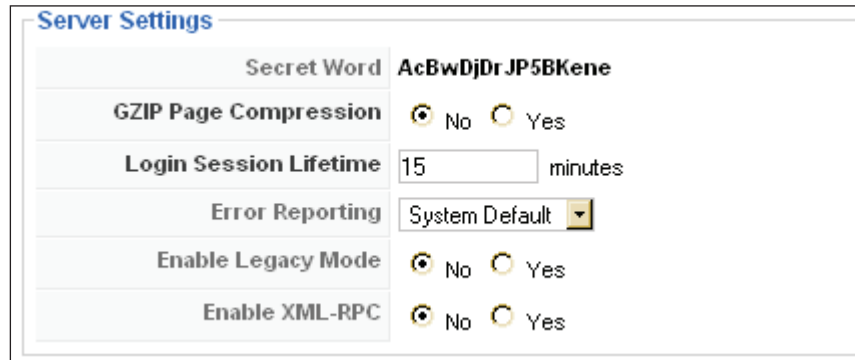


Figure 4.24: Site | Configuration | Server | Server Settings

- **Secret Word:** When you install Joomla!, a random order of characters is created. It is a type of seal. This secret word is used in connection with the XML/RPC interface. This is to ensure that in the future only registered and/or authorized remote systems can use this interface for access.
- **GZIP Page Compression:** You can enable compression of the pages with this. If both the browser and the web server support this function, the pages are sent in ZIP format and extracted at the client browser. This can significantly increase the speed of a page download, in particular with slow Internet connections. If your server handles a heavy load, you should only use this option if the respective GZIP libraries are directly installed at the server level. Depending on server configuration, it is possible that the server runs faster without this option!
- **Login Session Lifetime:** Whenever you log in as a user, you create a so-called session. If you don't log out, the session will be terminated after the number of minutes entered here.
- **Error Reporting:** PHP's own error reporting mechanism is activated with these switches. This is not related to logging in a file for future analysis but to displaying messages in the browser!

Option	Description
System Default	This adopts the settings from the <code>php.ini</code> configuration file.
None	Errors are not logged. This setting is recommended for working sites so that an attacker is not shown possible points of attack.

Option	Description
Simple	Errors and warning messages are logged. This setting complies with the parameter: <code>error_reporting (E_ERROR E_WARNING E_PARSE).</code>
Maximum	Errors, warning messages, and instructions are logged. This setting complies with the parameter: <code>error_reporting (E_ALL).</code>

Table 4.5: Error Report Categories

- **Enable Legacy Mode:** You need to enable this mode to get your templates in the old version of Joomla! working here.
- **Enable XML-RPC:** With this you can activate the interface of the same name, by means of which JExplorer for instance, can access your server or all kinds of other useful things like a third-party authentication (Google Mail), creation of content via a remote program (Flickr, w.Bloggar) etc., can be permitted. This does not affect outbound XML-RPC connections, only inbound ones!

Local Setting



Figure 4.25: Site | Configuration | Server | Local Setting

Time Offset: This setting can be used to display the correct time; for example, if the server that Joomla! runs on is located in the USA, but the site is meant for visitors in Germany. The base time is UTC.

Cache Settings

A **cache** is a temporary storage facility. Your browser, for example, has a picture cache, which makes pictures that have already been downloaded once available faster. Joomla! uses a similar mechanism on the server to cache pages generated by PHP. This option can drastically reduce response time for frequently visited pages.



The screenshot shows the 'Cache Settings' section of the Joomla! configuration interface. It contains four rows of settings:

Template Cache	<input checked="" type="radio"/> No <input type="radio"/> Yes
Extension Cache	Disabled (will be available in beta 2)
Page Cache	Disabled (will be available in beta 2)
Cache Time	<input type="text" value="900"/> seconds

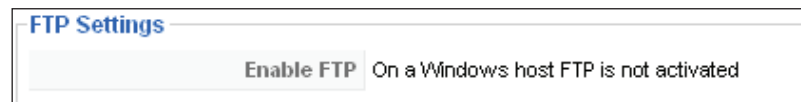
Figure 4.26: Site | Configuration | Server | Cache Settings

- **Template Cache:** Templates are temporarily stored here.
- **Extension Cache:** Extensions are temporarily stored here.
- **Page Cache:** Individual pages are temporarily stored here.
- **Cache Time:** You define the time that has to pass before content in the cache has to be renewed.

FTP Settings

File Transfer Protocol (FTP) is used to circumvent problems with security settings at providers. If it is not permitted to upload files with PHP, the FTP method is used instead. Enter the access data to your provider's FTP server here.

If you are on Linux host you will find this option accessible. In Windows, you will get to see a similar screenshot:



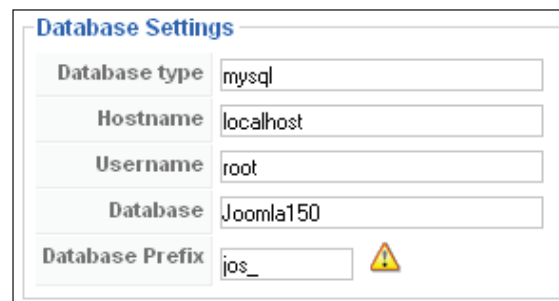
The screenshot shows the 'FTP Settings' section of the Joomla! configuration interface. It contains one row of settings:

Enable FTP	On a Windows host FTP is not activated
------------	--

Figure 4.27: Site | Configuration | Server | FTP

Database Settings

You can change the parameters of your database that you entered while installing your Joomla! here:



The screenshot shows the 'Database Settings' section of the Joomla! configuration interface. It contains five rows of settings:


Database type	<input type="text" value="mysql"/>
Hostname	<input type="text" value="localhost"/>
Username	<input type="text" value="root"/>
Database	<input type="text" value="Joomla150"/>
Database Prefix	<input type="text" value="jos_"/> 

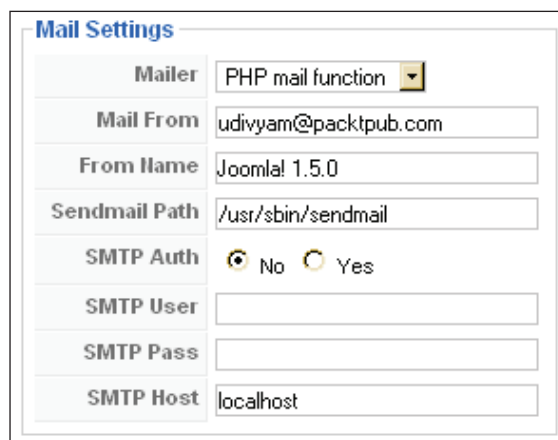
Figure 4.28: Site | Configuration | Server | Database Settings

Please keep in mind that you can corrupt Joomla! if you make a typing error. That means that you will no longer be able to access the back end. If something like that happens, you can edit the `configuration.php` file with a text editor. All of the settings that you make in the **Site | Configuration** menu are saved in variables in the `configuration.php` file:

```
<?php
class JConfig {
    var $dbtype = 'mysql';
    var $host = 'localhost';
    var $user = 'root';
    var $password = '';
    var $db = 'Joomla150';
    var $dbprefix = 'jos_';
    //
    ...
    // many other variables
    // ---
}
?>
```

Mail Settings

In this tab you can make all of the decisions about sending emails from Joomla!:



The screenshot shows the 'Mail Settings' configuration form in Joomla!. It includes the following fields and options:

Mail Settings	
Mailer	PHP mail function
Mail From	udivyam@packtpub.com
From Name	Joomla! 1.5.0
Sendmail Path	/usr/sbin/sendmail
SMTP Auth	<input checked="" type="radio"/> No <input type="radio"/> Yes
SMTP User	
SMTP Pass	
SMTP Host	localhost

Figure 4.29: Site | Configuration | Server | Mail Settings

- **Mailer:** Here you decide whether you want to use the built-in PHP mail function, the Sendmail program, or another email account such as Yahoo! or GMX.
- **Mail From:** This email address is automatically entered as sender of Joomla!-generated mail.

Customizing Joomla!

- **From Name:** This name is automatically entered as sender of Joomla!-generated mail.
- **Sendmail Path:** If you want to use the Sendmail program that can presumably be used on all Linux servers instead of the PHP mail function, you have to enter the path to the program here.
- **SMTP Auth:** Select if you want to use an external mail server (Yahoo!, GMX, your own server).
- **SMTP User:** Your username with this email provider.
- **SMTP Pass:** Your password with this email provider.
- **SMTP Host:** The SMTP server of this email provider.

Logout

By clicking on this menu item you log out from the back end and are automatically placed into the Joomla! Administration login page.

Menus Menu

There are numerous menus in the front end. They are often displayed as standalone boxes. The menu items are generally arranged one below the other:



Figure 4.30: Main Menu

Menus can also be integrated into the design horizontally so that at first sight they aren't even recognizable as cohesive menus.



Figure 4.31: Top Menu

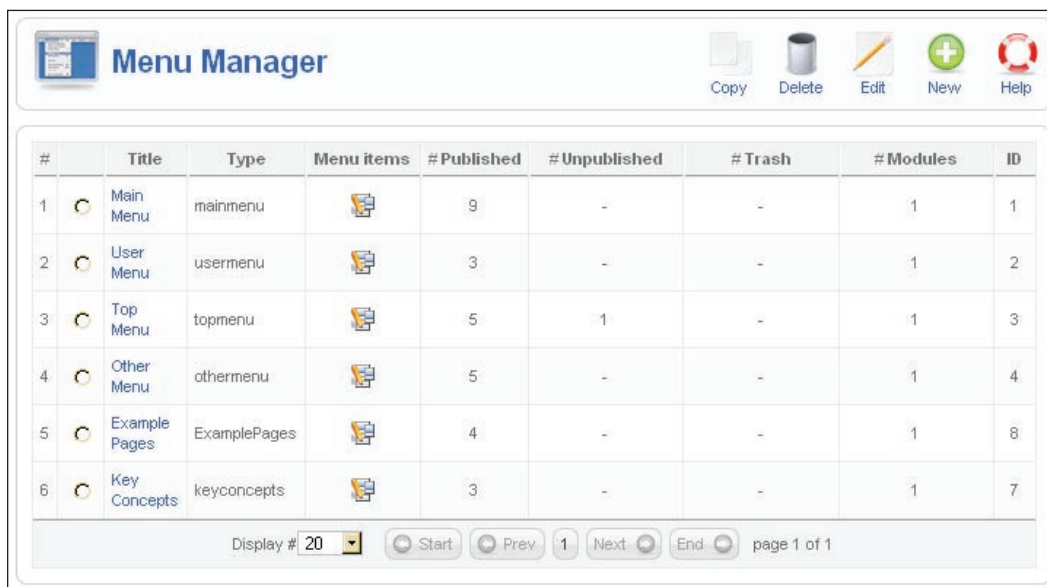
These menus are administered in the **Menus** work area. Joomla! has four different menus in the sample data (**Main Menu**, **Other Menu**, **Top Menu**, and **User Menu**).







The **Top Menu** is a horizontal menu, the other menus are vertical. Each menu is coupled with a so-called module, which is administered in the module manager (we will learn more about this in the following chapter).

Menu Manager

By clicking on **Menus | Menu Manager**, you get an overview of the available menus. You can also access the content of these menus by means of the menu bar – **Menus | Main Menu**, **Other Menu**, ... by clicking the respective menu link.

This **Menu Manager** area serves as an overview and shows you the number of published and unpublished menu items, the number of menu items in the trash, and the menu ID. Here you can, for instance, copy a menu or create a new one:



#	Title	Type	Menu items	# Published	# Unpublished	# Trash	# Modules	ID
1	Main Menu	mainmenu		9	-	-	1	1
2	User Menu	usermenu		3	-	-	1	2
3	Top Menu	topmenu		5	1	-	1	3
4	Other Menu	othermenu		5	-	-	1	4
5	Example Pages	ExamplePages		4	-	-	1	8
6	Key Concepts	keyconcepts		3	-	-	1	7

Display # 20 Start Prev 1 Next End page 1 of 1

Figure 4.32: Site | Menus | Menu Manager

Main Menu

Experiment a little with the menus to get a feel for things. The following edit steps are the same for all menus. Call up the menu item **Menus | Main Menu**. You will see a listing of the menu items in the **Main Menu**:

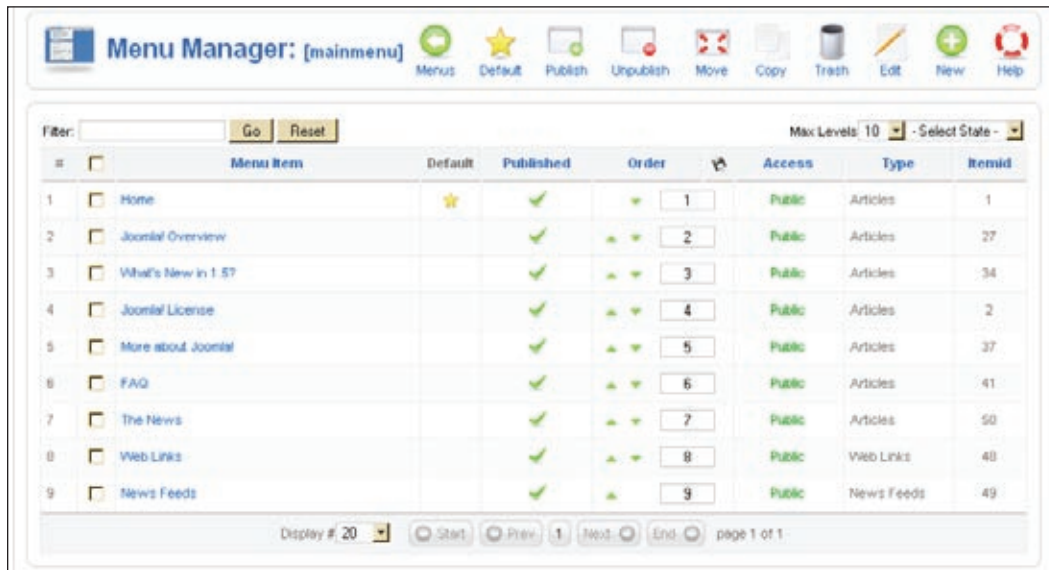


Figure 4.33: Site | Menus | Main Menu

Several functions can be executed in the table with a single mouse click. By clicking on the checkmark, you can assign a status of unpublished to the content item. You can change the order of the items by clicking on the triangles or by typing numbers into the fields under **Order**. If you use the numbers method, you have to click on the disk symbol in the header in order to effect the change.

In the **Access** column, you can decide via mouse click whether the menu is available to all users (**Public**), only to registered users (**Registered**), or only to a particular circle of users (**Special**).

Default Icon

The menu item that is marked as default with a yellow star is displayed as the start page when someone calls up the URL of your website. At the moment this is the menu item **Home**, but you can designate any element that you want as the start page. Just mark the checkbox and click on the **Default** icon.

Publish/Unpublish Icon

The state of a content element can either be published or unpublished. You can toggle this state individually by clicking the green checkmark and/or the red cross, or you can mark the checkbox and subsequently click on the appropriate icon. If you follow the latter method, you can toggle several menu items at the same time.

Move Icon

This entails the moving of menu entries. Select one or several menu elements and click on **Move**. This opens a form, listing the available menus. On the right you will see the elements that you want to move:

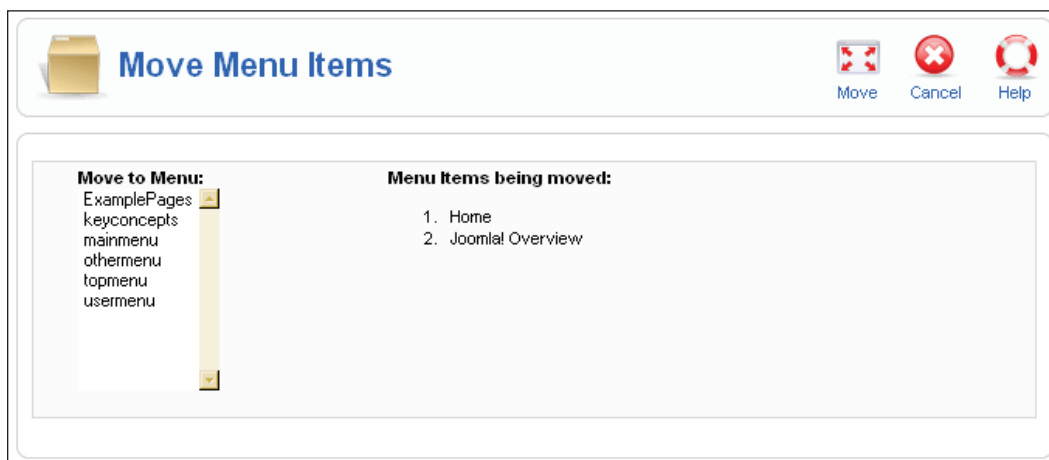


Figure 4.34: Site | Menus | Main Menu | Move

Select the menu into which you would like to move the marked menu items. Here, we have moved **What's New in 1.5** from **Main Menu** into the **Top Menu**. You can marvel at the results in the front end:



Figure 4.35: Top Menu with the Additional Item

Copy Icon

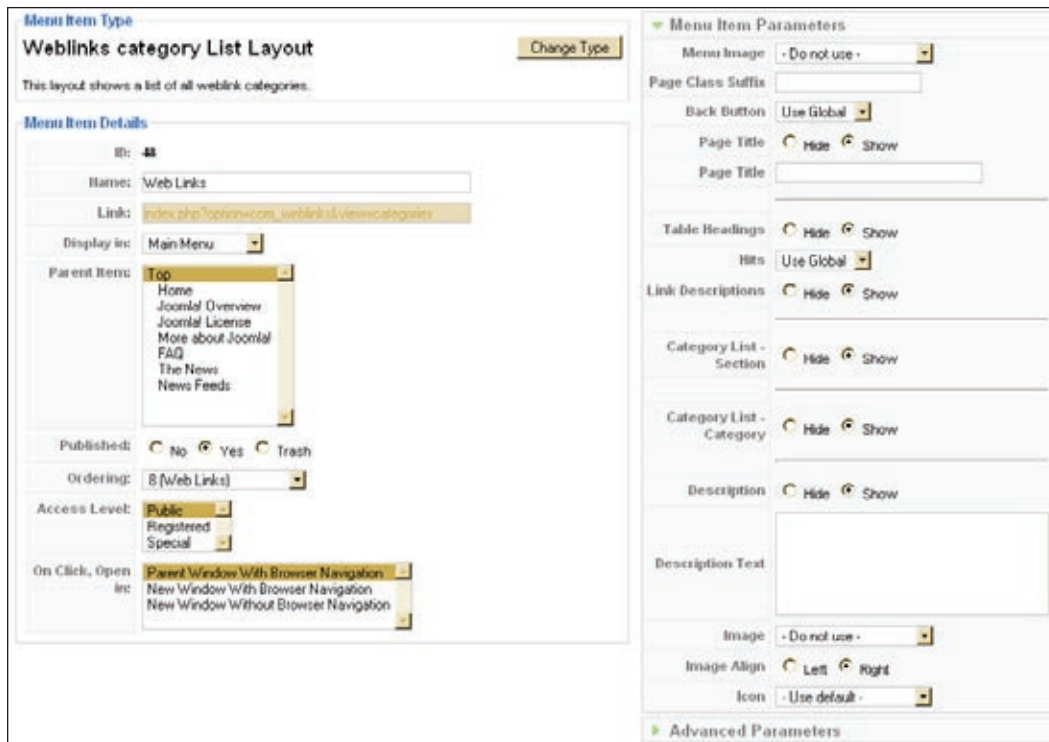
You can also copy menu items. To do that, select one or more menu items and click on the **Copy** icon. Just like with moving, a form with the available menus opens. Select the menu into which you want to copy the marked menu entries.

Trash Icon

In order to protect you from inadvertently deleting items, when editing them you cannot delete them immediately, you can only throw them in the trash. In order to throw menu entries into the wastepaper basket, select one or several menu elements and click on the **Trash** icon. The marked menu items are then dumped into the trash can. You can have the content of the trash displayed by clicking on **Menus | Trash Manager**.

Edit Icon

Here you can modify an existing menu, for instance **Web Links**. After clicking on **Web Links**, you will see the edit form for menu elements:



The screenshot shows the Joomla! menu editor interface for editing a menu item. The interface is divided into three main sections:

- Menu Item Type:** Shows the current menu item type as "Weblinks category List Layout" with a "Change Type" button.
- Menu Item Details:** Contains fields for:
 - Name:** Web Links
 - Link:** index.php?option=com_weblinks&view=categories
 - Display In:** Main Menu
 - Parent Item:** Top (with a dropdown menu showing options like Home, Joomla! Overview, Joomla! License, More about Joomla! FAQ, The News, News Feeds)
 - Published:** Radio buttons for No, Yes, and Trash.
 - Ordering:** 8 (Web Links)
 - Access Level:** Public (with options for Registered and Special)
 - On Click, Open In:** Parent Window With Browser Navigation (with options for New Window With Browser Navigation and New Window Without Browser Navigation)
- Menu Item Parameters:** Contains various settings:
 - Menu Image:** - Do not use -
 - Page Class Suffix:** (empty field)
 - Back Button:** Use Global
 - Page Title:** Hide (selected) / Show
 - Table Headings:** Hide (selected) / Show
 - Hits:** Use Global
 - Link Descriptions:** Hide (selected) / Show
 - Category List - Section:** Hide (selected) / Show
 - Category List - Category:** Hide (selected) / Show
 - Description:** Hide (selected) / Show
 - Description Text:** (empty text area)
 - Image:** - Do not use -
 - Image Align:** Left (selected) / Right
 - Icon:** - Use default -

Figure 4.36: Site | Menus | Main Menu | Web Links | Edit

The form is divided into three parts:

- **Menu Item Type**
- **Menu Item Details**
- **Menu Item Parameters**

We will be discussing these parts in the following sections.

Menu Item Type

Every menu item is of a particular type. We will go into more detail of this when we create new menus. For instance, a menu item can represent an installed Joomla! component, a content element, a link to an external website, or many other things. You can see what type of link is in this area, in our case it is a link to the Joomla! weblinks component, and you can also see a button with the label **Change Type**. If you click on that button, you get the following screen:

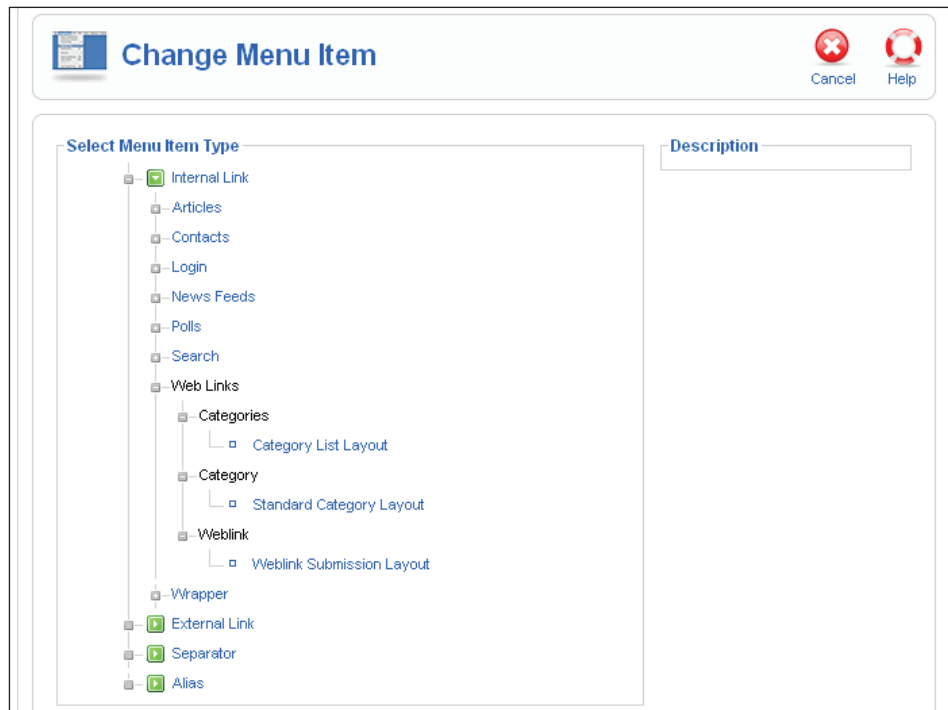


Figure 4.37: Edit Menu | Change Menu Item

This manager is new in Joomla! version 1.5 and really handy. In version 1.0.x there was no option to change the type of a menu item. You had to delete the old menu item and create a new one.

Now close this; we will get back to it when we create a new menu.

Menu Item Details

Menu Item Details contains the following options:

- **ID:** Everything in an administration requires an ID number and so does our menu item. In this case the menu item has the ID number 48. Joomla! assigns this number for internal administration purposes at the time that the item is created. This number cannot be changed.

Customizing Joomla!

- **Name:** This is the name of the menu and it will be displayed that way on your website.
- **Link:** This is the request for a component, in other words the part of the URL after the domain name with which you call up your website. In this case it is `index.php?option=com_weblinks&view=categories`.
- **Display in:** You can change the place where the item is displayed, in other words move it to another menu. The options field presents you with a list of the available menus.
- **Parent Item:** Of course menus can also contain nested, tree-like items. **Top** means that the item is at the uppermost level. The rest of the items represent existing menu items. If, for instance, you classify and save **Web Links** under **The News**, the display on the item list and the display on your website are changed:

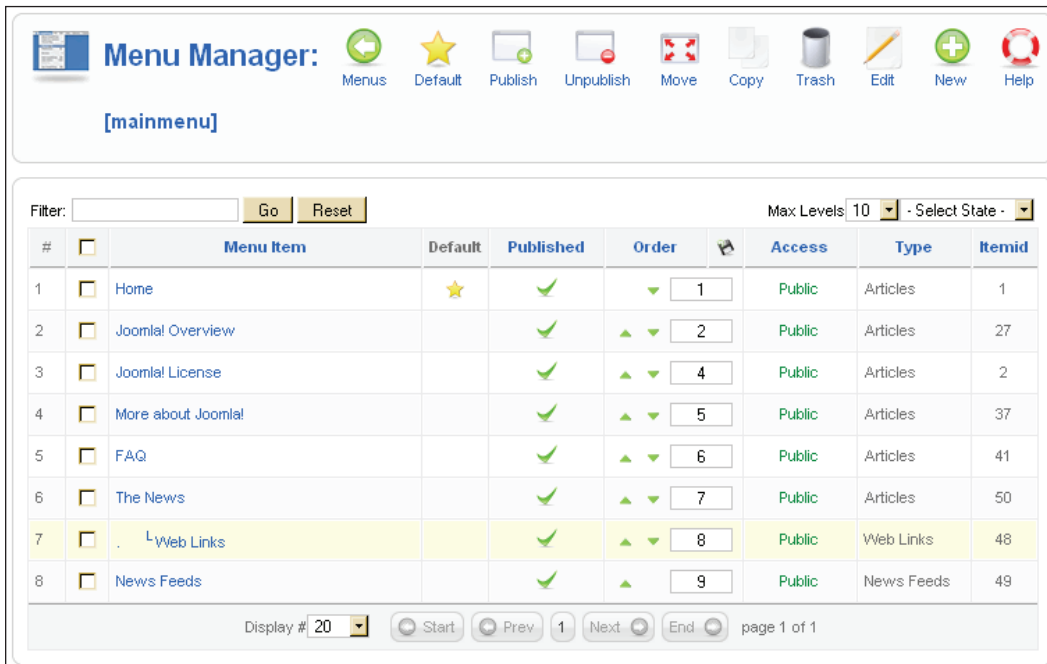


Figure 4.38: Tree Structures in the Menu Manager

The menu item **Web Links** has now moved into **The News** on your website. So you have to first click on **The News** in order to see the **Web Links** item. That way your website can easily and effectively be structured like a database tree:



Figure 4.39: Tree structures on the Website

- **Published:** With this you can publish a menu item or dispatch it to the trash.
- **Ordering:** From the options list, you can select after which link you want to position this link.
- **Access Level:** You can restrict which users that can see this list.
- **On Click, Open in:** A very handy option that influences the behavior of the link. The page is either opened in the existing window or in a new browser window after clicking. You can also define whether the new window will be displayed with or without navigation.

Menu Item Parameter

The possible parameters of a menu item depend on the type of the item. A simple link, of course, has fewer parameters than a configurable list.

In this case we have a link to a table display (these components will be described in detail in the next chapter).

#	Web Link	Hits
1	MySQL The database that Joomla! uses	0
2	php.net The language that Joomla! is developed in	5
3	Joomla! Home of Joomla!	2
4	OpenSourceMatters Home of OSM	10
5	Joomla! - Forums Joomla! Forums	3

« Start Prev 1 Next End »

Page 1 of 1

Figure 4.40: Table Display of the Web Link Component

Customizing Joomla!

The number and the type of parameters are dependent on the type of the menu item. You can open and collapse the parameter fields by clicking on the header. If the parameter fields are open, the green arrow next to header is pointed downward.

▼ Menu Item Parameters	
Menu Image	- Do not use -
Page Class Suffix	<input type="text"/>
Back Button	Use Global
Page Title	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Page Title	<input type="text"/>
Table Headings	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Hits	Use Global
Link Descriptions	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Category List - Section	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Category List - Category	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Description	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Description Text	<input type="text"/>
Image	- Do not use -
Image Align	<input type="radio"/> Left <input checked="" type="radio"/> Right
Icon	- Use default -
▶ Advanced Parameters	

Figure 4.41: Edit Menu Item Parameters

Menu Image: Here you can specify an image that must be in the root directory of the **Media Manager** (/images/stories/). Depending on the template, this picture is displayed on the left, next to the menu entry.

Page Class Suffix: You can define a text string with this, for example `my_menuitems`.

This label is then appended to the class name in the HTML code. You must have the respective class set up in your CSS file.

Page Title: You can display or hide the header of the page with this, or you can even assign customized headers – like that in the following screenshot:



Figure 4.42: Individual Web Link Page Title

Table Headings: You can display or hide the table headers directly above the link with this.

Hits: Your visitors' hits on the respective links of your pages are tracked in the hits column. You can display or hide this hits column.

Link Descriptions: With this you can display or hide the description that is below a link in the link list.

Other categories: When you click on the **Links Descriptions** link, you will either see the default text or the texts that you have set up in the parameters. Below that there is a list of usable categories and sections (if there are other categories). This list can be displayed or hidden.

Icon: You can select the icon that is to be displayed on the left of the list of links.

The **Advanced Parameters** block, in the case of web link components, is empty.

New Icon

This icon is used to create new menus. We will learn about this in the later section *Creating a New Menu*.

Trash Manager

The trash basket collects your deleted menu items:



Figure 4.43: Menus | Trash Manager

If you select the elements and click on the **Restore** icon, you can retrieve them all from the garbage can:



Figure 4.44: Menus | Trash Manager-Restore Items

If you click on the **Delete** icon, they are all irretrievably deleted.

Creating a New Menu

In this section, we will create a new menu named **Joomla! Book** and a link to <http://www.joomla.org/>. Call up **Menus | Menu Manager | New** and enter the data for the new menu.

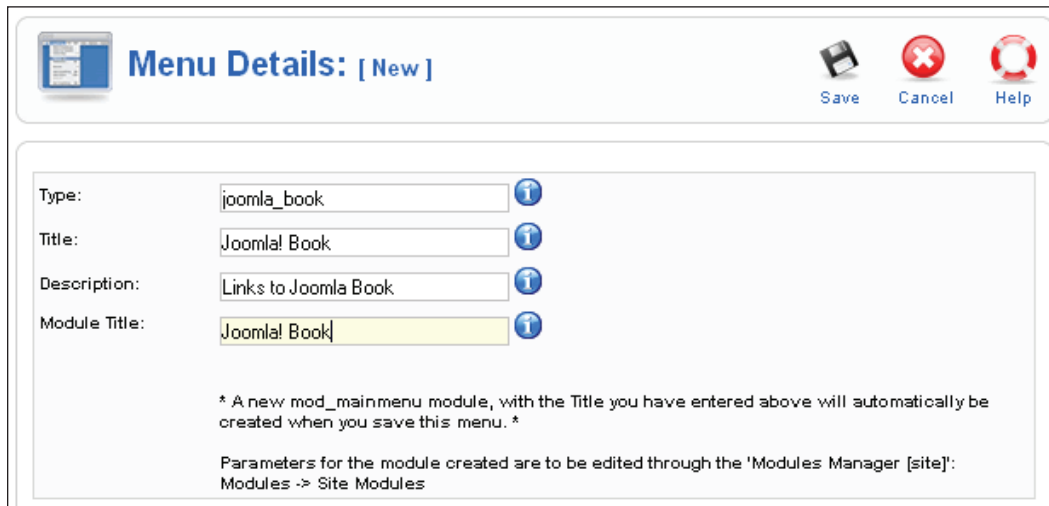


Figure 4.45: Site | Menu Manager | New

Type: This is the name that Joomla! uses in the code. You are not allowed to use any spaces. Here we are using `joomla_book`. This name will not be shown on the website; it strictly exists to create a link between the menu and the module.

Title: The name of the menu. Here it is Joomla! Book (with a space).

Description: A description of the menu. This description is only shown internally in the back end, for instance in the display of lists.

Module Title: The name of the module—Joomla! Book here as well.

Customizing Joomla!

After you click on **Save**, Joomla! creates a new module with the specified parameters. You are put back into the menu overview and you will see a new menu there that is still empty:

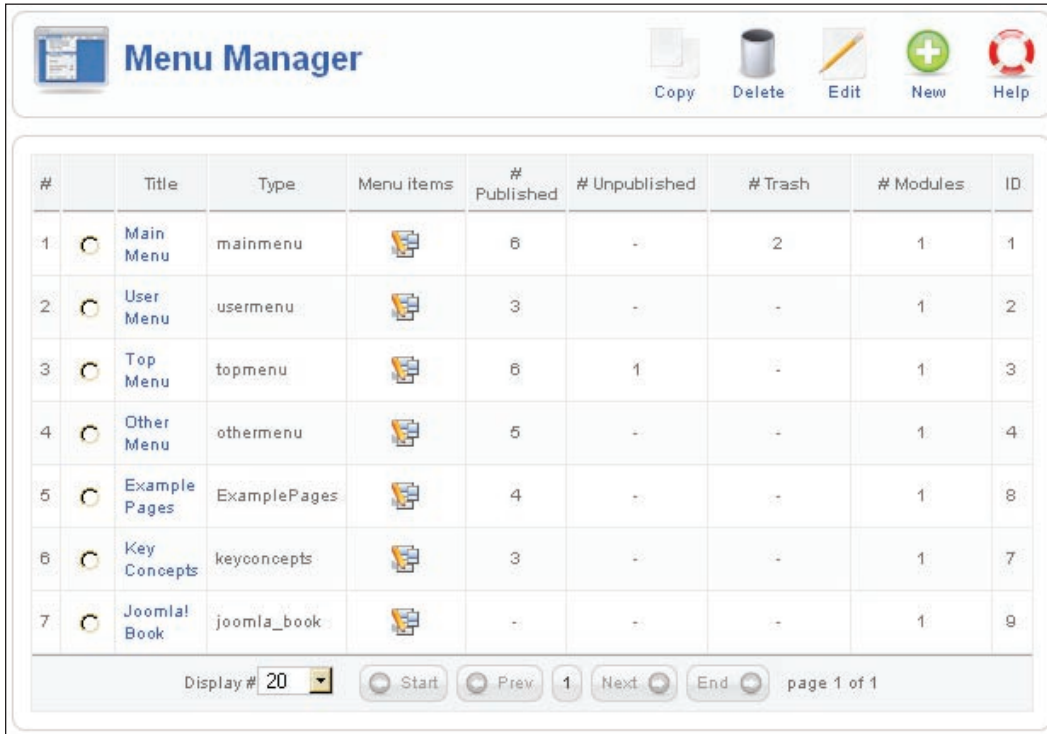


Figure 4.46: New Menu in the Menu Manager

The name now also appears in the menu by going to **Menus | Joomla! Book**. If you call up this menu item, you get an overview mask for the content of the **Joomla! Book** menu:

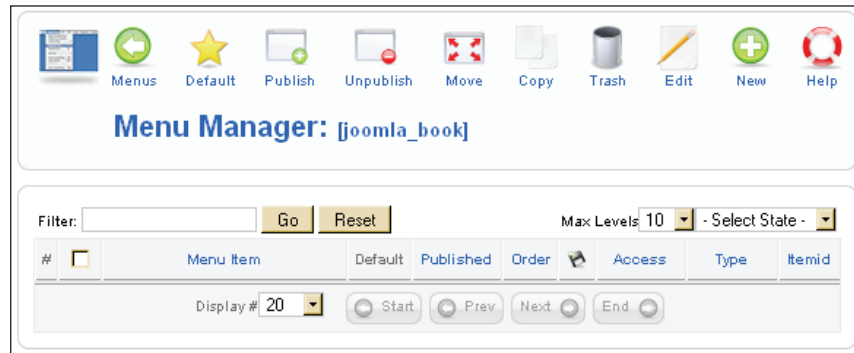


Figure 4.47: Menus | Joomla! Book

Since there isn't any content yet, click on the **New** icon. You can now select content from four different areas on the selection mask that appears on the screen. Since we are still at the threshold of our Joomla! knowledge, let's just insert a simple link to an external website.

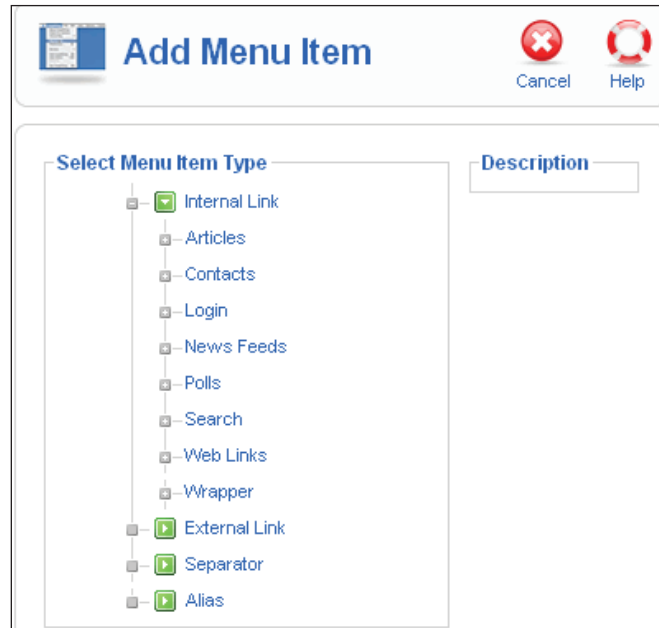


Figure 4.48: Site | Menu Manager | New Menu Item

Customizing Joomla!

To do that, click on the **External Link** button. You will then be taken to an already familiar form. Here you can define the details and the parameters of the link:

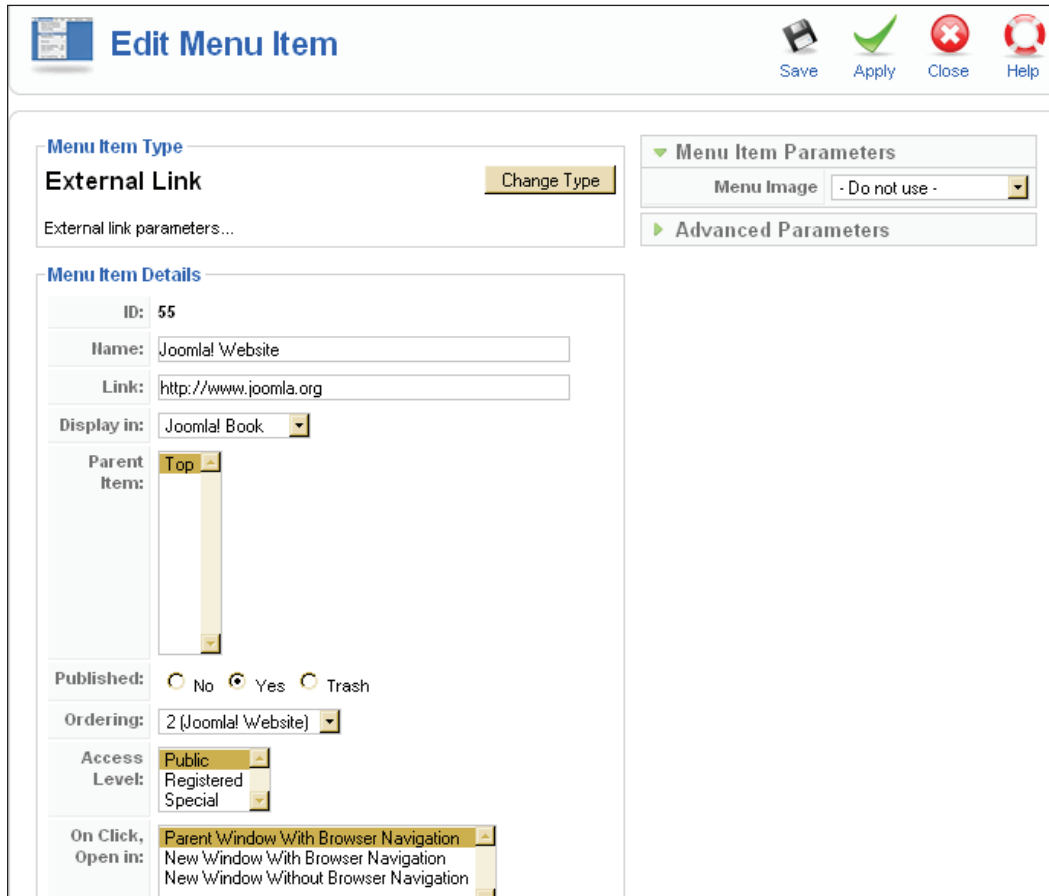


Figure 4.49: Site | Menus | Joomla! Book | Edit

Name: The name of the link that appears in the menu (Joomla! Website)

Link: The link to the page – this section is grayed out.

Display in: Of course **Joomla! Book** in our new menu.

Published: Click on **Yes** for the menu to be published.

Ordering: Since this is the first item, there is no order yet. New items are basically put at the end.

Access Level: Should this menu item be visible for our **Public** (visitors), **Registered** or **Special** groups?

On Click, Open in: What should happen when someone clicks on the link? Should the target be displayed in the same browser window, a new browser window with navigation, or a new browser window without navigation?

After this you can click on **Apply** or **Save**.

You have just created a menu. But before it can be displayed in the front end, you will have to publish the module. In the menu click on **Extensions | Modules** and there on the **Publish** icon in **Joomla! Book**. You can change the position of the menu by using the triangles.

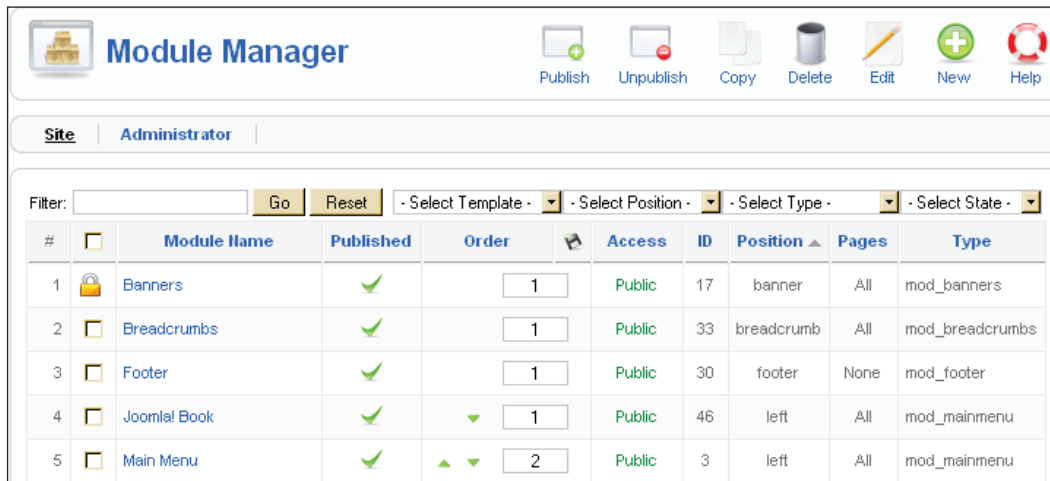


Figure 4.50: Modules – Site Modules

If you now call up your website, your new menu **Joomla! Book**, should appear above the **Main Menu**:



Figure 4.51: New Menu Joomla! Book

Customizing Joomla!

Unfortunately your menu looks different than the Main Menu underneath it. The blue border is missing. That attribute is defined in the CSS file of the template and can, if course, be modified. For that, the module has to invoke a particular CSS class. In our case, the class with which our menu is displayed is called `module`. You can check that by taking a look at the HTML source code of the website.

Source code for the new menu:

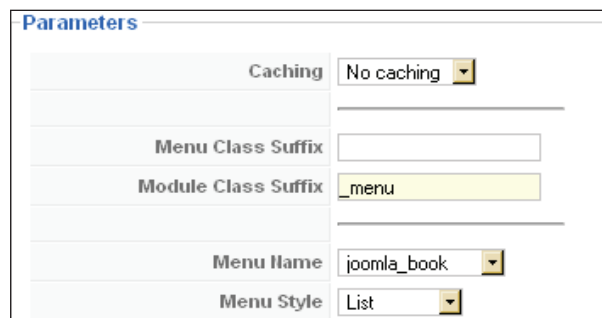
```
<div class="module">
  <h3>Joomla! Book</h3>
  <ul class="joomla_book">
    <li class="level1 item1"><a href="http://www.joomla.org"
      target="_blank">Joomla! Website</a></li>
  </ul>
</div>
```

The main menu, on the other hand, invokes the class `module_menu`.

Source code for Main Menu

```
<div class="module_menu">
<h3>Mainmenu</h3>
  <ul class="mainmenu">
    <li class="level1 item1"><a href=
      "http://localhost/Joomla150/index.php?option=com_frontpage&
      Itemid=1">Home</a></li>
    <li class="level1 item2"><a href=
      "http://localhost/Joomla150/index.php?option=com_content&
      Itemid=2">Joomla! License</a>
    </li>
  </ul>
</div>
```

During the editing of the module, you can ensure that the new menu also invokes that class. Click on **Extensions | Module Manager | Joomla! Book**. Simply enter the missing text, `_menu`, into the **Module Class Suffix** field.



The screenshot shows the 'Parameters' form for a Joomla! module. The fields are as follows:

Caching	No caching
Menu Class Suffix	
Module Class Suffix	_menu
Menu Name	joomla_book
Menu Style	List

Figure 4.52: Modification of the CSS Class

After this modification, your menu will be displayed the way you wanted. If you now click on the **Joomla! Book** link, a browser window with navigation should open and the website of the project should appear.



Figure 4.53: New Menu with a Modified CSS Class

Content Menu

The **Content** menu contains everything that is associated with content. Content in terms of text formatted with HTML is called an *article* in Joomla!. Joomla! articles are organized according to the following structure:

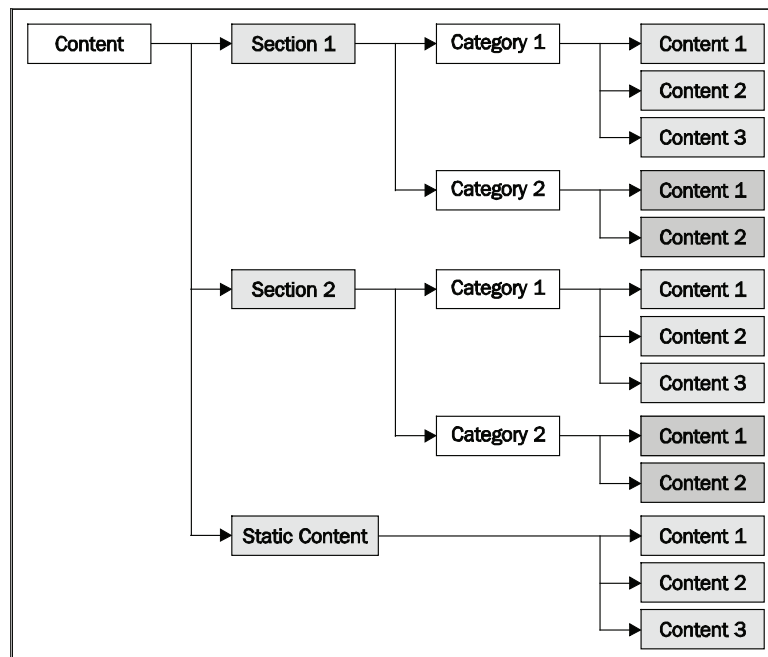


Figure 4.54: Content Structure

Customizing Joomla!

This structure can be compared to the directory tree of a hard disk. You can create as many articles and categories as you want. You can use this structure on an article but don't have to. You have the option of archiving elements. In that case, the structure is assumed by the archive.

This structure has its advantages and disadvantages. The biggest advantage, in my opinion, is that there is a structure and that the administrator therefore must abide by a certain hierarchy. This enhances the overview to begin with and serves to keep the website straightforward.

The disadvantage is that this two-stage structure cannot be changed and that therefore a number of administrators feel limited in individualizing their website. But this is mostly subjective. By using a clever combination of menus and content, every imaginable navigational structure can be produced.

The biggest problem usually is: *What do I want to present and in what context?*

The **Content** menu offers several workspaces for editing content and structures:

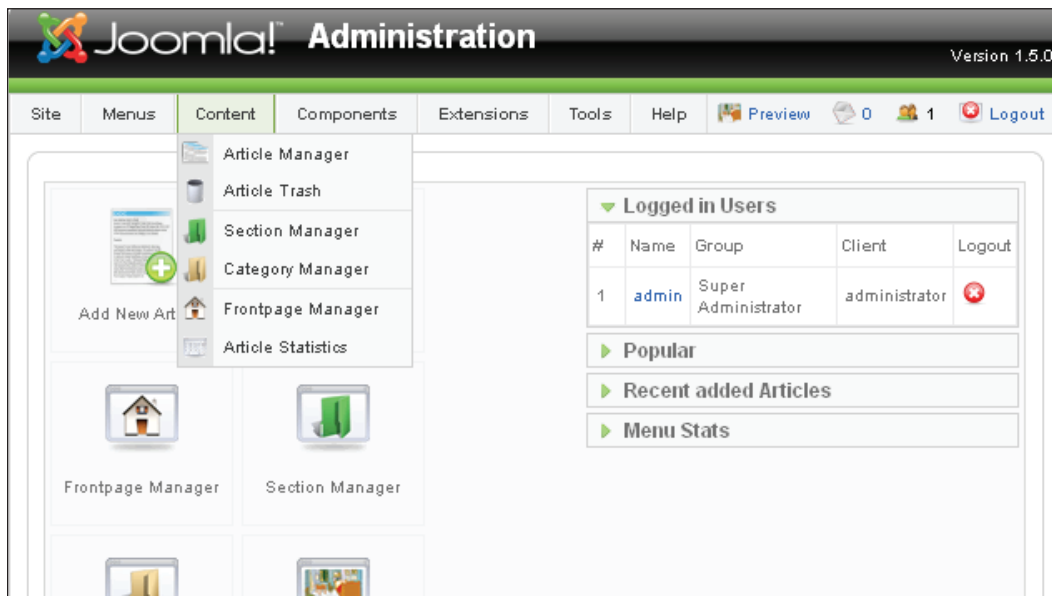


Figure 4.55: Content Menu

Article Manager

The menu item **Content | Article Manager** takes you to the overview mask for your articles. This is the central switch for the organization of the articles on your website.

You can filter the displayed content according to Section, Category, Author, and State in four option lists in the information space. There is also a search field with which you can search through the titles.

There is a navigation bar in the lower portion that you can use to leaf through the content. One of the option lists lets you define how many items you want to see in the list. The setting that you entered under **Site | Configuration** is the default.

The list is sorted by **Section, Category, Author, and State**:

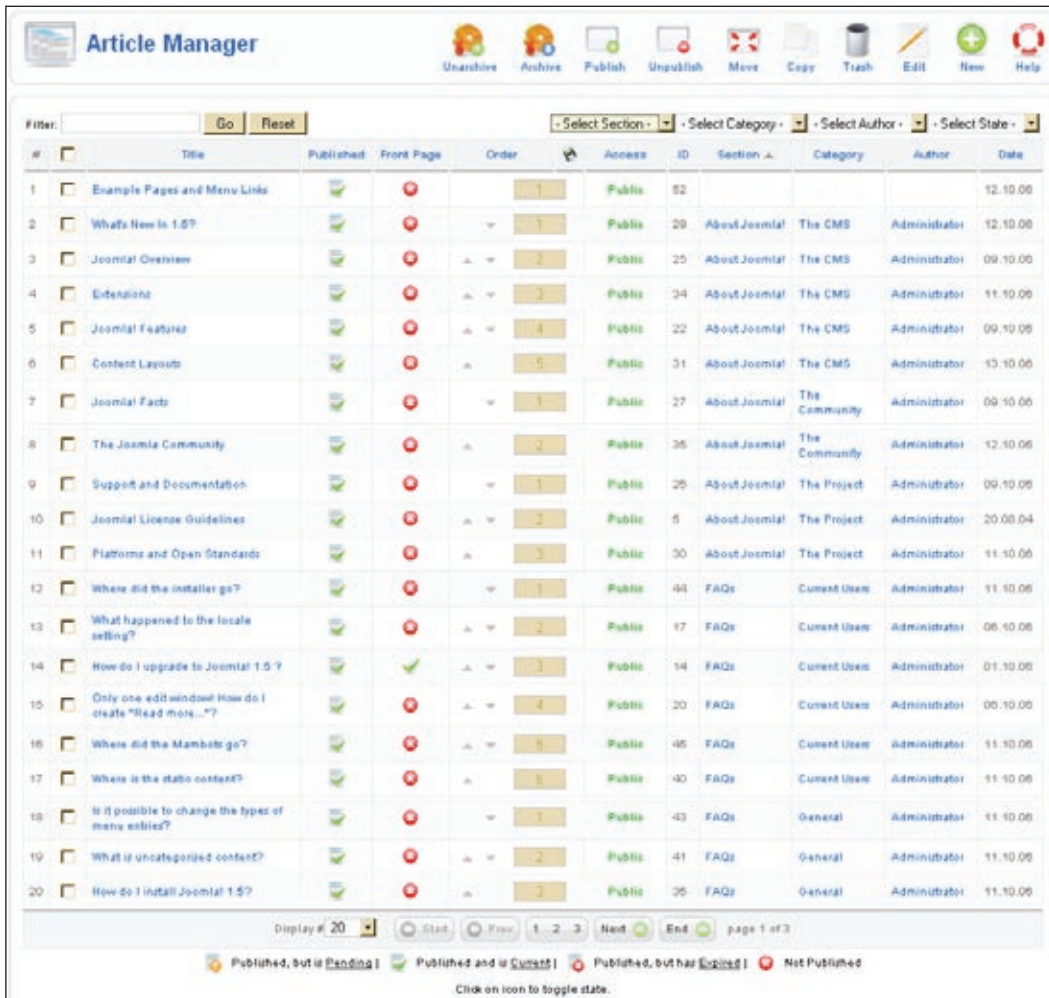


Figure 4.56: Content | Article Manager

There is a checkbox in front of the title that you can use to select the elements that you want to edit. If you select the checkbox in the header, all of the elements of the list are selected.

Customizing Joomla!

The **Title** is the link to the edit mode of the article. **Published** displays whether the article is published (green checkmark) or not (red cross). Besides these two symbols there is the possibility that the article is past its publication date and that the element is already in the archive. There is a legend for the icons in the footer of the page; the symbols for pending (in wait mode) and expired are also explained there.

Front Page displays whether the article is published (green checkmark) on the front page or not (red cross). In the **Order** column, you can move the items by clicking on the green arrows for each article. You can also sort the items by entering a number and subsequently clicking on the diskette icon in the header.

Under **Access**, you can see green **Public** links. You can change the access rights between the three groups of **Public**, **Registered**, and **Special** by clicking on one of these links. **ID** is the dataset number inside of the MySQL table. This ID will reappear in the URL to that item.

Section is the section that the article is assigned to. If you click on the link, you are taken to the **Section: [Edit]** screen. Articles without a section are considered *uncategorized*. **Category** is the structure underneath the sections to which this item has been assigned. Clicking on the link takes you to the **Category: [Edit]** screen. **Author** is the author of the article. Clicking on this link will take you to the **User Manager**. **Date** is the creation date of the article.

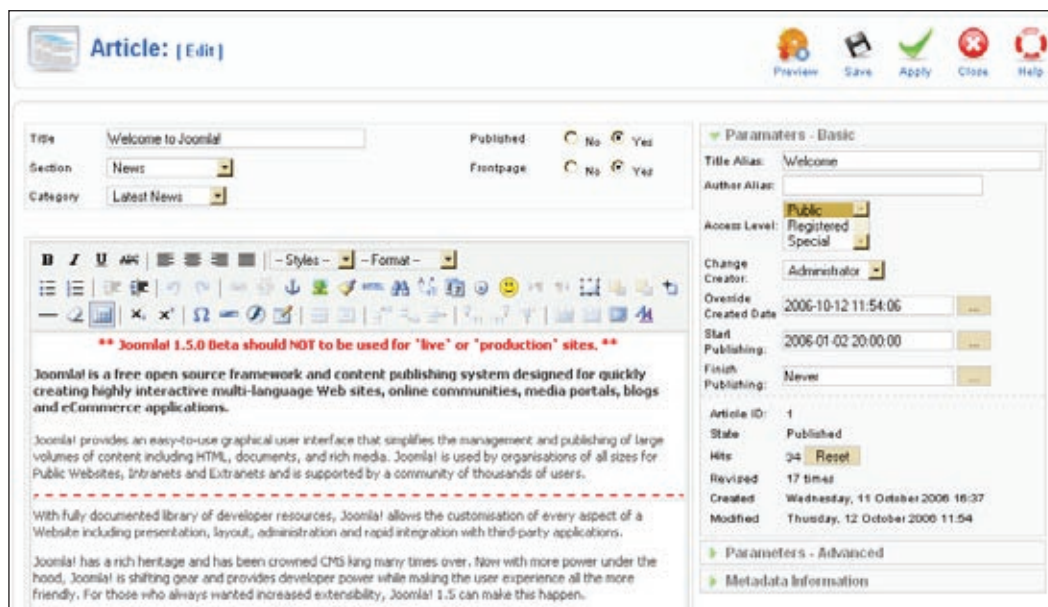


Figure 4.57: Article Edit

There were two text fields in Joomla! 1.0.x. One was for the hook and the other for the rest of the text. This separation has been discontinued in Joomla! 1.5, but the functionality remains. You can now insert the appropriate separators (continue, page break) with a mouse click in the one window. The separation of static and dynamic content elements has also been discontinued. Static articles are simply *uncategorized* articles and these can also be published on the front page.

The Archive

The idea behind an archive is that you don't delete older content, but preserve it for later use. You can archive the articles in **Article Manager** by clicking on the **Archive** icon. The article is then displayed in gray color in the overview:

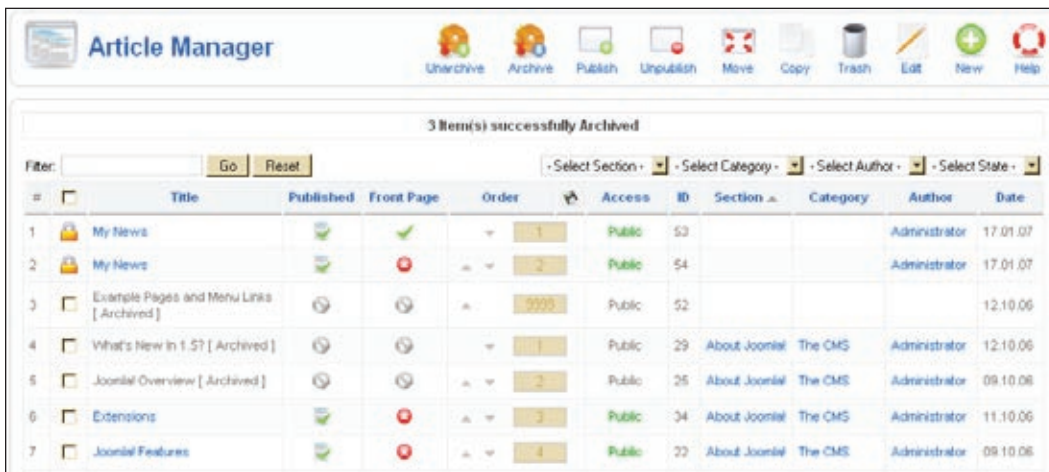


Figure 4.58: Archived Article

If you select **Archive** in the status filter, you can limit the display to archived articles. The archived articles can be shown in various situations on the website.

Creating a New Article

Let us use the example of a piece of *My News* that you want to display on the front page to go through the creation of an article. Click on **New**. An input template with a large editor text field appears:

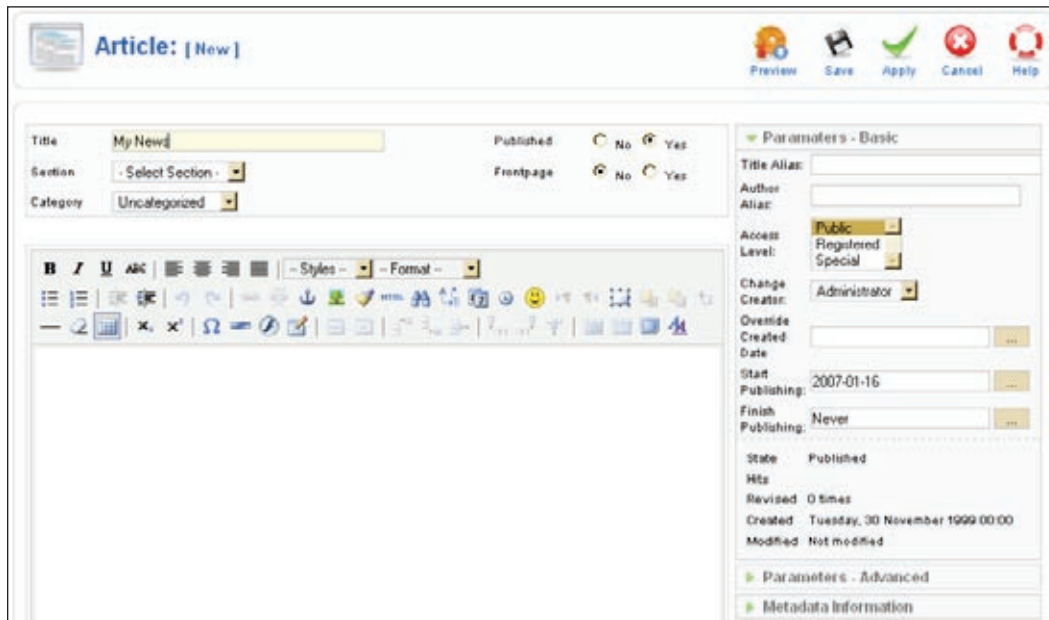


Figure 4.59: Template for a New Article

Enter a **Title** and some text. Select:

- Section: **uncategorized**
- Category: **uncategorized**
- Published: **Yes**
- Frontpage: **Yes**

Click on the **Apply** icon and take a look at your website. Your piece of news is indeed on the front page:

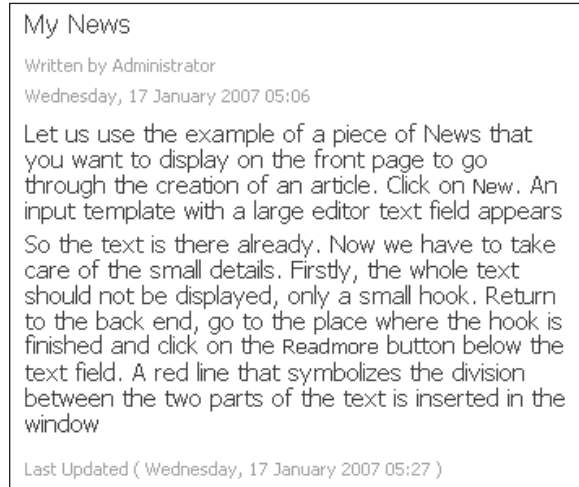


Figure 4.60: An Article on the Front Page

So the text is there already. Now we have to take care of the small details. Firstly, the whole text should not be displayed, only a small hook. Return to the back end, go to the place where the hook is finished and click on the **Readmore** button below the text field. A line that symbolizes the division between the two parts of the text is inserted in the window:

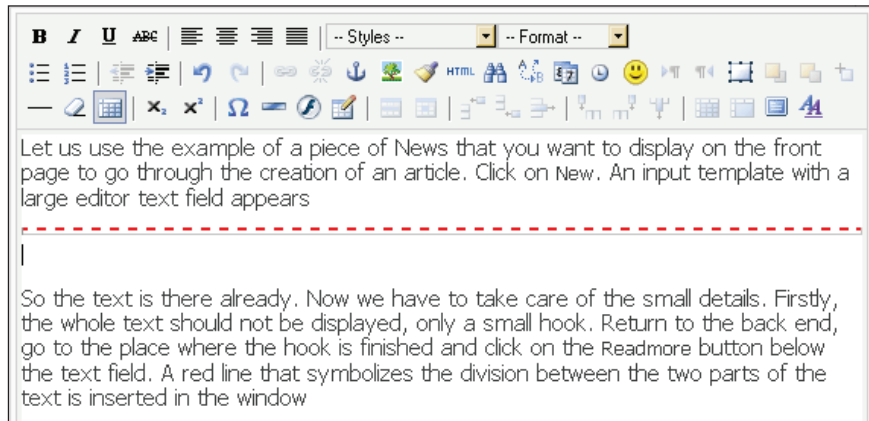


Figure 4.61: Continue Reading Section

Customizing Joomla!

You can also start to format the text with the Editor Tools. After you click on **Apply**, your front page looks like this:

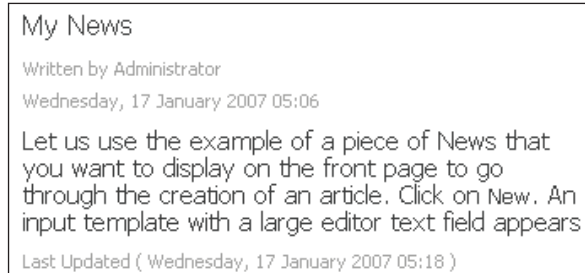


Figure 4.62: Article with the Continue Reading Section

Parameter Section

There are different parameter blocks here as well, just like in the creation of menus. The first block represents the basic parameters:

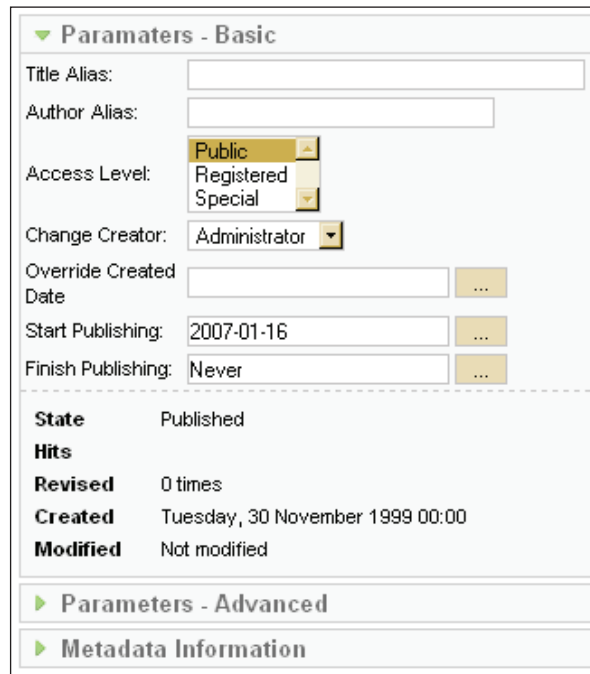


Figure 4.63: Basic Parameters

Title Alias: Here you enter an alternative title.

Author Alias: You can enter an author pseudonym here that will then be displayed on the website.

Access Level: This option is used for providing access to the three user groups.

Change Creator: You can change the writer of the news piece.

Override Created Date: At this point you can change the creation date of the article. If you click on the three dots, a calendar will appear to help you with the input.

Start Publishing: This lets you set a start date for publication. By default content is published immediately. If you click on the three dots, a calendar will appear to help you with the input.

Finish Publishing: Here you can enter an expiry date for the content. By default, content *never* expires. If you click on the three dots, a calendar will appear to help you with the input.

State: Current status (currently **Published**).

Hits: This helps you know how often this article has been accessed.

Revised: In the beginning, the article has version number 1. Every time it is saved, the version number increases by 1.

Created: This displays the date of creation.

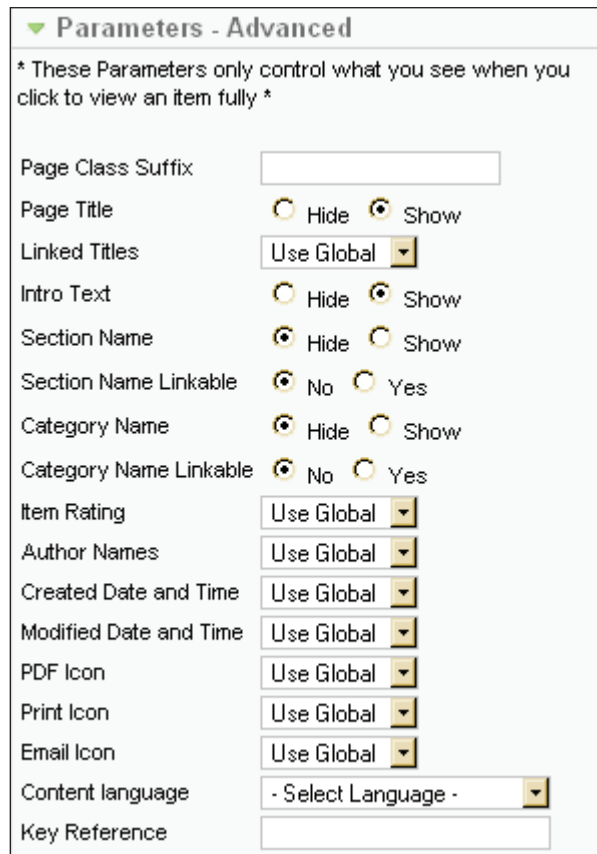
Edited: This displays the date of editing.



Figure 4.64: Calendar Element

Advanced Parameters

In the **Parameters–Advanced** section you can, among other things, overwrite the parameters that were defined for this content in **Site | Configuration**. These settings are only valid for the request of the complete article, in our case after clicking on the **Readmore** link:



▼ Parameters - Advanced

* These Parameters only control what you see when you click to view an item fully *

Page Class Suffix	<input type="text"/>
Page Title	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Linked Titles	Use Global ▼
Intro Text	<input type="radio"/> Hide <input checked="" type="radio"/> Show
Section Name	<input checked="" type="radio"/> Hide <input type="radio"/> Show
Section Name Linkable	<input checked="" type="radio"/> No <input type="radio"/> Yes
Category Name	<input checked="" type="radio"/> Hide <input type="radio"/> Show
Category Name Linkable	<input checked="" type="radio"/> No <input type="radio"/> Yes
Item Rating	Use Global ▼
Author Names	Use Global ▼
Created Date and Time	Use Global ▼
Modified Date and Time	Use Global ▼
PDF Icon	Use Global ▼
Print Icon	Use Global ▼
Email Icon	Use Global ▼
Content language	- Select Language - ▼
Key Reference	<input type="text"/>

Figure 4.65: Advanced Parameters

Page Class Suffix: As in the creation of menus, an individual class can be defined here.

Page Title: Use **Show** if you want to display the page title.

Linked Titles: Overwriting of the global settings.

Intro Text: Use to display or hide the Hook text.

Section Name: Use to display the name of the section.

Section Name Linkable: Use to display the section name as a link to all articles in this section.

Category Name: Use to display the category name.

Category Name Linkable: Use to display the category name as a link to all categories in this section.

The above seven parameters overwrite the global settings for this individual article.

Content language: The language of the article can be defined with this. This new feature offers interesting options with language control. Depending on the content, the website could be displayed in one or another language.

Key Reference: References can be defined here for export into the DocBook (<http://en.wikipedia.org/wiki/DocBook>) format. At the moment this is only relevant for developers in connection with a help system.

Metadata Information

In this tab you can enter a specific description and keywords for every page as metadata. The texts entered here are then inserted into the meta tags in the HTML source code of the article in addition to the metadata specified in the global configuration:



Figure 4.66: Metadata Information

When you click the **Add Sect/Cat/Title** button, these descriptions are automatically inserted into the keywords. This is, however, not particularly helpful with our example as the article is not categorized.

Images

How do images get into the content? The problem with images runs through the CMS world like a threat. The image has to be created (camera), transmitted (cable, WLAN), edited (image enhancement), loaded onto the server (FTP, PHP-Upload), and linked to the article.

Joomla! by default allows you to link images that are in the **Media Manager** into your article by means of a selection dialogue.

Position the cursor at the place within the text where you would like the image to be displayed. Then click on the **Image** button below the text window. The **Insert Image** dialogue appears:

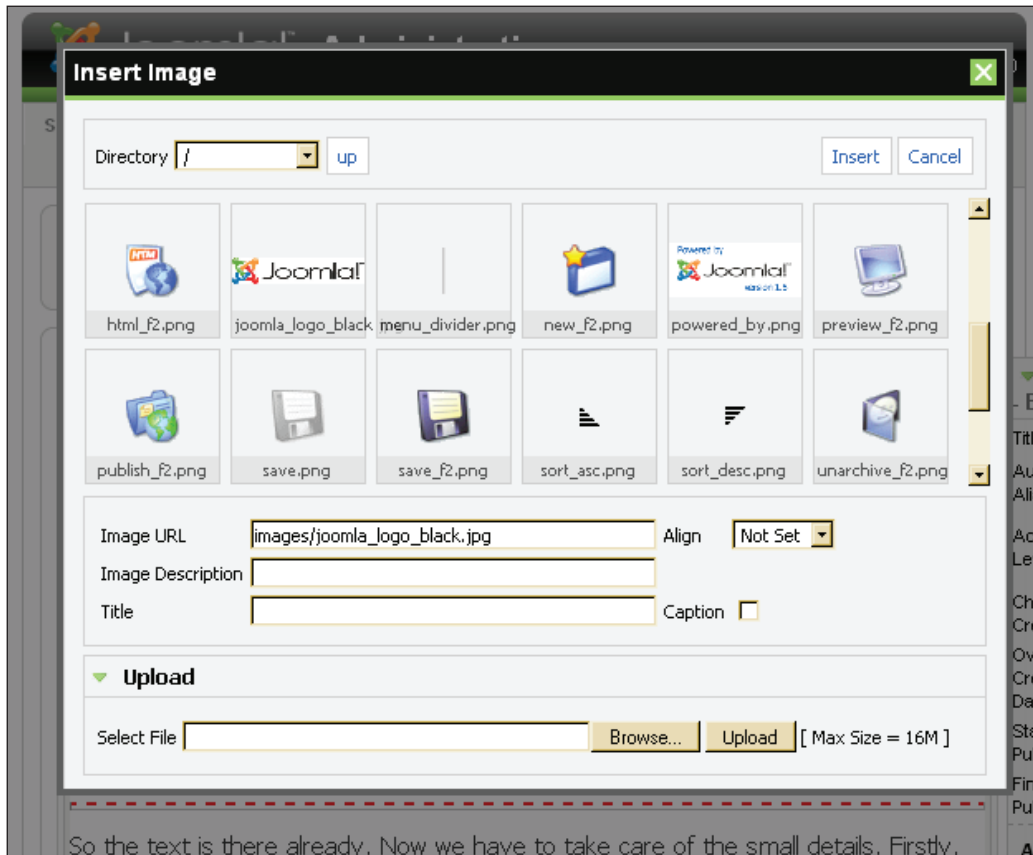


Figure 4.67: Inserting an Image

You can link whatever image you want from the media section into any content. If the image isn't in the media section yet, you can upload it by clicking the **Upload** icon.

Select a subdirectory and click on the image that you want to insert. Click on the **Insert** button. Your image now appears in the text window as seen in the following screenshot:

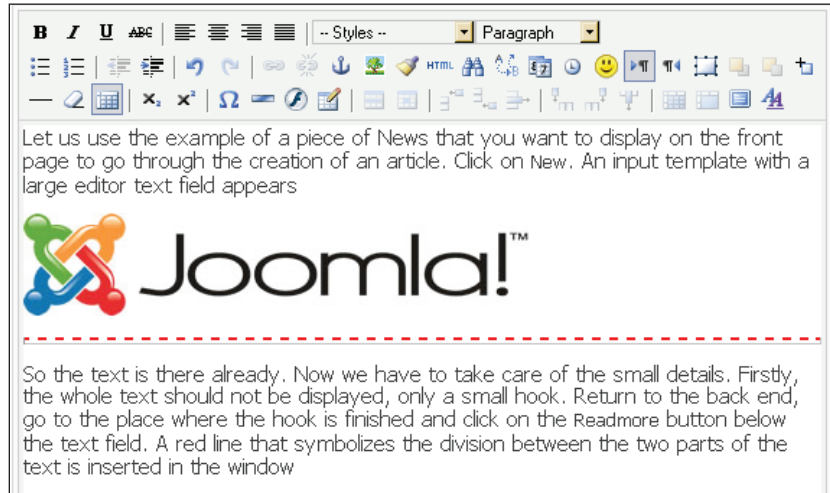


Figure 4.68: Inserted Image

Now we want to enhance the image a little. Mark the image by clicking on it. Then click on the icon with the tree above the text window – **Insert/edit image**. The image property dialog of the TinyMCE editor opens. You can add a description and in the **Appearance** tab you can set some other parameters. Here a 15-pixel space around the image is defined:

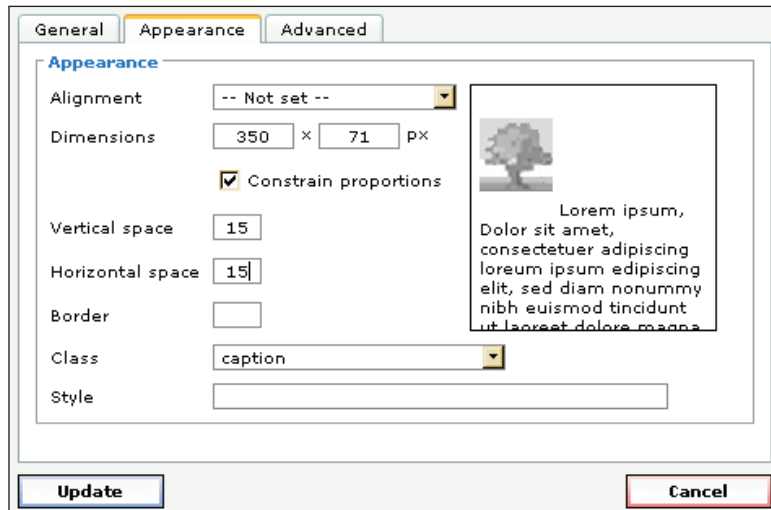


Figure 4.69: TinyMCE's Image Parameter Dialog

Customizing Joomla!

After you have clicked on the **Apply** icon, check the result either on your website or by clicking the **Preview** icon.

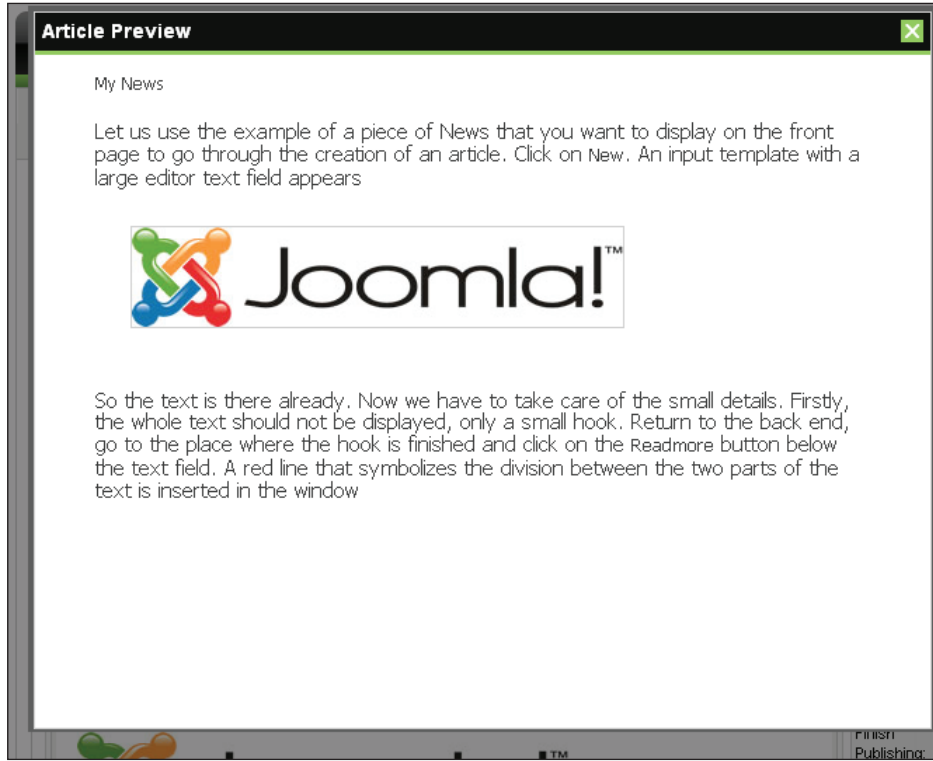


Figure 4.70: Content Preview

You can insert page breaks as easily as the **Readmore** link or the **Image**. Position the cursor where you want the page break to be and click on the **Pagebreak** button below the text field. A **Pagebreak** dialog box opens:

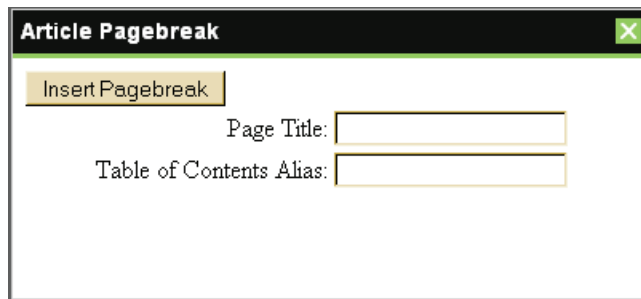


Figure 4.71: Page Break

You can enter a **Page Title** for the next page with this. In addition you can enter the text for the **Table of Contents Alias** that Joomla! automatically produces. Then click on **Insert Pagebreak**. The pagebreak is displayed as a gray line in the text window. Your article on the website now looks like the following screenshot:



Figure 4.72: Multiple Page Article Created by Joomla! - Page 2

Above the article text you can now see **Page x of y**, there is an **Article Index** at the top right with the text that you entered and below the text of the article are the links **Prev** and **Next** to navigate between the pages. You can integrate as many images and page breaks into a text as you want.

Article Trash

Just as with the menu items, you cannot delete articles directly. By clicking the **Trash** icon in article management, the articles are dispatched into the trash can. Clicking the **Restore** icon in the **Article Trash Manager** will restore the deleted articles; clicking the **Delete** icon will irretrievably delete them.



Figure 4.73: Article Trash Manager

Section Manager

The overview table of the **Section Manager** lists the information that you have already seen in other lists in a structured format. This time the categories that are contained in a section have been included as well as a count of how many are active and how many are in the trash basket:

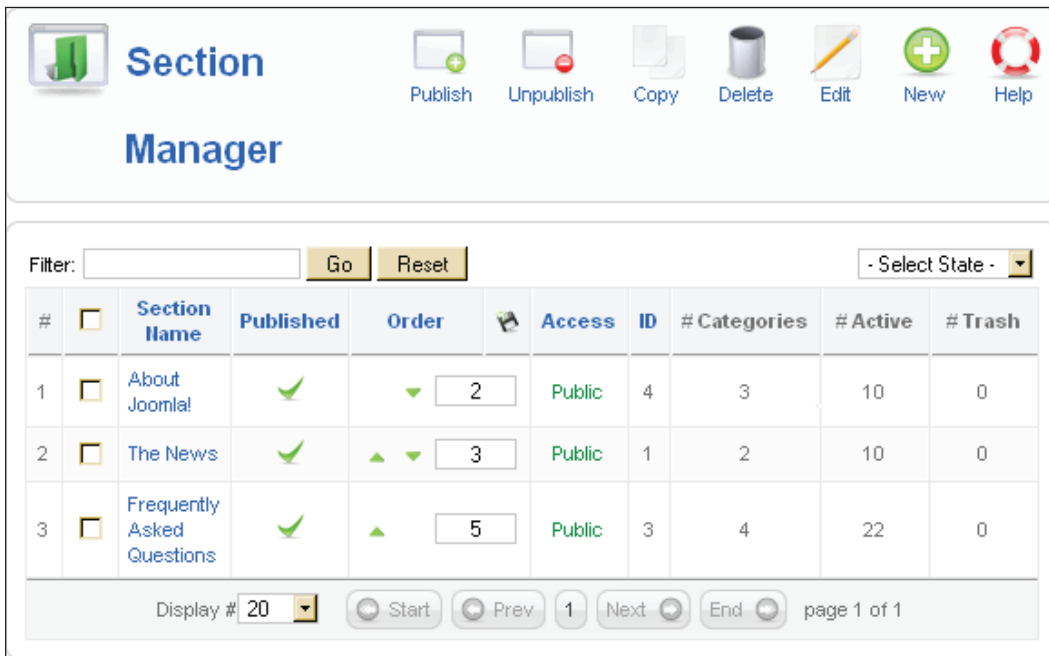


Figure 4.74: Section Manager

You can publish, copy, delete, and edit the available sections and create new ones here.

Editing Sections

As an example, let us modify the already existing section **The News** and link it to our **Joomla! Book** menu. Put a check on the checkbox in front of **The News** in the section list and click on the **Edit** icon:

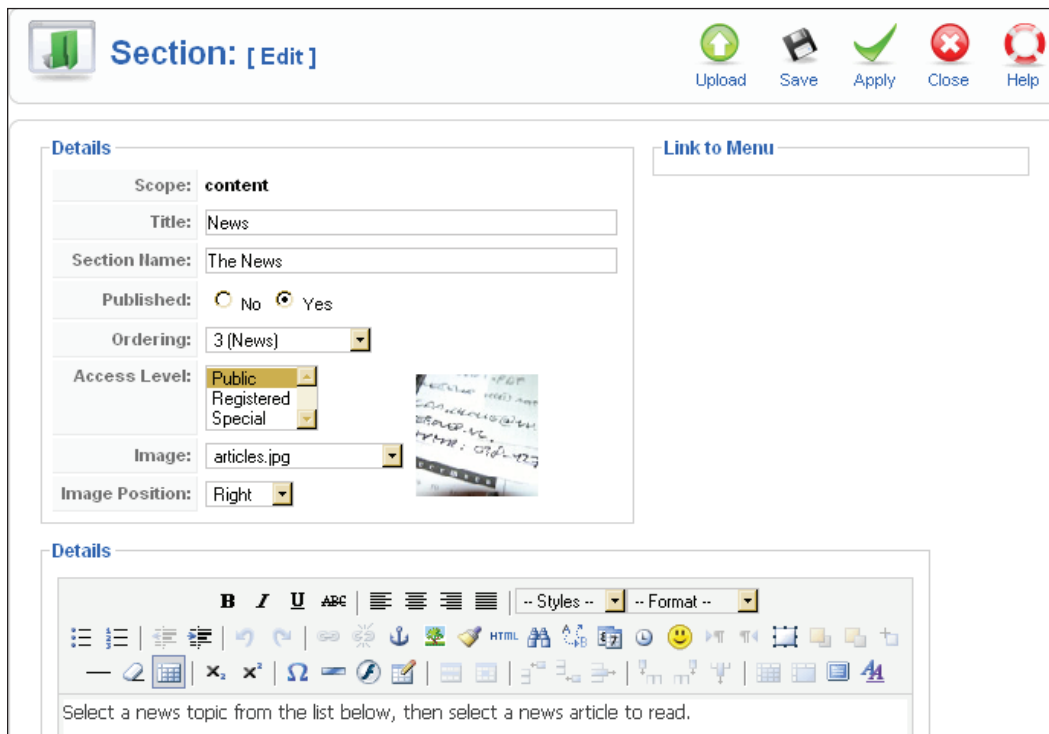


Figure 4.75: Content | Section Manager | Edit

Title: Title of the section (appears in the browser's title line).

Section Name: The name of the section that is displayed on the website.

Published: Whether the section should be published immediately or not.

Ordering: The sorting of the sections is defined here.

Access Level: User access to this element.

Image: Here you can select an image that will be displayed on the website when the category is requested. The image has to be in the root directory of the **Media**

Customizing Joomla!

Manager (/images/stories/)—here it is `articles.jpg`.

Image Position: You can define the alignment of the image here.

Details Window: If you have chosen a WYSIWYG editor in **Site | Configuration**, text editing is available here.

Link to Menu: This option is empty here, but you can link this section to some other menu, say Joomla! Book, using **Menus | Main Menu**. Here you can specify Joomla! book in the **Display in** option:

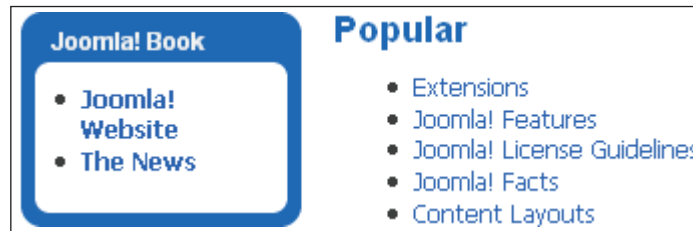


Figure 4.76: The News in a New Menu

Category Manager

You can edit categories in the same way as the sections. You can see the information structure in the overview table that you are already familiar with from the section display, but this time expanded by the assigned section, the number of articles it contains that are active, and those that are in the trash basket.

#	<input type="checkbox"/>	Category Name	Published	Order		Access	ID	Section	# Active	# Trash
1	<input type="checkbox"/>	Latest News	✓	1		Public	1	News	5	0
2	<input type="checkbox"/>	General	✓	1		Public	31	FAQs	8	0
3	<input type="checkbox"/>	The Project	✓	1		Public	25	About Joomla!	3	0
4	<input type="checkbox"/>	Newsflash	✓	2		Public	3	News	5	0
5	<input type="checkbox"/>	Current Users	✓	2		Public	28	FAQs	6	0
6	<input type="checkbox"/>	The CMS	✓	2		Public	29	About Joomla!	5	0

Figure 4.77: Content | Category Manager

When you allocate your newly created articles by means of the menu item **Content | Article Manager | Edit** to the **Latest News** category and the **News** section, they automatically appear in the newly created section lists.



The screenshot shows the Joomla! Category Manager interface. At the top, there's a title bar "Category Manager: [Content]" with several action icons: Publish, Unpublish, Delete, Edit, New, and Help. Below the title bar is a filter section with a "Filter:" label, a search input field, "Go" and "Reset" buttons, and dropdown menus for "Uncategorized" and "- Select State -". The main part of the interface is a table with the following columns: #, Category Name, Published, Order, Access, ID, Section, # Active, and # Trash. The table contains six rows of data.

#	Category Name	Published	Order	Access	ID	Section	# Active	# Trash
1	Latest News	✓	1	Public	1	News	5	0
2	General	✓	1	Public	31	FAQs	8	0
3	The Project	✓	1	Public	25	About Joomla!	3	0
4	Newsflash	✓	2	Public	3	News	5	0
5	Current Users	✓	2	Public	28	FAQs	6	0
6	The CMS	✓	2	Public	29	About Joomla!	5	0

Figure 4.78: Section and Category Allocation

Frontpage Manager

Frontpage Manager has a special mission. The front page is the title page of your website. When you are editing the individual articles, you can activate the **Frontpage** switch. The article is then shown in the **Frontpage Manager**. The construction of the list is similar to that of the article list. Here you can sort the articles within the front page. You can see the state, section, category, and author of the article. Uncategorized articles are called **Static Content** here. You can limit the display with the filter options. If you have more articles than will fit on the front page, navigation links are activated on the website below the front page articles.

The screenshot shows the Joomla! Frontpage Manager interface. At the top, there is a title bar with a home icon and the text "Frontpage Manager". To the right of the title bar are several icons: Archive, Publish, Unpublish, Remove, and Help. Below the title bar is a filter section with a "Filter:" label, a search input field, and a "Go" button. To the right of the search field are four dropdown menus: "Uncategorized", "- Select Category -", "- Select Author -", and "- Select State -". Below the filter section is a table with the following columns: #, Title, Published, Order, Access, ID, Section, Category, and Author. The table contains six rows of data. At the bottom of the table is a pagination section with "Display # 20", "Start", "Prev", "1", "Next", "End", and "page 1 of 1". Below the pagination section is a legend with four items: "Published, but is Pending", "Published and is Current", "Published, but has Expired", and "Not Published".

#	Title	Published	Order	Access	ID	Section	Category	Author
1	My News		1	Public	53	Static Content	Static Content	Administrator
2	Welcome to Joomla!		2	Public	1	The News	Latest News	Administrator
3	Example News Item 1		3	Public	6	The News	Latest News	Administrator
4	Example News Item 2		4	Public	7	The News	Latest News	Administrator
5	How do I upgrade to Joomla! 1.5 ?		5	Public	14	Frequently Asked Questions	Current Users	Administrator
6	Why does Joomla! 1.5 use utf-8 encoding?		6	Public	16	Frequently Asked Questions	General	Administrator

Figure 4.79: Frontpage Manager

Article Statistics

This section shows you the hit numbers of your articles. You can also display statistics about the browser, the operating system, and the domains of your visitors as well as a listing of the search terms with two additional tabs.

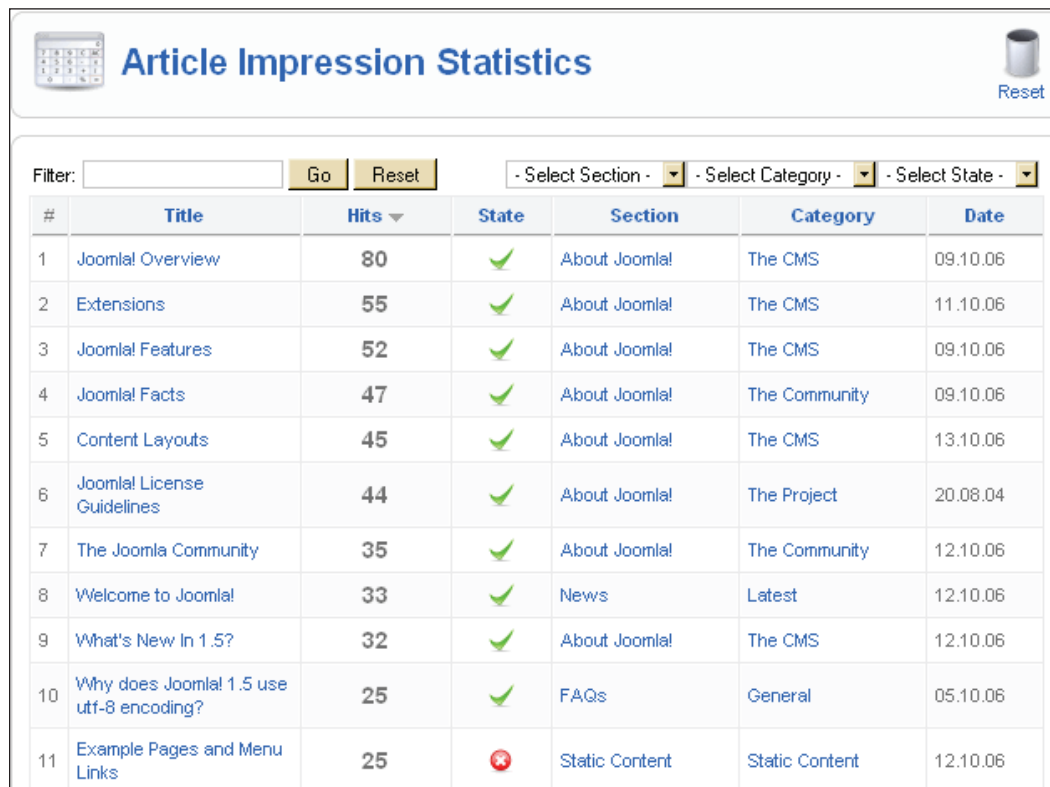


Figure 4.80: Article Statistics

Editing Content from the Front End

If you log in with a user name that has rights to change content, you will see a symbol of a pencil next to the articles:

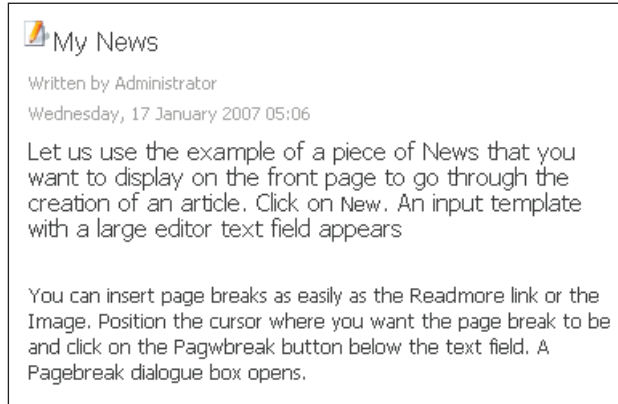


Figure 4.81: Pencil Symbol on the Website

If you click on this symbol, you can edit the content in a fashion similar to how you did it in the back end.

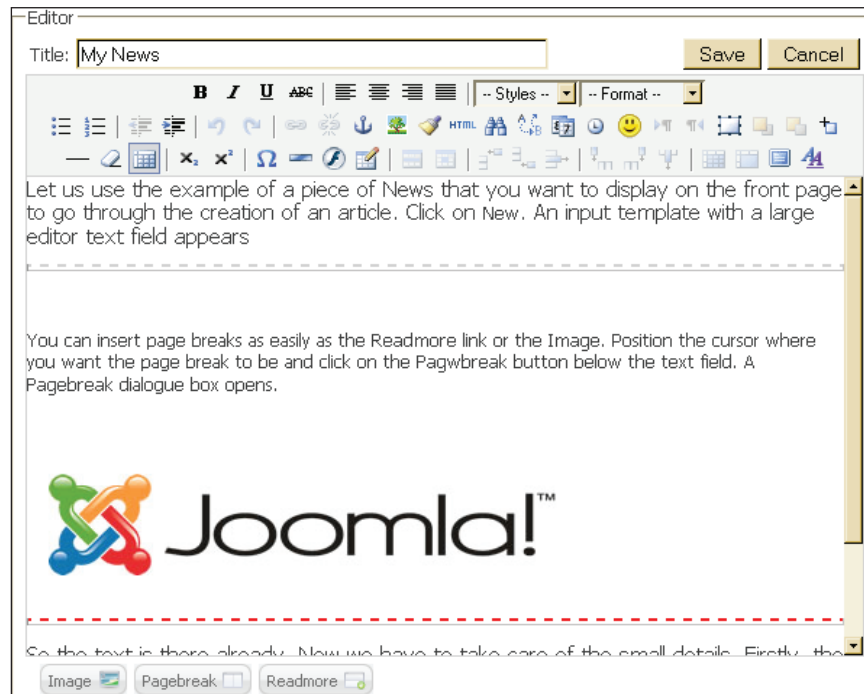


Figure 4.82: Editing Content from the Front end

The edit template is divided into three tabs, and permits image uploads and selection just like in the administration interface.

Tools

The **Tools** menu contains administrator tools: a private messaging system, a mass-mailing function, and the global checking-in of content elements.

Private Messaging

The **Read Messages** menu, contrary to its title, contains all messages, even those that have not been read. It is basically a mail inbox.

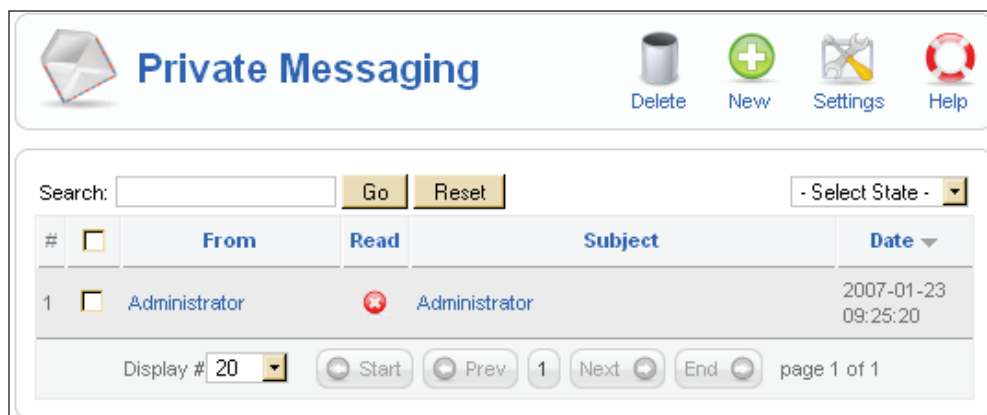


Figure 4.83: Private Messaging System

If you click on the subject matter, you can read it and the status is automatically set to read with a checkmark. System messages also get delivered, for example if someone has posted a new article. If you click on the **New** icon or the **Tools | Write Messages** menu, you can send a new message to users that have access to the Joomla! administration.

If you click on the **Settings** icon, you can configure the messaging system.

Lock Inbox: You can lock your inbox and stop any incoming mail. If you are the only administrator then this is OK, but otherwise you should leave your inbox open.

Mail me on new Message: This feature is really handy. Joomla! sends a notification to the email address that was set up in the user administration.

Auto Purge messages: You can have the system automatically delete messages when they reach a certain age in days.

Mass Mails

Delight your users with mass mailings!

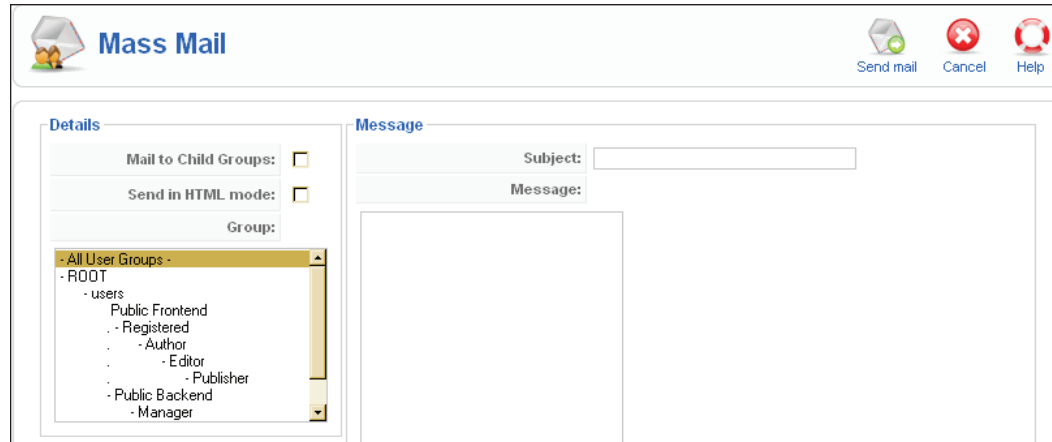


Figure 4.84: Mass Mail

As cynical as this may sound in our age of mass email spamming, it is the best way of contacting registered users. The **Mass Mail** component gives you the tool to do this.

Mail to Child Groups: If you check this, the subgroups of the selected user group also get mail.

Send in HTML mode: Check this if you want the mass mailing to be sent in the HTML format. HTML mail is getting more and more popular. Keep in mind, however, that many email clients can deactivate HTML display and that other recipients don't like HTML mail for various reasons.

Groups: Here you select which user group you want to target with the mass mailing.

Subject: The subject matter of your email.

Message: The actual text.

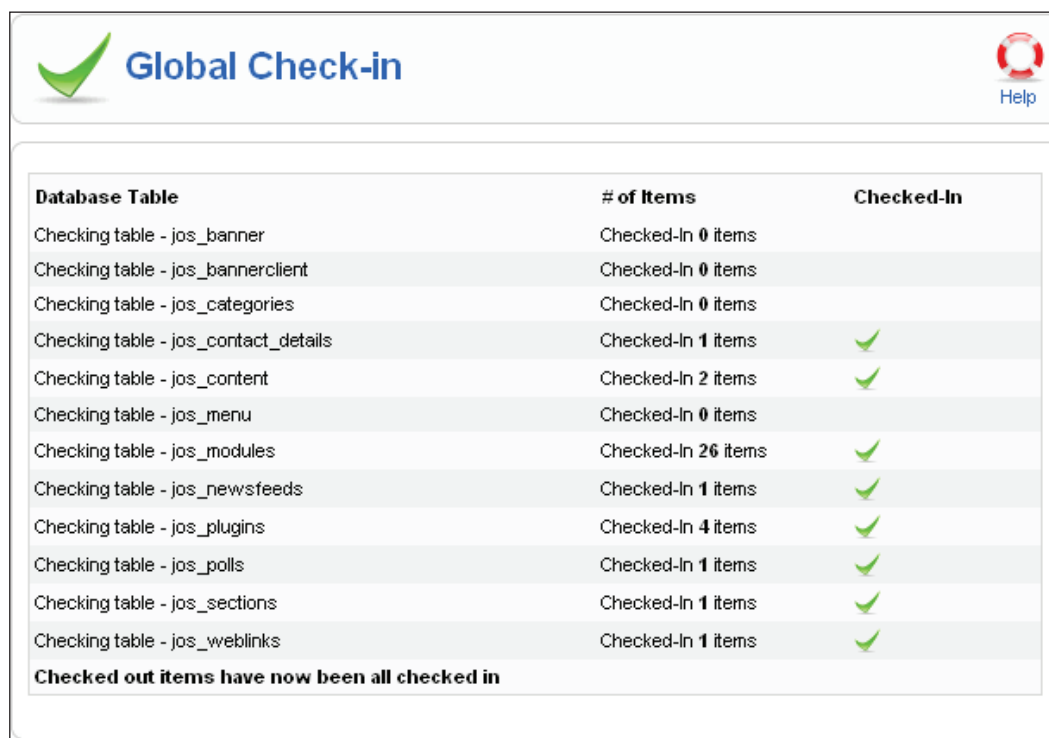
In order to be able to send your mass mailing, the mail settings in **Site | Configuration | Server | Mail Settings** have to be set correctly. If you want to send mass mailing from your local environment, but you don't have a mail server, you can enter the settings of your mail provider's SMTP server.

Global Check-In

If an authorized user requests the edit mode of a content element, this element is *checked out*. This just means that the user who has requested this element is allowed to edit it at that moment. During the edit, the other users will see an icon with a lock symbol in front of the name of the element. Once the document is saved again after the modification, it is automatically checked in again and the lock symbol disappears. But if the user, for example, closes his or her browser window or if the Internet connection goes down all of a sudden, the element remains checked out and can no longer be modified.

This is where the **Global Check-in** comes into play. If you click on this menu item, all elements that are being edited are checked in and you get a respective listing of the elements.

The disadvantage of this global check-in is the fact that indeed all elements get checked in. If someone is just about to make a modification, the element is checked in and somebody else can edit it as well. So be careful with this function and be aware who is online at the time.



Database Table	# of Items	Checked-In
Checking table - jos_banner	Checked-In 0 items	
Checking table - jos_bannerclient	Checked-In 0 items	
Checking table - jos_categories	Checked-In 0 items	
Checking table - jos_contact_details	Checked-In 1 items	✓
Checking table - jos_content	Checked-In 2 items	✓
Checking table - jos_menu	Checked-In 0 items	
Checking table - jos_modules	Checked-In 26 items	✓
Checking table - jos_newsfeeds	Checked-In 1 items	✓
Checking table - jos_plugins	Checked-In 4 items	✓
Checking table - jos_polls	Checked-In 1 items	✓
Checking table - jos_sections	Checked-In 1 items	✓
Checking table - jos_weblinks	Checked-In 1 items	✓

Checked out items have now been all checked in

Figure 4.85: Global Check-In

Summary

In this chapter we have seen how to customize Joomla! according to our needs. We have seen almost all options presented by Joomla! except the **Extension** and **Component** menus. These we will cover in the next chapter.

5

Components and Extensions

Extensions include components, modules, templates, and plug-ins. This chapter is going to cover the configuration of Joomla! based on extensions. All of the extension options that are available, other than components, are listed in the **Extensions** menu. Components are covered separately as a **Components** menu option. We will first take a look at the **Components** and then delve into the **Extensions**.

Components Menu

In software development, a component is a program that contains business logic, that can be accessed over defined interfaces, and that may possibly also have a user interface. Simply think of a component as a *Black box*. I put something in the front and something comes out of the back: You don't have to know what happens inside, what matters is that you can use the black box for various completely different purposes.

Components can be designed in a generic way and bundled into handy packages. This idea of a software component is similar in Joomla!. Business logic, such as banner administration or a forum, is written in a generic way and then works in Joomla! in concert with all of the templates and the Joomla! administration. A module often assumes the display of a website and a plug-in adds additional functionality to existing text (similar to a scripting language).

Banners

The banner component makes the display of advertising banners on your site possible. Banner switching with Joomla! is accounted for on the basis of purchased banner impressions. A banner can consist of graphics or text. Just like the wildly successful Google model, Joomla! websites can also sell advertising links. Every time that your site is accessed, a different banner is displayed from your banner administration. Every display counts as an impression. The text link is clickable and links to the site of the customer. In addition to text links, graphic banners can also be placed.

Components and Extensions

The **Banner** component offers customer, category, and banner administrations. With graphics banners, often so-called full banners are switched. A full banner has a size of 468 x 60 pixels and should not substantially exceed 20 KB in file size. The format is .gif, .jpg, or .png.

Let's walk through a banner switch. Create or copy a banner with the dimensions of 468 x 60 pixels:

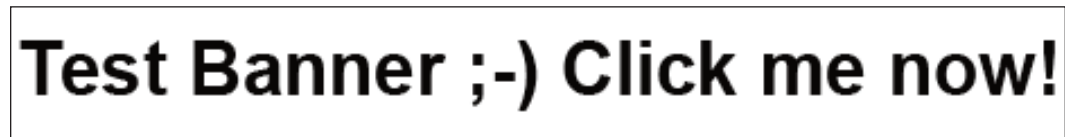


Figure 5.1: Test Banner with 468 * 60 Pixels

Banner Clients

Before you can switch a banner, you need a customer. Click on **Components | Banners | Clients | New**, open a new customer account and save it by clicking on the **Save** icon:

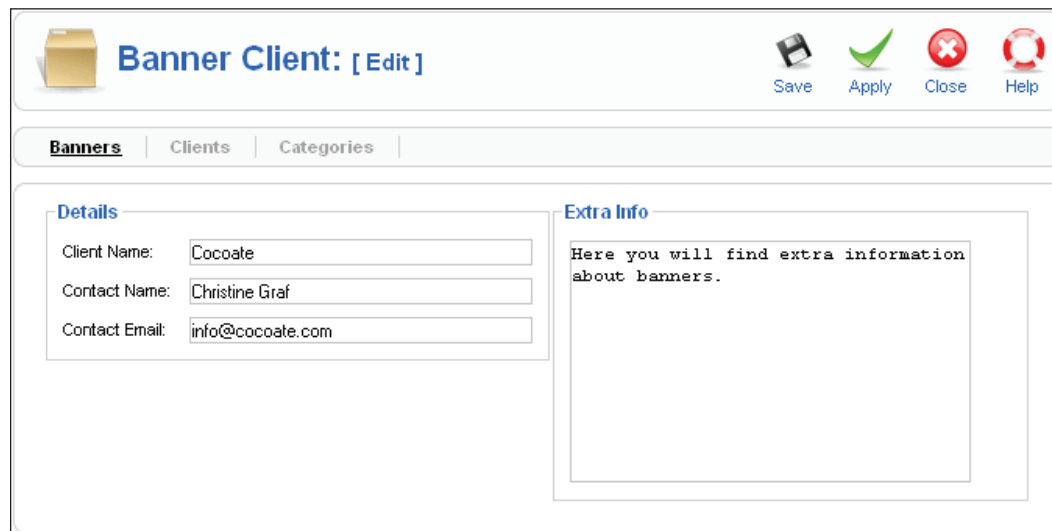
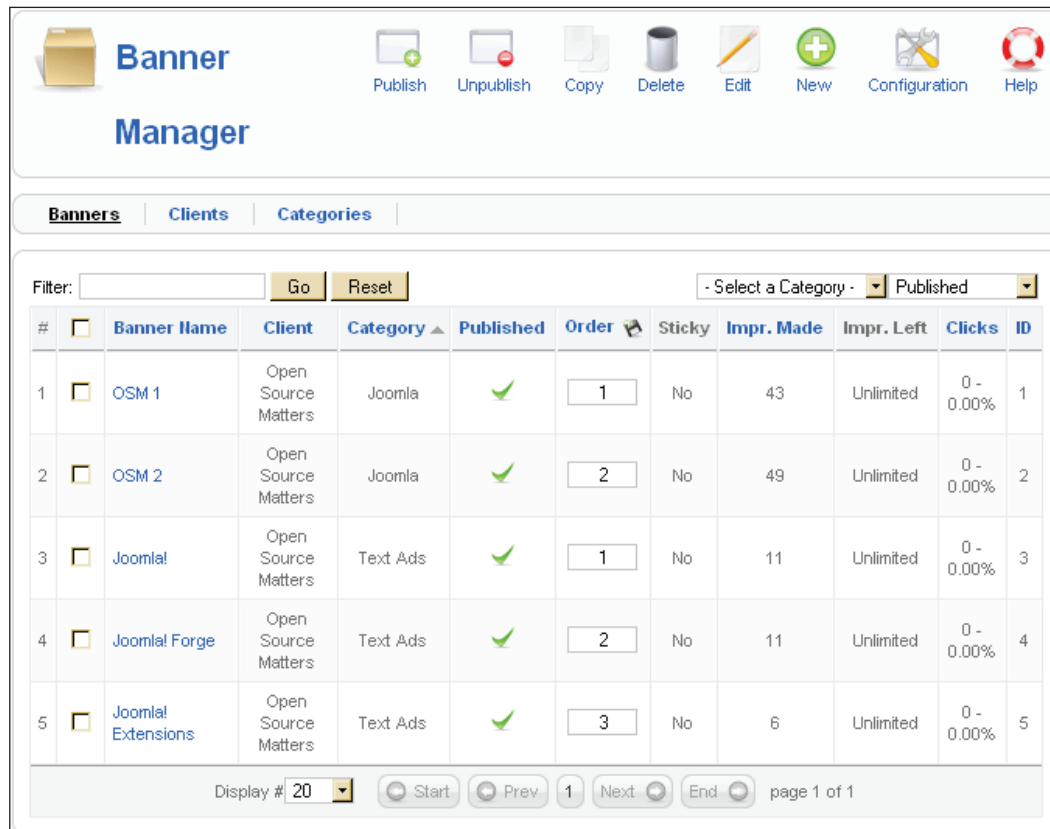


Figure 5.2: New Client

The **Banner Client** section, where you wind up after saving, now displays your new customer as well as the number of active banners of this customer (right now 0).

Manage Banners

Now the banner and/or the text link of this client has to be incorporated. Click on **Components | Banners | Banners**. You can see the overview of the existing banners:



#	<input type="checkbox"/>	Banner Name	Client	Category ▲	Published	Order	Sticky	Impr. Made	Impr. Left	Clicks	ID
1	<input type="checkbox"/>	OSM 1	Open Source Matters	Joomla	✓	1	No	43	Unlimited	0 - 0.00%	1
2	<input type="checkbox"/>	OSM 2	Open Source Matters	Joomla	✓	2	No	49	Unlimited	0 - 0.00%	2
3	<input type="checkbox"/>	Joomla!	Open Source Matters	Text Ads	✓	1	No	11	Unlimited	0 - 0.00%	3
4	<input type="checkbox"/>	Joomla! Forge	Open Source Matters	Text Ads	✓	2	No	11	Unlimited	0 - 0.00%	4
5	<input type="checkbox"/>	Joomla! Extensions	Open Source Matters	Text Ads	✓	3	No	6	Unlimited	0 - 0.00%	5

Figure 5.3: Banner Manager

You will see the following categories:

Banner Name: Name of the banner

Client: Client of the banner

Category: Category assigned to the banner

Published: Banner published or not

Order: Define the order in which the banners are switched with this

Sticky: Sticky banners are preferred or not

Components and Extensions

Impr. Made: The number of successful impressions

Impr. Left: The number of remaining impressions

Clicks: Clicks on the banner in numbers and as a percentage of impressions

ID: Dataset key

Graphic Banners

To switch to a new full banner, first upload the file from the media area (**Site | Media Manager**) into the `/images/banner/` directory. (If you want to delete a banner file, switch to **Detail View** in the media section and click on the cross.)

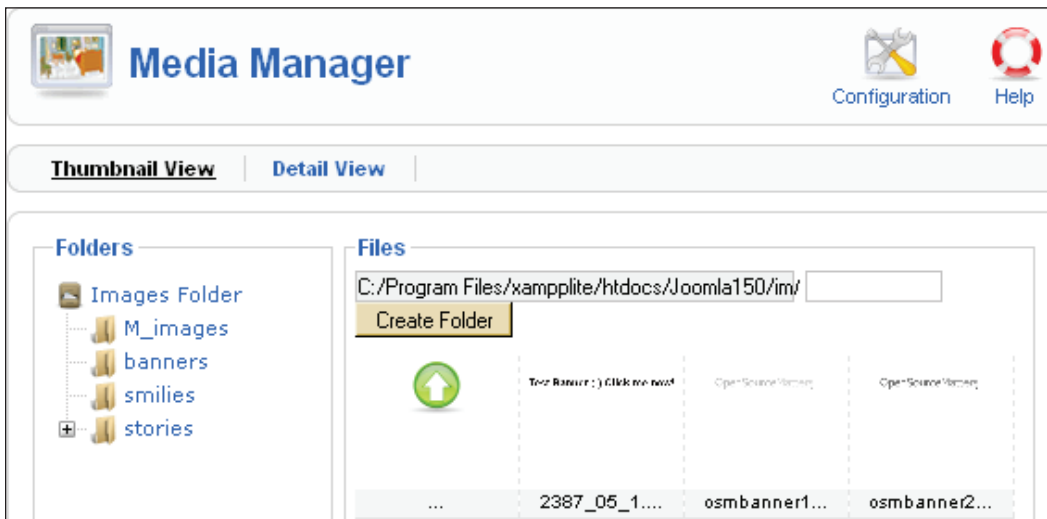


Figure 5.4: Upload Banner

After you have uploaded the banner, click on the **New** icon in the banner section (**Components | Banners | Banners** menu) and fill out the form:

Details

Banner Name:	<input type="text" value="Text Banner"/>
Show Banner:	<input type="radio"/> No <input checked="" type="radio"/> Yes
Sticky:	<input checked="" type="radio"/> No <input type="radio"/> Yes
Ordering:	<input type="text"/>
Category:	- Select a Category -
Client Name:	Cocoate
Impressions Purchased:	<input type="text"/> Unlimited <input checked="" type="checkbox"/>
Click URL:	<input type="text" value="http://www.cocoate.com"/>
Clicks	0 <input type="button" value="Reset Clicks"/>
Custom banner code:	<div style="border: 1px solid gray; height: 100px;"></div>
Description/Notes:	<div style="border: 1px solid gray; height: 40px;"></div>
Banner Image Selector:	2387_05_1.png
Banner Image:	Test Banner ;-) Click me now!

Figure 5.5: New Banner

You will need to give the following details:

Banner Name: Give the banner a meaningful name so that you can recognize it in the banner manager.

Show Banner: You can take a banner out of rotation here.

Sticky: Banners that are marked **Sticky** are displayed first.

Ordering: Here you define the order in which the banners are to be displayed.

Category: You can assign categories and then display banners from particular categories on the website.

Client Name: Select the client from the option list of existing clients.

Impressions Purchased: Enter the number of impressions purchased or check the **Unlimited** checkbox.

Click URL: Enter the URL of the website that the banner should link to.

Customer Banner Code: Here you enter the special banner code from the Affiliate Network (http://en.wikipedia.org/wiki/Affiliate_Networks).

Since this template is also designed to edit banners, you will also find a display of successful clicks and a **Reset Clicks** button that sets the clicks back to 0.

After a click on the **Save** icon, your banner should now be in rotation and be displayed on your website:



Figure 5.6: Client Banners on the Website

As the category selected for our banner was **Text Ads**, the banner is displayed in the advertisement section.

Text Links

To integrate a text link, simply enter the link in HTML format into the **Custom Banner Code** field. To be able to differentiate between different banner types, you should preset respective categories (i.e. *Text Ads*).

Banner Modules

The banners are displayed on the website by means of modules. You will learn more about banner modules later in this chapter under the **Modules** section.

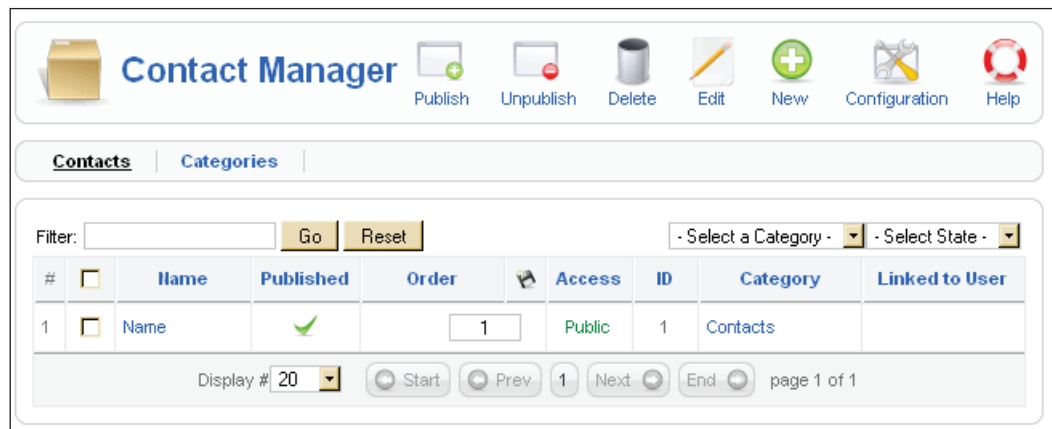
Contacts

It is often difficult for a customer surfing your site to contact you. Many employees normally work in different departments in companies and often only one email address (for example, `info@company.com`) is shown on the homepage or on a form and the customer has no idea who receives it.

To avoid this, Joomla! makes it possible to specify contact categories. You can register contact persons for your company. Joomla! then, depending on configuration, produces a table of coworkers and/or a contact form for every coworker on the site.

Contact Manager

This is where the individual employee contact forms are administered. You will find a co-worker by the name of **Name** in the sample data, who is assigned to the group **Contacts**. To follow this functionality, simply set up a new staff member. Click on **Components | Contacts | Contacts**. You can see the **Contact Manager** section and the previously mentioned contact from the sample data:



#	Name	Published	Order	Access	ID	Category	Linked to User
1	Name	✓	1	Public	1	Contacts	

Figure 5.7: Contacts

Click on the **New** icon and set up a new contact.

Components and Extensions

The template that pops up now is divided into three parts and is very extensive:

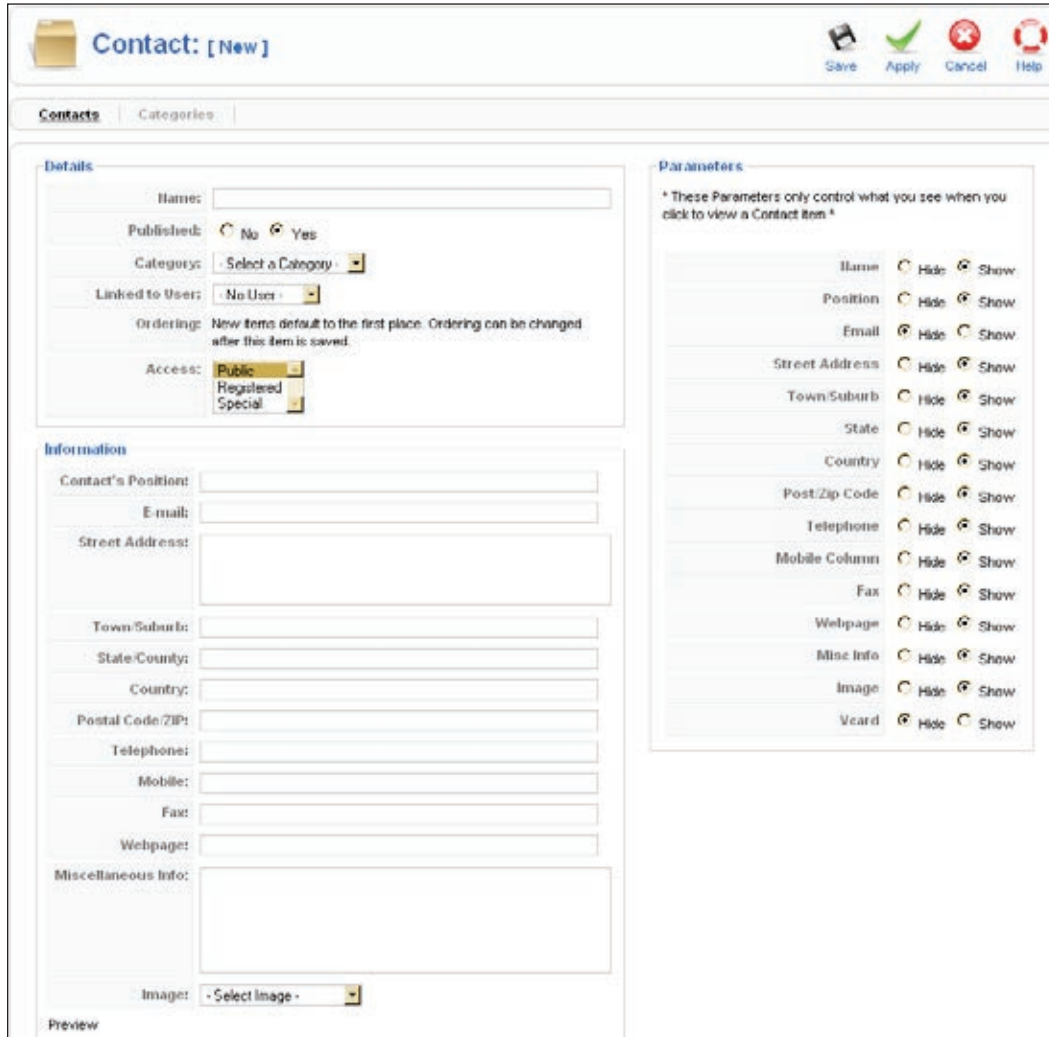


Figure 5.8: New

Details

This section has the following options:

Name: Name of the contact

Published: Yes or no

Category: Select the contact category. At the moment we have the sample category **Contacts**.

Linked to User: This option allows you to link a contact with a user account.

Access: Access to desired user

ID: Dataset number of the contact

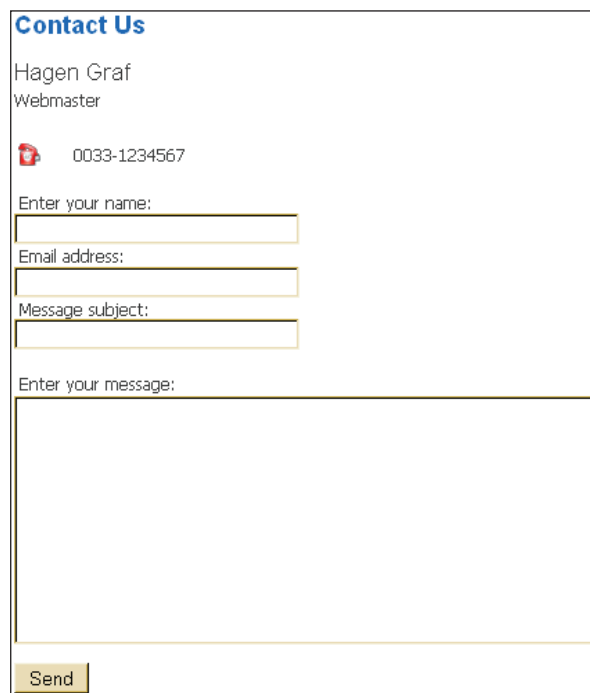
Information

Here are the information fields for this contact. Fields that don't have any values in them are automatically hidden. The **Miscellaneous Info** field can be used for individual descriptions. You can display an image of the contact from the `/images/stories/` subdirectory in the **Image** field. You can upload the image in the **Site | Media** menu.

Parameters


Here you can decide which fields you want to display and which you want to hide. A **Vcard** is a file format that can automatically be entered into numerous address directories (<http://en.wikipedia.org/wiki/VCard>).

Save the new contact by clicking on the **Save** icon. The new contact will now be displayed in the list. Now take a look at the website and click on the **Contact Us** link in **Main Menu**. You will see a form displayed:



Contact Us

Hagen Graf
Webmaster

 0033-1234567

Enter your name:

Email address:

Message subject:

Enter your message:

Figure 5.9: Contact Form

Components and Extensions

If you don't see any **Contact Us** link in your website, it means that you need to create a new item in **Main Menu** in your Admin area and select the **Contacts** type from the tree structure:

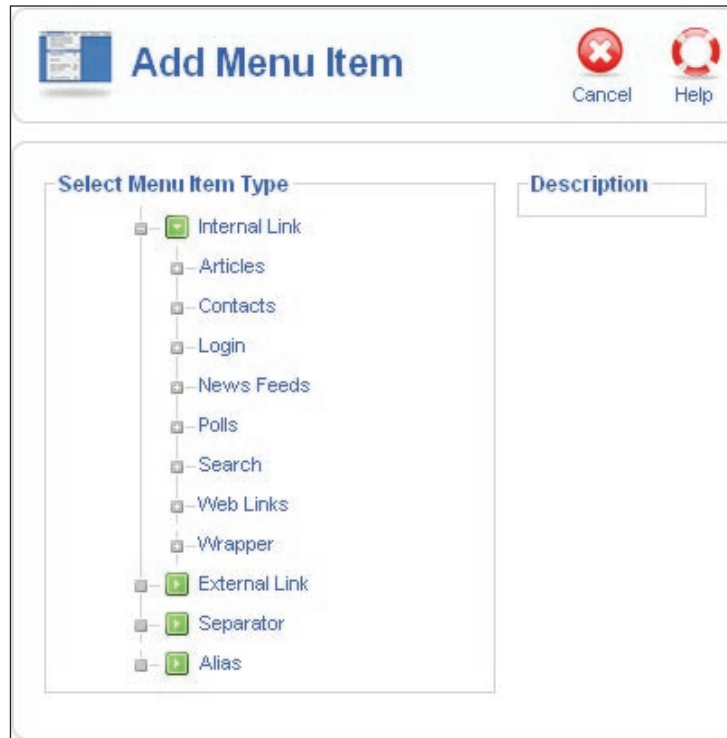


Figure 5.10: Contact Tree Structure

This will then bring the **Contact** item into the editing state. You can now configure the table by means of the parameter column to your heart's content.

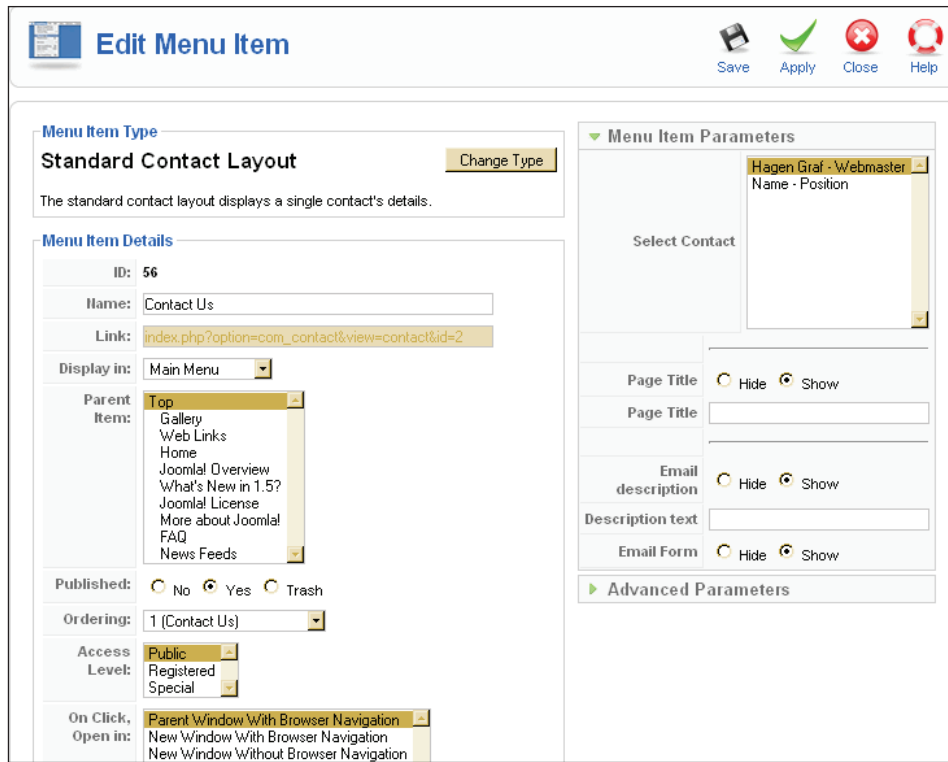


Figure 5.11: Contact Edit

You can also eliminate certain email addresses and words from the section.

Category Manager

In the **Categories** section you can set up new categories and edit existing ones. This section can be accessed via the **Component | Contacts | Categories** menu and via the **Categories** tab in **Contact Manager**:

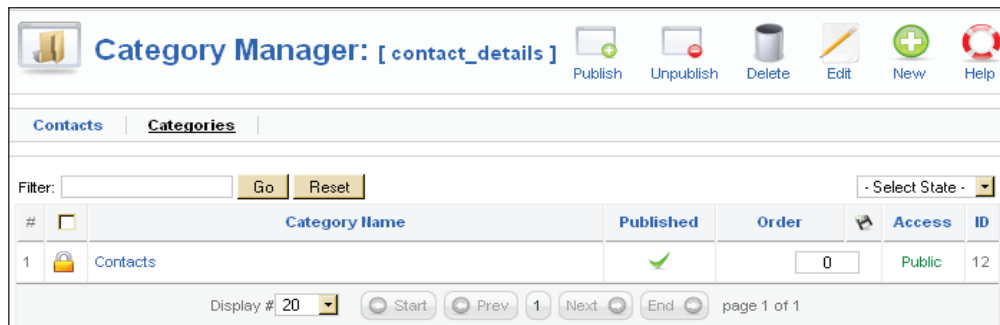


Figure 5.12: Contacts | Categories

The edit form is displayed after you click on the **Edit** icon or on the category name. You can also integrate an image and edit the description with the WYSIWYG editor.

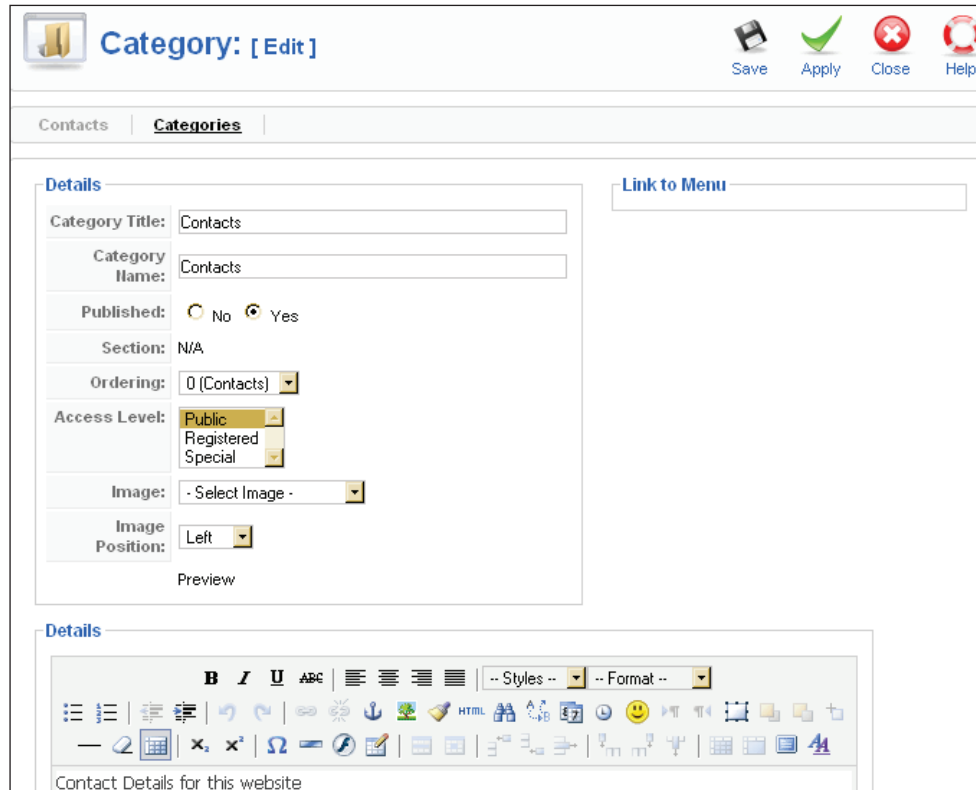


Figure 5.13: Contacts | Edit

News Feeds

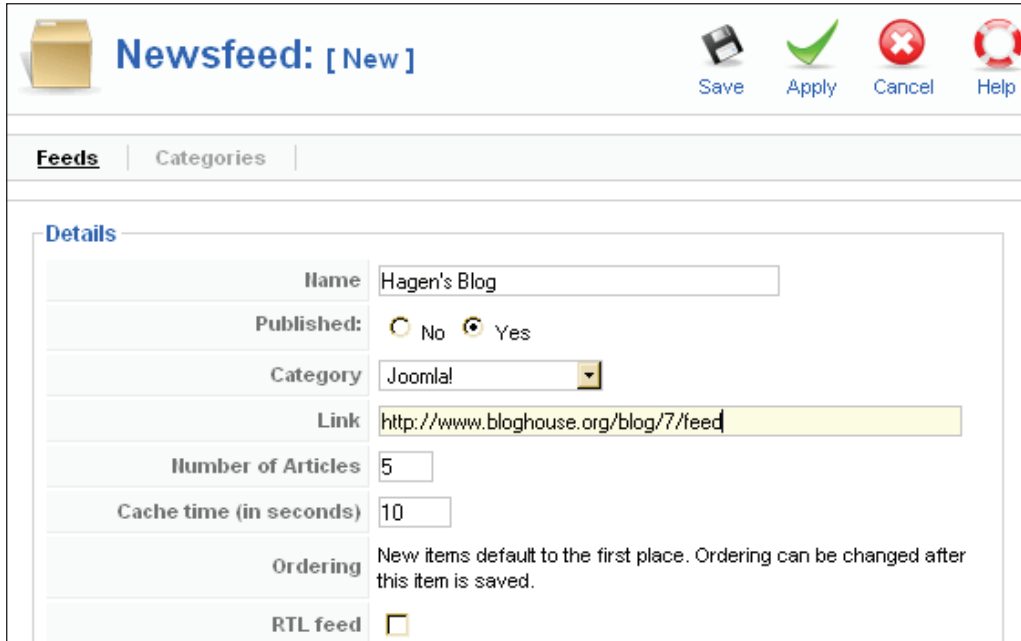
News feeds are a terrific thing. The ever-growing information abundance on the Internet makes it necessary to use effective organizational methods. If you regularly have to visit twenty web pages to check what's new, it takes up too much time. With a fifty or a hundred, it is absolutely hopeless to try to keep an overview. News feeds are an attempt to solve this problem.

With the news feed component you can merge feeds from other pages into your pages. To do that, a Category Manager and a Content Manager are at your disposal. The sample data already has several categories and numerous news feeds incorporated.

To integrate your own news feed, you can use a search engine or look for the small orange RSS button at the sites that you visit.

Feeds

In order to show you an example, we will enter the feed from the blog <http://www.bloghouse.org/blog/7/feed>. Click on **Components | News feeds | Feeds | New:**



Newsfeed: [New]	
<p>Feeds Categories</p>	
<p>Details</p>	
Name	Hagen's Blog
Published:	<input type="radio"/> No <input checked="" type="radio"/> Yes
Category	Joomla!
Link	http://www.bloghouse.org/blog/7/feed
Number of Articles	5
Cache time (in seconds)	10
Ordering	New items default to the first place. Ordering can be changed after this item is saved.
RTL feed	<input type="checkbox"/>

Figure 5.14: Integrating a New News Feed

These are some of the options:

Name: This is the name of the news feed that appears on your page.

Published: For it to be published immediately – **Yes**.

Category: Select a suitable category from the existing ones.

Link: This is the link to the news feed, the blog link given above.

Number of Articles: This refers to the number of articles that are to be merged.

Cache (in seconds): How long should the break between refreshes be (in seconds)?

Ordering: New news feeds are generally at the beginning. The sequence can be changed after it has been saved once.

RTL feed: Check this if you wish to use middle-eastern languages such as Hebrew and Arabic.

Components and Extensions

Your new news feed, assuming that you have Internet access, is now displayed on your website:



Figure 5.15: News Feeds on your Website

Categories

This is where you administer the news feed categories. The administration works the same as the **Category Manager** in **Contacts** (refer to Chapter 4).

Polls

The integrated poll module makes it possible for you to publish polls on your site. One poll is already included in the sample data. You can create a new poll by clicking on the **New** icon. You will need to fill in the following field:

Title: This is the title of your poll.

Lag: This determines the time in seconds that has to elapse before another selection can be made. This lag offers some kind of protection from the falsification of survey data.

Menu Items Link(s): You can select in which area of the site the poll is to be displayed. Multiple options are possible by holding down the *Ctrl* key and clicking the left mouse button.

Options: Here you can enter up to twelve answer options.

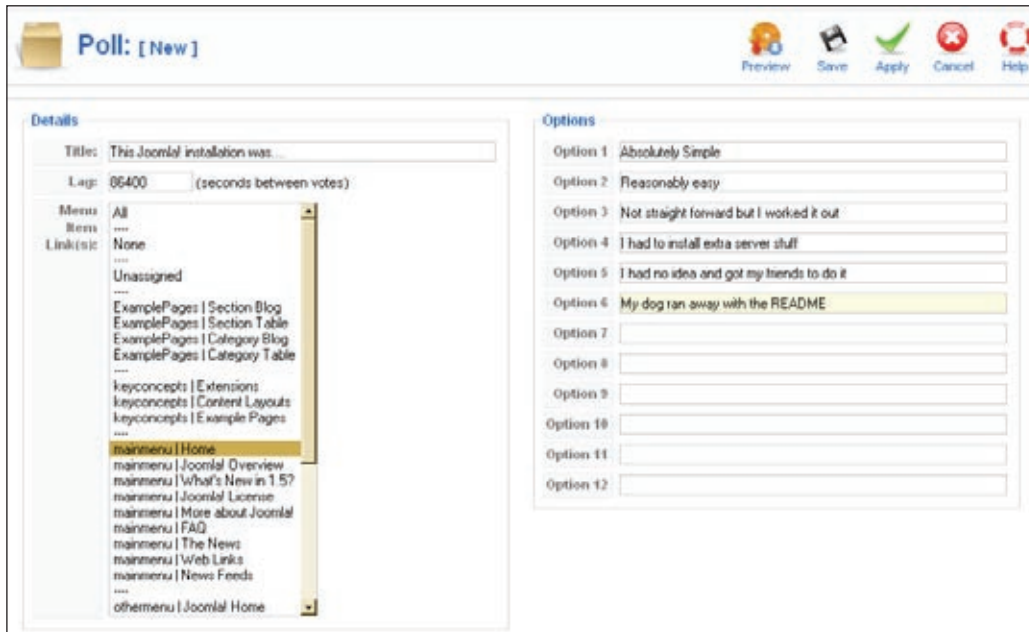


Figure 5.16: Components | Polls | Edit

Click the **Preview** icon to get a preview of your poll. To display it on your site, make sure that polls are provided in the current template (the poll you wish to see on your site should be published). You can now see the poll on your site:

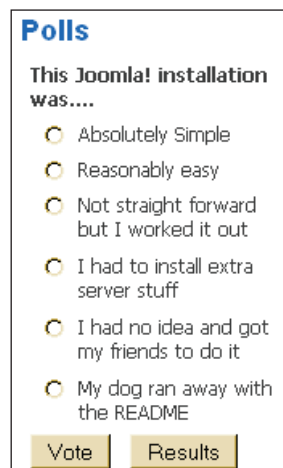


Figure 5.17: Polls on the Website

If you try the poll out and select an answer, a summary results template appears. The poll itself is no longer displayed, since by default it was attached only to the front page.



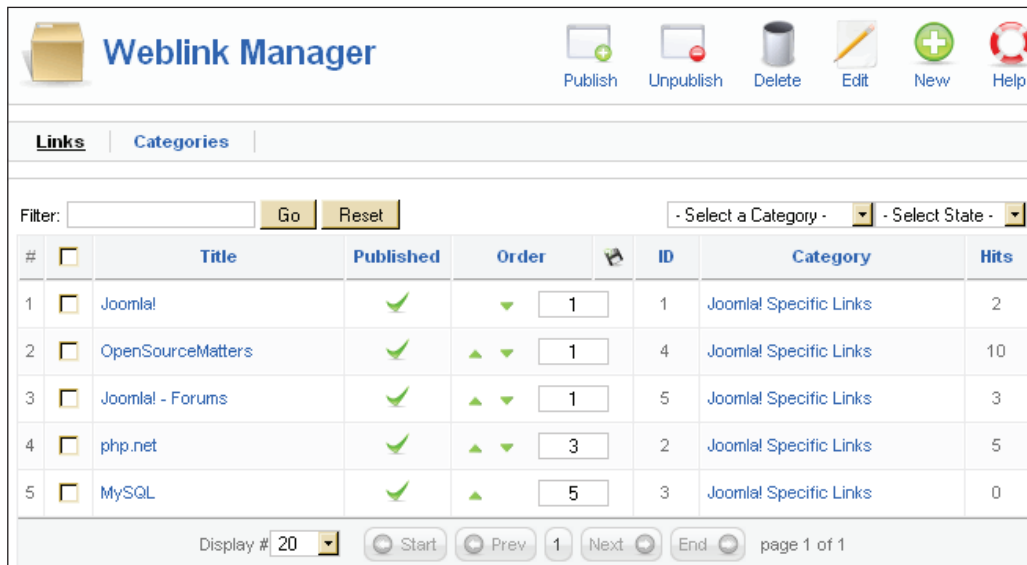
Figure 5.18: Summary Results on the Website

Web Links

Here you are constructing a link list or a download section that you can integrate with your website. Joomla! provides the categories again and counts the hits on the links. In the **User Manager**, you can have your registered visitors suggest links to be added to the list. These links will then appear in the list, which can be requested from the **Components | Web Links | Links** menu. These links, depending on user rights, may still have to be published.

Links

You can see all of the links that have been entered so far with their assigned categories and the hits up to now in the links overview template.



#	<input type="checkbox"/>	Title	Published	Order	ID	Category	Hits
1	<input type="checkbox"/>	Joomla!	✓	1	1	Joomla! Specific Links	2
2	<input type="checkbox"/>	OpenSourceMatters	✓	1	4	Joomla! Specific Links	10
3	<input type="checkbox"/>	Joomla! - Forums	✓	1	5	Joomla! Specific Links	3
4	<input type="checkbox"/>	php.net	✓	3	2	Joomla! Specific Links	5
5	<input type="checkbox"/>	MySQL	✓	5	3	Joomla! Specific Links	0

Figure 5.19: Links

Here you can enter individual links from scratch, and decide about publishing them and editing them:

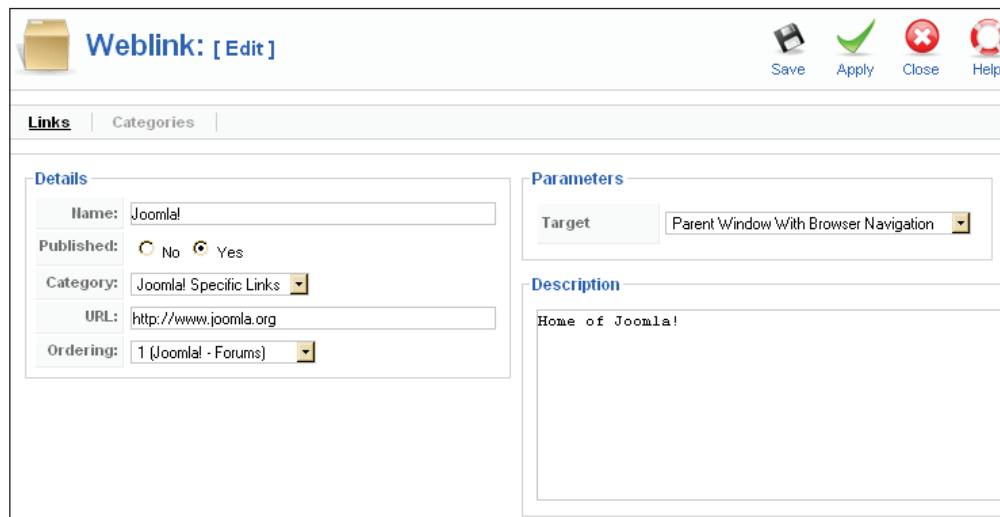


Figure 5.20: Links | Edit

Components and Extensions

The details include:

Name: The name of the link that appears on your website.

Published: Should the link be published immediately?

Category: Selection of an existing link category.

URL: The URL of the link.

Ordering: Sorting of the links.

Description: You can enter a comprehensive description of the link.

Target: You can make a decision in the **Parameters** section whether the link should appear in a new window (with or without navigation) or in the same window.

Categories

The link categories, which can be accessed in the respective **Category Manager**, are administered in the **Components | Web Links | Categories** menu:

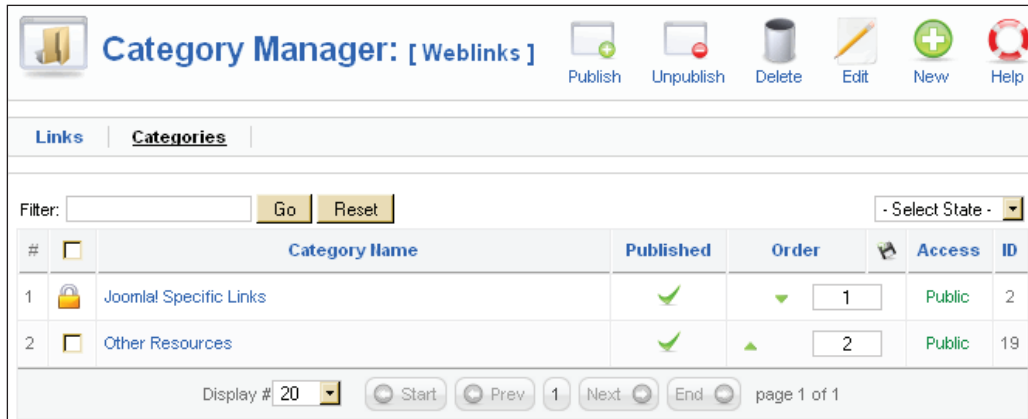


Figure 5.21: Components | Web Links | Categories

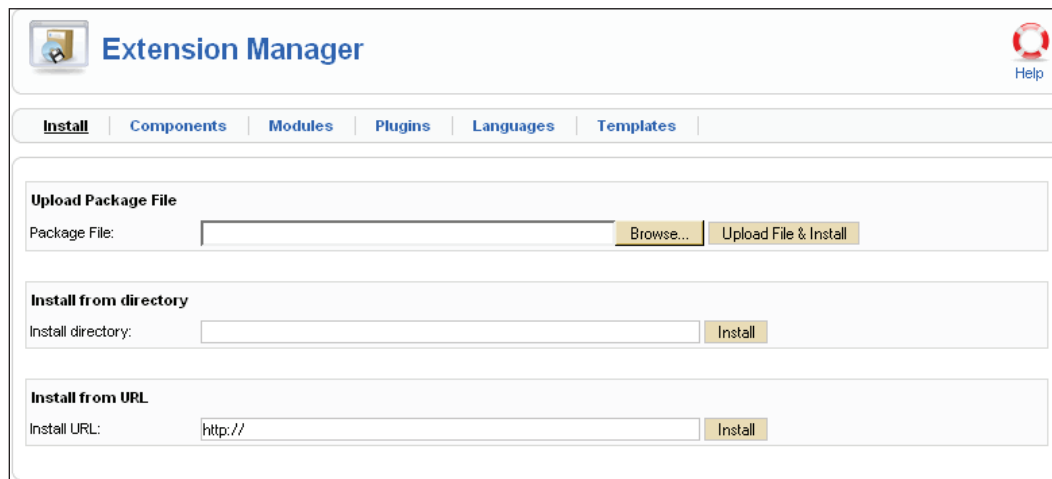
Extensions Menu

Now that we have the **Components** menu covered, let us learn about the **Extensions** menu. Extensions include modules, plug-ins, templates, and languages. There is also a central installer here that makes it possible to install and uninstall Joomla! extensions with just a few mouse clicks.

Install/Uninstall

In principle, anyone can write a Joomla! extension, pack it according to set rules, and integrate it into the website. You will find an overview of the installed extensions in this section. When you call up the **Extensions | Install/Uninstall** menu item, a template appears that gives you three options from installing extensions (also see Chapter 7):

- Uploading a file package
- Installation from your own directory
- Installation by entering a URL



The screenshot shows the Joomla! Extension Manager interface. At the top, there is a navigation bar with tabs for **Install**, **Components**, **Modules**, **Plugins**, **Languages**, and **Templates**. The **Install** tab is currently selected. Below the navigation bar, there are three main sections for installing extensions:

- Upload Package File:** This section contains a text input field for the "Package File", a "Browse..." button, and an "Upload File & Install" button.
- Install from directory:** This section contains a text input field for the "Install directory" and an "Install" button.
- Install from URL:** This section contains a text input field for the "Install URL" with the text "http://" pre-filled, and an "Install" button.


Figure 5.22: Install/Uninstall

Components and Extensions

The **Components**, **Modules**, **Plugins**, **Languages**, and **Templates** tabs each contain listings of the installed extensions. We will discuss the installation of additional components in a later section.

Each of the tabs, for instance the **Components** tab, displays a list of the components that have been installed by default. You will see the version information, the date of creation, and the author's information.

To uninstall a component, select it and click on the **Uninstall** icon in the toolbar.

 Careful! After uninstalling, the component is really gone, and with it all of the stored data. There is no such thing as a component trash basket!

Module Manager

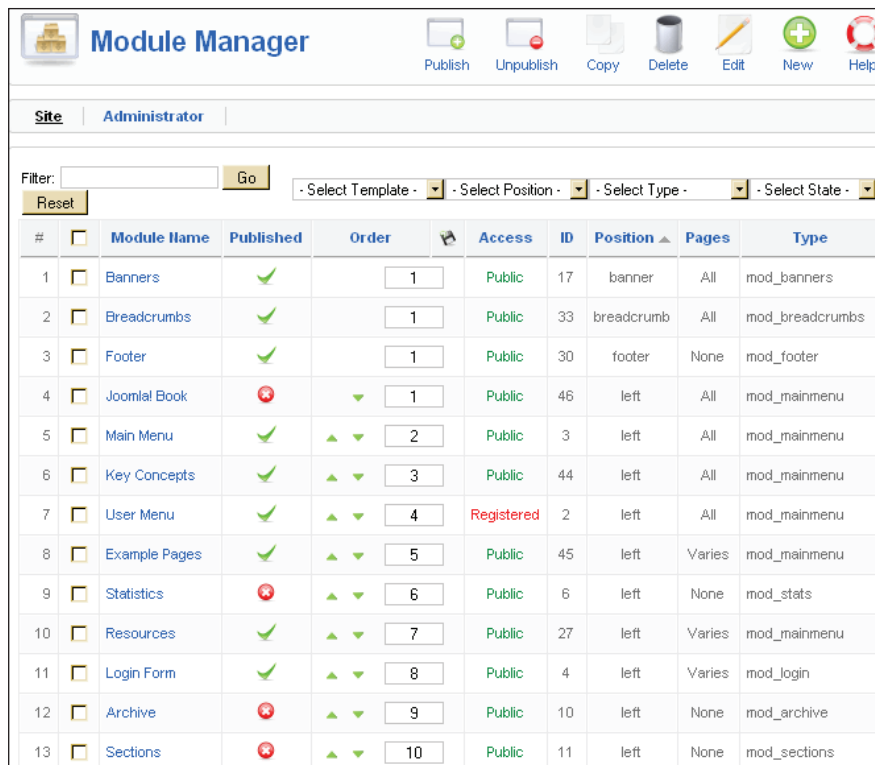
The structure of a module is a lot simpler than that of a component. It is a code fragment that is integrated and interpreted by another part of the program. Due to the scripting language PHP's capabilities, modules can collect data from all kinds of sources. The sources could be your own website (the last five articles) or a weather, stock quotation, Amazon, or ebay web service.

A module contains business logic and a user interface. It does not have its own administration section as most components do. Modules can, however, be controlled by parameters. Your website's template addresses the various modules directly and integrates them.

Since modules are self-contained programs, they can handle a dedicated task in this template domain, such as displaying a banner. A template does nothing but to group several different modules in a visually appealing manner. The module structure gives you the advantage of being able to extend your website in a simple way.

Site Modules

Since you can use templates for both your website and for the Joomla! administration, various modules are also available for these templates. You administer modules in the central module section by means of the **Extensions | Module Manager** menu:



The screenshot shows the Joomla! Module Manager interface. At the top, there are icons for Publish, Unpublish, Copy, Delete, Edit, New, and Help. Below that, the site name 'Administrator' is displayed. A filter section includes a 'Filter:' input field, a 'Go' button, and dropdown menus for 'Select Template', 'Select Position', 'Select Type', and 'Select State'. A 'Reset' button is also present. The main content is a table with the following columns: #, Module Name, Published, Order, Access, ID, Position, Pages, and Type. The table lists 13 modules with their respective status and configuration options.

#	Module Name	Published	Order	Access	ID	Position	Pages	Type
1	Banners	✓	1	Public	17	banner	All	mod_banners
2	Breadcrumbs	✓	1	Public	33	breadcrumb	All	mod_breadcrumbs
3	Footer	✓	1	Public	30	footer	None	mod_footer
4	Joomla! Book	✗	1	Public	46	left	All	mod_mainmenu
5	Main Menu	✓	2	Public	3	left	All	mod_mainmenu
6	Key Concepts	✓	3	Public	44	left	All	mod_mainmenu
7	User Menu	✓	4	Registered	2	left	All	mod_mainmenu
8	Example Pages	✓	5	Public	45	left	Varies	mod_mainmenu
9	Statistics	✗	6	Public	6	left	None	mod_stats
10	Resources	✓	7	Public	27	left	Varies	mod_mainmenu
11	Login Form	✓	8	Public	4	left	Varies	mod_login
12	Archive	✗	9	Public	10	left	None	mod_archive
13	Sections	✗	10	Public	11	left	None	mod_sections

Figure 5.23: Module Manager

The different sections that you see here are:

Module Name: Name of the module and header on the website.

Published: Displays whether the module is published or not.

Order: You can change the order with the green arrows next to the input box. That way you can control whether the **Joomla! Book** menu is above or below the **Main Menu**. Sorting directly by entering the position in the input box can be done by clicking once on the **Order** icon; this circumvents the tedious clicking on the green arrows.

Access: Access rights for this module (**Public**, **Registered**, or **Special**).

Position: Position tells the template where you want the module to be displayed. There are default positions defined in the template, such as: `banner` (advertising section), `left` (left side), `right` (right side), `top` (top), `user1` (user defined 1), `user2` (user defined 2), `user3` (user defined 3), and `user4` (user defined 4).

With the help of the option list on the top, you can filter the display according to these positions for a better overview.

ID: The dataset number from the database.

Pages: The module can be displayed on all pages or only particular ones.

Type: There are various types of modules. The `mod_mainmenu` type, for instance, occurs frequently since every menu belongs to that type. The individual menus are only differentiated by their parameters. With the help of the option list at the top, you can filter the display according to these types for a better overview.

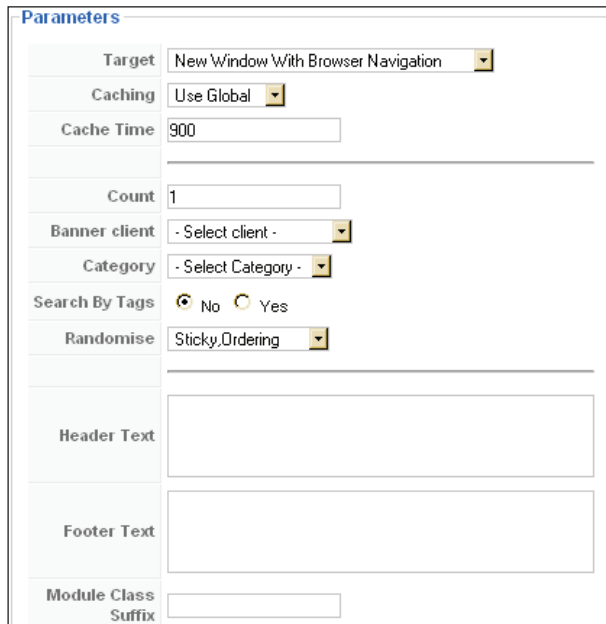
The template has numerous filter options with which you can individualize the display. These filter mechanisms are very useful and Joomla! already has more than 20 of them as standard equipment.

The handling of these modules is pretty uniform. Besides entering the name, access information, etc., you also have to decide on which pages the module is to be displayed and at what position.

The parameter list is especially important with modules. We will emphasize on each of them as they come into use. You can copy modules at will and display them on your website with modified parameters and titles and in different positions.

Banner

This module controls the display of the banners. Besides the settings for details and page allocation that you already know, you can configure the following parameters:



The screenshot shows the 'Parameters' configuration form for a Joomla! Banner module. The form is organized into several sections separated by horizontal lines. The parameters and their values are as follows:

- Target:** New Window With Browser Navigation (dropdown menu)
- Caching:** Use Global (dropdown menu)
- Cache Time:** 900 (text input)
- Count:** 1 (text input)
- Banner client:** - Select client - (dropdown menu)
- Category:** - Select Category - (dropdown menu)
- Search By Tags:** No Yes (radio buttons)
- Randomise:** Sticky,Ordering (dropdown menu)
- Header Text:** (empty text area)
- Footer Text:** (empty text area)
- Module Class Suffix:** (empty text input)

Figure 5.24: Banner Parameters

You can enter the dataset ID of the client (**Banner client**) as a parameter and thereby ensure that only banners from that client are displayed.

Target: Where should the target URL be displayed – the same window or a new window, with or without navigation?

Caching: Individual caching.

Cache Time: The time-period that the data should remain in the cache.

Count: Number of impressions.

Banner Client: If this module is to display banners of only one client, you select the client here.

Category: If only banners of one particular category are to be displayed (for example only text links), select the appropriate category here.

Search By Tags: Tags can be assigned to individual banners. You can decide here whether you only want to display banners with, for instance, the tag Joomla!.

Randomize: You can control the rotation of the banners with this. **Sticky, Ordering** means that those banners that are marked sticky are displayed first and then the others in the order of the sorting as is defined in the banner. **Sticky, Randomize** means that those banners that are marked sticky are displayed first and then the others in randomized order.

Header Text: Text before the banner.

Footer Text: Text after the banner.

Module Class Suffix: You can enter a suffix for the CSS class with this. For instance, if you just enter `table` in the field, the class `module_table` is then activated. You then have to implement your class in the respective CSS file of the template.

Breadcrumbs

Breadcrumbs provide navigational information in articles. The superordinate concepts of the article, the **News** section and the **Latest** category are linked and thereby give the visitor orientation and navigation options. You can define whether **Home** should be displayed in the parameters by setting **Show Home** to **Yes** or **No**. If you deactivate this module, the breadcrumbs are no longer displayed.



Figure 5.25: Breadcrumbs

Footer

This module is supposed to display the Joomla! copyright information in the footer. You can deactivate it and suppress this information on the front page.

Main Menu

The `mod_mainmenu` is used for all menu modules. There are vertical (Main Menu) and horizontal (Top Menu) menus. With vertical menus you also have the option of a *flat list*. A flat list is simply a listing of individual items. The parameters in **Main Menu** are:

Caching: You can cache the menu content to save load time.

Menu Class Suffix: You can enter a special CSS class here for the visual configuration of the menu.

Module Class Suffix: You can enter a special CSS class here for the visual configuration of the module (menu content).

Menu Name: Name of the menu

Menu Style: You can specify vertical, horizontal, or flat list.

Start Level: The nesting level at which this menu should start.

End Level: The nesting level that this menu should include for its last items.

Always show submenu items: You can have the menu items always be open, even if another item is clicked. This function only makes sense with nested menu structures:



Figure 5.26: Menu Nesting

Full Active highlighting: This parameter refers to parental elements and whether to mark them active. Normally this is only done to the individual link. Careful: Marking more than one link active is contradictory to the HTML standard.

Show Menu Icons: You can display icons for the menu items; display of the icons is in each case dependent on the active template.

Menu Icon Alignment: You can have the menu icons on the left or on the right.

Expand Menu: This makes the submenu items visible.

Activate Parent: This will activate your parent menus.

Indent Image: You can specify the item to be displayed with the substructures of a menu item. You have the option to use Joomla! default values, to supply a separate image for each hierarchical level, or to display no icons at all.

Indent Image (1-6): You can define six icons for six hierarchical steps.

Spacer: Here you define the spacer that should be displayed between the menu items of a horizontal menu.

End Spacer: You can display an end character at the end of the items of a horizontal menu. If this is what you want, you can define it here.

Target Position: You can position the pop-up window by defining the value (JavaScript) here.

Login Form

There are two available views for the login module. If the visitor is not logged in, he or she gets a login form where he or she can enter his or her username and password. Depending on the settings in **Site | Configuration | Users** it is possible to register as a new user.

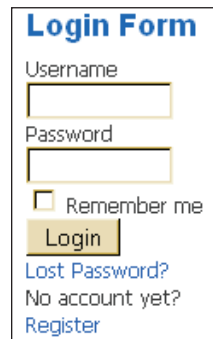


Figure 5.27: Login Module

After a successful login, the display changes to give a logout option:

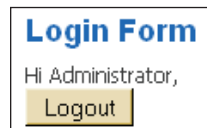


Figure 5.28: Logout Option

The parameters includes the following options:

Caching: You can cache the menu content to save load time.

Module Class Suffix: A special CSS class for the visual configuration of the menu can be entered here.

Pre-text: The text that you enter here is displayed before the form in Login mode.

Post-text: The text that you enter here is displayed after the form in Login mode.

Login Redirection URL: Here you define the URL that the user is sent to after a successful login.

Logout Redirection URL: Here you define the URL that the user is sent to after a successful logout.

Greeting: After a successful login, the look of the module changes and displays a greeting and a **Logout** button. You decide here whether you want "Hello, [User name]" text to be displayed.

Name/Username: Here you decide whether the user should be addressed with his or her real name or the user name.

Statistics

By default, the **Statistics** module is deactivated. If you activate it, you have to define the pages where it will be displayed. This module displays information about your server and your website.

The parameter options other than the usual ones are:

Server Info: You can set this to display server information.

Site Info: You can set this to display website information.

Hit Counter: Here you can activate or deactivate your visitor counter.

Increase Counter: You can set the starting number of the visitor counter with this.

Archive

By default, the **Archive** module is deactivated. If you activate it, you have to define the pages where it will be displayed. This module displays information about the content of your archive. The display of this is grouped by month. With the parameter **Count** you can define the number of the items to be displayed.

Sections

By default, the **Sections** module is deactivated. This module displays the various sections of the website. If you activate it, you have to define the pages where it will be displayed. With the parameter **Count** you can define the number of the sections to be displayed.



Figure 5.29: Sections

Related Items

The **Related Items** module (similar articles) displays the headers of other articles that are related to this article. The relationship is based on the keywords that have been entered in the metadata. All of the keywords of the currently displayed article are compared with those of the rest of the published articles.

If, for instance, you have entered `test` as a keyword in **Joomla! License Guidelines** and also in your newly created news article, the news article and the license conditions are displayed as Related Items when one of them is requested. You can also set a parameter so that the date of creation of the related article will be displayed.

Wrapper

The **Wrapper** module links external, non-Joomla! created contents inside of a so-called `iframe`. An `iframe` is an HTML tag and represents a scrollable area within a website. With this module you can integrate entire websites that are located online on other servers into Joomla! content. Here is an example using Google's PDA-Portal:

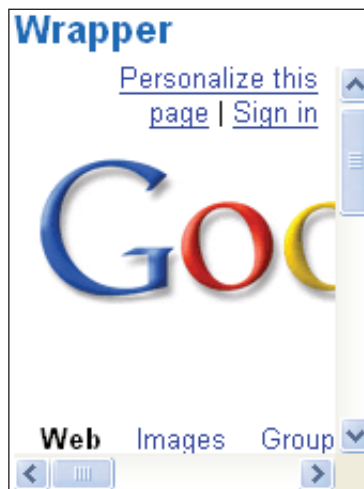


Figure 5.30: Wrapper linked to Google Website

Some of the parameter options are:

Url: Enter the URL of the desired website.

Scrollbars: If you want scrollbars to be displayed in the `iframe`, you have a choice of **Yes**, **No**, and **Auto** for automatic activation.

Width/Height: Width and height of the `iframe` in percentage or pixels.

Auto Height: Set to **Yes** if you want the height to be automatically adjusted.

Auto Add: By default, an `http://` is inserted before the URL if no `http://` or `https://` was found. This switch can be enabled or disabled here.

Feed Display

With this module you can display a news feed on the website.

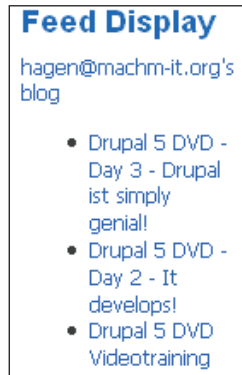


Figure 5.31: Newsfeed as a Module

The parameters involved are:

Feed URL: You can enter the URL of the desired Newsfeeds here.

RTL Feed: You can set the direction of the scroll to go from right to left.

Feed Title: You can set the title of the feed to be displayed here.

Feed Description: You can set this to display the description of the newsfeeds.

Feed Image: You can set this to display the logo of the newsfeeds.

Items: Number of articles that should be displayed.

Item Description: You can have the text of the articles displayed.

Word Count: Here you define how many words of the description of the article are to be displayed.

Joomla! Book

This is the **Menu** module that we created in Chapter 4 in the *Creating a New Menu* section. Do you remember?

Polls

The functionality for the display of polls is globally enabled or disabled with this. The polls themselves are configured in the **Polls** component (see **Components** menu). You can set a parameter as to whether the content of the module should be cached or not.

Who's Online

The **Who's Online** module displays who happens to be on the site at the time. The module also differentiates between guests and registered users. The **Display** parameter helps you make decisions about the display of the module. You can choose from: Number of guests and members, Usernames of the logged users, or both.

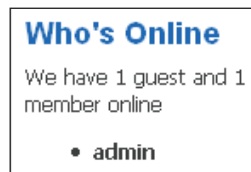


Figure 5.32: Who's Online Module

Random Image

This module will display randomly selected images from a folder of your choice. This module is activated by default but not allocated to a page. You have to allocate pages to it before you can see it on your website.



Figure 5.33: Random Image Module

The parameters include:

Image Type: Here you define the type of the image (`jpg/png/gif`). You can only select one type at a time.

Image Folder: Here you have to enter the folder where the images are located. For this example we have selected `images/stories` without a leading slash.

Link: If you enter a URL here, the image is clickable. The target of the link is the URL that you define here.

Width (px)/Height (px): Width and height of the displayed images in pixels. If you don't define these, the images are displayed as well as possible.

Advertisement

This module is a copy of the banner module. This, by default, has to do with banners that are in the Text Ads category and contain text links.

Syndication

The website's newsfeed is delivered by the **Syndication** module. You can set the standards that you are offering in the parameters. The articles on the front page represent the content of the offered newsfeed:

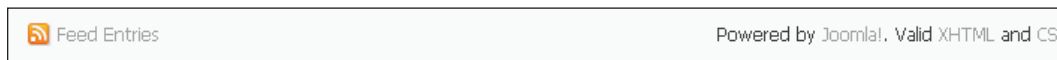


Figure 5.34: Syndication Module

Newsflash

The **Newsflash** module displays randomized intro texts of your articles:



Figure 5.35: Newsflash Module

The parameters are as follows:

Category: By selecting from a list of categories, you can decide whether the content should come from one or all categories.

Layout: You decide between a horizontal and a vertical presentation.

Show Images: Should the images that are contained in the contents be displayed or not?

Linked Titles: You can decide if the title should also be a link to the entire article.

Read More: Activate **Read More** link or not.

Article Title: You can display the title of the article.

No. of Articles: You define how many articles are to be displayed simultaneously.

Latest News

This module displays the latest (newest) articles. By default it is assigned to the `user1` position. You can assign a different position to it, for instance on the right.



Figure 5.36: Latest News Module

The parameters include:

Module Mode: The decision here is whether you want categorized or non-categorized (static) articles or both in the list.

Frontpage Items: When you are in Content Items only mode, you can define whether elements of the front page should be included.

Count: This is the number of the elements to be displayed.

Category ID: If you enter the dataset numbers of the categories to be displayed in a comma separated list, you can force the selection of content to come only from those categories.

Section ID: If you enter the dataset numbers of the sections to be displayed in a comma separated list, you can force the selection of content to come only from those sections.

Popular

This module displays the most popular articles:



Figure 5.37: Popular Module

The parameters are exactly the same as those of the **Latest News** module.

Search

In the default template, the **Search** module is only an input template.

This template does not change all of the parameters. Here the text has been modified from **Search** to **You will find...**

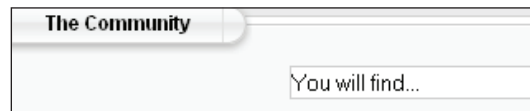


Figure 5.38: Search Module

Parameters in this module include:

Set Itemid: An `Itemid` can be entered that is then passed to the URL that is created by the search field. This is helpful with third-party components.

Box Width: Set the size of the textbox in characters, the example has 30 characters.

Text: Enter the text that is displayed in the search field – like we changed **Search** to **You will find...**

Search Button: Here you can decide whether you want to have a search button or not.

Button Position: If you have selected a search button, you define its position with this (right, left, top, bottom).

Search Button as Image: You can use a graphic as search button.

Button Text: This is the description of the search button.

Copy a Module

Imagine that you would like to display two random images. One module is to display images from folder A and another module should get the images from folder B. In a case like this, select the **Random Images** module by clicking the checkbox in front of the name and then click the **Copy** icon.

It is possible that you will get an error message; nonetheless, you'll notice that the copy of your desired module is created.

A new module with the name of **Copy of Random Image** appears in the list. Change the information the way you want and you have a new module:



Figure 5.39: Copying a Module

If you deactivate the Latest News and Popular modules and switch the two image modules to positions `user1` and `user2`, the new modules are displayed in the content section of the template above the news and/or the frontpage.

Administrator Modules

In the same template, in the **Administrator** tab, you can see a structurally identical section, with modules, however, that are useful for the administration area.

#	Module Name	Published	Order	ID	Position	Pages	Type
1	Logged in Users	✓	0	29	cpanel	None	mod_logged
2	Popular	✓	3	19	cpanel	None	mod_popular
3	Recent added Articles	✓	4	20	cpanel	None	mod_latest
4	Menu Stats	✓	5	21	cpanel	None	mod_stats
5	Cpanel Shell	✓	1	40	cp_shell	All	mod_cpanel
6	Footer	✓	0	31	footer	All	mod_footer
7	Unread Messages	✓	1	22	header	None	mod_unread
8	Online Users	✓	2	23	header	None	mod_online
9	Quick Icons	✓	1	26	icon	None	mod_quickicon
10	Login	✓	1	9	login	None	mod_login

Figure 5.40: Administrator Module

Logged in Users

This module displays a list of the currently logged in users at position cpanel, as a list in the tab in the **Control Panel**.

#	Name	Group	Client	Logout
1	admin	Super Administrator	administrator	

Figure 5.41: Admin Modules in the Control Panel

Popular

The **Popular** module presents a list of the most often visited content as a tab in the **Control Panel**.

Recent added Articles

This module presents a list of the most recently published articles as a tab in the **Control Panel**. Articles that were published on the front page are not included in this list.

Menu Stats

The **Menu Stats** module displays statistics about the layout of the individual menu elements as a tab in the **Control Panel**.

Footer

This module displays the Joomla! copyright information in the back end.

Unread Messages

The **Unread Messages** module informs you about the number of unread administrator messages at the header position, or any other position you wish.

Online Users

The **Online Users** module displays the count of logged-in users at the header position.

Quick Icons

The **Quick Icons** module offers quick access icons in the control panel.

Login

This module displays the login form for the administration area. Do not deactivate it!

Admin Menu

This module displays the JavaScript navigation in the back end. Do not deactivate it!

User Status

The **User Status** module switches the top right information section completely on or off. You shouldn't deactivate this either!

Admin Submenu

The **Admin Submenu** module turns off the section in which the tabs (submenus) are displayed. Do not deactivate it!

Title

The **Title** module determines the display of the description and the icon on the left of the toolbar. Do not deactivate it!

Toolbar

The **Toolbar** module sets the display of the toolbar. Do not deactivate it!

Cpanel Shell

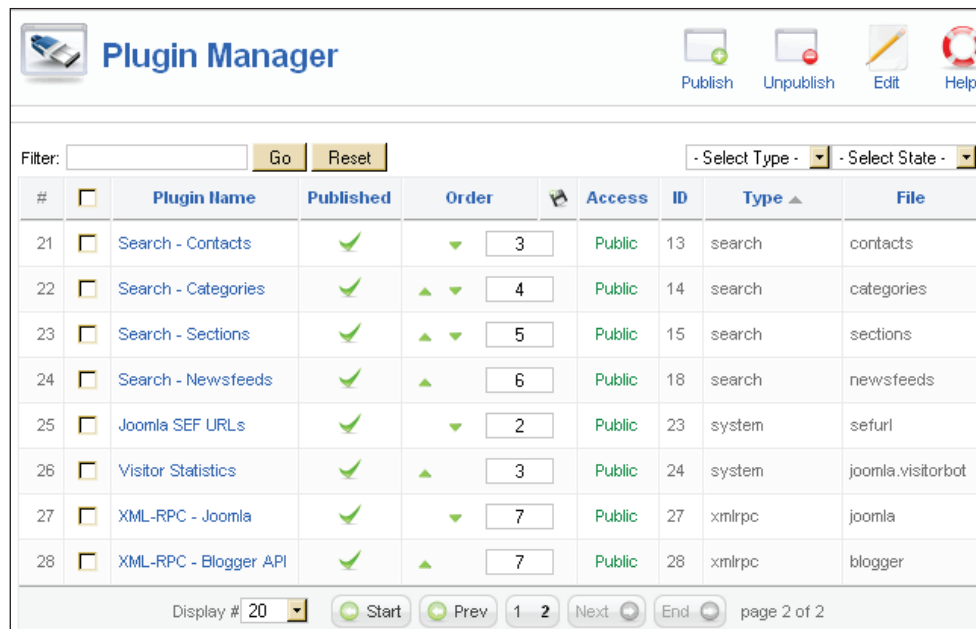
The **Cpanel Shell** module is responsible for the display of the modules using the `cpanel` location on the Administrator control panel.

Plug-In Manager

Plug-ins can absolutely be compared with a Joomla! scripting language. You have already run across an integrated plug-in when you worked with the TinyMCE-Editor.

Plug-ins always belong to a particular type. At the moment, there are plug-ins from the `authentication`, `content`, `editors`, `editors-xtd`, `search`, `system`, and `xmlrpc` sections.

You can activate and deactivate the individual functions in the **Plugin Manager** section. Plug-ins have few or even no changeable settings since they are mostly programmed and optimized for a very specific purpose.



#	<input type="checkbox"/>	Plugin Name	Published	Order	Access	ID	Type	File
21	<input type="checkbox"/>	Search - Contacts	✓	3	Public	13	search	contacts
22	<input type="checkbox"/>	Search - Categories	✓	4	Public	14	search	categories
23	<input type="checkbox"/>	Search - Sections	✓	5	Public	15	search	sections
24	<input type="checkbox"/>	Search - Newsfeeds	✓	6	Public	18	search	newsfeeds
25	<input type="checkbox"/>	Joomla SEF URLs	✓	2	Public	23	system	sefurl
26	<input type="checkbox"/>	Visitor Statistics	✓	3	Public	24	system	joomla.visitorbot
27	<input type="checkbox"/>	XML-RPC - Joomla	✓	7	Public	27	xmlrpc	joomla
28	<input type="checkbox"/>	XML-RPC - Blogger API	✓	7	Public	28	xmlrpc	blogger

Figure 5.42: Plug-Ins

Authentication Plug-Ins

In Joomla! 1.5, users can authenticate themselves in a number of different ways. This has the advantage that you can do without the cumbersome process of registration on the website and that you don't have to remember another username and password. In order that these various methods of authentication can be used, you have to publish the plug-in.

Gmail

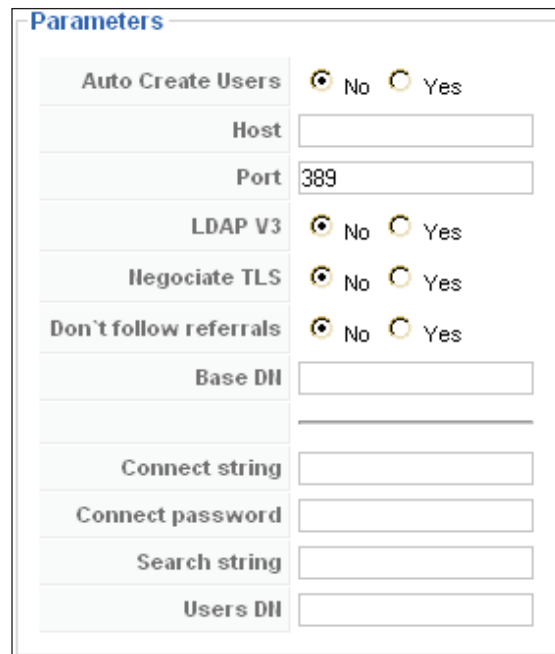
Google's GMail Service comes with the option of authentication via email address.

Joomla!

The "normal" authentication subsequent to a prior registration on the website.

LDAP

LDAP is a network protocol that is used with so-called directory services. It transmits the communication between the LDAP client (in our case the Joomla! website) and the directory from which person-related data is read. LDAP directories (<http://en.wikipedia.org/wiki/LDAP>) are common in companies and therefore it stands to reason that one could access this type of person-related data over the company's intranet.



Parameters	
Auto Create Users	<input checked="" type="radio"/> No <input type="radio"/> Yes
Host	<input type="text"/>
Port	<input type="text" value="389"/>
LDAP V3	<input checked="" type="radio"/> No <input type="radio"/> Yes
Negotiate TLS	<input checked="" type="radio"/> No <input type="radio"/> Yes
Don't follow referrals	<input checked="" type="radio"/> No <input type="radio"/> Yes
Base DN	<input type="text"/>
Connect string	<input type="text"/>
Connect password	<input type="text"/>
Search string	<input type="text"/>
Users DN	<input type="text"/>

Figure 5.43: LDAP Parameters

Some of the options in the **Parameters** section include:

Host: Host name of the directory service.

Port: Port number.

LDAP V3: You can use the latest version of OpenLDAP (version 3); by default version 2 is used.

Base DN: The Base DN defines where in the directory tree the downward search for particular objects should be started.

Connect string: You pass the data that the user enters in Joomla! on to the LDAP Server.

Connect password: Password of the directory manager.

Search string: Here you enter what is being searched for, for example an MD5-coded password (`md5{password}`).

Users DN: The primary key of the directory manager.

Content Plug-Ins

Here we will discuss the plug-ins available for content.

Image

This plug-in has been kept for reasons of compatibility with Joomla! 1.0.x. It interprets the `{mosimage}` command in the text of an article and then displays the assigned image. You can define the margin and border spaces of these images in the parameters.

Code Highlighter (Joomla!)

Content elements that contain source code can be formatted with the **Code Highlighter** plug-in.

Use of the Code Highlighter plug-in:

```
<code>{code}
  if ($number > 0){
    echo $number;
  } else{
    $number++;
  }
{/code}</code>
```

Page Navigation

This plug-in integrates the **Next** and **Previous** functions under the articles. If you want to use it, it has to be activated.

SEF

SEF is the acronym for **Search Engine Friendly**. This plug-in creates search-engine friendly URLs for content elements. If you wish to use that feature, this plug-in has to be activated.

Rating

This is the plug-in that displays the rating bar above the contents. If you want to use it, it has to be activated.

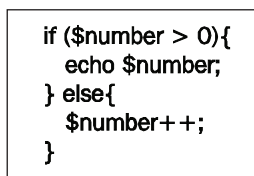
Email Cloaking

This plug-in changes an email address that you enter into a content element in the form of `hagen@example.com` into a link and cloaks the email address using JavaScript. This has the advantage that email collection programs cannot read your email address very easily.

Code Highlighter (GeSHi)

The GeSHi MAMBOT formats just like the Code Highlighter plug-in. GeSHi, however, can perform syntax highlighting and creates an impressive listing on the website if you embed the code to be formatted within `<pre>` and `</pre>` HTML tags:

```
<pre>
  if ($number > 0){
    echo $number;
  } else{
    $number++;
  }
</pre>
```



```
if ($number > 0){
  echo $number;
} else{
  $number++;
}
```

Figure 5.44: Source Code formatted with GeSHi

Load Module

The **Load Module** plug-in makes it possible to load modules inside of articles. It can be called up, for instance, with `{loadposition user1}`.

Pagebreak

The **Pagebreak** plug-in takes care of Joomla! 1.0.x-pagebreaks in articles. Just like the image plug-in, it is simply integrated in the content. Besides a simple word-wrap, various headers and page titles can also be defined. You have used page break in Chapter 4 in the *Creating a New Article* section.

Syntax:

```
{pagebreak}
{pagebreak title=Page title}
{pagebreak heading=First page}
{pagebreak title=Page title titleheading=First page}
{pagebreak heading=First page title=page title}
```

Editor Plug-Ins

There are three editors available for Joomla!

No Editor

This plug-in has to be activated if you want to offer `textarea` fields without an editor.

TinyMCE 2.0

This plug-in has to be activated if you want to offer `textarea` fields with the TinyMCE 2.0 editor.

XStandard Lite 1.7

This plug-in has to be activated if you want to offer `textarea` fields with the XStandard Lite 1.7 Editor.

Editors-Xtd Plug-Ins

The editors-xttd type contains the **Editor Button-Image**, **Pagebreak**, and **Readmore** plug-ins. These three plug-ins generate the three buttons below the editor window. Clicking on these buttons launches the **Insert Image** dialogue, the **Pagebreak** dialogue, and the **Readmore** dialogue.

Search Plug-Ins

The Search plug-ins for **Contents**, **Weblinks**, **Contacts**, **Categories**, **Sections**, and **Newsfeeds** can be activated as you wish. They control the search function of the **Search** module. They have to be activated if you want to get search results from the respective sections. If you want to search additional components, the respective plug-ins for them have to be available.

System Plug-Ins

The system type has **Joomla SEF URLs** and **Visitor Statistics** plug-ins

Joomla SEF URLs

Like the **SEF** plug-in, this too creates the search engine friendly URLs for content elements. If you want to use this feature, this plug-in has to be activated. What is interesting is that this plug-in also makes URLs available even if the `mod_rewrite` Apache web server module is not installed.

Visitor Statistics

This plug-in provides statistical functions (browser, operating system, etc.).

XML-RPC Plug-Ins

The XML-RPC interface gives you the ability to operate Joomla! remotely.

XML-RPC

This plug-in enables the use of content from other platforms, for example Flickr photos from the popular Blogger gateway. Joomla is another gateway that is being offered here. In the plug-in's parameters you can select the section and category into which these articles are allocated.

Template Manager

Now that everything looks a bit more familiar, you may want your site to have a completely different design (you will learn more about templates in Chapter 6). The following screenshot appears when you click on **Extensions | Template Manager**:

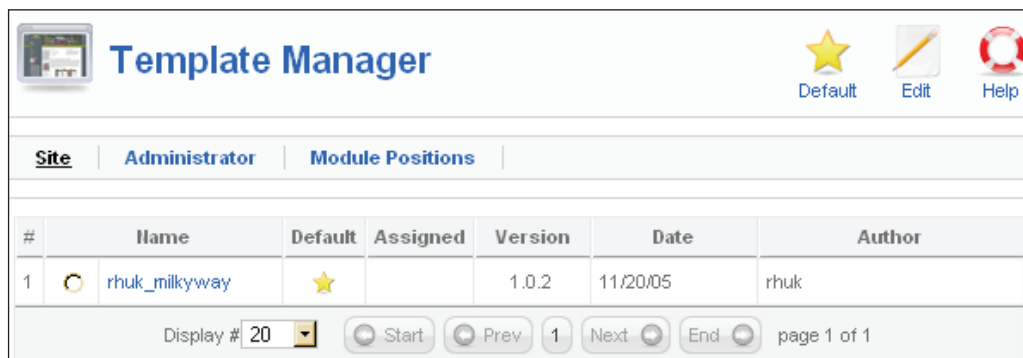


Figure 5.45: Site Template Manager



On clicking **Extensions | Template Manager**, you might come across an error in your administrator\components\com_templates\controller.php file on line 36. The code reads:

```
$limit = $mainframe->getUserStateFromRequest
("$option.$client.limit", 'limit',
    $mainframe->getCfg('list_limit'), 0);
```

Replace \$client with client.

Joomla! forums will provide you with help on similar bug-related issues.

Joomla! differentiates between **Site** and **Administrator** templates. But before we explore the **Template Manager**, we'll take a look at how to add a new template for your website.

Site Templates

You can see the installed templates here. You can define one template as default and assign other templates for particular menu links. To make a template the default one, mark the checkbox next to the template and click on the **Default** icon. Your website will then change its appearance respectively.

Module Position

You can see all of the available module positions for your Joomla! website here. You can display the module contents at these positions in the template. You can extend this list with individual positions for self-made templates. Normally, however, the positions being offered are sufficient.

Administrator Templates

Whatever is valid for your website is, of course, also valid for your administration interface. You can assign other templates to your administration interface just as you were able to assign site templates.

Adding a New Template for Your Website

At the moment, Joomla! 1.5.0 comes with only one template. In order to have a choice, simply install a new template. You will find the Bertrand template with this book; you'll need to download it from <http://www.packtpub.com/support>.

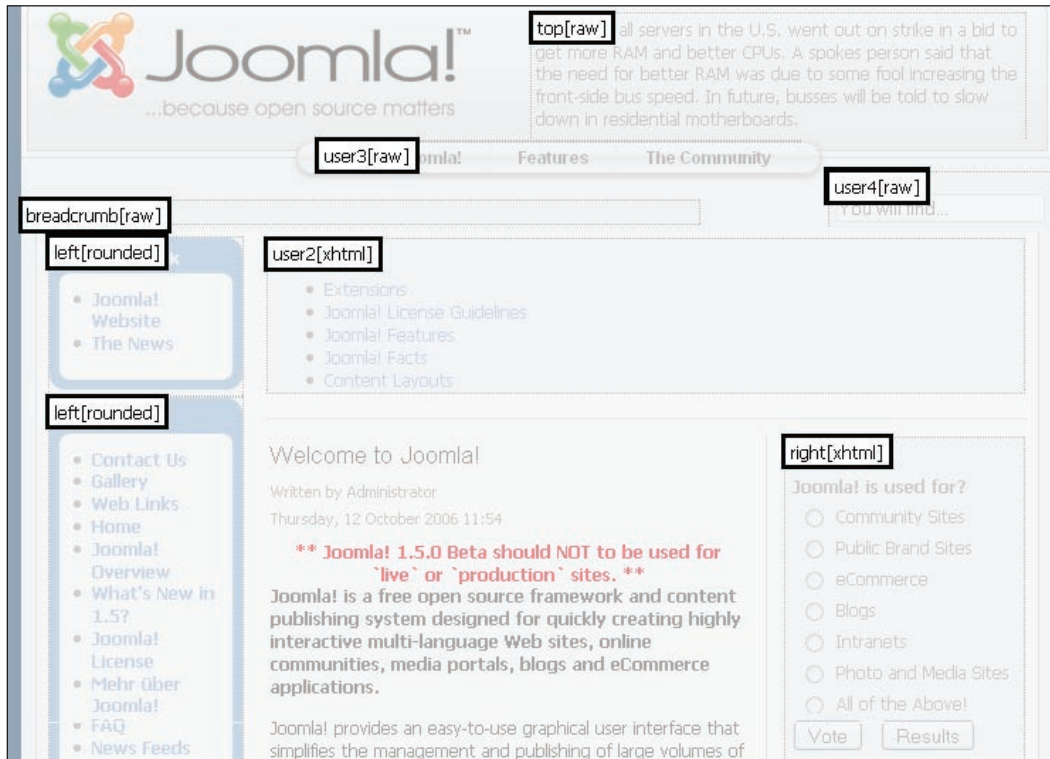
Go to the Joomla! installer from the **Extensions | Install/Uninstall** menu. Click on the **Search** button and upload the `tmpl_bertrand.zip` file from your PC. Click on the **Upload & Install** file button. The new template will be installed and the installer reports the result.

Use **Extensions | Template Manager** to get to the template area. The template that is currently active is marked with a yellow star. If you slide your mouse cursor over the link with the name of the template, a small thumbnail view appears:



Figure 5.46: Template Manager

To assign this template to your site, select the radio button before the name of the desired template and click **Default** in the menu bar. The yellow star is now beside the selected template. Switch to your website and click the **Update** button in the browser. Now you already have a different layout and a completely new appearance. The creation of your own templates is covered in Chapter 6 and that of Mr.Bertrand's website is covered in Chapter 8. (The following screenshot has been split for a better view.)



Components and Extensions



Figure 5.47: Default Template Preview

After this first round of satisfying our urge to play, we now get to some more elaborate customization.

Modifying a Template

After you check a template and click on the **Edit** icon, you can modify and configure all of the properties of an individual template. The parameter list will, among other things, show you menu links that have been allocated to this template:

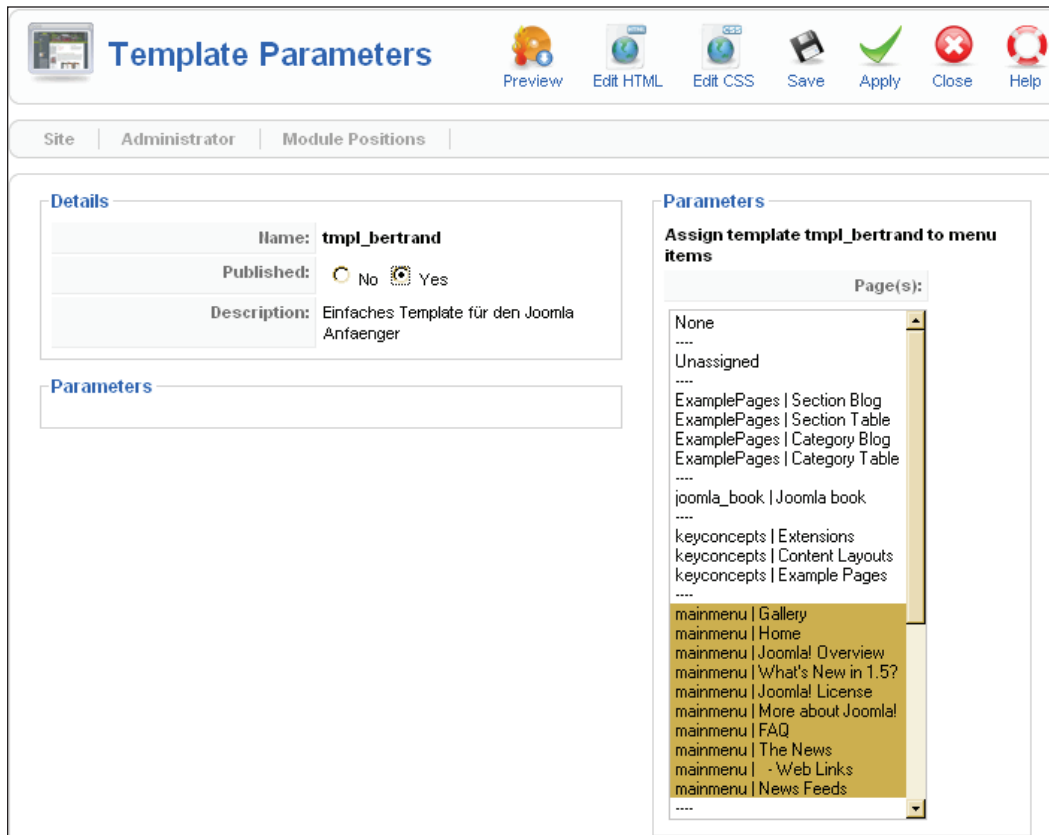



Figure 5.48: Edit Templates

Mark the menu link that you want to allocate to the template. You can make multiple selections by clicking the desired elements one after the other while holding down the *Ctrl* key.

You can now edit the HTML source code of the selected template. Templates always consist of an HTML file. There is a box with template snippets in the top section. These *Template Snippets* are predefined commands for certain actions in the template. We will get back to this in Chapter 6.

Components and Extensions

 You have to know what you are doing if you try your hand at this. HTML and CSS knowledge is essential here. It is, nonetheless, interesting for the beginner to see how a template is constructed.

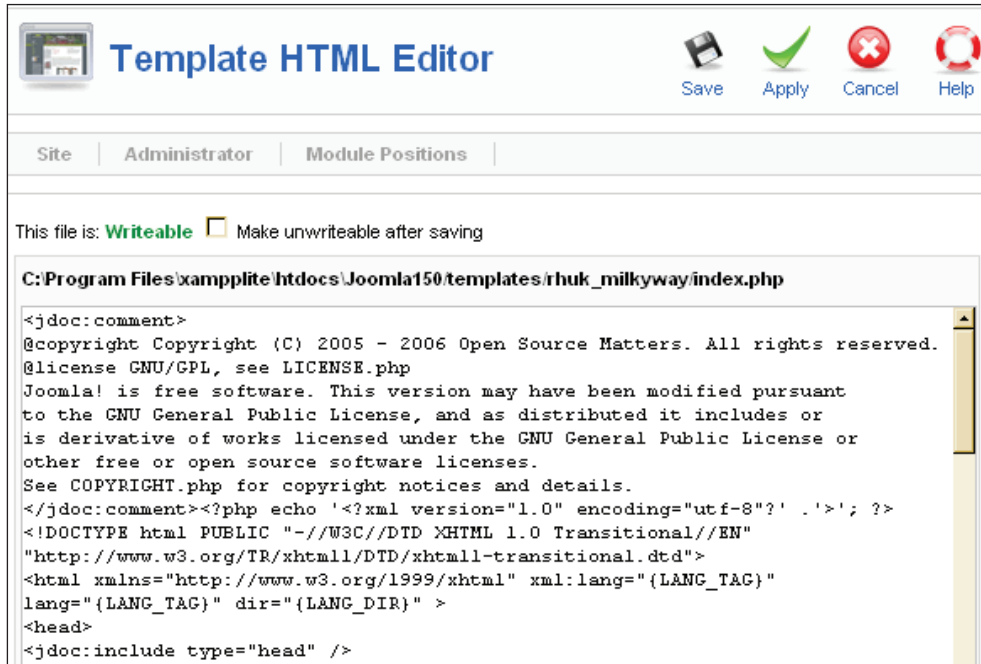


Figure 5.49: HTML Editor

You can also edit the CSS source code of the selected template. Templates can have any number of CSS files. This template, for instance, has four CSS files:



Figure 5.50: CSS Files of a Templates

You have to select the desired file and click on the **Edit** icon again.

Language Manager

Here you have a **Site** and an **Administrator** tab with which you can select the languages for your website and for the back-end section:

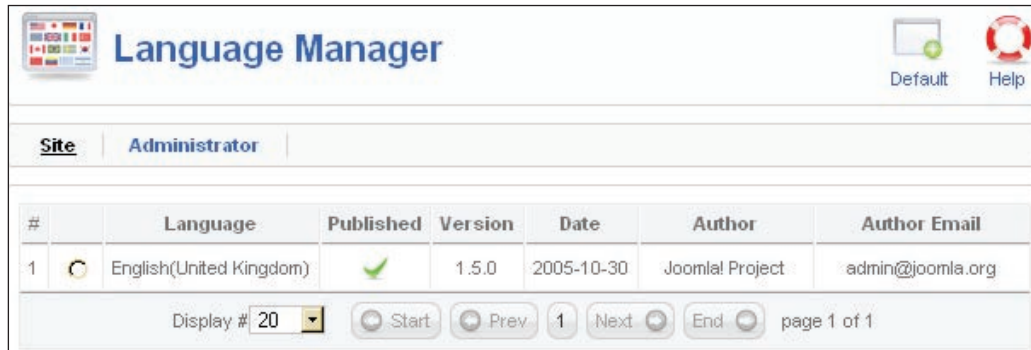


Figure 5.51: Language Manager

By default there is only one language – here English. In the following section we will see how to add another language – here German.

A Different Language for the Website and the Administrator

If you want to reach a linguistic target group with your site, you should understand the language of that target group and build the site in that language. If you want to address an international audience, you should prefer English as the default language. Regardless of which language you choose, you need a customized language file.

Installation of a Different Language File

If you want to reach a German target group, for example, save the language file for the website (Joomla! 1.5 frontend language pack de-DE.zip) and the language file for the administration area (Joomla! 1.5 admin language pack de-DE.zip) to your PC (you will find the download options in the Appendix). These language files were created by the GTT, German Translation Team, under the project direction of Antonio Cambule (http://www.joomla.de/willkommen_beim_deutschen_ubersetzungsteam.html).

Components and Extensions

To install the files click on **Extensions | Install/Uninstall**.

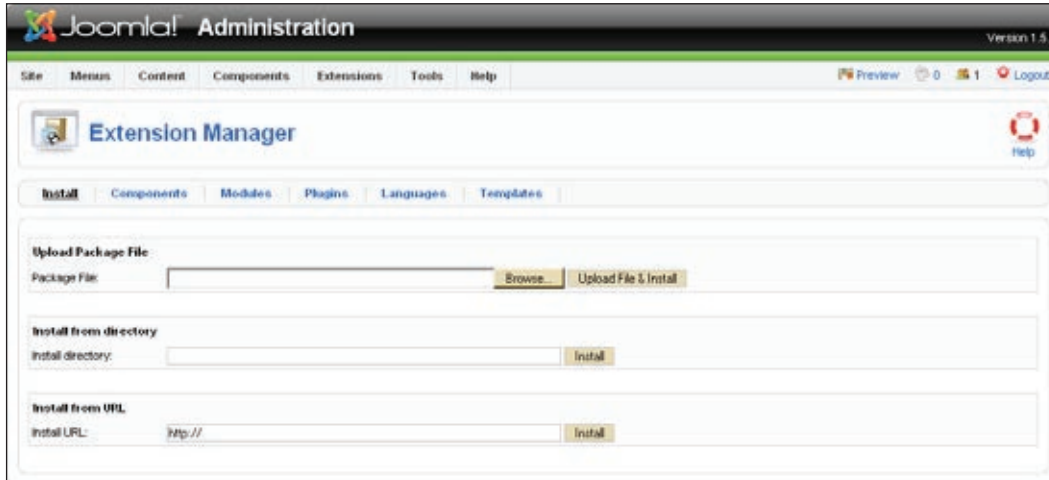


Figure 5.52: Installing the Language File

For our purpose, choose the first option. Click on **Browse...** next to the **Package File** field and first select the Joomla! 1.5 frontend language pack de-DE.zip file. Now click on **Upload File & Install**. If all rights are correctly set, you will receive the message: **Install Language:Success**.

Repeat this procedure with the other file.

Now go to the **Language Manager** by clicking on **Extension | Language Manager**. You will see the available languages here, separated into the **Site** and **Administrator** tabs. Besides **English (United Kingdom)**, now there is also **German (Germany)** and information about the creator of the language file.

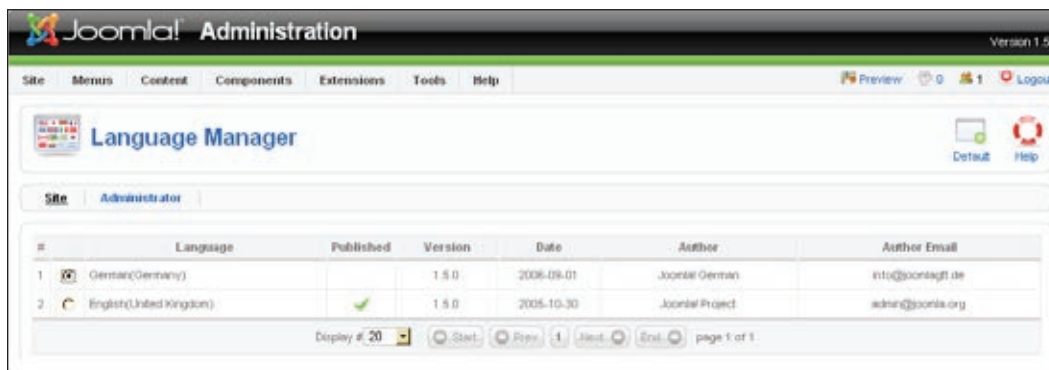


Figure 5.53: Selecting a Language (Language Manager)

The green checkmark is for the standard language; however, it is still on **English**. Select the **German** option with the radio button and then click on the **Default** icon, which is on the top right.

Don't forget to switch the administration interface to **German** as well. This setting will be active next time you log in. So log out, log in again and Joomla! will converse with you in German in the back end.

Translation of a Menu Entry

Your website now speaks German as well. Go to <http://localhost/joomla150/> and take a good look at it. This page displays both German and English languages. Everything that is programmed to function automatically, like the search procedure or the login procedure, is in German; everything else is in English.

Why?

The answer is quite simple. Only the words and sentences that are in code can be translated. A big part of the page, however, consists of user-entered content. This content appears in the language in which it was entered. In our case, the sample data was provided in English.



Joomla! version 1.0.x had an option to change the contents of the language package by means of the language editor. This language editor is gone from the default version of Joomla! 1.5. It is, however, being offered as an installable add-on.

So what do you have to do to change, for example, the menu entry **More about Joomla!** to **Mehr über Joomla!** or to rename the **Main Menu** to **Hauptmenü**?

Components and Extensions

For **More about Joomla!**, click on **Menüs | Main Menu** in the **Administration**. You are now in the Menus area:

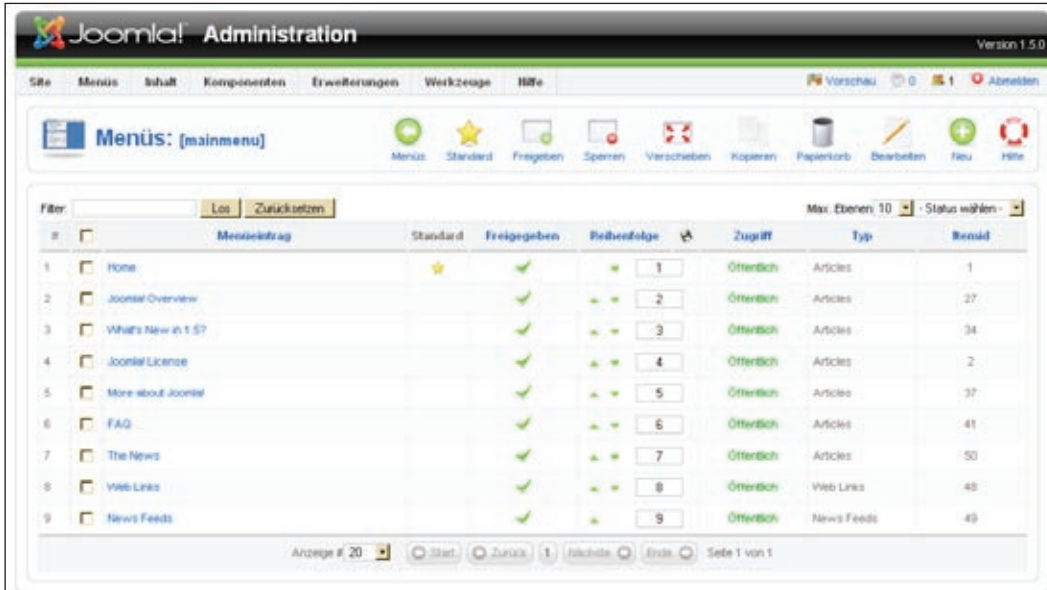


Figure 5.54: Menu Manager

Click on the **More about Joomla!** link and edit it in the form that appears. At this time, we won't pay any attention to the numerous configuration options being offered; simply replace the text **More about Joomla!** with **Mehr über Joomla!** and click on **Anwenden** (Apply). **Mehr über Joomla!** will now appear in your **Main Menu** on the website!:



Figure 5.55: Translated Main Menu

Modifying the Menu Name

In order to make a **Hauptmenü** from the **Main Menu**, open the **Module** by clicking **Erweiterungen | Module**:

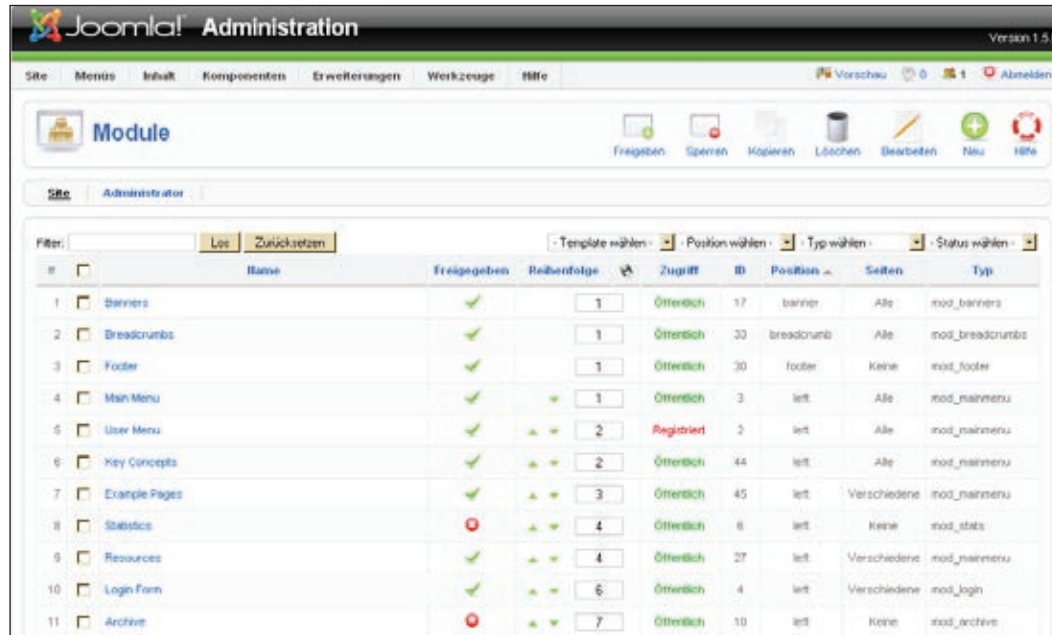


Figure 5.56: Module Area

Click on the **Main Menu** link to get a form, just as we did for the menu entry. Simply change the text, click on **Anwendun** (Apply) and/or **Speichern** (Save) and **Main Menu** becomes **Hauptmenü**.

More on Extensions

It is easy to extend Joomla!'s functionality. Due to the popularity of the Joomla! 1.0.x system, numerous customized extensions are available. The Joomla! Project team has opened its own extensions section on its website, which you can access at the URL <http://extensions.joomla.org/>. Every visitor can download extensions from there, and every registered user can submit his or her extensions to that section.

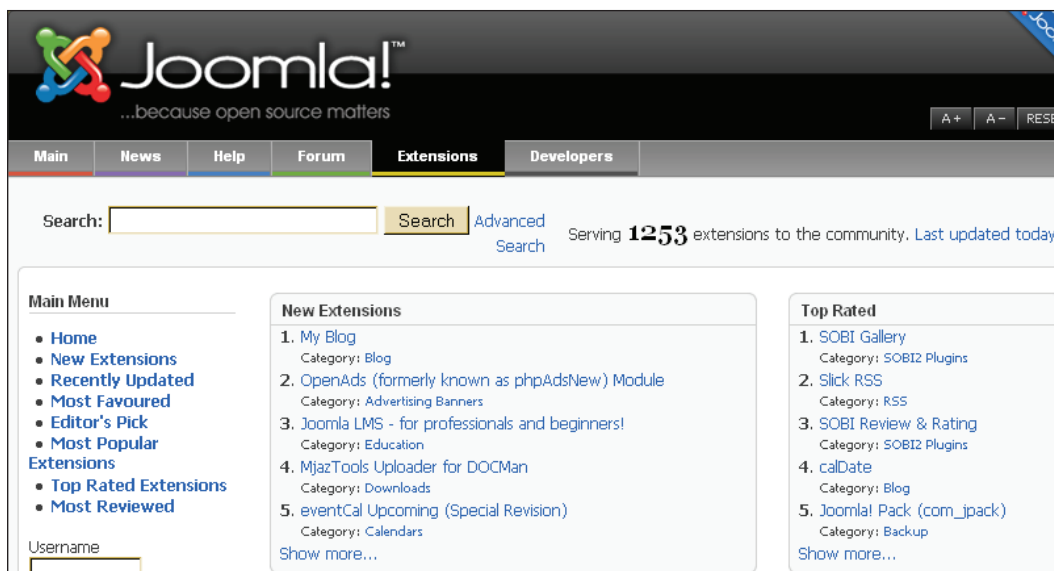


Figure 5.57: Joomla! Extension Section

You will learn how to create your own extensions in Chapter 7. Once you have developed a functional extension, why not offer it on this site? Other users will download it, evaluate it, and if what you have developed is also useful to others, you will soon get some feedback and perhaps even offers of help for your continued development.

It is very easy to install extensions due to Joomla!'s modular structure and the easy-to-use installer. As of January 2007, a total of 1253 extensions were being offered, 48 of which were components that are said to be also working with Joomla! 1.5. Unfortunately some of the most popular components are not yet available for Joomla! 1.5 at the time of printing. All of these popular extensions, however, are being upgraded for Joomla! 1.5 and will hopefully be ready soon. In Joomla! 1.5, the operation of these extensions will be identical to the Joomla! 1.0.x versions. The modifications that are being programmed in will mainly show up in the source code and in improved integration.

For you to get an impression of the operation of a component, I will introduce you to **Exposé**, an image gallery that already works with Joomla 1.5.

Exposé Flash Gallery

Image galleries and video uploads are becoming more and more popular in today's new Web 2.0 world. Applications such as Flickr's photo platform (<http://www.flickr.com/>) and Youtube's video platform (<http://www.youtube.com/>) make it easier for an individual to publish multimedia content in today's increasingly broad-banded Internet. The expectations of design are also getting higher and higher. A simple listing of photographs, in the meantime, is gathering a bit of dust.

Exposé is one of the Adobe Flash-Format-based galleries for pictures, graphics, and videos and is visually outstanding. *Exposé* can:

- Create as many picture albums as you want
- Evaluate images during the upload for different image sizes
- Create thumbnails for the albums
- Play videos with the Flash Player in a compressed format
- Add audio commentary to the individual images

A version of *Exposé* that can be installed on your PC allows you to assemble pictures on your own computer. Due to the compact Flash format, the pictures can also be viewed without a broadband connection.

The Album Manager, which administers the data in the administration section, makes a few demands on the PHP interpreter and the GD, DOMXML, and iconv extensions have to be installed. These preconditions were taken care of by the server in our local xampplite environment. It is possible that your web-space provider doesn't offer these PHP features but there is a solution for that as well (see *Album Manager in the Administration Section*). When it comes to the front end, visitors must at least have a Flash Player version 8 installed in their Internet browser.

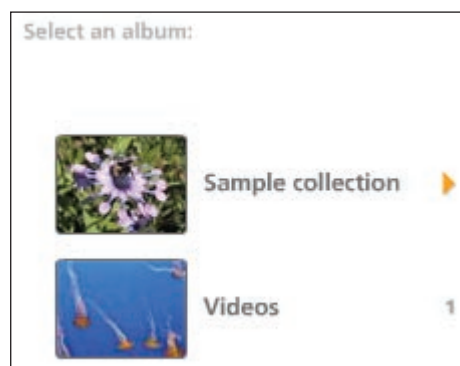


Figure 5.58: Exposé Gallery

Installation

A component is installed in the **Extensions | Install/Uninstall** menu just like language packets, templates, modules, and plug-ins. Every extension package consists of a compressed archive with various files and at least one XML instruction file for the Joomla! installer. The XML file reports what type of extension you are dealing with. Because of this installation method, all types of extensions can be installed from the same installation template. You will create some packages like this in Chapters 6 and 7.

In the case of the Exposé gallery, load the `expose_php5_200608.zip` package from the project's website (<http://www.gotgtek.com>). In the installation template, click on the **Browse** button, select the file, and click on **Upload & Install file**. The Joomla! installer uploads the files, deals with the menu items, and displays a success message as in the following screenshot. Read these messages carefully and remember the password!

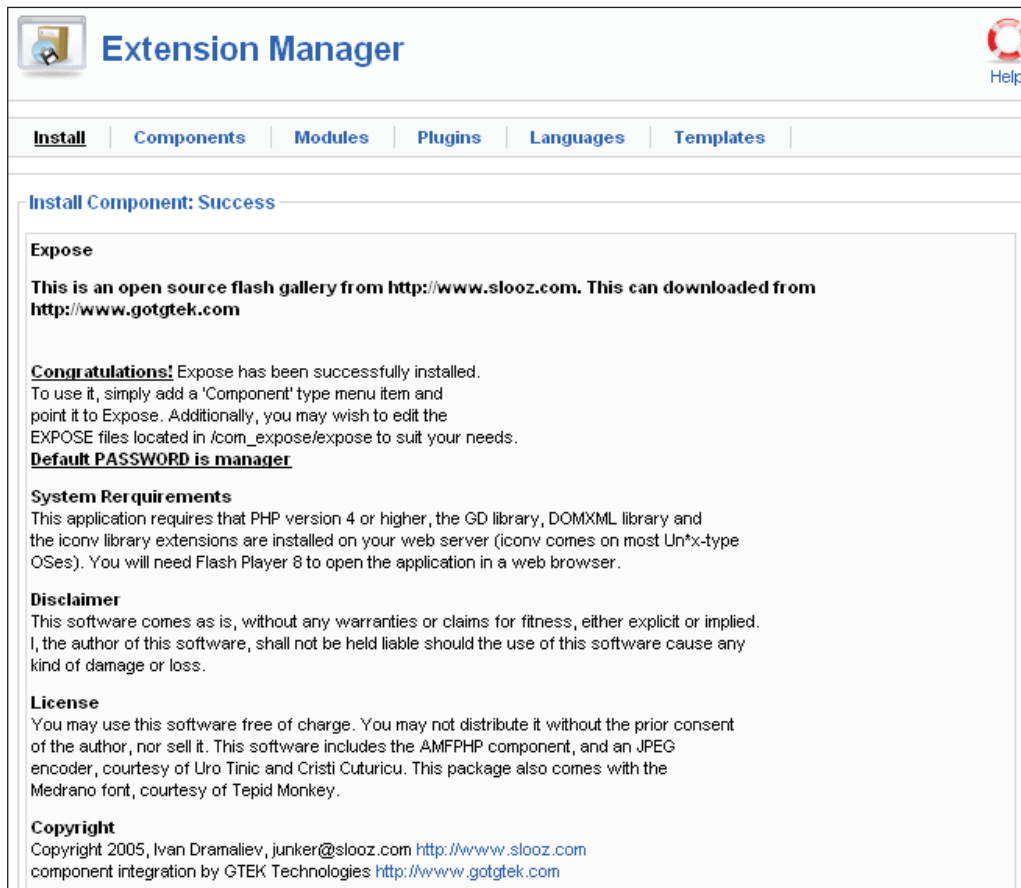


Figure 5.59: Installer Message

Integration into the Website

The component is now installed with sample data and still needs to get a menu link in the front end. Let us put it in the **Main Menu**.

Click on **Menus | Main Menu** and then on the **New** icon. The **Add New Menu Item** is already offering you the new Exposé component:

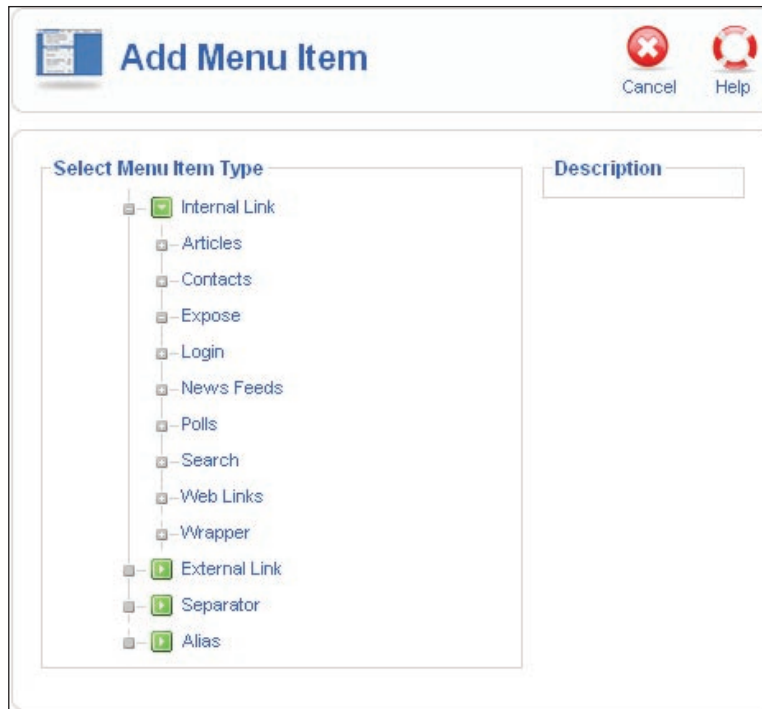


Figure 5.60: Integration into the Main Menu

Clicking on it takes you to the edit template for menu items. You still have to enter a title for the link, set the **Published** radio button to **Yes**, and save your work. (If you come across XML Parsing Error, take a look at this thread in Joomla! forums: <http://forum.joomla.org/index.php?topic=130864.msg646816>.)

Menu Item Type

Expose Change Type

This is an open source flash gallery from <http://www.slooz.com>. This can be downloaded from <http://www.gotgtek.com>

Menu Item Details

Name:	Gallery
Link:	index.php?option=com_expose
Display in:	Main Menu
Parent Item:	<ul style="list-style-type: none"> Top Home Web Links Joomla! Overview What's New in 1.5? Joomla! License More about Joomla! FAQ The News News Feeds
Published:	<input type="radio"/> No <input checked="" type="radio"/> Yes
Ordering:	New items default to the last place. Ordering can be changed after this item is saved.
Access Level:	Public
On Click, Open in:	Parent Window With Browser Navigation

Menu Item Parameters

There are no parameters for this item

Advanced Parameters

Figure 5.61: Integration into the Main Menu II

If you go to your website now, you will see the **Gallery** link in the **Main Menu**. If you click on this link you will see two sample folders. One is called **Sample Collection** and contains additional albums, and the other one is a **Videos** album.

If there is an arrow beside the preview picture, then it contains additional albums, if there is a number there, the number refers to the number of media items contained in it. If you click on **Sample collection**, the albums that lie under it appear:

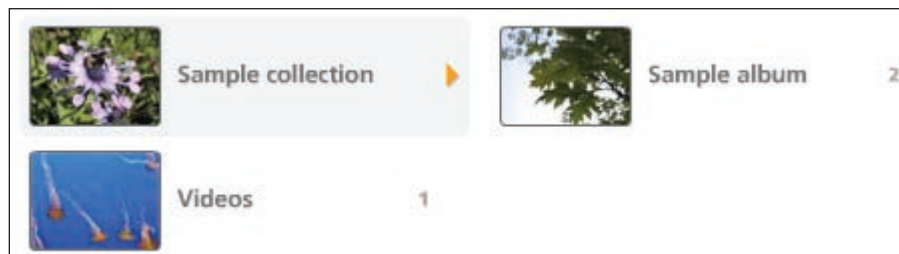


Figure 5.62: Gallery Overview

Components and Extensions

Clicking on **Sample album** takes you to the so-called *Image Strip*:

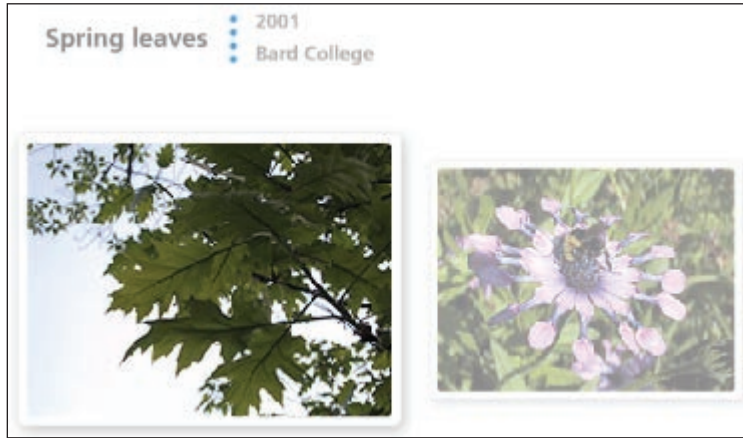


Figure 5.63: Image Strip

This strip can be moved horizontally with the mouse button; the picture in the middle is shown enlarged.

If you click on the picture, you get an individual view:



Figure 5.64: Individual View

At the bottom there are a number of navigation triangles on the right and on the left of the Image Strip:

- (1) This lets you scroll forward and backward through the pictures.
- (2) This lets you request an automatic slideshow.
- (3) This displays the album you are in.
- (4) This takes you back to the Image Strip.
- (5) This takes you back to the albums.
- There is also a question-mark symbol that leads you to the help screen that explains the symbols.

Videos

Go back to the albums and request the video view. You can see an Image Strip here as well with a preview picture of the video. If you click the picture, you get an individual view of the video:

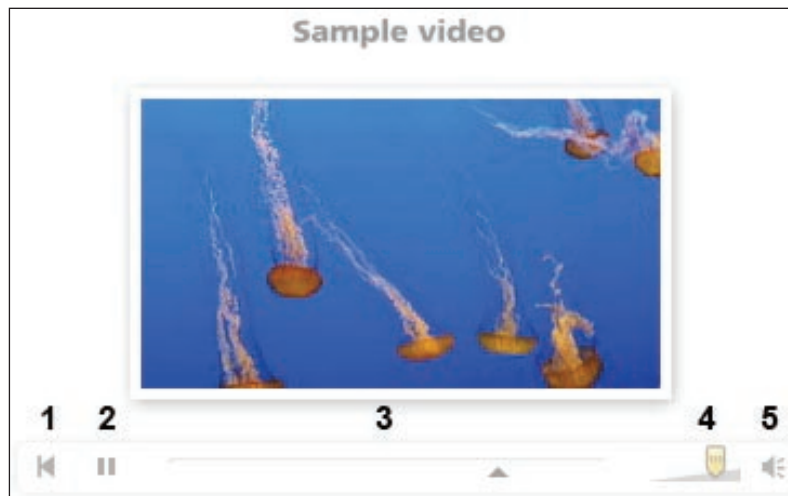


Figure 5.65: Video Player

- (1) This takes you to the beginning of the video.
- (2) This is the Play/Pause button.
- (3) You can use this arrow to move within the video (forward and backward).
- (4) Volume control.
- (5) Clicking the speaker symbol will turn the sound on or off.

Album Manager in the Administration Section

To load your own pictures and videos into the gallery, go to the Administration section and request the **Components | Expose | Manage Albums** menu. You will see a login screen asking you for a password. The default password is **manager**. You can (and you should) change the password by clicking on the **Change Password** button. If you make a mark in the checkbox, the password is stored and you won't have to enter it again.

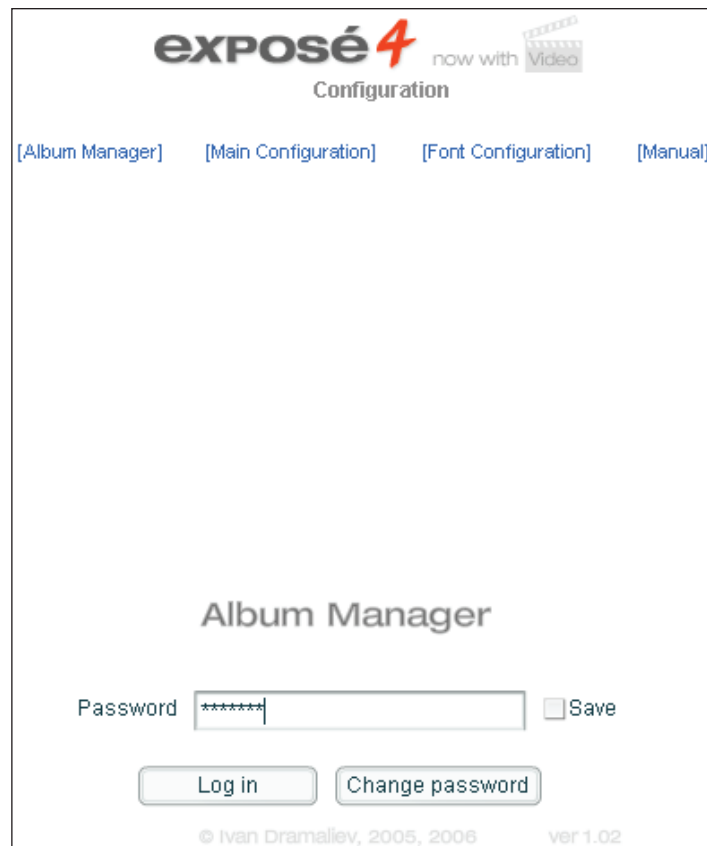


Figure 5.66: Album Manager | Login

The administration section is divided into several areas:

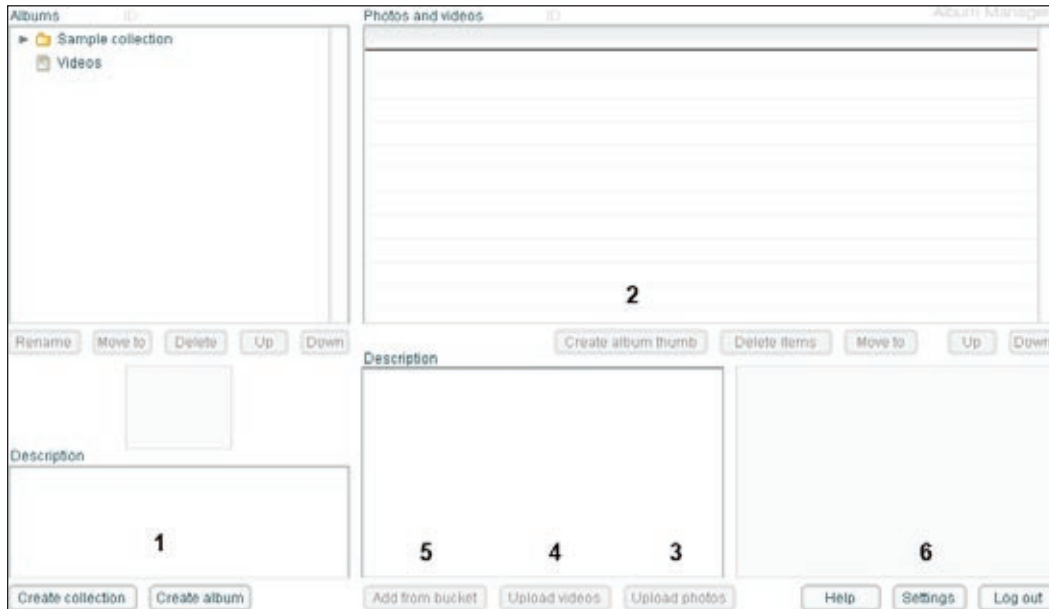


Figure 5.67: Album Manager

- (1) Here you can create a collection and individual albums. In the area above, you can move (**Move to**) albums, rename them (**Rename**) and delete them (**Delete**).
- (2) To create a preview picture for an album, select the picture you want and click on the **Create album thumb** button.
- (3) If you click the **Upload photos** button, you can start a dialogue with which you can upload a picture from the hard drive to the gallery. The picture is automatically compressed, meaning that you can upload the pictures in the size that they are in the camera.
- (4) You can load videos into the gallery in the flv format. To convert your videos to that format, you can use the trial version of the **DREAMWEAVER 8** (<http://www.rivavx.com/>) software or you can encode it with the **Riva FLV Encoder 2** freeware encoder.
- (5) **Add from bucket** is a useful feature. You can use FTP to move files into the [PathtoJoomla!]components/com_expose/expose/manager/bucket directory. You can later add these files with the **Add from bucket** function.
- (6) You can use **Settings** to configure with which parameters the pictures and videos are to be uploaded. You can even integrate a watermark into the pictures.

The following screenshot is of Photo Settings:

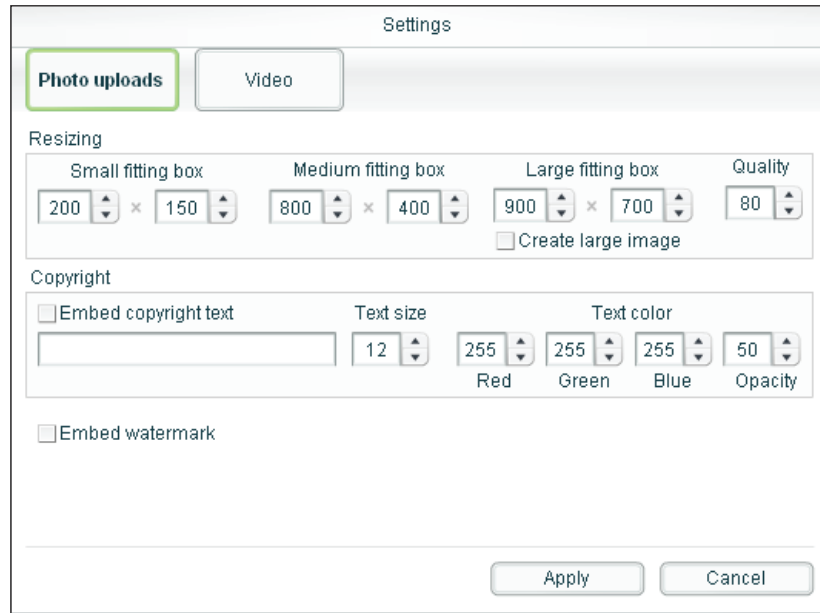


Figure 5.68: Settings

Finally, when you can finish your management of images click on **Log out**.

Uninstallation

If you want to remove this component from your system, go to the **Extensions** | **Install/Uninstall** menu and remove it from the **Components** tab. Select the component and click on the **Uninstall** icon:

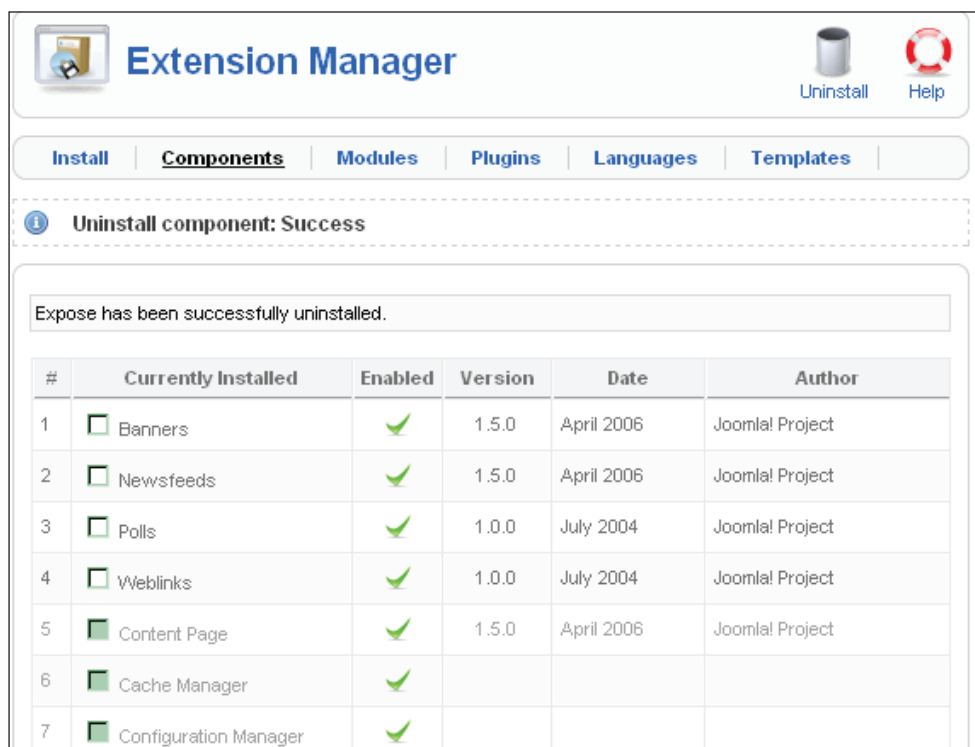


Figure 5.69: Uninstalling a Component

The component will really be totally removed – with all of your pictures!

Album Manager as a Stand-Alone Program

Since working online is sometimes tedious with large files, a *Stand-Alone* version of the Album Manager is available for the Windows and Mac OS X operating systems.

You can install the program in Windows by double-clicking the AlbumManager-2.7-SetupWin.msi file (http://www.gotgtek.com/cms/downloads/cat_view-12.html). The Album Manager works with the [PathtoJoomla!]/components/com_expose/expose directory. If you are working in a local server environment, you can just refer to this directory when starting the Album Manager. The Album Manager will then let you work with the existing structure. The operation is then exactly like the online version (see figure 5.70):

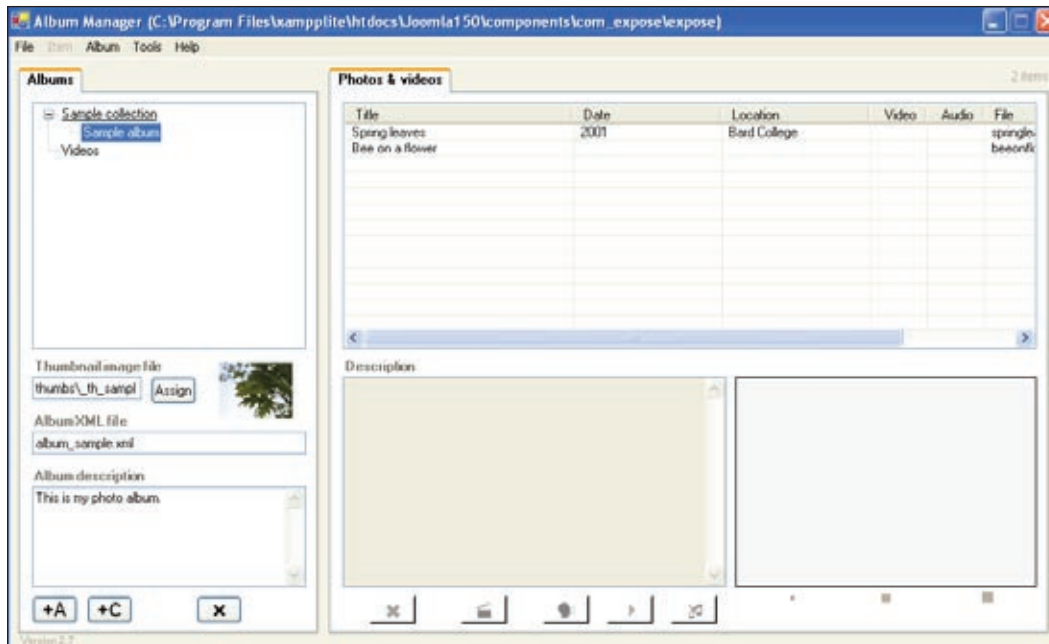


Figure 5.70: Album Manager as a Stand-Alone Program on your PC

If you want to use the Album Manager for a website on the Internet, simply download the directory before you start working with it and upload it again to the server when you are finished.

Integration into the Joomla! Framework

Because Joomla! has changed from a pure content management system to a framework, it is now possible to write Joomla! components that no longer look like Joomla!. The gallery component, for example, has a very loose link to Joomla! 1.5. The gallery is displayed in a wrapper component similar to an IFrame. The back-end administration also works over an IFrame; that's why there is a request for a password in the back end. Other components, for instance the Community Builder (<http://www.joomlapolis.com/>), Facile Forms (<http://www.facileforms.biz/>), or the JoomlaBoard (<http://www.tsmf.net/>), are integrated deeper into the system; and in the case of the Community Builder, for instance, swap the login procedure.

With Joomla! 1.5, a lot of these features can now be solved more elegantly and all of the third-party teams have already been working on the integration of their components since July of 2006. To accelerate the migration, Johan Janssens, Joomla!'s project leader, readied an example of the integration of Facile Forms into Joomla! 1.5 as early as the first of May last year (http://dev.joomla.org/component/option,com_jd-wp/Itemid,33/p,89/).

Problems with Third-Party Components

Update problems become more of an issue as the popularity of Joomla! grows.

Updates

This past year, a new improved version of Joomla! was released on the average of once per month. A lot of Joomla!'s old security problems, among other things, were successfully defeated.

As long as you are operating a website that consists of only Joomla! core code, you are on the safe side. Download the package, overwrite the old files, and you have a "fresh" system. There are rarely any table updates for the database. Even with the upgrade from Joomla! 1.0.x to Joomla! 1.5.x, only two fields have been changed! (see Appendix).

But there are often none or only cursory updates for additional components. As an operator of a website, one is often afraid of the new installation and therefore stays with the older version.

Security

If you are using a lot of additional components, you will often find yourself in a pickle. For example, the Joomla! development team recommends the use of the PHP safe mode to set `register global` on OFF, as well as other measures that will make Joomla! very secure. But some components don't work with these settings. You now have to make a decision to abstain from the use of such components or to live with a certain amount of insecurity.

What to Do?

When you make these decisions, take into account what a broken website (and server) means to your business. How important and how sensitive is the information on your server?

In some cases it may have been acceptable to have a third party operate a website or email address system for you or to at least have them store all of your information from the user registration; but not anymore. With the complexity of the components, there is still an update and security issue and sometimes this is often neglected by third-party developers and website operators.

A Google Summer of Code program team started last year is tackling the development of an update system for Joomla! that downloads updates when needed and perhaps also installs them. An update system of this type would also do a world of good for extensions.



I am not going to tell you to not use Release Candidates (RC) or even Beta versions online. I live in the real world as well and sometimes clients demand it. Depending on the project, some Beta versions are very stable. MySQL (database) ran for a long time as a Beta version and so did the Apache web server. Just be aware of the basic risks! If at all possible, try to make do with the standard version of Joomla!.

Summary

In this chapter we have covered the remaining two important menu items—Components and Extensions. We have also learned how to use a customized component available for Joomla!. Finally we discussed the issues concerning the use of third-party components. In the next chapter we will learn about creating our own templates.

6

Writing Your Own Joomla! Templates

In order to customize the appearance of your site to your company colors, you have to modify an existing template or create a new one. In this chapter, you will learn the basics of building your own Joomla! templates.

Corporate Identity

Corporate Identity (CI) refers to the self-image and the appearance of an enterprise. This appearance, the identity, either results from the enterprise's tradition or is completely invented for a newly created enterprise. This identity is important to give the customer a feel for the enterprise and to enable recognition.

Corporate Identity includes:

- Corporate Image (price, product, and advertising strategy)
- Corporate Design (visual appearance)
- Corporate Communication
- Corporate Behavior (behavior of employees towards each other and towards the outside world)

These areas have to be considered when developing a website. In this chapter, we will examine Corporate Design – at a minimum, it consists of a logo, a character font, and the house colors that the enterprise uses. The visitors to your website should recognize your enterprise on the first visit.

HTML/XHTML, CSS, XML

The abbreviations HTML/XHTML, CSS, and XML stand for Internet technologies that Joomla! works with. The World Wide Web Consortium (W3C) standardizes these technologies.

HTML/XHTML

The World Wide Web (WWW) is based on HTML. **HTML** is not a programming language, but a text-description language. All of the text consists of structures like headings, lists, bold and italic areas, tables, and much more. HTML works with so-called tags. A **tag** has an opening form and a closing form. For example, a first-level heading looks like this:

```
<h1>This is a heading</h1>
```

The tags are interpreted in a browser and the text is displayed according to their meaning. HTML is easy to learn and many online tutorials can be found. HTML is not being developed any further; the successor to HTML is XHTML version 1.0.

CSS

Cascading Style Sheets (CSS) are an extension to HTML. CSS is not a programming language either, but a vocabulary for defining the format properties of HTML elements.

With the help of CSS commands, you can define, for instance, that first-level headings should have a character size of 18 points in the character font Arial, are not bold, and have a spacing of 1.9 cm to the next paragraph. Such options are not possible with pure HTML and were not necessary when HTML was being developed. With the progressive commercialization of the Internet, additional formatting possibilities do, however, become more and more important.

CSS data can be integrated into HTML the following three ways:

- **In the Central HTML File:** The CSS commands are defined in the head section of the HTML file like this:

```
<head>
  <title>title of the file</title>
  <style type="text/css">
    <!--
      /* ... this is where the CSS commands are defined ... */
    -->
  </style>
</head>
```

- **In a Separate CSS File:** If the CSS commands apply to several HTML files, they can be stored in a separate file and the path to this file can be specified in the HTML head section. This is the version that Joomla! uses:

```
<head>
  <title>title of the file</title>
  <link rel="stylesheet" type="text/css" href="formate.css">
</head>
```

- **Within an HTML Tag:** CSS commands can also be integrated within an HTML tag:

```
<body>
  <h1 style="... CSS command ...">...</h1>
</body>
```

These three methods can be combined without any problem in an HTML file. It is, for instance, possible to subsequently overwrite CSS commands that were defined in a central file in the additional source code of an HTML page. This practice, however, quickly results in confusing structures; it is better to customize the central file.

XML

The **Extended Markup Language (XML)** is a universe in itself. It represents a meta-language and derives from the earlier and much more complex **SGML (Standardized Generalized Markup Language)**. XML is often used for configuration files and as an interchange format. For our purposes, you need XML as the description language for the metadata of the templates that you want to create. These metadata are primarily important for the **Template Installer** and the display in the **Template Manager**.

In principle, these data consist of elements with opening and closing tags. For example:

```
<name>Joomla_book</name>
```

The difference between HTML and XML is that in XML no tags are predefined. Because of that, you are totally free in the organization of the structures and the naming of the tags.

Creating Your Own Templates

Now we will shift our focus on how to create our own template. There are several things to consider before we have a finished template package. Let's take it one step at a time.

Concept

Before you start working, you have to create a concept. The work starts with a sketch or a diagram, especially when producing templates. It is up to you whether you create this sketch with programs like Adobe Photoshop, Microsoft Paint that comes with Windows, the open-source program Gimp, or with a piece of paper and markers.

Fixed Size or Variable

You can create two kinds of templates. Templates that adapt their structure to the size of the browser window and templates that have a fixed size.

An example for the flexible layout: if you have 2048 pixels on your screen and the browser window is maximized, then your page is stretched accordingly. That can look strange if you use graphical, non-scalable elements like logos and signatures in your template. You have no control on what it is going to look like.

Your other choice is to decide on a certain resolution and to position all elements exactly on the pixels in the template. The advantage is that your webpage always looks the way you want. Unfortunately, you do not know the resolution of the monitor viewing your page. If that monitor has a resolution of 800 x 600 pixels, then your page fills the screen. On a large screen with a resolution of 1400 x 1050 pixels, it occupies about a quarter of the screen surface and looks a little lost.

There is no real solution for this dilemma. You have to weigh the pros and cons and make a decision. If you prefer the fixed size, you should select a size that looks presentable on most screens (800 x 600 pixels). Since the browser has a scroll bar on the right side and the browser window is framed, the available width is even smaller, so you have a maximum of 780 pixels to work with.

Structure

You are dealing with structured data and first have to determine a general allocation. Joomla! normally uses a structure as shown in the following figure:

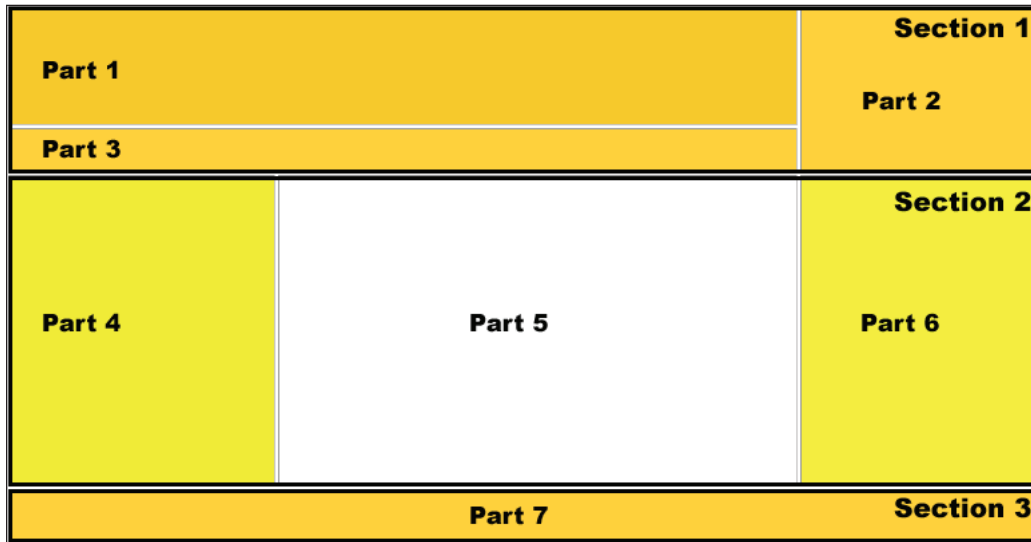


Figure 6.1: Structure

Section 1:

- Part 1: This is where your logo or a picture and the page name go.
- Part 2: This is where the search field is.
- Part 3: Here is the linked navigation path (breadcrumbs).

Section 2:

- Part 4: The most important menus are shown in the left column.
- Part 5: The actual page content goes here.
- Part 6: The right column is a place for additional menus.

Section 3:

- Part 7: The bottom part is the footer.

HTML Conversion

Now you have to convert the concept into HTML. Depending on the graphics program that you used to create it, there is a possibility that the picture can be automatically converted to HTML code. You can also do the conversion manually in a text editor, in an HTML editor like Macromedia Dreamweaver (<http://www.macromedia.com/software/dreamweaver/>), or in one of the numerous free HTML editors (<http://www.thefreecountry.com/webmaster/htmleditors.shtml>).

The source code of the HTML conversion looks somewhat like the following:

Listing 6.1: HTML file

```
<html>
  <head>
    <link rel="stylesheet"
      href="templates/_system/css/general.css" type="text/css" />
    <link rel="stylesheet"
      href="templates/{TEMPLATE}/css/template.css" type="text/css" />
  </head>
  <body bgcolor="#FFFFFF" text="#000000">
    <table width="780" border="1">
      <!-- Section 1 -->
      <tr>
        <!-- Part 1 -->
        <td colspan="2" height="89" bgcolor="#F5C228">&nbsp;</td>
        <!-- Part 2 -->
        <td width="178" height="120" rowspan="2" bgcolor="#FFCC33">
          &nbsp;</td>
      </tr>
      <tr>
        <!-- Part 3 -->
        <td colspan="2" height="33" bgcolor="#FFCC33">&nbsp;</td>
      </tr>
      <!-- Section 2 -->
      <tr>
        <!-- Part 4 -->
        <td width="197" height="233" bgcolor="#F5EE28"> &nbsp;</td>
        <!-- Part 5 -->
        <td width="389" height="233">&nbsp;</td>
        <!-- Part 6 -->
        <td width="178" height="233" bgcolor="#FFFF33"> &nbsp;</td>
      </tr>
      <!-- Section 3 -->
      <tr bgcolor="#FFCC33">
        <!-- Part 7 -->
        <td colspan="3" height="40">&nbsp;</td>
```

```

        </tr>
    </table>
</body>
</html>

```

Listing 6.2: CSS file

```

body {
    font-family: Arial, Helvetica, Sans Serif;
}

```

The code is kept simple and is not consistent with the XHTML standard in the header. To see the division better, the table has a border (attribute `border="1"`). Here you can give your creative juices free reign when it comes to colors and logos. The file name of this layout file has to be `index.php` since Joomla!, by means of PHP, searches it for embedded commands.

A standard CSS file from the Joomla! system and the subsequent CSS file from the individual template are integrated into the header area of the code. This CSS file at the moment only contains one command, which determines the typeface.

You will create this first template completely manually in the Joomla! directory. When the template is ready, you can turn it into a compressed installation package that can then be installed by a third party (or by you yourself) using the Joomla! installer. Save the HTML layout file by the name of `index.php` in the newly created `[PathtoJoomla]/templates/joomla150_book/` directory. Save the `template.css` file in the `[PathtoJoomla]/templates/joomla150_book/css/` directory.

The basic structure of your template is done. Now you have to define it with the help of an XHTML file so that Joomla! will display it in the administration section.

Directory Structure of the Template

Now it's time to take care of certain conventions. As previously discussed, the template has to be stored in a special directory structure:

```

[PathtoJoomla]/templates/[name of the template]/
[PathtoJoomla]/templates/[name of the template]/css/
[PathtoJoomla]/templates/[name of the template]/images/

```

The name of the template cannot contain blanks and other special characters. When this template is later installed as a package, the template installer has to create a directory from this name. Depending on the operating system, exotic combinations of characters can cause problems. In addition, the name should be meaningful. Here `joomla150_book` is the name of the template.

Various files with certain names have to be present in the template directories:

- **Layout File:** This is the basic version of the HTML file that we created earlier: `/templates/joomla150_book/index.php`. It has to have the `.php` ending, since the dynamic Joomla! module elements that we will insert later have to be interpreted by PHP.
- **Preview Picture:** The file `/templates/joomla150_book/template_thumbnail.png` contains a preview picture of your template for selection in the Joomla! administration in the **Extensions | Template Manager** menu. Preview pictures have a format of approximately 227 x 162 pixels. You can create this file later when you see your template.
- **Metadata of the Template:** The `/templates/joomla150_book/templateDetails.xml` file represents the construction manual for the template installer. Here you specify the location where the files are to be copied, who the author is, and additional metadata of the file. During subsequent installation of this template by the Joomla! installer, PHP reads this file out and copies the data to the place specified by the XML file. For the example template, you can use the file from Listing 10.3 and provide your own data. For every file that you use in the template, an appropriate XML element has to be filled with the file name and the correct path.

```
<files>
  <filename> ... enter the filename of a file in the
                    TemplateRoot directory ...
  </filename>
  <filename> ... for every file a filename-Container
  </filename>
</files>
```

The other elements of the XML file are there for the description of the template. Here is the complete functional listing of the XML file:

Listing 6.3: templateDetails.xml

```
templateDetails.xml
<install version="1.5" type="template">
  <name>joomla150_book</name>
  <version>0.1</version>
  <creationDate>28.07.2006</creationDate>
  <author>Hagen Graf</author>
  <copyright>GNU/GPL</copyright>
  <authorEmail>hagen@cocoate.com</authorEmail>
  <authorUrl>http://www.cocoate.com</authorUrl>
  <version>0.1</version>
  <description>... Description</description>
  <files>
```

```

<filename>index.php</filename>
<filename>templateDetails.xml</filename>
<filename>template_thumbnail.png</filename>
<filename>css/template.css</filename>
</files>
</install>

```

Create the `templateDetails.xml` file in the `[PathtoJoomla]/templates/joomla150_book/` directory as well.

- **CSS File:** You can use several CSS files for your template. What name you give the CSS file and how you create it is up to you. There are, however, standard descriptions for various CSS elements. You can find a table of these elements in the Appendix. For your first attempt, you need a CSS file with the name `/templates/joomla150_book/css/template.css` (Listing 6.2).
- **Graphics, Pictures:** Here you can enter user-defined image files that you need in your template. The installer then copies the files into the `images` folder. The file name appears as `/templates/joomla150_book/images/[user-defined image files]`.

First Trial Run

Once you have reproduced all of the structures in the `[pathtoJoomla!]/templates/` subdirectory, you can already see your new template in the **Extensions | Template** menu of your Joomla! administration and you can make it the default:

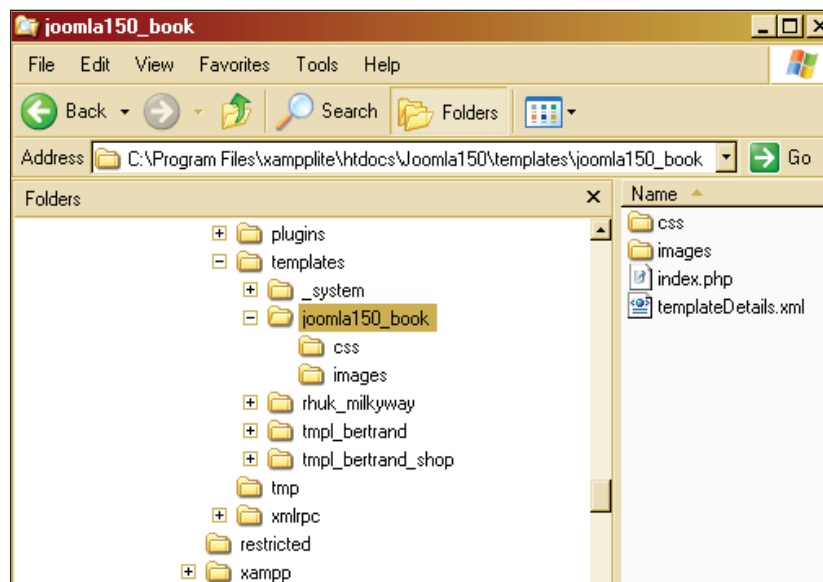


Figure 6.2: Directory Structure and Files of the Joomla150_book Template

Writing Your Own Joomla! Templates



Figure 6.3: New Template in the Manager

When you call up your website, you will see your new template. Unfortunately, no content is shown yet. Since this content is produced dynamically, you have to integrate it piece by piece into your new template.

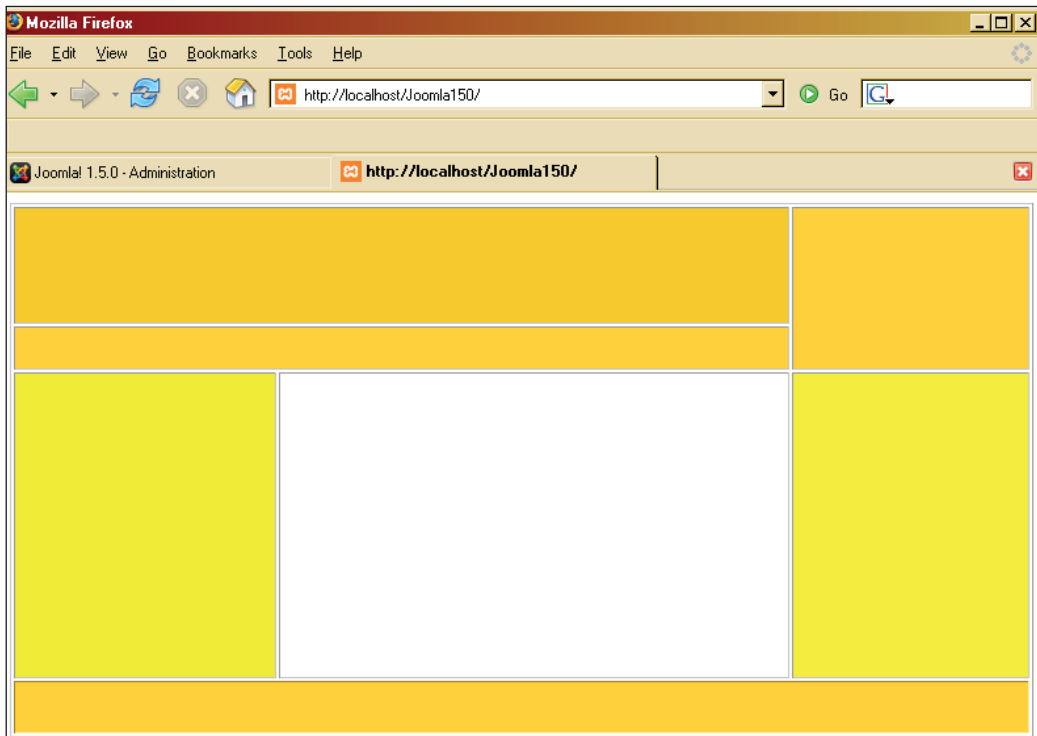


Figure 6.4: View in Browser

Integration of the Joomla! Module

The integration of the Joomla! module takes place by means of commands embedded into the HTML code. These could be the "old" Joomla! 1.0.x or Mambo 4.5.x commands or the new `patTemplate` commands. The `patTemplate` class is there to differentiate between PHP and HTML code. Joomla! uses the namespace `jdoc` in order to be able to integrate various elements into the template. If you insert the following line in place of the `<title>` tag in the head section of the `index.php` file, the favicon, the title of the page, and the Newsfeed symbol are already correctly displayed:

```
<head>
  <jdoc:include type="head" />
</head>
```

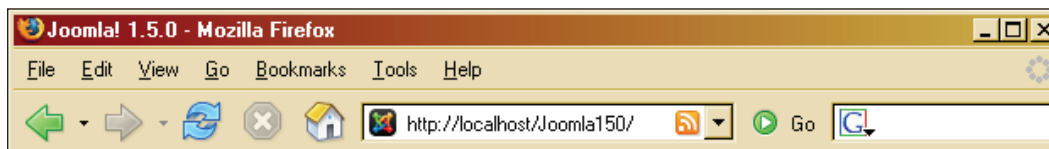


Figure 6.5: Favicon and Page Title

If you call up the source code of this page, you will notice that Joomla! has copied the entire metadata that you entered in administration into the HTML code. In addition, the RSS Feeds have been integrated by means of link tags and these will be displayed as a Newsfeed symbol in browsers like Firefox that support this technology.

Listing 6.4: Joomla! Metadata

```
...
<head>
  <title>Joomla 1.5.0 - Home</title>
  <meta name="generator" content="Joomla! 1.5" />
  <meta name="description" content="Joomla! - the dynamic portal
    engine and content management system" />
  <meta http-equiv="Content-Type" content="text/html;
    charset=utf-8" />
  <meta name="robots" content="index, follow" />
  <meta name="keywords" content="joomla, Joomla" />
  <link
    href="http://localhost/Joomla150/feed.php?option=com_frontpage&
    Itemid=1&format=rss" rel="alternate" type="application/rss+xml"
    title="RSS 2.0" />
  <link
    href="http://localhost/Joomla150/feed.php?option=com_frontpage&
    Itemid=1&format=atom" rel="alternate"
```

Writing Your Own Joomla! Templates

```

type="application/atom+xml" title="Atom 1.0" />
<link href="favicon.ico" rel="shortcut ico" rel="shortcut icon"
type="image/x-icon" />
</head>
...

```

Since this has worked so well, we will waste no time and get to the other relevant `jdoc` tag insertions. The command `<jdoc:include type="modules" name="top" style="-1" />`, for example, expects the type of insertions as the `type` parameter, in our case `modules`. The `name` parameter marks the position of the module (`top`, `right`, `left`, `user1`, etc.) on the website. You can assign this position to your module from the **Extensions | Module Manager** menu in the **Position** category. You can get an overview of all of the possible positions under the menu item **Extensions | Templates | Module positions**. The `style` parameter, finally, contains a value that pertains to the type of HTML code that is being delivered by the module. For example, `-1` delivers pure HTML output without surrounding `<div>` tags.

Parameter	Output
1	Horizontal menu
-1	Raw – without surrounding layout elements
-2	XHTML-compliant output
-3	Option of formatting round corners

Table 6.1: Module Parameters

You can get a description of the `jdoc` insertions in the `index.php` file in the following listing.

Listing 6.5: index.php with jdoc insertions

```

<html>
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="templates/_system/css/general.css"
type="text/css" />
<link rel="stylesheet"
href="templates/{TEMPLATE}/css/template.css" type="text/css" />
</head>
<body bgcolor="#FFFFFF" text="#000000">
<table width="780" border="1">
<!-- Section 1 -->
<tr>
<!-- Part 1 -->
<td colspan="2" height="89" bgcolor="#F5C228">&nbsp;  </td>
<!-- Part 2 -->

```

```

        <td width="178" height="120" rowspan="2" bgcolor="#FFCC33">
        <jdoc:include type="modules" name="user4" style="-1" />
        </td>
    </tr>
    <tr>
        <!-- Part 3 -->
        <td colspan="2" height="33" bgcolor="#FFCC33">
        <jdoc:include type="modules" name="breadcrumb" style="-1" />
        </td>
    </tr>
    <!-- Section 2 -->
    <tr>
        <!-- Part 4 -->
        <td width="197" height="233" bgcolor="#F5EE28">
        <jdoc:include type="modules" name="left" style="-1" />
        </td>
        <!-- Part 5 -->
        <td width="389" height="233">
        <jdoc:include type="component" style="-1" />
        </td>
        <!-- Part 6 -->
        <td width="178" height="233" bgcolor="#FFFF33">
        <jdoc:include type="modules" name="right" style="-1" />
        </td>
    </tr>
    <!-- Section 3 -->
    <tr bgcolor="#FFCC33">
        <!-- Part 7 -->
        <td colspan="3" height="40">
        <jdoc:include type="modules" name="footer" style="-1" />
        </td>
    </tr>
</table>
</body>
</html>

```

If you call up the website with the modified HTML code on the local server, you already see the dynamic content. Your new template has already been filled with all of the data.

Writing Your Own Joomla! Templates

The visual aspect of the result could still use some improvements, but nonetheless the concept works:

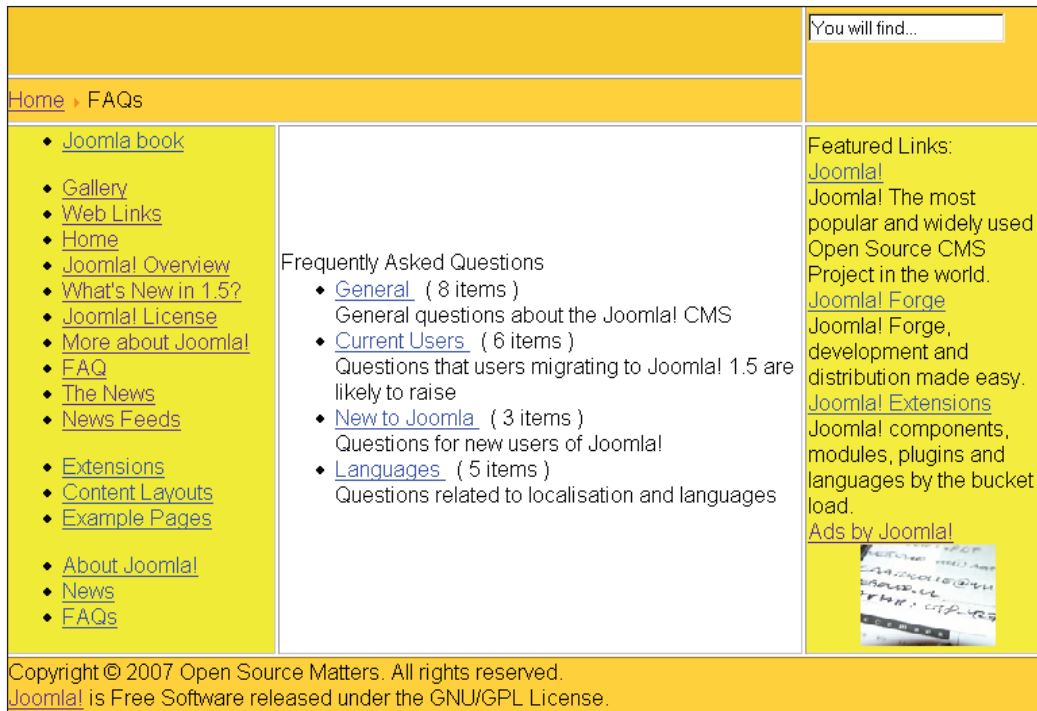


Figure 6.6: Template with Dynamic Data

Let's jump right into CSS formatting to improve the visual aspect. Copy the following code into your `template.css` file. All that we define here is that the default font is Arial, that the links are not to be underlined, and that the script will be displayed in another color and in bold font if you roll over a link with your mouse pointer.

Listing 6.6: template.css

```
body {
    font-family: Arial, Helvetica, Sans Serif;
}
a:link, a:visited {
    color: #ff6600;
    text-decoration: none;
    font-weight: bold;
    font-size: 15px;
}
a:hover {
    color: #C43C03;
    text-decoration: none;
```

```
font-weight: bold;
font-size: 15px;
}
```

These modifications have already improved the attractiveness of your template. The mouse pointer just happens to be on the **More about Joomla!** link in the main menu. The link becomes bold and changes color:

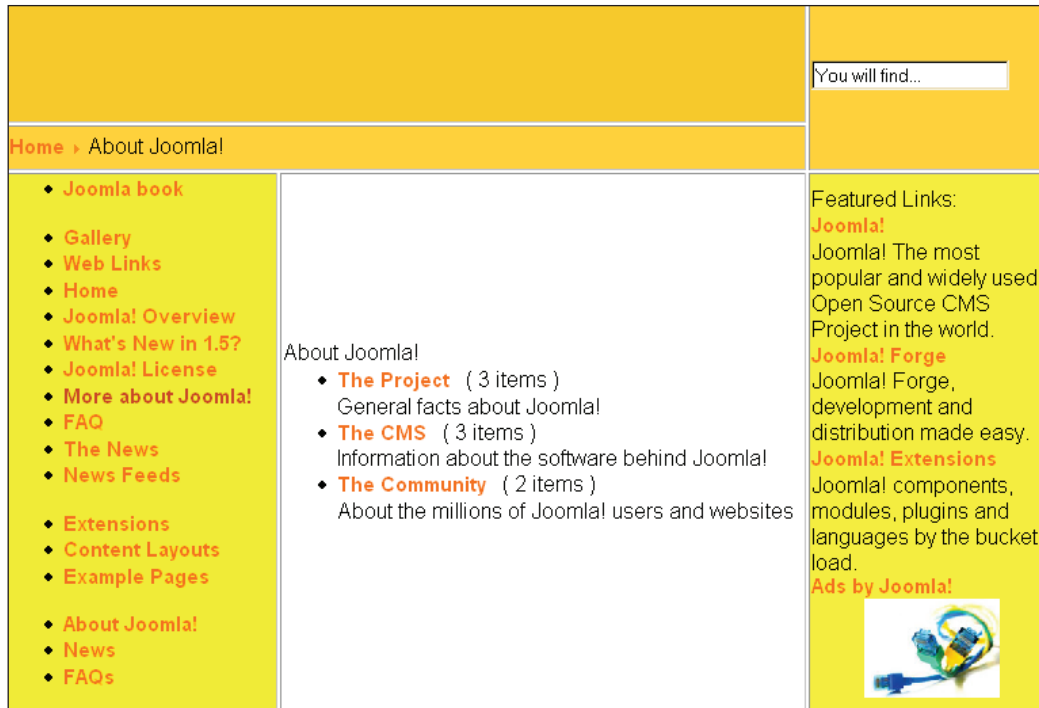


Figure 6.7: Template with CSS File

This template consists of HTML tables that should really be avoided nowadays, since they make barrier freedom impossible if used as layout elements. In addition, no images have been inserted yet. But we have solved all of the core problems. Now we still have to compress the template into an installation packet.

Creating a Template Package

In order to pass your template to others, you have to compress it into a ZIP archive. Before you do that, create a current preview image of your template (template_thumbnail.png) with a size of 227 x 162 pixels. This thumbnail should now be displayed if you pass your mouse pointer across the name link in the template section:

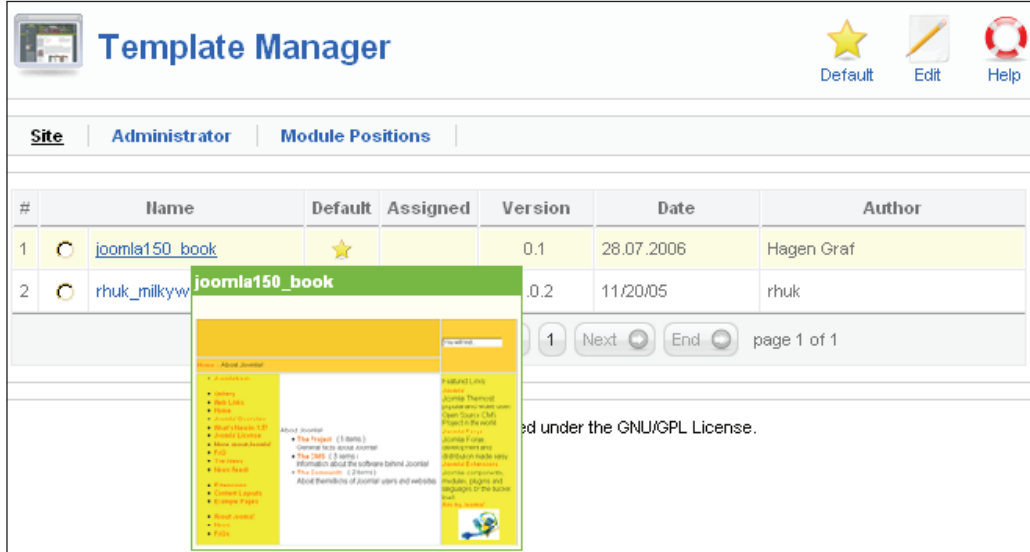


Figure 6.8: Preview Picture of the joomla150_book Template

Now pack all of the joomla150_book files with subdirectories into a ZIP archive. In addition, select all of the files and folders in the [PathtoJoomla]/templates/joomla150_book/ and pack them all into the joomla150_book.zip file. Make a backup of this file and the ZIP file.

You can now pass this package to others or install it yourself. If you want to try the installation, you have to remove the just created template from your Joomla! system.

To be able to remove it, you first have to designate a different template as the default template. Click on the **Extensions | Template Manager** menu, select the desired template, and click on the **Default** icon.

Now you can uninstall your template. Go to the **Extensions | Install/Uninstall** menu and click on the **Templates** tab. Select the just created `joomla150_book` template and click on the **Uninstall** icon:

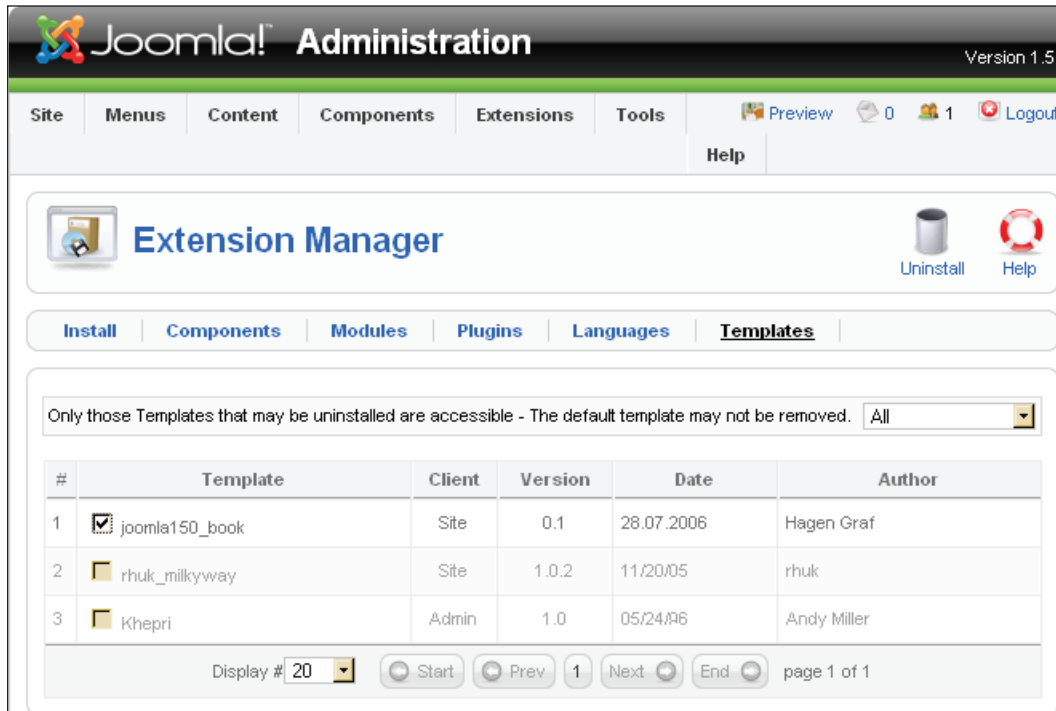


Figure 6.9: Uninstallation of the Template

You can also delete a template from its directory, here `[pathtoJoomla!]\templates`.

Installation with the Joomla! Template Installer

After you have eliminated all traces of your development and have backed up your work, go to the **Extensions | Install/Uninstall** menu.

You can install your ZIP package from here. Select the `joomla150_book.zip` file and install it by clicking on the **Upload File & Install** button. The installer will report that the installation was successful and this success message will also display the description of the XML file.

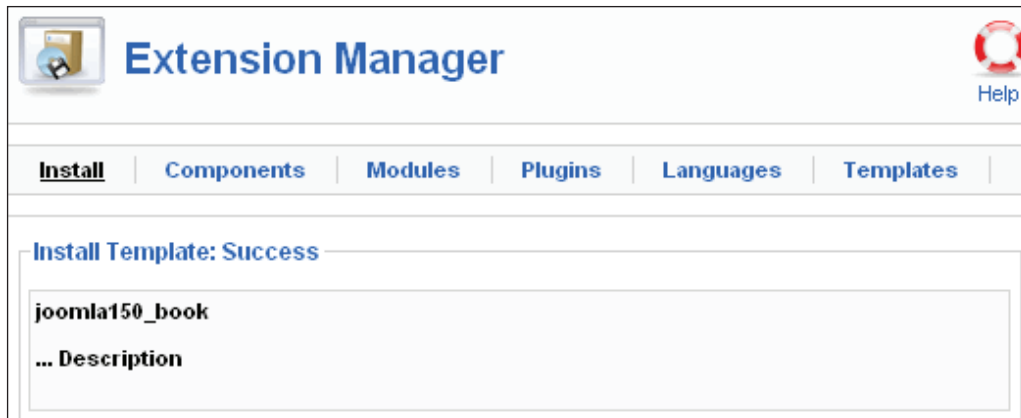


Figure 6.10: Installation of the Template Package File

If you now change to the template section, the `joomla150_book` template will be installed and selectable.

Creating Templates with the Dreamweaver Extension

There is an extension for the HTML editor Dreamweaver with which you can produce a template completely within Dreamweaver. You need a current version of Dreamweaver (MX 2004 and up) and you can install the **Template Builder Extension** there. The program is written for Joomla! 1.0.x and does not incorporate the Joomla 1.5.x tags. But since Joomla! 1.5.x is fully backward compatible, this is not a disadvantage.

Installation

Download the file `joomlasolutions1.0.mxp` (http://developer.joomla.org/sf/frs/do/viewRelease/projects.joomlasolutions/frs.dreamweaver_joomla_template_exte.joomla_dreamweaver_ext_1_0). Start Dreamweaver, click on **Commands | Manage Extensions**. The **Macromedia Extension Manager** opens. Select the file and click on **Install**:

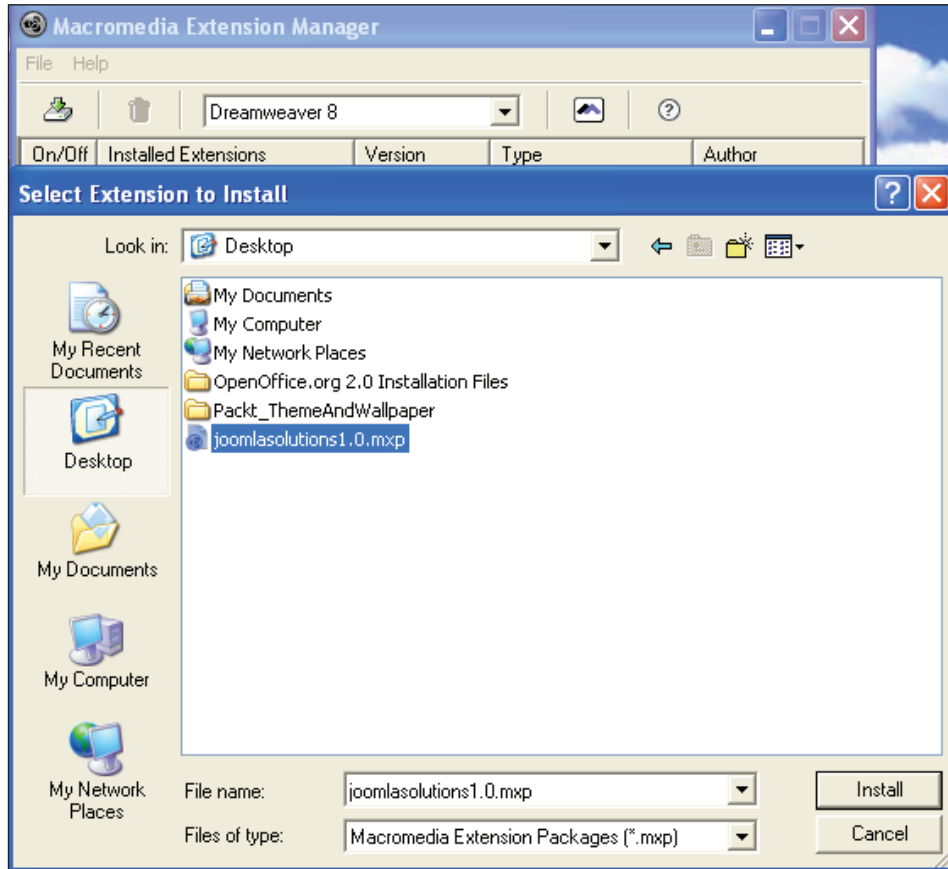


Figure 6.11: Installing the Dreamweaver Extension

Create a New Template File

In order to activate the extension, you have to restart Dreamweaver. Open a new document of the dynamic page type. Click on **File | New | Dynamic Page | PHP:**

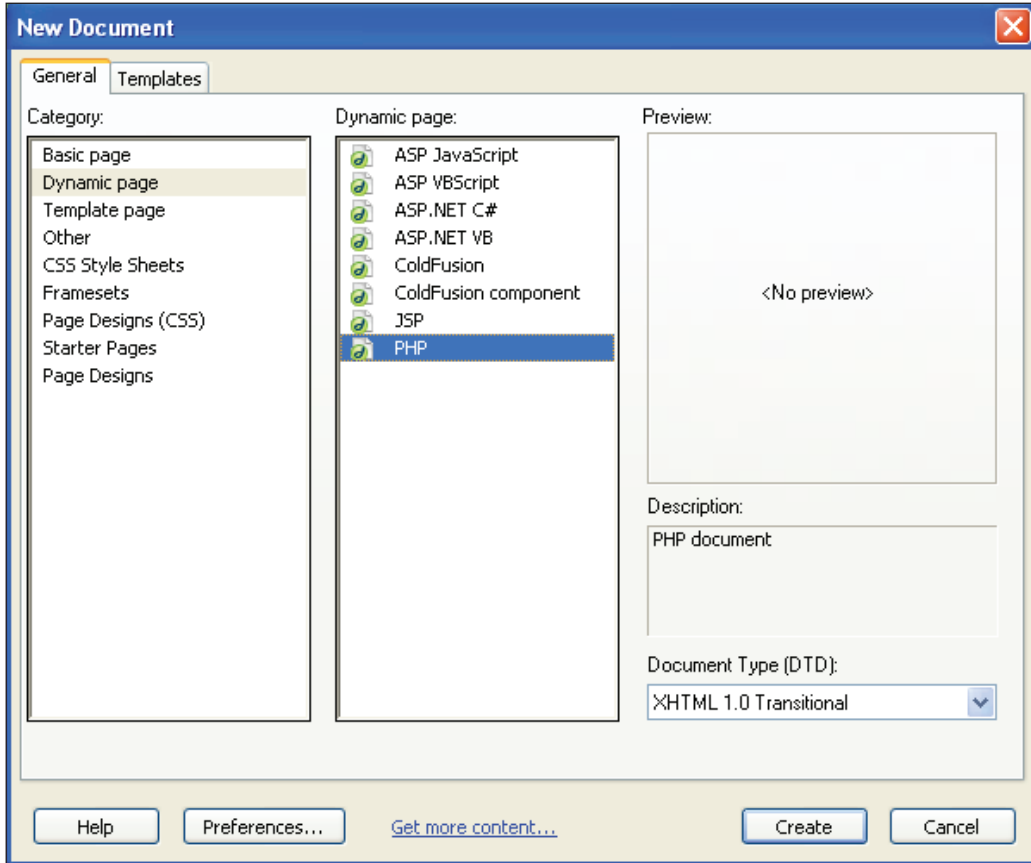


Figure 6.12: Setting Up a New Dreamweaver Document

After the file has been created, you will see a small triangle beside **Common** in the top section that will let you choose various palettes. Select **Joomlasolutions1**:

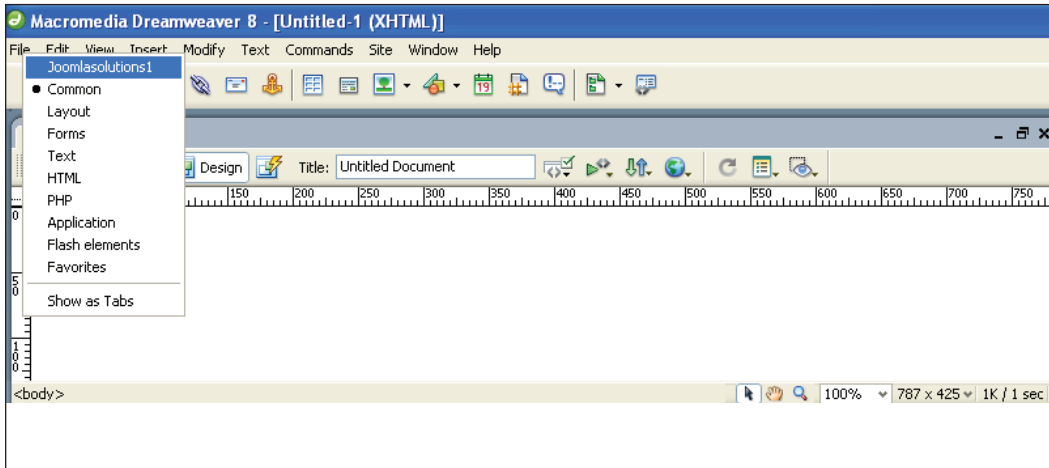


Figure 6.13: Template Palette in Dreamweaver

This palette has various buttons that you can use to add various template elements. The first thing you have to do is to click on the **Code** tab and delete the header in the source code window that was produced by Dreamweaver and replace it with the Joomla!-specific one.

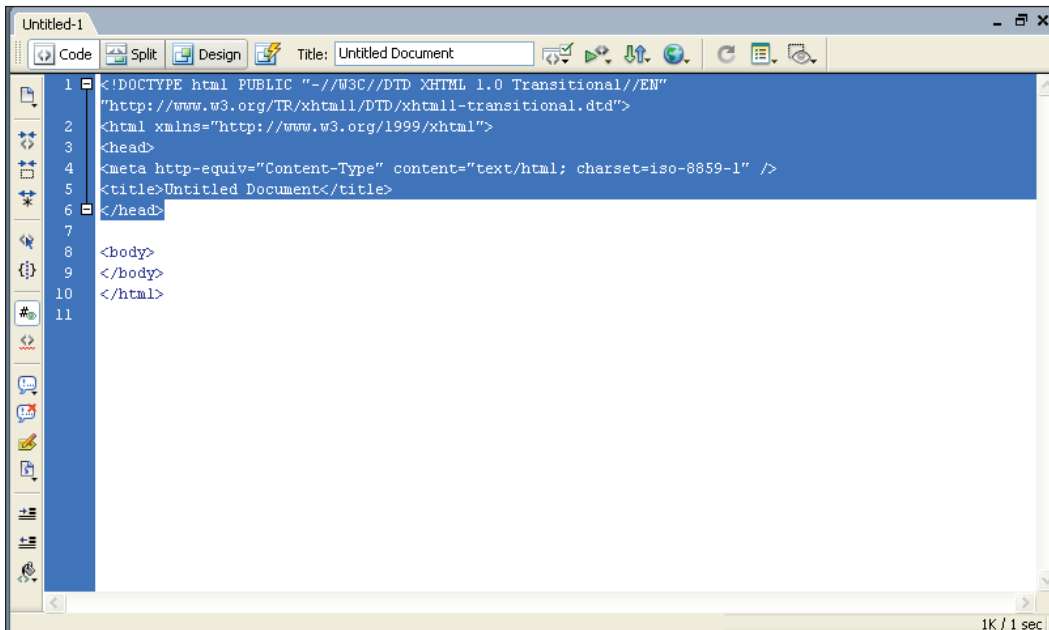


Figure 6.14: Delete Header

Writing Your Own Joomla! Templates

After that, click on the **Insert Head Code** icon. This will insert the source code that Joomla! needs.

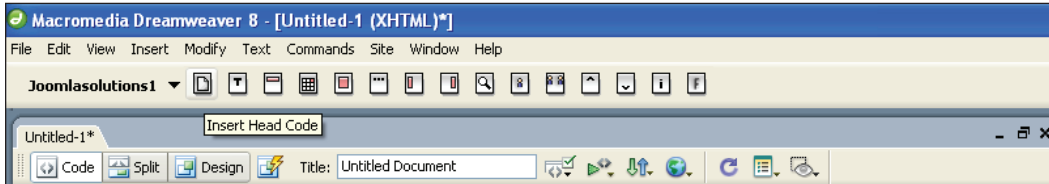


Figure 6.15: Inserting Joomla! Head Code

Now you have the source code in your header that a Joomla! template requires and that we had to type in manually before.

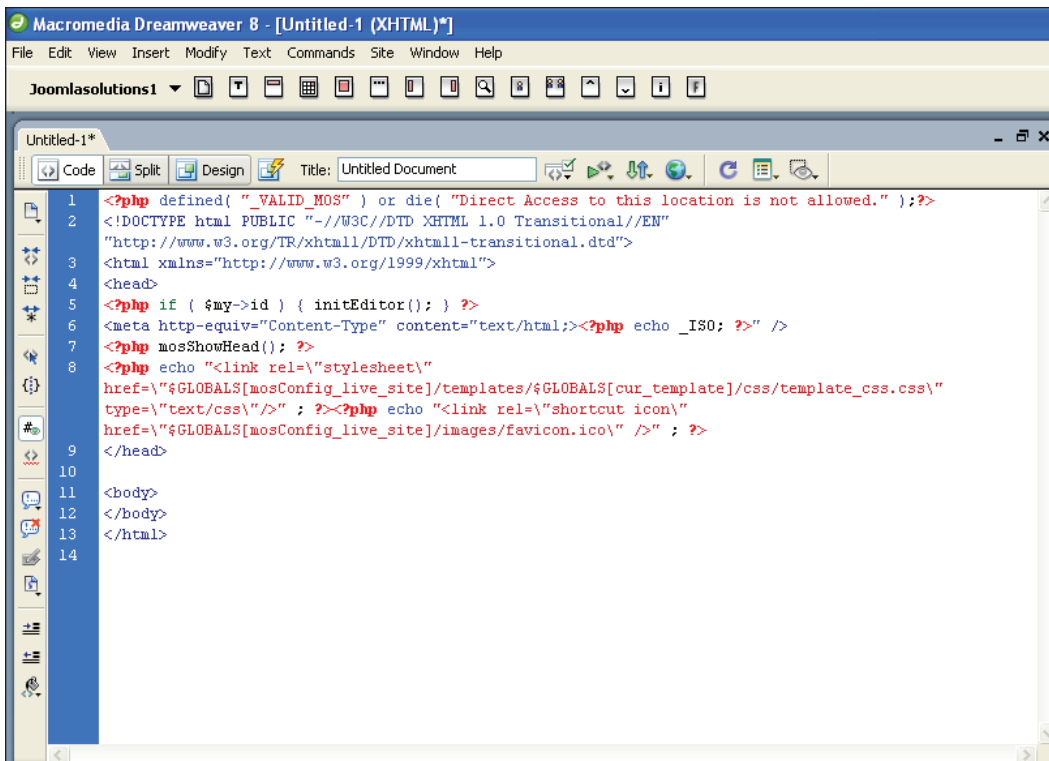



Figure 6.16: The Necessary Head Code for Joomla!

 The inserted code at the moment is based on Joomla! 1.0.x. Joomla! 1.5 is compatible with this code, but in long run it would be expedient and also clearer to use the new commands.

Template Structure

Other than the special header, you now need a structure for your website. You can create this structure with the help of a table or with HTML `<div>` tags. We will describe the table version here. If the layout mode is still activated in your Dreamweaver, please terminate it.

With Table

Create a table structure as described in the last example. You can create the desired table structure by clicking on **Insert | Table**. You have numerous options to create complex table structures with Dreamweaver and to have tables nest within each other. We will demonstrate a simple version on purpose. Color the table as you want.

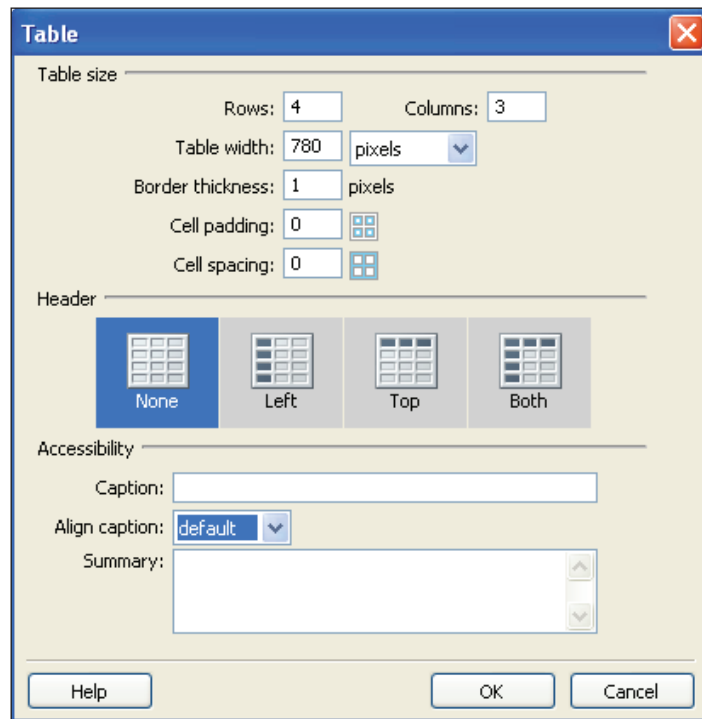


Figure 6.17: Setting up a Simple Table

Writing Your Own Joomla! Templates

With some complex inputs into the tables, you will get a table similar to the following:

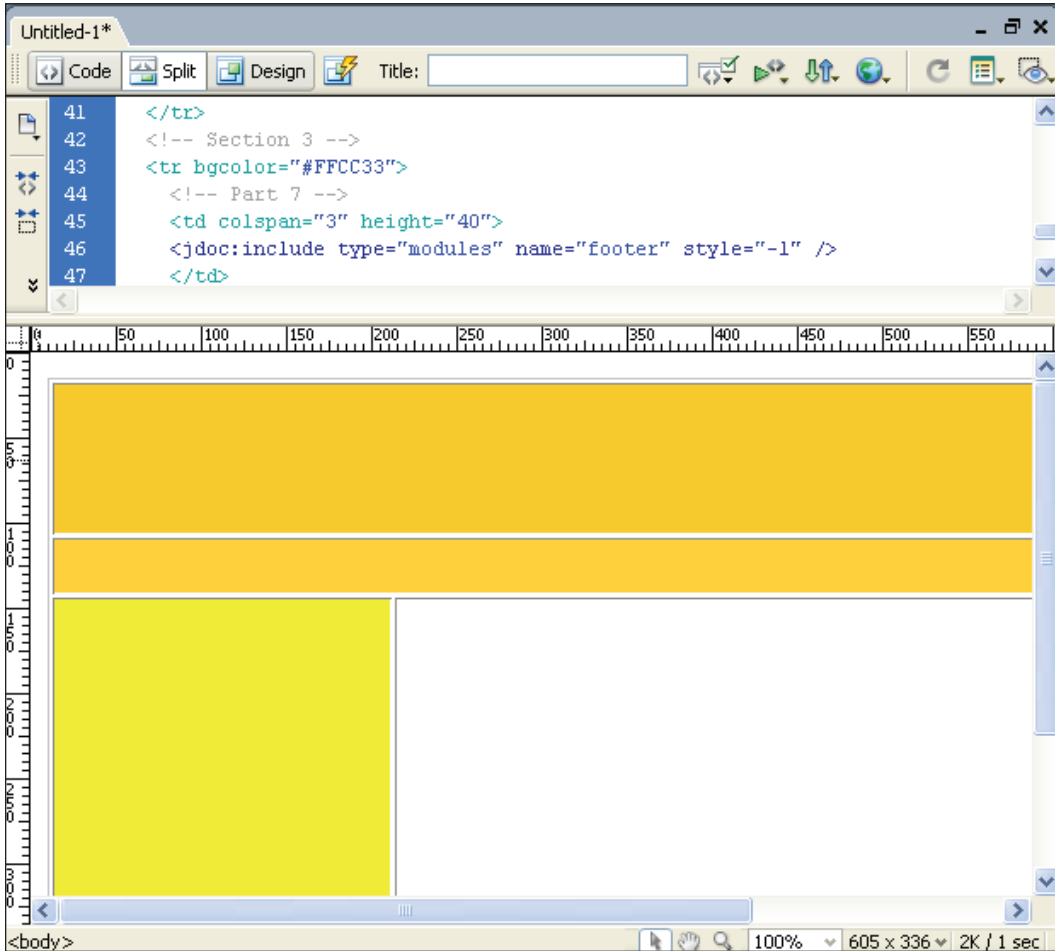


Figure 6.18: Table Structure

Now save the file. Set up a new subdirectory under [pathtoJoomla!]/templates with the name `joomla150_book_dw`. Save the file in it with the name of `index.php`. In addition set up a `css` directory and an `images` directory.

Deposit all of the images and graphics that you will use in your template into the `images` directory. You can create the graphics with a commercial graphics program such as Photoshop or an open-source solution such as Gimp, and place it in your template with Dreamweaver.

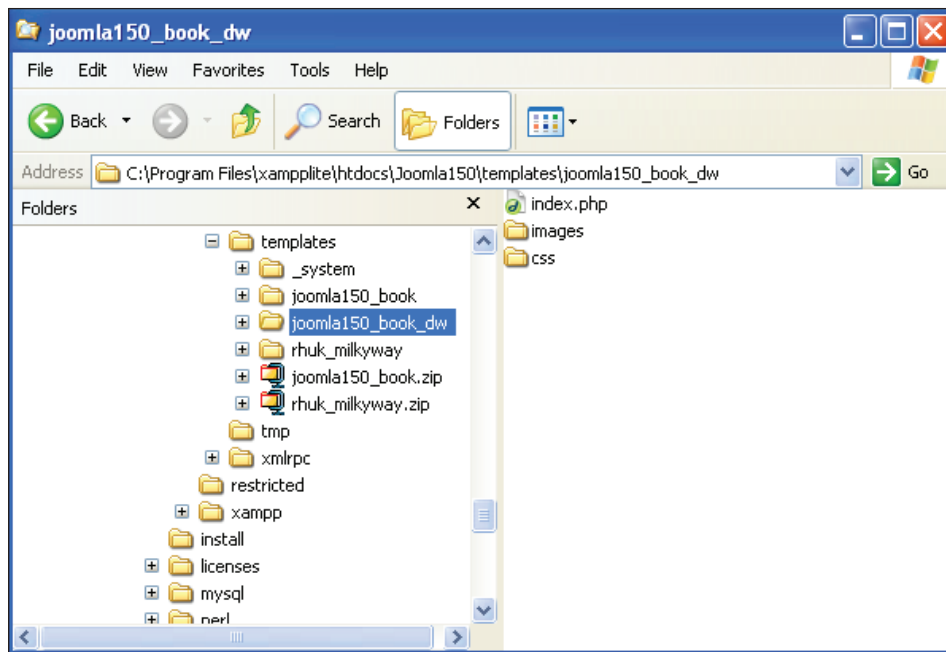


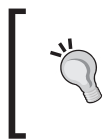
Figure 6.19: Directory Structure of your Template

The CSS file that is valid for the entire template is in the `css` directory, as the name implies. It has to have the name `template_css`. You can create the file with Dreamweaver or copy an already existing CSS file from another Joomla! template. If you create the file with Dreamweaver, Dreamweaver changes the header of your template by inserting a link to the newly created CSS file there:

```
<link href="css/template_css.css" rel="stylesheet" type="text/css" />
```

You will need the absolute path and thus the version that we already inserted a while ago in your subsequent template within Joomla!. Dreamweaver does not evaluate the PHP tags in our situation. The PHP variables do not contain a value and therefore the CSS link that was set up by the extension leads nowhere.

Thus use the link inserted by Dreamweaver before and remove it again later.



I recommend to you to use an already existing CSS file since the Joomla!-specific classes are already defined in it. If you want to give that a try, copy the CSS file or its contents from the `rhuk_milkyway` template into your freshly created CSS file. It is really easy to edit the CSS file with Dreamweaver.

Insertion of the Joomla! Modules

Now you have the basis for inserting the Joomla! modules. Move the cursor to the spot in your table where you want to have, for example, a search field and click on the **Insert Search** icon in the option bar.

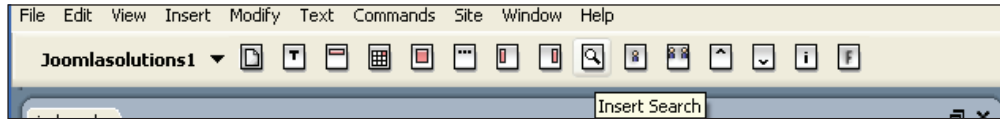


Figure 6.20: Inserting a Search Field

You can now see a search field in the design window and all kinds of HTML and JavaScript code that was added to the source code:

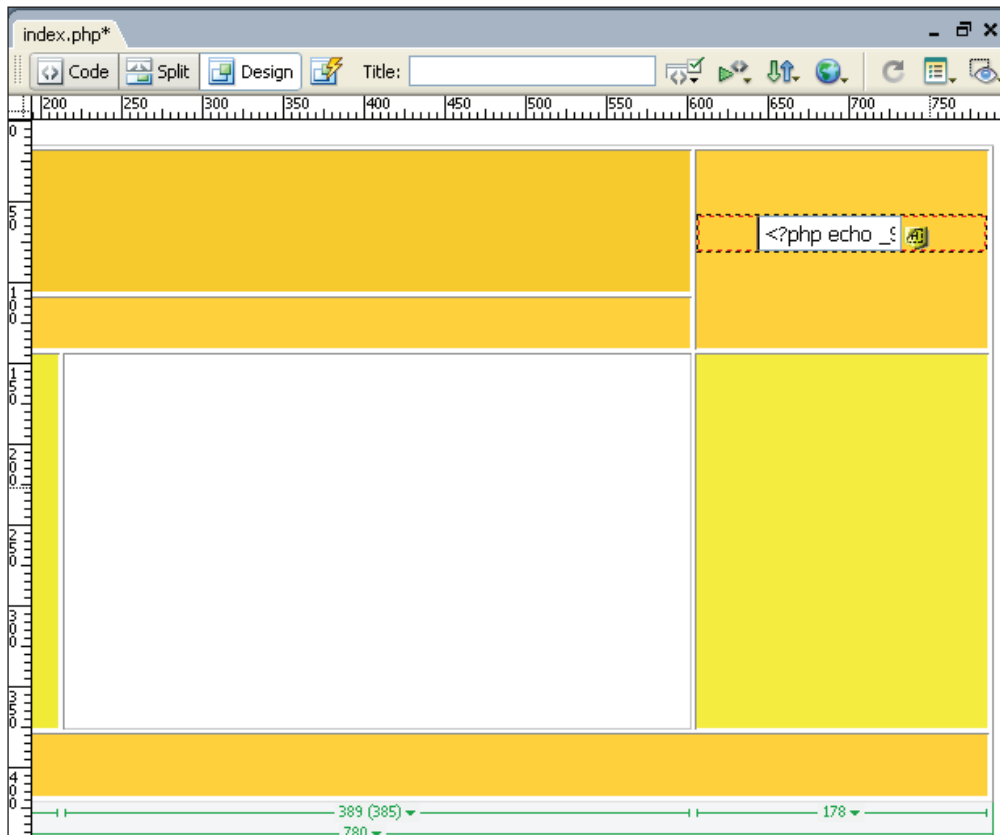


Figure 6.21: Search Field in the Template

Now you can gradually insert the dynamic elements into the appropriate table fields by clicking on the respective icons.

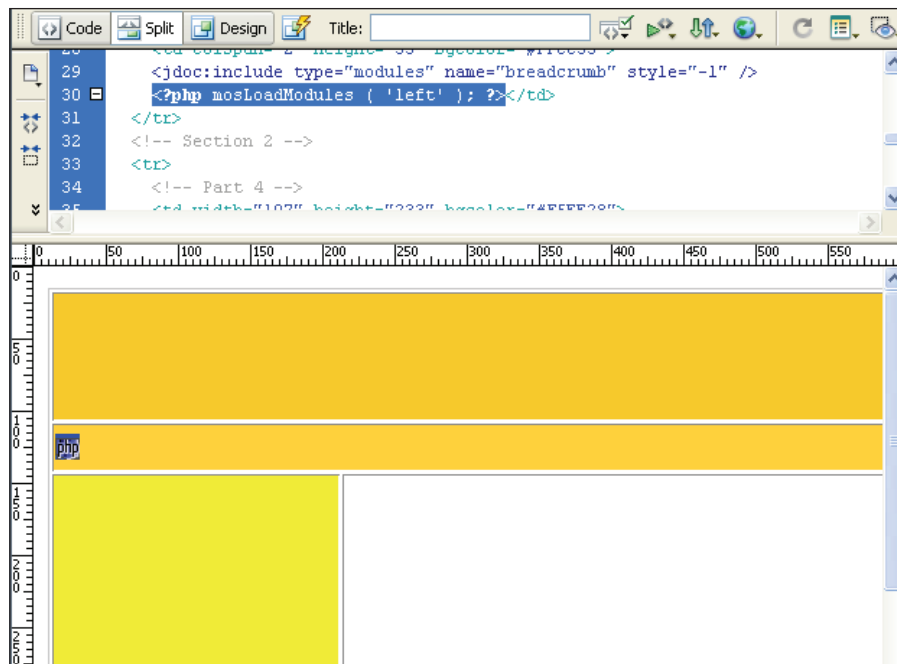


Figure 6.22: Template with Joomla! Modules

Live Site

In order to be able to view the template in your Joomla! installation, you have to create the XML file with the name `templateDetails.xml`. A basic version without pictures looks like this:

```
<install version="1.5" type="template">
  <name>joomla150_book_dw</name>
  <version>0.1</version>
  <creationDate>28.07.2006</creationDate>
  <author>Hagen Graf</author>
  <copyright>GNU/GPL</copyright>
  <authorEmail>hagen@cocoate.com</authorEmail>
  <authorUrl>http://www.cocoate.com</authorUrl>
  <version>0.1</version>
  <description>... Description ... Dreamweaver Version </description>
  <files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <filename>template_thumbnail.png</filename>
    <filename>css/template.css</filename>
  </files>
</install>
```

Writing Your Own Joomla! Templates

The descriptions from *Creating a Template Package* also apply for general publication and the automatic installation from the Joomla! template installer.

After the XML file has been created, you can select your new template in Joomla! administration, specify it as the default, and view your Joomla! site live:



Figure 6.23: Template Manager with your own Template

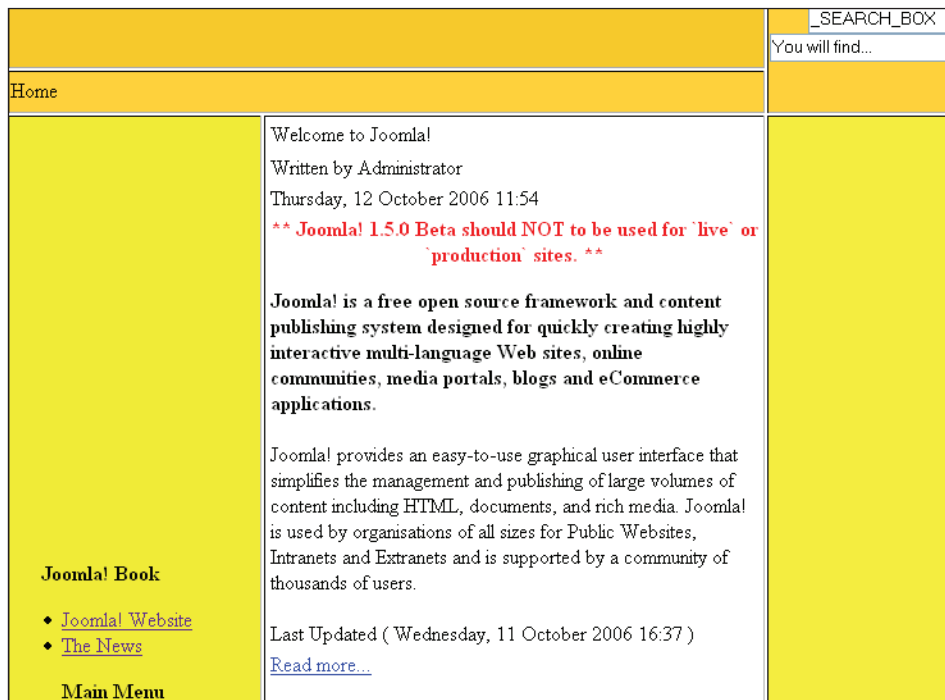


Figure 6.24: Your Website with your Template

With this construct, you can now work in Dreamweaver, use the design mode and the many features of Dreamweaver, and examine the live result on your website immediately after saving. You can even configure Dreamweaver in such a way that your just-modified template file is automatically updated on the distant FTP or WebDAV server of your provider. To take advantage of that, set up a Dreamweaver site with the root path to your Joomla! files and the appropriate access data to your live site.

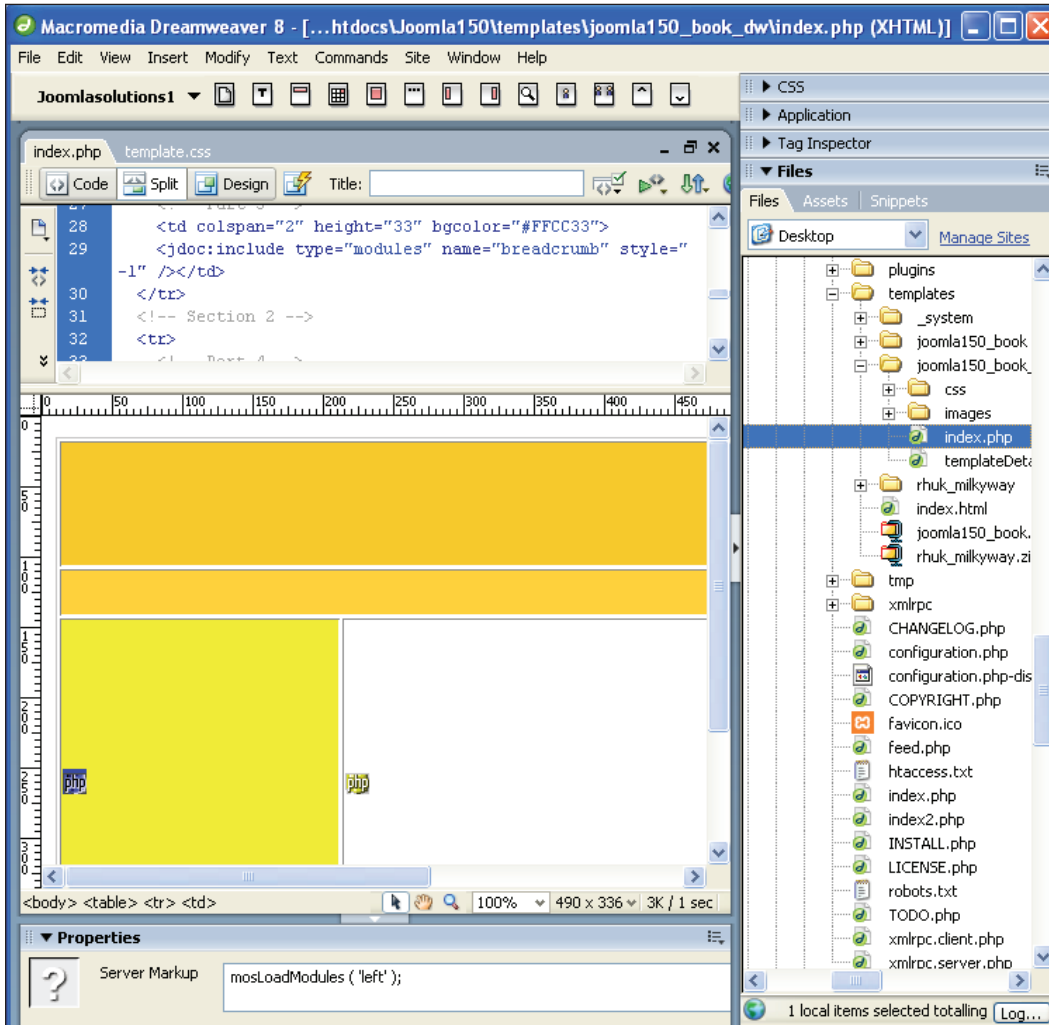


Figure 6.25: Dreamweaver with Site Utilization

Templates and <div> Tags

Joomla! version 1.5.0 works with HTML table representations as well. In the next versions of Joomla!, this process will change in order to accommodate the requirements of barrier freedom. The <div> tag will replace and supplement the <table> tag in HTML.

You can thus include several HTML elements such as text and graphics in a shared area. This area at first doesn't affect anything except that it starts in a new line of the text flow. The <div> tag does not have any other properties. The big advantage, however, results from the combination of <div> tags with CSS statements. <div> was invented for exactly this purpose: In order to be formatted with the help of CSS statements. These properties of the <div> tags make it interesting to produce templates that are controlled by a CSS file.

The included `rhuk_milkyway` template uses this technology. It no longer contains table tags but creates the entire table structure by means of <div> tags. Take a look at the original source code of this template to get familiar with this technology. Dreamweaver also supports this technology.

The Left Module Position in Detail

Let's take a look at the left module bar in more detail:

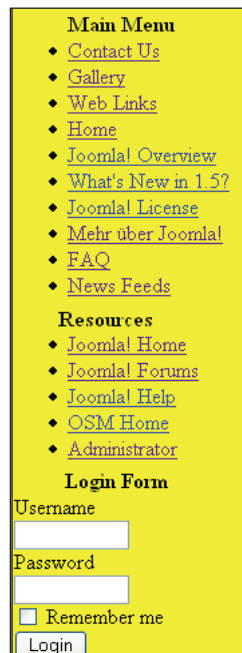


Figure 6.26: Module at the Left Position

This is the HTML code in the `index.php` layout file:

```
<div id="leftcolumn">
  <jdoc:exists type="modules" condition="left" >
    <jdoc:include type="modules" name="left" style="-3" />
  </jdoc:exists>
</div>
```

The `<div>` tag is formatted by a CSS ID with the name of `leftcolumn`. The content between the `<jdocs:exists>` and `</jdoc:exists>` tags is only displayed if the conditions in the parameters returns true. In this case the decision will depend on whether there are any modules present at the `left` position that should be displayed. If there are no modules there, the layout could be modified appropriately to fill any existing spaces.

In our case, there are three modules.

The CSS `leftcolumn` id looks like this in the CSS file:

```
#leftcolumn {
  padding: 0;
  margin: 0;
  width: 20%;
  float:left;
}
```

If you take a look in a browser at the HTML code that was output (right-click on your site and then click **View Page Source**), you will, of course, see a lot more code that was created by Joomla! with the `jdocs` request.

Listing 6.7: Excerpt from the output HTML source code

```
... additional HTML commands
<div id="leftcolumn">
  <div class="module_menu">
    <div>
      <div>
        <div>
          <h3>Main Menu</h3><ul class="mainmenu">
            <li class="level1 item1 active current">...</li>
            <li class="level1 item2">...</li>
          </div>
        </div>
      </div>
    </div>
  <div class="module_menu">
    <div>
      <div>
```

```

        <div>
        <table ...>
            <tr ><td>...</td></tr>
            <tr ><td>...</td></tr>
        </table>
    </div>
</div>
</div>
</div>
... additional HTML commands

```

The `module_menu` CSS class is conspicuous in this code. (This code is based on the alpha version of Joomla! 1.5.) We covered the additional CSS suffixes in the module parameters in Chapter 5. This apparently is one such case. They can also immediately be found in the CSS file:

```

div.module_menu {
    background: url(../images/mw_box_blue_br.png)
                100% 100% no-repeat;
    ... additional commands ...
}
div.module_menu div {
    background: url(../images/mw_box_blue_bl.png)
                0 100% no-repeat;
}
div.module_menu div div {
    background: url(../images/mw_box_blue_tr.png)
                100% 0 no-repeat;
}
div.module_menu div div div {
    background: url(../images/mw_box_blue_tl.png)
                0 0 no-repeat;
    padding: 10px;
    padding-top: 30px;
    padding-bottom: 15px;
    width: auto;
}
div.module_menu div div div div {
    background: none;
    padding: 0;
}

```

These four CSS blocks activate a particular background picture with every nested `<div>` tag. In this case, the result is the rounded edges of the menus. There are also additional entries in the CSS file that format the third-order header (`<h3></h3>`) and an unordered list (the menu links) in this case.

So far, so good. This type of formatting is desirable, and the output of the necessary `<div>` can still be changed with the `style` attribute (you will find a list of the parameters in the appendix). We used the `-1` code in our sample template; in this template, the `-3` code is used. It delivers the respective four-part `<div>` tag. This type of coding is very suited for formatting each CSS. If you continue reading in Listing 6.7, you will stumble across tables again! The login module still works with tables! At this point you run into a problem if you want to customize the login module without tables. But Joomla! has a solution here as well.

Modifying HTML Output without Changing the Core Files

In order to get a grip on the table-output problem, you would normally have to change Joomla!'s core files. There wouldn't be any tables in your login, but it is possible that the file will be overwritten with the next Joomla! update.

Ever since Joomla! 1.5.0 was released, the so-called `views` are the solution for this problem. Every component receives an additional `views` folder, in which you will find various numbers of views of the output of the logic component. With the `com_content` content component, these are, for example, `archive`, `article`, `category`, `frontpage`, and `section`. These five folders contain files that supply different views of the content. I am sure you remember the table and blog view. Every one of these folders also contains a `tmpl` folder and it in turn contains a PHP file with the default name of the view (and the superordinate directory), in this example it is `article.php`.

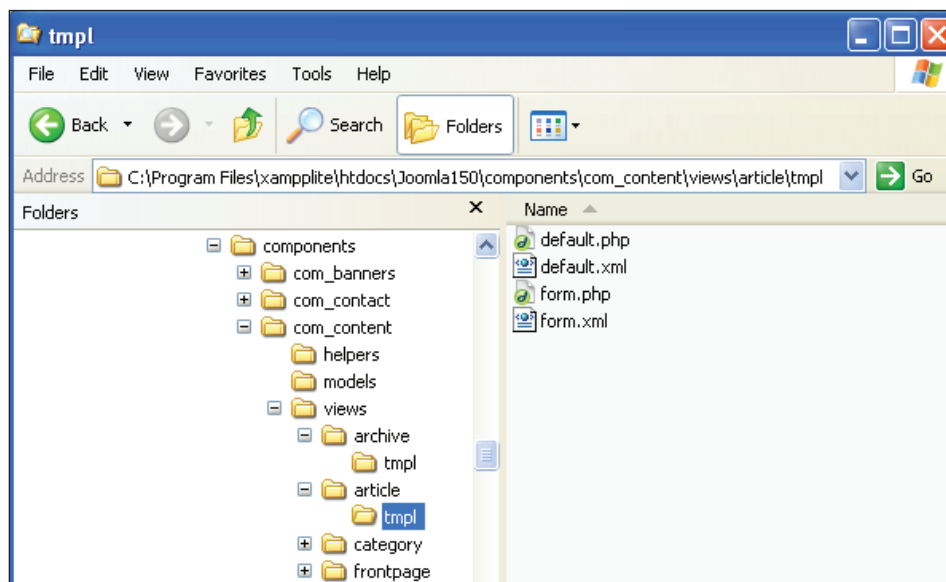


Figure 6.27: Views Directory Structure

If you want to use this method, you will need an additional directory in your template directory. In this example, the directory would have to be called `com_content`, since you want to change the layout of the content component:
`[PathtoJoomla]/templates/joomla150_book_dw/com_content/`.

If you now copy the `[PathtoJoomla]/components/com_content/view/article/tmpl/default.php` file into your new template directory, you can customize it to your heart's content. Joomla! first checks the template directory for these files. If they are there, they are used; if they are not there, the core files from the component are used. This type of modification does indeed make customization of the HTML code possible!

Web Accessibility for Joomla!

Web accessibility starts in the head! In many countries this is a specific topic that has even been regulated. Complete web accessibility and fundamental accessibility are difficult to achieve on the Internet, but, of course, not impossible in principle. The term accessibility is often used in this connection. When authorities offer their services for sale on the Internet, they must be accessible for as large a number of inhabitants as possible.

Barrier-free access to a website is the same as a ramp for wheelchair users at a bus or a crossing light that beeps when the light changes to green, so that the blind can hear it.

Accessibility is a distinct advantage for a company and/or anyone offering a public service. Every visitor to a website is also a potential customer. This topic transcends far beyond handicapped or older people and will win worldwide importance.

Criteria for Accessible Websites

In order to give you an idea of the necessary changes to your website, here are a few criteria:

- **Clarity:** Every website should be clear and fast to comprehend. Information should preferably be in those places in which the user expects it.
- **Browser compatibility:** There are different kinds of Internet browsers. From the text-based Lynx on the Linux console, through browsers on cell phones, PDAs and other mobile terminals to various browsers on various operating systems (OSX, Linux, Unix, Windows, OS2). All of these terminals have different resolutions and can or cannot display graphics, can or cannot run JavaScript code, can or cannot play Flash movies, etc.

Remember that there are also character recognition programs and, for example, Braille keyboards for blind people. These programs are also browsers!

- **Valid source code and logically structured page architecture:** Navigation, layout and contents are the important facets of a website. They must be developed logically content-wise and be semantically correct. XHTML is a standard. The source code that you use should be valid!
- **Contrasts:** A high-contrast representation of website content must be ensured to accommodate vision-impaired users.
- **Graphics and Pictures:** A lot of output devices for handicapped people cannot represent graphics. Therefore it is absolutely necessary to have an alternative way of representing content.
- **Font Sizes:** The writing must also be readable on older and alternative systems.



In order to get a feeling for the genuine requirements, take a quick look at Section 508 (<http://www.section508.gov/>).

The Reality

At first, the reality is a little frightening. There is hardly a website that conforms to these simple and plausible rules. "Eye Candy", animations, and complicated navigation rule most websites. A process of rethinking is, however, taking place with the increased popularity of content management systems and also because of the regulatory laws.

Companies that offer their goods on the Internet, are slowly realizing that accessible websites are also good for their business. Customers simply find it easier to navigate! The websites of barrier-free online banking sites are, by the way, extremely popular with customers compared with other online banking websites!

In real life you have to deal with different user groups, various user terminals, and different requirements. Thus, your website will always be a compromise! Don't set your requirements too high and start simple.

Is Joomla! 1.5.0 Web Accessible?

In order to make it short: No. The current Joomla! 1.5.0 does, however, lay a good foundation that, according to the Joomla! Roadmap will be implemented with the next version.

Is it Possible to make Joomla! Web Accessible?

In order to answer this question briefly: Yes!

The Technology

Joomla! 1.5.0 still uses an XHTML table layout. Table tags are insurmountable for a barrier-free setup and may not be used. Therefore the programming code in the Joomla! system that is responsible for output has to be changed. You can accomplish this with the view system that we touched on in *Modifying HTML Output without Changing the Core Files*. By using a flexible access to the views, barrier-free templates will be here very soon.

If you now build a clear, table-free structure, valid XHTML, scalable fonts, appropriate color choices, and sufficient contrast into your structure, accessibility will start to look pretty good already. From a purely technical perspective, you are now well prepared for the future.

You can test the functionality with various tools. You can find an overview of the usual tools, among other places, on Jan Eric Hellbusch's website.

The People

Now we come to the people that set up contents on the website – the editors. Here you have to wake up their sense of responsibility.

This, for example, is a valid and also semantically correct text input:

```
<h1>first order heading</h1>
<p>first paragraph</p>
<p>second paragraph</p>
```

What looks so simple is so difficult to get into the brains of those who ultimately write the texts. One could let the heading look like a heading even with the change of script attributes. The following command has a similar effect, however, it is no longer a heading:

```
<span class="heading">first order heading</span>
```

The following command would be the correct version:

```
<h1 class="heading">first order heading</h1>
```

How can you get an editor to learn this?

There are two programs in the text-processing world that represent the standard in editing of texts: OpenOffice.org and Microsoft Word. It is absolutely necessary for a similar tool with similar capabilities to make text input possible for web pages. The WYSIWYG editor used by Joomla! is only a small start.

Here is another example of the correct formulation of a graphic with the `title` and `longdesc` attributes.

```


```

The `longdesc` attribute refers to a separate description.

Many such "little things" must be considered. The WYSIWYG editor that comes with Joomla! is not really usable for these tasks. Nevertheless, in order to enable easy editing, you can deactivate functions of the editor in the source code. However, detailed knowledge of the JavaScript language is necessary for this.

The correct phrasing is also important when it comes to the people that write the text! A small summary about the subject matter of the article has to be in the opening text. The language must be easily understandable and proper for the target group. Everything you learned in your journalist courses suddenly becomes important again.

Web Accessible Sites with Joomla!

Here is an example of a barrier-free website of 3tc, a not-for-profit website
(www.3tc4u.co.uk):



Figure 6.28: Barrier Free Website

You can find additional examples of barrier-free sites on the website of the W3C site.

Summary

This chapter has helped you create your own first template. The chapter also discussed on the necessity of making Joomla! pages accessible to people with disabilities and different ways to do so. In the next chapter we will learn to make our own extensions.

7

Your Own Components, Modules, and Plug-Ins



The code in this chapter is based on the Alpha version of Joomla! 1.5. The code in the download section of the Packt website has been modified to contain the `models` and `views` folders, and may work with the Beta 1 version.

Let's say you want to solve a problem with Joomla!, but there are no ready-made components for it yet. For instance, you are a car dealer and you need a listing of your used vehicles on your website that you can administer with the Joomla! administration or you need a list of your branches. Simply extend Joomla!'s functionality with new components, modules, and plug-ins. What looks quite difficult at first can actually be accomplished with beginner-level knowledge of PHP.

I am sure you can figure out what you have to do after your experience with building templates. The emphasis this time, however, is not on layout but on programming. But before we really get going, let's work on a simple "Hello World" component.

Sample Helloworld Component

Components are split into front-end and back-end components. Front-end components are displayed on your website, back-end components are developed for the administration section, mostly to manage front-end components. From the viewpoint of a visitor to your site, you can recognize a component by the way a Joomla! page is requested.

The URL `http://localhost/Joomla150/index.php?option=com_contact`, for instance, calls up the `com_contact` front-end component.

If you take a look at your database system, there are a whole bunch of components in the [PathtoJoomla] /components subdirectory and one of them is the com_contact component:

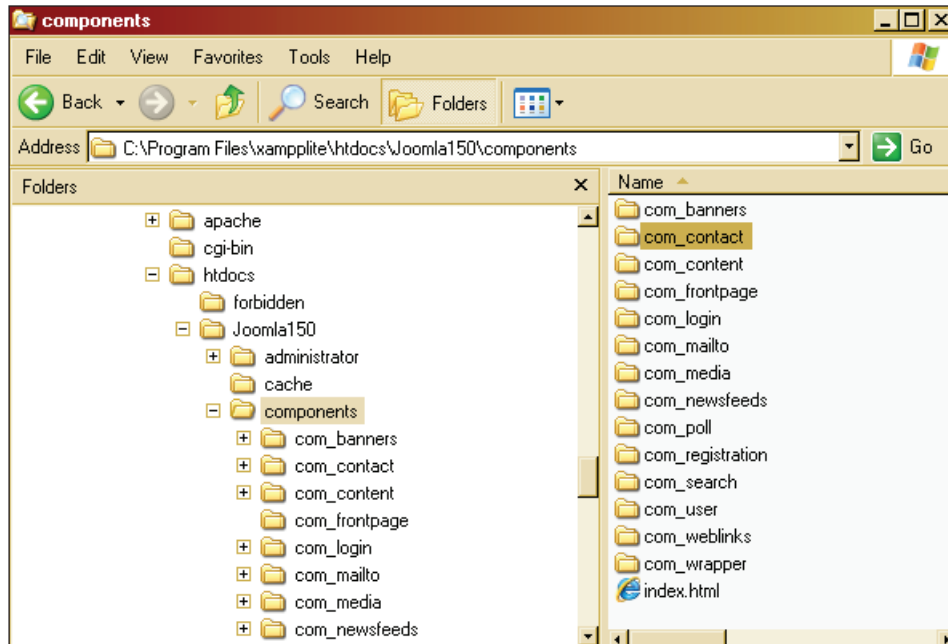


Figure 7.1: Joomla! Components

Depending on the complexity of the components, there are additional directories with files contained within this directory. But there is only one really important file right now. In the case of the com_contact component, this is the `contact.php` file. It controls the rest of the components.

A Home for Helloworld

Every component inhabits its directory and Joomla! suggests the following sequence:

1. Joomla! interprets the rendered values in the URL:
`/index.php?option=com_helloworld.`
2. It searches the component table for the `com_helloworld` component.
3. It looks for a `com_helloworld` directory.
4. In that directory it looks for a `helloworld.php` file.
5. It interprets this file.

So, you need to set up a `com_helloworld` directory and create the `helloworld.php` file in a text editor. Of course, you still need a little source code for the component to be able to greet us.

Listing 7.1: helloworld.php file

```
<?php
echo 'Hello World!';
?>
```

Check the source code in the URL `http://localhost/Joomla150/components/com_helloworld/helloworld.php`.

You should get an output like that in the following screenshot:

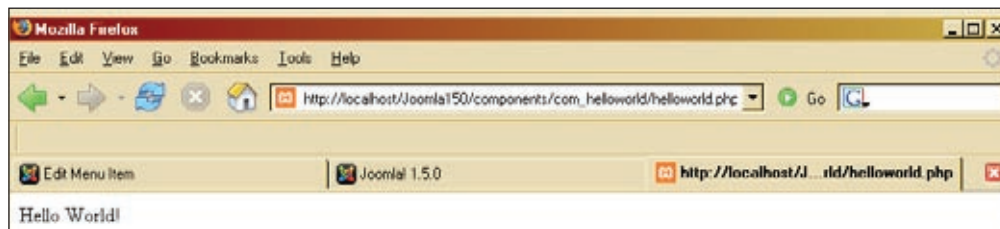


Figure 7.2: Test of the Simple Source Code

Now you are certain that your source code is correct. But you really would like to display the component in terms of the Joomla! framework (in other words inside of Joomla!).

Making Use of the Joomla! Framework

The Joomla! framework consists of numerous functions, methods, and classes that are all there to help you develop components. Previously, we called the new component directly with its file name. For security reasons, this should be prevented as soon as possible. To do that, extend the `helloworld.php` file with the following commands:

Listing 7.2: helloworld.php

```
<?php
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
echo 'Hello World!';
?>
```

If you now try to access this file directly using the URL `http://localhost/Joomla150/components/com_helloworld/helloworld.php`, this attempt will be prevented by the Joomla! framework.

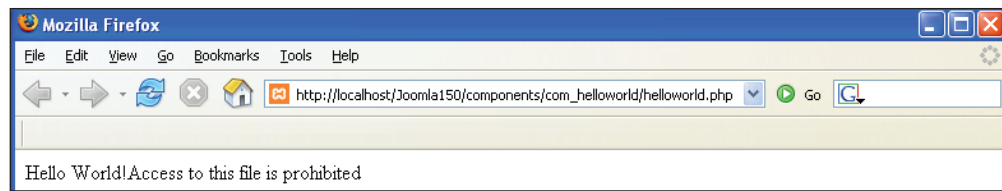


Figure 7.3: Direct Access to the Component is Not Possible

You can continue to access the component from the `http://localhost/Joomla150/index.php?option=com_helloworld` URL.

The `_JEXEC` constant is only set if the file is called from the Joomla! URL. Joomla! 1.5.0 is backward compatible to Mambo and Joomla! 1.0.x. With these, the respective constant is called `_VALID_MOS`.

A Little more Functionality

In order to put a little more interactivity into the world, we will now pass parameters to the component.

Listing 7.3: helloworld.php

```
<?php
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
$task = JRequest::getVar('task');
$name = JRequest::getVar('name', 'Joomla! 1.5.0');
switch ($task) {
    case 'show':
    default:
        echo 'Welcome ' . $name . ' to our World!';
        break;
}
?>
```

To make Joomla! more secure, the `JREQUEST::getVar()` method was adopted. It reads the passed parameters from the URL and tests them for possible attacks such as, for instance, SQL injections. In our case, the `task` and `name` parameters are read out and tested. Based on the content of `task`, a `switch` command will branch into different directions. If `task` has no value, the `default` case is assumed. A text will be combined with the content of the `name` parameter. If `name` has no value, Joomla! 1.5.0 is used.

The URL, `http://localhost/Joomla150/index.php?option=com_helloworld`, outputs:

Welcome Joomla! 1.5.0 to our World!

The URL, `http://localhost/Joomla150/index.php?option=com_helloworld&name=Hagen`, displays: **Welcome Hagen to our World!**

Now you have a tool that makes the component react dynamically.

Separation of HTML Output and Programming Logic

One goal of all software developers is, among others, the separation of logic and presentation. Programmers and designers have differing levels of expertise and ideas, and with the separation of logic and presentation they don't get into each other's way, making coordinated software development possible. PHP code is characterized by a colorful mixture of PHP commands and HTML enhancement. This is good for a beginner because the learning curve is flat. But with larger projects, the code soon becomes unintelligible as one loses sight of where the HTML code is. Joomla! 1.0.x already had a separation, and in Joomla! 1.5.0 it is even more consistent.

In order to make this separation already clear at the programming level, an additional `helloworld.html.php` file is added to our `helloworld.php` logic file. In `helloworld.php`, all necessary data are collected, edited, and created; in `helloworld.html.php`, they are brought into a structured format.

Listing 7.4: helloworld.php

```
<?php
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
// load the HTML View class
require_once (JApplicationHelper :: getPath('front_html',
                                           'com_hello'));

$task = JRequest::getVar('task');
$name = JRequest::getVar('name', 'Joomla! 1.5.0');
switch ($task) {
    case 'show':
    default:
        hello_HTML::show($name);
        break;
}
?>
```

Listing 7.5: helloworld.html.php

```
<?php
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
class hello_HTML {
    function show($name) {
        echo 'Welcome ' . $name . ' to our World!';
    }
}
?>
```

The principle is simple, the `hello_HTML::show($name);` command in the `helloworld.php` file requests the `show()` method in the `hello_HTML` class and passes the `$name` variable. The `hello_HTML` class is implemented in the `helloworld.html.php` file. How the passed parameter should be displayed is defined in the `show()` method. Even if this looks a little confusing at first glance, the advantages of this separation become easier and easier to see the more complex the code becomes.

An Example Component

We want to set up a used-car table that can be maintained from the back end. For this you will need: a new front-end component, a back-end component, a special table in the database, an additional module to display the items on the website, and a search plug-in so that you can search your new content.

So that you get an idea of what I am talking about and what the list should look like on the website, here is the finished example and an additional link. When you click on the link, a simple list of vehicles should appear:



Figure 7.4: Your Own Component on the Website

In our case, this list contains automobile types. If you want to, you can fill the example with other types of data (branches, offers, etc.). It is usually sufficient for visitors to be able to view the list; they don't have to be able to edit the items. The administrator, on the other hand, has to manage the list. Management in this connection means being able to:

- Enter new items
- Modify existing items
- Delete existing items

So that we don't make this example too complicated, let's just display and edit one single field. The principle of programming a component in Joomla! will become clear that way and you can easily extend this example by a few more fields. The component should fit into the existing structure. The administration interface in the administration section should look like the following figure:



Figure 7.5: Your Own Component in the Joomla! Administration

It should be possible to execute the above named functions. In addition, you will of course need a toolbar for the display of the list and another one for the edit mode. You need to have options to publish items and to unpublish them, and of course you want the **Publish** and **Unpublish** options to be able to affect several items at once by marking the checkboxes in front of the items.

Let's start with the component; here it is called `auto`. If you want to duplicate this demonstration, you can download the finished example and install it like any component. You can then manually modify the component. Modification and conversion has the advantage that you will take a good look at the structures and that, maybe, you will want to do more. (I admit that more complex code examples quickly become confusing. In this case I have tried to keep the files as small as possible without leaving out important things.)

The MySQL Table

So that you can start right at the beginning, you need a new table in MySQL for your entries in your relevant database (here, `Joomla150`). Set up a new table with the `CREATE TABLE` command. You have to label the individual fields (name, size, type) and set up an index. Every dataset gets a unique number by means of this index and can be explicitly addressed in a later search.

With the `AUTO_INCREMENT` function you ensure that the number (of the key) is automatically incremented by one every time a new dataset is inserted. The `INSERT` command inserts the datasets. With it you enter data into the individual fields of the table.

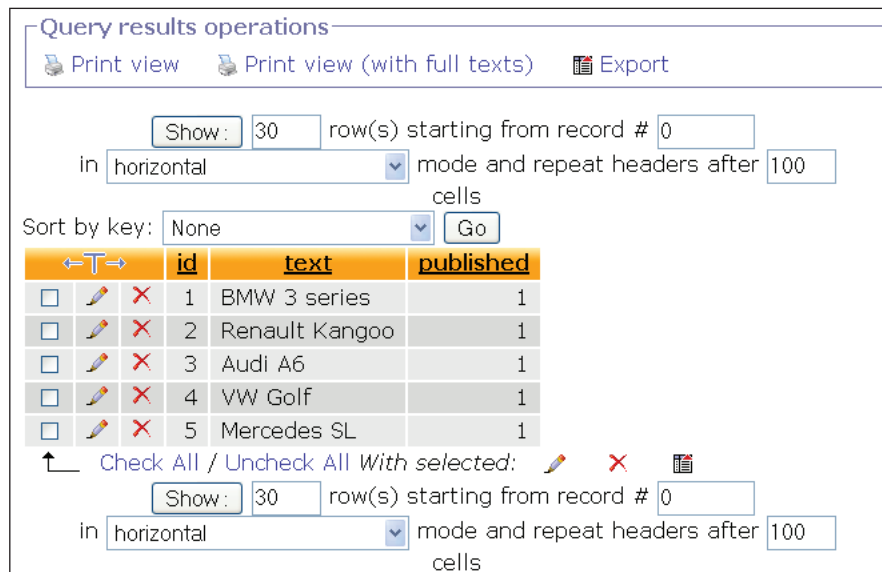
You need the following SQL commands to create the table and to insert the data in this component example:

Listing 7.6: SQL commands for the example table

```
-- Table structure for 'jos_auto' table
CREATE TABLE 'jos_auto' (
  'id' INT NOT NULL AUTO_INCREMENT, 'text' TEXT NOT NULL,
  'published' TINYINT(1) NOT NULL, PRIMARY KEY ('id') );
-- data for 'jos_auto' table
INSERT INTO 'jos_auto' VALUES ('', 'BMW 3 series', 1);
INSERT INTO 'jos_auto' VALUES ('', 'Renault Kangoo', 1);
INSERT INTO 'jos_auto' VALUES ('', 'Audi A6', 1);
INSERT INTO 'jos_auto' VALUES ('', 'VW Golf', 1);
INSERT INTO 'jos_auto' VALUES ('', 'Mercedes SL', 1);
INSERT INTO 'jos_components' ('name', 'link', 'option', 'enabled')
VALUES ('Auto', 'option=com_auto', 'com_auto', 1);
```

To pass these SQL commands to the database, select your database in phpMyAdmin, and click on the SQL link. An HTML form appears into which you can enter the SQL commands and send them directly to the database. Enter the commands from the above listing into the form and click on **GO**.

phpMyAdmin sets up the new `jos_auto` table and inserts five datasets. If you now click on the new `jos_auto` table and the **Browse** link, you will see the result in a clearly laid out format. The description is in the `text` field and the `published` field is used to administer the publishing status. A value of 1 means that the respective item is published on the website. A value of 0 prevents publishing.



Query results operations

Print view Print view (with full texts) Export

Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

	id	text	published
<input type="checkbox"/>	1	BMW 3 series	1
<input type="checkbox"/>	2	Renault Kangoo	1
<input type="checkbox"/>	3	Audi A6	1
<input type="checkbox"/>	4	VW Golf	1
<input type="checkbox"/>	5	Mercedes SL	1

Check All / Uncheck All With selected:

Show: 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Figure 7.6: `jos_auto` Table

We have set up the table and can leave phpMyAdmin for the time being.

We manually created the table and inserted the values into the database, since our component that will assume this job with automatic installation in Joomla! in the future is not ready yet and therefore cannot be installed by means of the installer. You have to define the SQL commands for the correct initialization of your component in a separate XML configuration file.

The Front End

Now we start programming the list in the front end.

Set up a new subdirectory by the name of `[PathtoJoomla!]/components/com_auto`. Set up two files in this subdirectory:

- `auto.php`: This file again contains the logic of the component in pure PHP code. The database is queried here and the result is transferred to an array.
- `auto.html.php`: This file handles the presentation of the data. PHP is used in it as well, but the bulk of the commands are in HTML and CSS code.

auto.php

This file consists of four sections and a few lines.

The first section ensures that this file can not be accessed directly with the filename.

```
defined('_JEXEC') or die('Access to this file is prohibited');
```

This measure is for security as this component cannot run by itself without the Joomla! framework. In addition, the source code of the second file (`auto.html.php`) is integrated into it.

```
require_once (JApplicationHelper::getPath('front_html', 'com_auto'));
```

Since this integration of the HTML interface is necessary with every component, a method is available for it by the name of `JApplicationHelper::getPath()`. It searches the same directory for the corresponding data presentation file and integrates it. With it, the name is dynamically created from the name of the directory and the `.html.php` suffix.

In the second section, the parameters (here the HTML title of the page) are passed. The title is displayed in the blue title bar of the browser window.

```
$document =&$mainframe->getDocument();
$document->setTitle( "Example componenet Auto" );
```

There is a method for this purpose as well: `setTitle`.

The third section contains a conditional switch. You can request the component with different parameters, which you then evaluate here. For example, you can differentiate between a table view and an individual view of the datasets. When we apply this to the used car list, the name of the item could be structured as a link; clicking on that link could display a detailed description of the vehicle.

```
switch( $task ) {
    case 'free':
        // more display options
        break;
    default:
        listAuto();
        break;
}
```

The fourth section contains the actual logic. The database is queried here, a results array is created, and the `showtable()` method in the `HTML_auto` class is called up. This class is in the integrated `auto.html.php` file.

Here is the complete source code:

Listing 7.7: auto.php

```
<?php
// Section 1
// no direct access
defined('_JEXEC') or die('Access to this file is prohibited');
// loading of the HTML View class
require_once (JApplicationHelper::getPath ('front_html',
                                             'com_auto'));

$task = JRequest::getVar ('task');
$name = JRequest::getVar('name', 'Joomla! 1.5.0');
// Section 2
$document =&$mainframe->getDocument();
$document->setTitle( "Example component Auto" );
// Section 3
switch( $task ) {
case 'free':
    // more display options
    break;
default:
    listAuto();
    break;
}
// Section 4
function listAuto( ) {
    global $mainframe;
    $db =& $mainframe->getDBO();
    /* SQL query for all published items */
    $query = "SELECT * FROM #__auto WHERE published='1'";
    $db->setQuery( $query );
    $rows = $db->loadObjectList();
    HTML_auto::showTable( $rows );
}
?>
```

auto.html.php

The `auto.html.php` file contains the `HTML_auto` class. A class is a sort of construction plan with numerous methods and properties. We have already come across two methods of a class previously (`JApplicationHelper::getPath()` and `JRequest::getVar()`). While those methods had been predefined in Joomla!, we have to write this method and class ourselves. Our small class only contains the `showTable()` method.

An array (`$rows`) is passed to the `showTable()` method as parameter. The array is the result of a database query and contains the list of all of the datasets that have been entered. This list is to be displayed in a table. In order to achieve this, a `foreach` loop is executed in the array until no elements are left.

```
...
foreach ($rows as $row) {
...

```

A different value is created for the `$k` variable with every loop through the array. The value in `$k` controls the selection of the CSS class used to format a table line (`class="sectiontableentry<?php echo ($k+1);`). The result of this, when the CSS class has been implemented in the CSS file, is that alternate lines are displayed in two different colors.

Listing 7.8: Implemented CSS classes

[PathtoJoomla]/templates/rhuk_milkyway/css/template.css.

```
td.sectiontableentry0{
    background-color : #fff;
    padding: 3px;
}
td.sectiontableentry1{
    background-color : #ddd;
    padding: 3px;
}

```

In principle, you can implement as many methods as you want in the class. As already mentioned, there could be an additional individual view besides the table view:

```
HTML_auto::showtable($rows);
```

This is a matter of a static class (`HTML_auto`), whose method (`showtable($rows)`) can be called up directly. Class name and method name are separated by two colons.

Listing 7.9: auto.html.php

```
<?php
// no direct access
defined('_JEXEC') or die('Access to the file is prohibited');
// Start of the class definition
class HTML_auto {
    // Start of the method definition
    function showTable( $rows ) {
?>
<table width="100%" border="0" cellspacing="0" cellpadding="0"
        align="center">
```

```

<tr>
  <td height="20" class="sectiontableheader">
    <?php echo "&Uuml;header"; ?>
  </td>
</tr>
<?php
  $k = 0;
  // Selection of the datasets in the array
  foreach ($rows as $row) {
    ?>
    <tr>
      <td height="20" class="sectiontableentry"<?php echo ($k+1); ?>">
        <?php echo $row->text; ?>
      </td>
      <?php
        $k = 1 - $k;
      }
    ?>
  </table>
<?php
  }
}
?>

```

Integration into the Main Menu

Since we have not yet installed the component properly at this point, Joomla! doesn't recognize it to be there. You can, however, call it up from the URL:

http://localhost/joomla150/index.php?option=com_auto.

You can insert a menu link in the component by means of the **Add Menu Item** dialog. In the Joomla! administration, click on the **Menus | Main Menu** item and then on the **New** icon. Click the **Auto** component and in the next step you can position it in the menu.

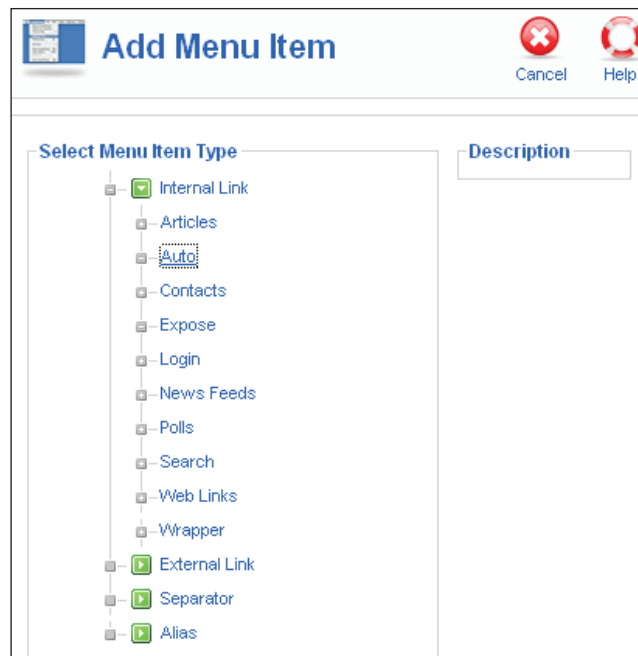


Figure 7.7: Menu Entry per URL

If you call up your website now, you should be able to see the **Auto** menu link, and if you click on it, the auto table (as seen in Figure 7.8).

You can see that the table fits itself to the template automatically and that the entire job wasn't that difficult, right? This is a great argument to get seriously involved in object-oriented programming and in PHP, HTML, and CSS syntax.

The Joomla! Administration

The pure display of data on the website was relatively simple; administration of the data, by nature, is a little more complicated. As an administrator, you have to be able to display, modify, insert, delete, and publish data. This involves significantly more interactivity than there was in the simple listing on the website.

The Component Table

Joomla! doesn't use the components in the `jos_components` table just for the front end. The menu items of all of the components in the back end also have to be noted here. Since you didn't use an installer, but built everything manually, you will again need at least one, preferably two, new menu items. The first menu item sees to it that your new component is displayed in the **Components** menu and the second one defines the (**Edit**) sub-item.

Call phpMyAdmin again, select the `jos_components` table, click on the SQL link, insert the two SQL commands, and confirm it with **GO**. (Here the provided ID number is 100.) The second item refers to the first item. It is therefore important that the ID numbers agree. At the moment there should be a maximum of 35 items in the table, so this method should work:

Listing 7.10: SQL commands menu entry

```
INSERT INTO 'jos_components' VALUES ('100', 'Auto',  
'option=com_auto', 0, 0, '', 'Auto', 'com_auto', 0,  
'js/ThemeOffice/component.png', 0, '', 1);  
INSERT INTO 'jos_components' VALUES ('101', 'Edit Auto', '', 0, 100,  
'option=com_auto', 'Edit Autos', 'com_auto', 0,  
'js/ThemeOffice/edit.png', 0, '', 1);
```

By inserting these two datasets, you are extending the **Components** menu by one main item (**Auto**) and one sub-item (**Edit Auto**).

Furthermore, you are defining what each of these two menu items requests (`option=com_auto&act=all'`) and you are defining a graphic that will appear next to the menu (`'js/ThemeOffice/component.png'`). The graphics are in the `[PathtoJoomla]/includes/js/ThemeOffice` folder.

Now in the admin you will see a screen similar to this screenshot:

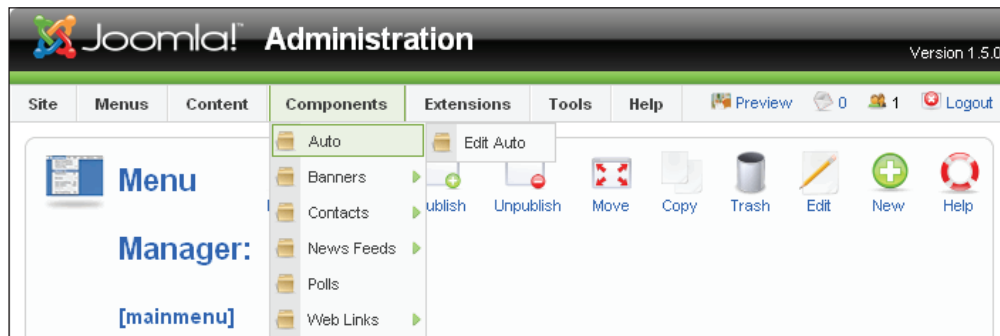


Figure 7.8: Auto Component in the Component Menu

The edit template for the individual items looks like the following screenshot:

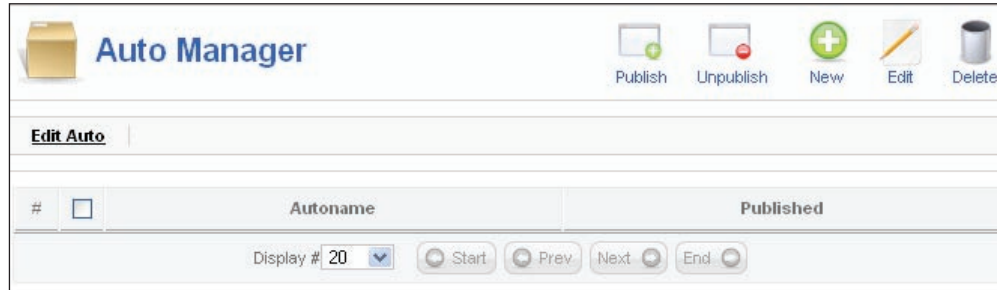


Figure 7.9: Edit Template for the Auto Component

To set up the administration component, you don't just need two, but five files. You first have to set up a subdirectory by the name of `com_auto` in the `[PathtoJoomla]/administration/components/` directory. Create the following five files in this subdirectory:

- `admin.auto.php`
- `auto.class.php`
- `admin.auto.html.php`
- `toolbar.auto.html.php`
- `toolbar.auto.php`

admin.auto.php

This file contains the logic of your component. We are going a step further than in the front end and are putting some of the database queries into the `auto.class.php` file. In principle, the configuration is the same as we have already described in the front end. The file is larger because there are more options, each of which has to be implemented with a function.

In addition, access rights are checked at the start since we are now in the Joomla! administration.

Listing 7.11: admin.auto.php

```
<?php
defined( '_JEXEC' ) or die( 'Direct access to this file is
prohibited.' );
global $mainframe;
// A check is performed here , whether the logged-in user is
// authorized
```

```
// The next two lines will be used in Joomla! 1.5.x, once the user
// rights are extended in the next version.
// $user = & $mainframe->getUser();
// if (!$user->authorize( 'com_auto', 'manage' )){
// The "Old" method
if (!( $acl->acl_check( 'administration', 'edit', 'users',
    $my->usertype, 'components', 'all' ) | $acl->acl_check(
    'administration', 'edit', 'users', $my->usertype, 'components',
    'com_joomlabook' ))) {
    josRedirect( 'index2.php', JText::_('ALERTNOTAUTH' ) );
}
// Importing the Admin HTML class
require_once( JApplicationHelper :: getPath('admin_html'));
// Importing the Admin database class
require_once( JApplicationHelper :: getPath('class'));
// Fetch the parameters
$id    = JRequest::getVar( 'id', 0, 'get', 'int' );
$cid   = JRequest::getVar( 'cid', array(0), 'post', 'array' );
if ( !is_array( $cid ) ) {
    $cid = array(0);
}
// Case differentiation
switch ( $task ) {
    case "publish":
        publishAuto( $cid, 1, $option );
        break;
    case "unpublish":
        publishAuto( $cid, 0, $option );
        break;
    case "new":
        editAuto( 0, $option );
        break;
    case "edit":
        editAuto( $cid[0], $option );
        break;
    case "remove":
        removeAuto( $cid, $option );
        break;
    case "save":
        saveAuto( $option );
        break;
    case "cancel":
        cancelAuto( $option );

```

```

        break;
    default:
        showAuto( $option );
        break;
    }
// Publishing the items
function publishAuto( $cid, $publish, $option ) {
    global $mainframe;
    $db    =& $mainframe->getDBO();
    if (count( $cid ) < 1) {
        $action = $publish ? 'publish' : 'unpublish';
        echo "<script> alert('". JText::_( 'Select an item to',
        true ) ." ". $action ."');
        window.history.go(-1);</script>\n";
        exit;
    }
    $cids = implode( ',', $cid );
    //
    $db->setQuery( "UPDATE #__auto SET published=".$publish."
        WHERE id IN ($cids)");
    if (!$db->query()) {
        echo "<script> alert('". $database->getErrorMsg()."');
        window.history.go(-1); </script>\n";
        exit();
    }
    if (count( $cid ) == 1) {
        $row = new josAuto( $database );
        $row->checkin( $cid[0] );
    }
    josRedirect( "index2.php?option=$option" );
}
// Setting up a new item (id = 0)
// or editing item with id = n
function editAuto( $id, $option ) {
    global $mainframe;
    $db    =& $mainframe->getDBO();
    $row = new josAuto( $db );
    $row->load( $id );

    HTML_Auto::editAuto( $row, $option );
}
// Deletion of items
function removeAuto( $cid, $option ) {

```

```

global $mainframe;
$db =& $mainframe->getDBO();
if (!is_array( $cid ) || count( $cid ) < 1) {
    echo "<script> alert('Please select an item to be
    deleted'); window.history.go(-1);</script>\n";
    exit;
}
$cids = implode( ',', $cid );
$db->setQuery( "DELETE FROM #__auto WHERE id IN ($cids)" );
if (!$db->query()) {
    echo "<script> alert('".$database->getErrorMsg()."');
    window.history.go(-1); </script>\n";
}
josRedirect( "index2.php?option=$option" );
}
// Saving an item
function saveAuto( $option ) {
    global $mainframe;
    $db =& $mainframe->getDBO();
    $row = new josAuto($db );
    if (!$row->bind( $_POST )) {
        echo "<script> alert('".$row->getError()."');
        window.history.go(-1); </script>\n";
        exit();
    }
    if (!$row->store()) {
        echo "<script> alert('".$row->getError()."');
        window.history.go(-1); </script>\n";
        exit();
    }
    josRedirect( "index2.php?option=$option" );
}
// Aborting the current action
function cancelAuto( $option ) {
    global $mainframe;
    $db =& $mainframe->getDBO();
    $row = new josAuto( $db );
    $row->bind( $_POST );
    $row->checkin();
    josRedirect( "index2.php?option=$option" );
}
// Listing of the items
function showAuto($option) {
    global $mainframe;

```



```

$db =& $mainframe->getDBO();
$limit = $mainframe->getUserStateFromRequest(
"limit", 'limit', $mainframe->getCfg('list_limit') );
$limitstart = $mainframe->getUserStateFromRequest(
"$option.view.limitstart", 'limitstart', 0 );
// Counting items
$db->setQuery( "SELECT count(*) FROM #__auto" );
$total = $db->loadResult();
echo $db->getErrorMsg();
jimport('joomla.presentation.pagination');
$pageNav = new JPagination( $total, $limitstart, $limit );
# Do the main database query
$db->setQuery( "SELECT * FROM #__auto ORDER BY id LIMIT
$pageNav->limitstart,$pageNav->limit" );
$rows = $db->loadObjectList();
if ($db->getErrorNum()) {
    echo $db->stderr();
    return false;
}
HTML_Auto::showAuto( $rows, $pageNav, $option );
}
?>

```

auto.class.php

This is the transfer of a dataset structure into its own class, which is derived from the JTable class. JTable is a class in the /includes/database.php file that makes basic database access methods available:

Listing 7.12: auto.class.php

```

<?php
/* ensuring that this file is called from another file */
defined( '_JEXEC' ) or die( 'Direct access to this file is
                                                                    prohibited.' );

class josAuto extends JTable {
    // INT(11) AUTO_INCREMENT
    var $id=null;
    // TEXT
    var $text=null;
    // TINYINT(1)
    var $published=null;
    function __construct( &$db ) {
        parent::__construct( '#__auto', 'id', $db );
    }
}
?>

```

admin.auto.html.php

The `HTML_auto` class, which refers to the presentation of the data and which has already been described in the front-end component, is in this file. The HTML and the CSS structures are created here on the basis of the data transferred by the `$row` array from `admin.joomlabook.php`.

Listing 7.13: admin.auto.html.php

```
<?php
/* ensuring that this file is being called by another file */
defined( '_JEXEC' ) or die( 'Direct access to this file is
                                prohibited.' );

class HTML_Auto {
    function showAuto( &$rows, &$pageNav, $option ) {
        global $mainframe;
        $user =& $mainframe->getUser();
        mosCommonHTML::loadOverlib();
    }
}

<form action="index2.php" method="post" name="adminForm">
    <table class="adminlist">
        <thead>
            <tr>
                <th width="20" class='title'>#</th>
                <th width="20" class='title'><input type="checkbox"
                    name="toggle" value=""
                    onclick="checkAll(<?php echo count($rows);
                                ?>);" /></th>
                <th class='title'>Autoname</th>
                <th class='title'><?php echo JText::_('Published' ); ?>
                </th>
            </tr>
        </thead>
        <?php
        $k = 0;
        for ($i=0, $n=count($rows); $i < $n; $i++) {
            $row = $rows[$i];
            $link = 'index2.php?option=com_joomlabook&task=
                editA&hidemainmenu=1&id='.$row->id;
            $img = $row->published ? 'tick.png' : 'publish_x.png';
            $task = $row->published ? 'unpublish' : 'publish';
            $alt = $row->published ? 'Published' : 'Unpublished';
            $checked = mosCommonHTML::CheckedOutProcessing( $row,
                $i );
        }
    ?>
```

```

<tr class="<?php echo "row$k"; ?>">
  <td align="center">
    <?php echo $i+$pageNav->limitstart+1;?></td>
    <td><input type="checkbox" id="cb<?php echo $i;?>"
      name="cid[]" value="<?php echo $row->id; ?>"
      onclick="isChecked(this.checked);" /></td>
    <td><a href="#edit" onclick="return
      listItemTask('cb<?php echo $i;?>', 'edit')">
      <?php echo $row->text;
      ?></a></td>
    <td width="10%" align="center"><a href="javascript:
      void(0);" onclick="return listItemTask('cb<?php echo
      $i;?>', '<?php echo $task;?>')"></a></td>
  </tr>
<?php
  $k = 1 - $k;
}
?>
<tfoot>
<td colspan="10">
<?php echo $pageNav->getListFooter(); ?>
</td>
</tfoot>
</table>
<input type="hidden" name="option" value="<?php echo
  $option; ?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="boxchecked" value="0" />
<input type="hidden" name="hidemainmenu" value="0">
</form>
<?php
}
function editAuto( &$row, $option ) {
  if ($row->image == '') {
    $row->image = 'blank.png';
  }
  mosCommonHTML::loadOverlib();
  mosMakeHtmlSafe( $row, ENT_QUOTES, 'misc' );
  ?>
<script language="javascript" type="text/javascript">
function submitbutton(pressbutton) {
  var form = document.adminForm;

```

```

    if (pressbutton == "cancel") {
        submitform( pressbutton );
        return;
    }
    // do field validation
    if (form.text.value == '') {
        alert( "Please enter something into this field." );
    } else {
        submitform( pressbutton );
    }
}
</script>
<div class="col60">
    <form action="index2.php" method="post" name="adminForm"
        id="adminForm" class="adminForm">
        <fieldset class="adminform">
            <legend><?php echo JText::_ ( 'Details' ); ?></legend>
            <table class="admintable">
                <tr>
                    <td class="key">Item: </td>
                    <td ><input class="inputbox" type="text" size="50"
                        maxlength="100" name="text" value="<?php echo
                            $row->text; ?>" /></td>
                </tr>
            </table>
            <input type="hidden" name="id" value="<?php echo
                $row->id; ?>" />
            <input type="hidden" name="option" value="<?php echo
                $option; ?>" />
            <input type="hidden" name="task" value="" />
        </fieldset>
    </form>
</div>
<?php
}
}
?>

```

toolbar.auto.php

Depending on the function, this file calls up the various toolbar symbols that represent the logic of the toolbar. The `toolbar.auto.html.php` file with the presentation information is linked.

Listing 7.14: toolbar.auto.php

```
<?php
/* ensuring that this file is called from another file */
defined( '_JEXEC' ) or die( 'Direct access to this file is
                                prohibited.' );

require_once( JApplicationHelper::getPath( 'toolbar_html' ) );
switch($task) {
    case "new":
    case "edit":
        Toolbar_Auto::_Edit();
        break;
    default:
        Toolbar_Auto::_Default();
        break;
}
?>
```

toolbar.auto.html.php

The various toolbars are predefined here. Methods of the JMenuBar static class are called, once for the list view and once again for the edit view.



Figure 7.10: Toolbar in List Mode



Figure 7.11: Toolbar in Edit Mode

Listing 7.15: toolbar.auto.html.php

```
<?php
/* ensuring that this file is called from another file */
defined( '_JEXEC' ) or die( 'Direct access to this file is
                                prohibited.' );

class Toolbar_Auto {
    function _Edit() {
        JMenuBar::title( JText::_ ( 'Edit Auto' ), 'generic.png' );
        jMenuBar::startTable();
        jMenuBar::save();
        jMenuBar::cancel();
        jMenuBar::spacer();
    }
}
```

```

        jMenuBar::endTable();
    }
    function _Default() {
        JMenuBar::title( JText::_( 'Auto Manager' ), 'generic.png' );
        jMenuBar::startTable();
        jMenuBar::publishList();
        jMenuBar::unpublishList();
        jMenuBar::divider();
        jMenuBar::addNew();
        jMenuBar::editList();
        jMenuBar::deleteList();
        jMenuBar::spacer();
        jMenuBar::endTable();
    }
}
?>

```

Test

After you have set up all of the files, you can test your component and then administer the datasets from the Joomla! administration. You can set up new items, delete existing ones, modify them, and publish them. If you mark the items first, you can publish or delete several of them at the same time.

Creating an Installation Package

To create an installation package for your new component, you will need two files in addition to the programming files with the installation and uninstallation texts (and an XML file with the instructions for the Joomla! installer).

Listing 7.16: install.auto.php

```

<?php
function com_install() {
    echo "Thank you for the installation. If you have problems,
    simply contact hagen.graf@gmail.com.";
}
?>

```

Listing 7.17 : uninstall.auto.php

```

<?
function com_uninstall() {
    echo "Thank you for using this component. The component has been
    uninstalled. If you have any questions or comments, please contact
    hagen.graf@gmail.com.";
}
?>

```

auto.xml

Here you are describing your component for the Joomla! installer. You have to enclose all information like metadata, all file names, and the necessary SQL commands (queries) in XML tags. The Joomla! installer imports this file, creates new subdirectories, copies the files to the proper place, and sets up the necessary tables. In other words, everything that you needed to do manually before.

Listing 7.18: auto.xml

```
<?xml version="1.0" ?>
<install version="1.5.0" type="component">
  <name>Auto</name>
  <creationDate>30.07.2006</creationDate>
  <author>Hagen Graf</author>
  <copyright>GNU/GPL</copyright>
  <authorEmail>hagen.graf@gmail.com</authorEmail>
  <authorUrl>http://hagen.take-part.org</authorUrl>
  <version>0.1</version>
  <files>
    <filename>auto.php</filename>
    <filename>auto.html.php</filename>
  </files>
  <install>
    <queries>
      <query>DROP TABLE IF EXISTS 'jos_auto';</query>
      <query>CREATE TABLE 'jos_auto' (
        'id' INT NOT NULL AUTO_INCREMENT,
        'text' TEXT NOT NULL,
        'published' TINYINT(1) NOT NULL,
        PRIMARY KEY ('id') )
    </query>
    </queries>
  </install>
  <uninstall>
    <queries>
      <query>DROP TABLE IF EXISTS 'jos_auto';</query>
    </queries>
  </uninstall>
  <installfile>
    <filename>install.auto.php</filename>
  </installfile>
  <uninstallfile>
    <filename>uninstall.auto.php</filename>
  </uninstallfile>
</installfile>
```

```

<administration>
  <menu>Auto</menu>
  <submenu>
    <menu act="all">Edit items</menu>
  </submenu>
  <files>
    <filename>admin.auto.php</filename>
    <filename>admin.auto.html.php</filename>
    <filename>joomlabook.class.php</filename>
    <filename>toolbar.auto.php</filename>
    <filename>toolbar.auto.html.php</filename>
  </files>
</administration>
</install>

```

In order to create an installation package, you have to copy all of the new files into a directory and compress this directory into a ZIP package by the name of the component, in this case `com_auto.zip`.



Figure 7.12: Files of the com_auto Component

Now you can install this ZIP file with the Joomla! installer as usual, and if you want, you can offer it to others for download. Before you do that in your own installation, use the Joomla! installer to uninstall the version that you set up manually. To do that, click on **Extensions | Install/Uninstall**, check your component, and click on the **Uninstall** icon.

Modules

Modules are a lot simpler. Modules don't usually have a real administration interface, but now and then they have parameters. Modules are all about the presentation on your website and the integration into your template. Modules usually attach to existing components. It is therefore assumed that the particular tables and the content already exist and can be maintained.

You need two files to program your own module. One is for the logic and the presentation, and the other one is an XML file for the Joomla! installer. Both file names start with the label `mod_`.

Source code

Let's take a look at the source code of these files. The `mod_auto.php` file is the control file for the module. The file `auto.functions.php` that contains the functionality of the module is loaded into it. This separation is not absolutely necessary but makes sense with more complex modules.

Listing 7.19: `mod_auto.php`

```
<?php
    defined( '_JEXEC' ) or die( 'Restricted access' );
    jimport( 'joomla.application.controller' );
    require_once (dirname(__FILE__).DS.'auto.functions.php');
    $controller = new JModStatsController( $mainframe );
    $controller->params = &$params;
    $controller->execute( 'display' );
?>
```

In the `auto.functions.php` field, a database query that is dependent on the defined parameter is executed. The presentation of the data is not yet clearly separated from the logic here. Future releases of Joomla! will certainly also introduce views here.

Listing 7.20: `auto.functions.php`

```
<?php
    defined( '_JEXEC' ) or die( 'Restricted access' );
    class JModStatsController extends JController{
        var $params;
        function display(){
            $mainframe = &$this->getApplication();
            $db = &$this->getDBO();
            $anzahl = $this->params->get( 'number' );
            $query = "SELECT *"
                . "\n FROM #__auto"
```

```

        . "\n WHERE published = 1"
        . "\n LIMIT 0,$number";
echo $query;
$db->setQuery( $query );
$rows = $db->loadObjectList();;
echo "<ul>\n";
if ($rows) {
    foreach ($rows as $row) {
        echo " <li>" . $row->text . "</li>\n";
    }
}
echo "</ul>\n";
}
}
?>

```

mod_auto.xml

To be able to install the module, you will need all of the relevant data for the Joomla! installer in an XML file, just as with the component. I have made the description in the description container more detailed in this case.

Listing 7.21: mod_auto.xml

```

<?xml version="1.0" encoding="iso-8859-1"?>
<install type="module" version="1.0.0">
  <name>Auto</name>
  <author>Hagen Graf</author>
  <creationDate>30.07.2006</creationDate>
  <license>http://www.gnu.org/copyleft/gpl.html
  GNU/GPL</license>
  <authorEmail>hagen.graf@gmail.com</authorEmail>
  <authorUrl>http://hagen.take-part.org</authorUrl>
  <version>0.1</version>
  <description><![CDATA[<p>This module displays an adjustable
  number of items from the com_auto component.</p>
  <p>Created by <a
  href="mailto:hagen.graf@gmail.com">hagen.graf@gmail.com</a></p>
  <p>Have fun with this module!</p>
  <p><i><a href="http://hagen.take-part.org"
  target="_blank">http://hagen.take-part.org</a></i></p>
  <p>30.07.2006</p>]]></description>
  <files>
    <filename module="mod_auto">mod_auto.php</filename>
    <filename module="mod_auto">auto.functions.php</filename>
  </files>
</install>

```

```

</files>
<params>
  <param name="number" type="text" default="5"
    label="Number of items" description="Define the number of
    displayed items here " />
</params>
</install>

```

Installation

Copy the `mod_auto.php`, `auto.functions.php`, and `mod_auto.xml` files into a sub-directory and compress this directory into a ZIP package by the name of `mod_auto.zip`. As usual, you can now install this package with the Joomla! installer and if you want to, you can let others download it. Before you do that in your own installation, use the Joomla! installer to uninstall the version that you set up manually. To do that, click on **Extensions | Install/Uninstall**, mark your component, and click on the **Uninstall** icon. You can now install your-brand new `mod_auto.zip` package from the **Install** tab:

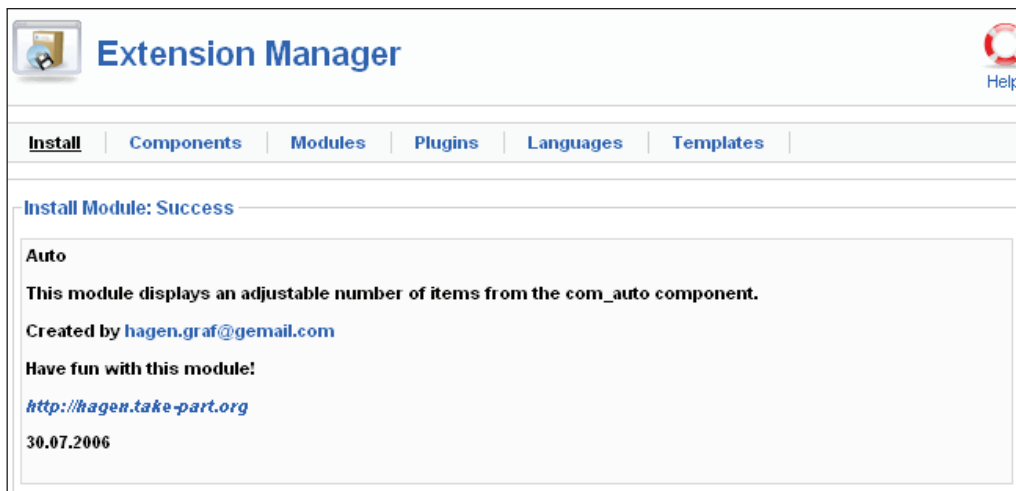


Figure 7.13: Installation Message of the Auto Module

There is a more extensive description this time than with the last installation. Make a habit of giving your users information about the next steps and/or links with additional documentation in this description.

Now you have to activate the module in the **Extensions | Module Manager** menu.
You can also configure the number parameter as well as the other parameters
(position, title, display on certain pages, etc.):

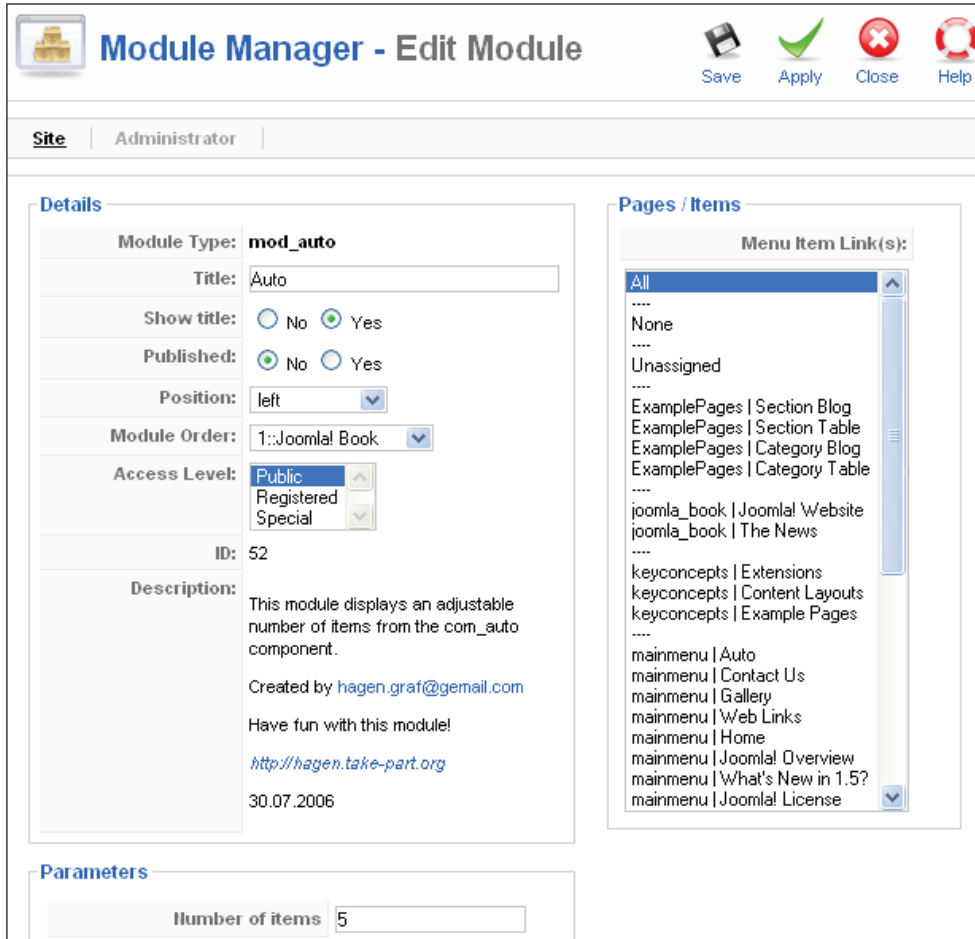


Figure 7.14: Editing the mod_auto Module

View on the Website

You can now see the items from the `jos_auto` table at your selected position:



Figure 7.15: Auto Module on the Website

Plug-Ins

Last but not least, we want to integrate the component into Joomla!'s general search function. You will need a plug-in of the `search` type to make your table searchable. In this case not every plug-in has a subdirectory, but every plug-in type does. For that reason we will be working in the `[PathtoJoomla]/plugins/search` directory.

Source Code

You need at least one PHP file with the logic and the XML file with the description for the plug-in. The names for these should derive from the component, thus `auto.php` and `auto.xml`. If you are writing a user plug-in, for example, you will conform to the function of the plug-in when you are naming it. In this case, there is a concrete relationship with the `com_auto` component. The plug-ins also have to be announced in a table. In this case, the `jos_plugins` table. Here is the respective SQL command:

```
INSERT INTO 'jos_plugins' ('id', 'name', 'element', 'folder', 'access',
'ordering', 'published', 'iscore', 'client_id', 'checked_out',
'checked_out_time', 'params') VALUES (100, 'Search - Autos', 'auto',
'search', 0, 2, 1, 0, 0, 0, '0000-00-00 00:00:00', '');
```

The search function is very extensive and can be supplied with a wide variety of parameters. The source code from will give you an impression of the possibilities.

Since our component doesn't track when an item was added or how often an item has been accessed (we don't have a vehicle detail page yet), a lot of these possibilities will remain unused for now.

Listing 7.22: auto.php plug-in

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );
$mainframe->registerEvent( 'onSearch', 'botSearchWeblinks' );
$mainframe->registerEvent( 'onSearchAreas',
                           'botSearchWeblinksAreas' );

function &botSearchWeblinksAreas() {
    static $areas = array(
        'auto' => 'Autos'
    );
    return $areas;
}

function botSearchWeblinks( $text, $phrase='', $ordering='',
                           $areas=null ){

    global $mainframe;
    $db =& $mainframe->getDBO();
    $user =& $mainframe->getUser();
    if (is_array( $areas )) {
        if (!array_intersect( $areas, array_keys(
            botSearchWeblinksAreas() ) )) {
            return array();
        }
    }
    $plugin =& JPluginHelper::getPlugin('search', 'auto');
    $pluginParams = new JParameter( $plugin->params );
    $limit = $pluginParams->def( 'search_limit', 50 );
    $text = trim( $text );
    if ($text == '') {
        return array();
    }
    $section      = JText::_ ( 'Autos' );
    $wheres       = array();
    switch ($phrase) {
        case 'exact':
            $wheres2 = array();
            $wheres2[] = "LOWER(text) LIKE '%$text%'";
            $wheres2[] = "LOWER(text) LIKE '%$text%'";
            $where = '(' . implode( ' ) OR (', $wheres2 ) . ')';
            break;
        case 'all':
```

```

        case 'any':
        default:
            $words = explode( ' ', $text );
            $wheres = array();
            foreach ( $words as $word ) {
                $wheres2 = array();
                $wheres2[] = "LOWER(text) LIKE '%$word%'";
                $wheres2[] = "LOWER(text) LIKE '%$word%'";
                $wheres[] = implode( ' OR ', $wheres2 );
            }
            $where = '(' . implode( ($phrase == 'all' ?
                ') AND ( ' : ') OR ( ' ), $wheres ) . ')';

            break;
        }
        switch ( $ordering ) {
            case 'newest':
                $order = 'text ASC';
            default:
                $order = 'text ASC';
        }
        $query = "SELECT * FROM #__auto"
        . "\n WHERE ( $where )"
        . "\n AND published = '1'"
        . "\n ORDER BY $order";
        $db->setQuery( $query, 0, $limit );
        $rows = $db->loadObjectList();
        return $rows;
    }
?>

```

The XML code is again there for the installer to organize the files and the metadata.

Listing 7.23: auto.xml

```

<?xml version="1.0" encoding="iso-8859-1"?>
<install version="1.5" type="plugin" group="search">
    <name>Search - Autos</name>
    <author>Hagen Graf</author>
    <creationDate>30.07.2006</creationDate>
    <license>http://www.gnu.org/copyleft/gpl.html GNU/GPL</license>
    <authorEmail>hagen.graf@gmail.com</authorEmail>

```

```
<authorUrl>http://hagen.take-part.org</authorUrl>
<version>1.5</version>
<description>Enables searching of the data in the auto
                                component</description>

<files>
  filename plugin="auto">auto.php</filename>
</files>
<params>
  <param name="search_limit" type="text" size="5"
    default="50" label="Search Limit" description="number of the
    returned search results"/>
</params>
</install>
```

Pack both files into a `plu_auto.zip` ZIP package and install the package using Joomla! installer:

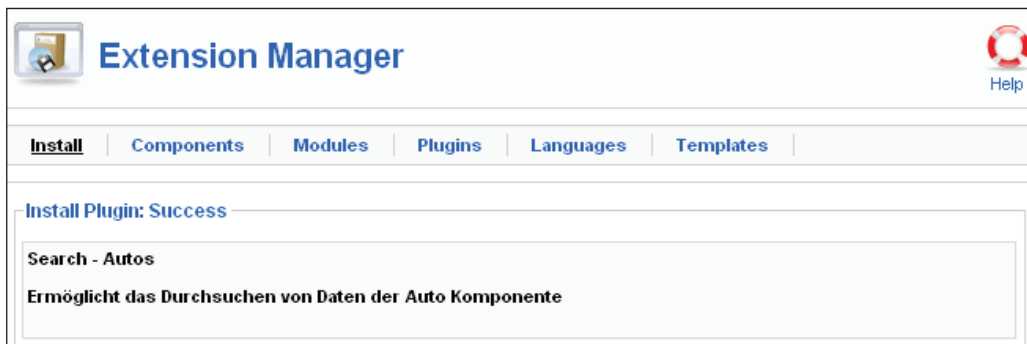


Figure 7.16: Installation Message for the Auto Search Plug-in

View on the Website

After you activate the plug-in in the **Extensions | Plugin Manager** menu, your list is searchable by means of the search field on the website. On entering a search term, the text field in the database is searched and the results are displayed in the general search template.

Your Own Components, Modules, and Plug-Ins

The search plug-in is being kept simple on purpose. A link to an individual view of the list element should be placed at the positions where the search results are found, so that the user doing the search can go there. But since we did not program an individual view into our component, we naturally cannot put the link there:

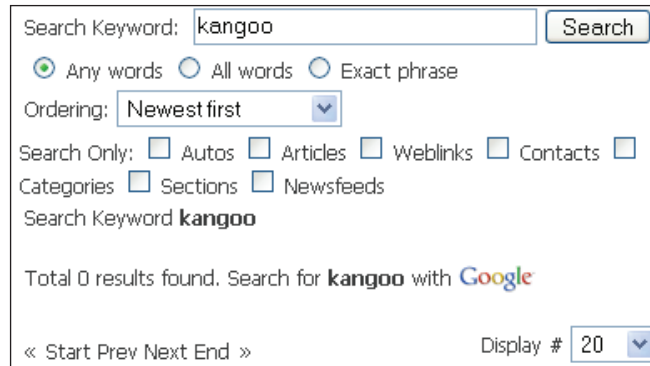


Figure 7.17: The Auto Search Plug-In in Action

Summary

This chapter was written to give you an overview of the creation of components, modules, and plug-ins. You can easily derive further development from comparable components. Our `auto` component, for example, only has one table view. Look for a component with an individual view, as for instance `com_contact`, and extend `auto` with this functionality. The same is true with parameter assignments in modules. Look for a master and create your own module.

Things that look complicated at first will reveal themselves as totally transparent when you look at them again. Have fun exploring!

8

A Website with Joomla!

Now perhaps you have read the entire book up to this point. You have seen dozens of administration pages. You have racked your brain about the connection between web technologies and Joomla! structures. You have heard about all kinds of mnemonics like HTTP, HTML, CSS, SQL, PHP, SEF, div, and others.

But all you wanted was a website! And perhaps you have come to this chapter first because of precisely that reason. It doesn't matter – welcome to a concrete example. This chapter describes the building of a website from idea to realization.

Idea

The site that will be described here is the website of the vintner family Bertrand. Pascal Bertrand, who manages the winery, is the third generation of his family involved in the business. His product offering includes wines from several types of grapes and vintages.

Up until now he has been delivering his wine to a vintners' cooperative and in the summer he sells directly to consumers. Now M. Bertrand would like to sell his wine over the Internet as well and, of course, he wants to do that with the help of Joomla!

The website should:

- Present the Bertrand family business
- Disseminate information about the vineyard and the wine
- Allow online ordering
- Give M. Bertrand the option to promote new products on the website
- Contain a gallery with pictures and videos of the harvesting of grapes and events

A Website with Joomla!

- Contain an internal area where registered users can access special offers and a newsletter
- Offer a way for visitors to contact the vintner
- Allow insertion of Newsfeeds from the wine industry

All of this should be accomplished in two days.

Preparations

Some preparations are necessary to meet all of these demands.

Logo and Appearance

The Bertrand family meets with Ruth Prantz, a friend and designer, to discuss the layout and the content that should be promoted on the website. Ruth asks whether there is an existing logo on printed material. The available printed material includes pamphlets, flyers, letterhead, and a store sign. There is a logo that the grandfather had drawn a while ago:



Figure 8.1: Grandfather Bertrand's Logo

The logo has been used by the family for various purposes over and over again. But there is no consistent appearance. It has been printed in different colors, with different typefaces, and various styles of images and graphics.

Ruth also wants to know what the goals of the website are and what target groups are to be addressed with it.

The Bertrand winery is an ecological company that pays special attention to the quality of the wine it produces and is trendy for the times. The family would like to sell 5 % of their product from the website. Their target group is wine lovers and their friends, age 35 and up.

Ruth suggests that the logo, the colors, and the typefaces should be updated for the present day in order to differentiate it from the competition.

The first draft that is visible within a half an hour looks like this:

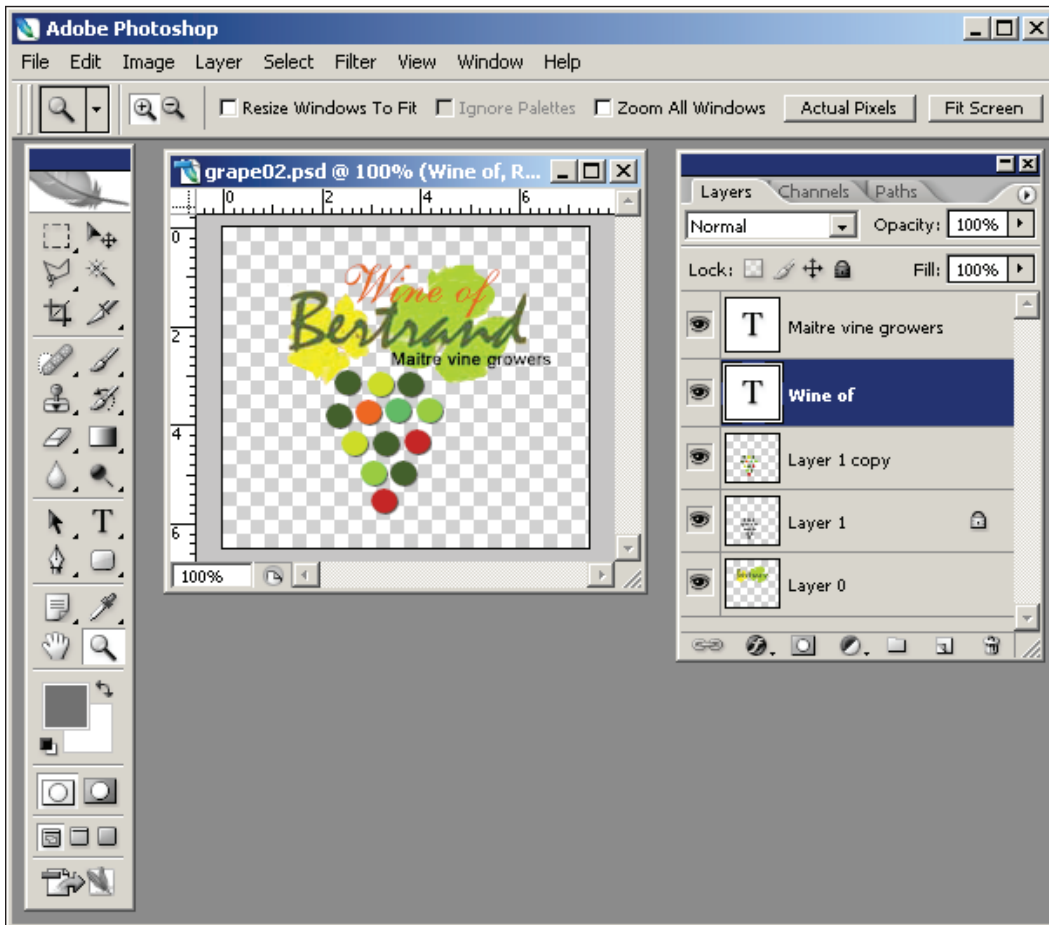


Figure 8.2: First Draft of the Logo

Ruth is using Photoshop Elements (<http://www.adobe.com/products/photoshop/family.html>) to design the logo. Many providers include this program at no charge with a hosting contract. You can, of course, use other graphics programs such as the open-source program Gimp (<http://www.gimp.org/>).

A Website with Joomla!

The family really likes the draft and Ruth bids her farewell. She will base the development of the Joomla! template on this basic draft and these colors. The Bertrand family, in the meantime, will collect materials for the website and define the structure of the Joomla! system.

Photographs

Photos are needed to make an impression of the wine, the area, the vines, the work, and the family. Didier and Marlene, the son and daughter, are given the job of searching through their comprehensive archive of photos and videos for usable material and to also take their cameras to today's wine festival to capture more images and video footage.

Texts

Monsieur Bertrand will personally take care of the text that will appear on the website. He has written the text for countless flyers and also articles for the local press and trade magazines in the past. He also has numerous documents about his land and his wine, which he will scan and offer as PDFs on the site.

Technical Conversion

While the children are busy collecting materials, M. Bertrand is going to start preparing the structure of the website.

Local Installation

He installs a local Joomla! in an xampplite environment as described in Chapter 2. Instead of using the subdirectory `[PathtoJoomla]/Joomla150`, he uses `[PathtoJoomla]/bertrand`.

The connection information for the MySQL server is as follows:

- Hostname: `localhost`
- Username: `root`
- Password: no password (he leaves it empty)
- Name of the database: `bertrand`

He runs the installation without loading sample data since he will create his own content. He stores the particulars in the sixth step of the installation procedures; the screen cues are a little misleading here. The radio button next to the **Install sample data** is active by default. But if you don't click on it, the data is not installed.

The screenshot shows the Joomla! installation configuration window with three radio button options:

- Install default sample data** (selected): *Installing this is strongly recommended for beginners. It will install default sample content that is included in the Joomla! installation package.* Below it is an "Install sample data" button.
- Load local Joomla! 1.5 SQL script**: *The SQL scripts need to be Joomla! 1.5 compatible and should have the appropriate table prefix.* Below it is a text input field, a "Browse..." button, and an "Upload and execute" button.
- Load migration script**: *The migration script needs to be created on the old site by the com_migrator tool or created manually to conform. Enter the table prefix of the old site and enter the encoding used in old site (_ISO setting in language file or as seen in browser info/encoding/source).* Below it are fields for "Old table prefix", "Old site encoding" (set to iso-8859-1), a "Migration Script" field with a "Browse..." button, and an "Upload and execute" button.

Figure 8.3: Joomla! Installation Configuration

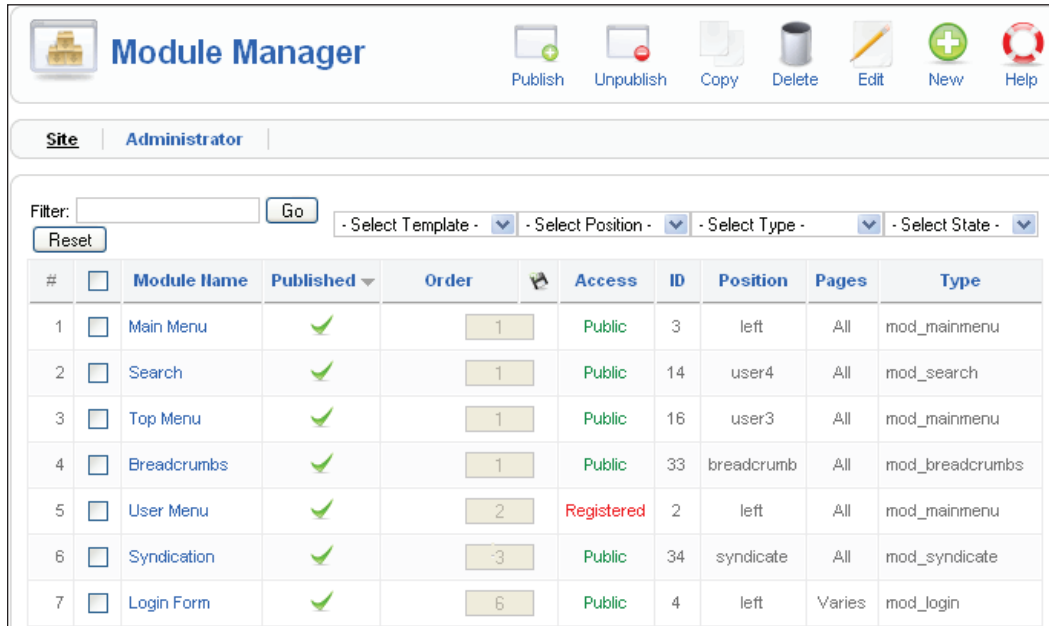
He directly clicks on the **Next** button. The installer creates an empty Joomla! site. He can already see **Wines of Bertrand** in the title bar window:



Figure 8.4: Website without Sample Data

A Website with Joomla!

But the website is not completely empty. M. Bertrand now clicks on the **Extensions | Module Manager** menu in the administration area. He can see 16 activated modules here, many of which he doesn't need right at this time. He deactivates all the modules except these seven: **Breadcrumbs, Main Menu, Login Form, User Menu, Syndication, Top Menu, and Search**. The **User Menu** can only be seen by registered users.



#	<input type="checkbox"/>	Module Name	Published	Order	Access	ID	Position	Pages	Type
1	<input type="checkbox"/>	Main Menu	✓	1	Public	3	left	All	mod_mainmenu
2	<input type="checkbox"/>	Search	✓	1	Public	14	user4	All	mod_search
3	<input type="checkbox"/>	Top Menu	✓	1	Public	16	user3	All	mod_mainmenu
4	<input type="checkbox"/>	Breadcrumbs	✓	1	Public	33	breadcrumb	All	mod_breadcrumbs
5	<input type="checkbox"/>	User Menu	✓	2	Registered	2	left	All	mod_mainmenu
6	<input type="checkbox"/>	Syndication	✓	3	Public	34	syndicate	All	mod_syndicate
7	<input type="checkbox"/>	Login Form	✓	6	Public	4	left	Varies	mod_login

Figure 8.5: Activated Modules

The First Few Articles

Now it is time to work on the structure. M. Bertrand would like to see the news in blog-form on the front page of the website. He first wants to set up three pieces of news to see what it looks like. The news doesn't have to be categorized.

He goes to the **Content | Article Manager | New** menu and sets up three articles. He positions the cursor at the place where he wants the **Readmore** link and clicks on the **Readmore** button below the editor; a red line is inserted. **Section** and **Category** are **uncategorized**, he selects the **Yes** button next to **Frontpage**:

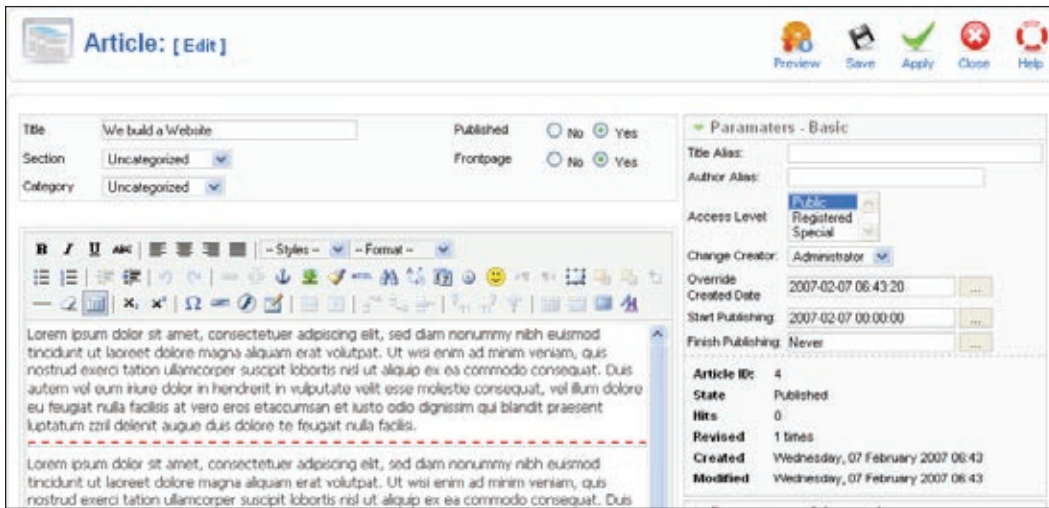


Figure 8.6: Setting Up some News

The three pieces of news (articles) are displayed on the website:

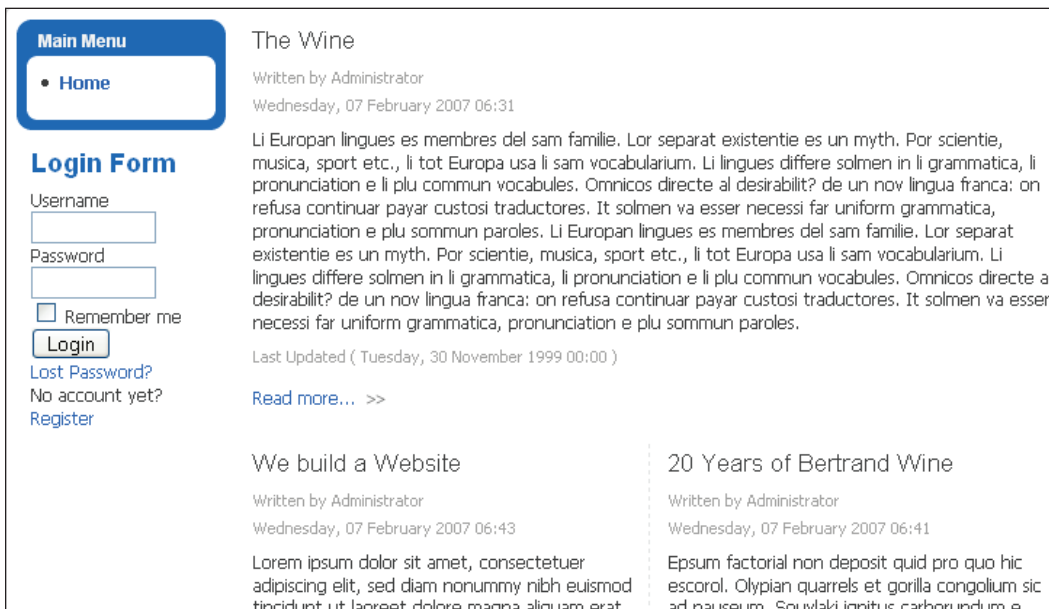


Figure 8.7: Front Page with Three Articles

A Website with Joomla!

M. Bertrand just wants to see the date of creation under the headline of the piece of news. He wants to hide the name of the author and the updated date. Since he will be the only one adding articles to the website for the time being, he changes the settings in the **Author Name** and **Modified Date and Time** fields in the **Site | Configuration | Content** menu:

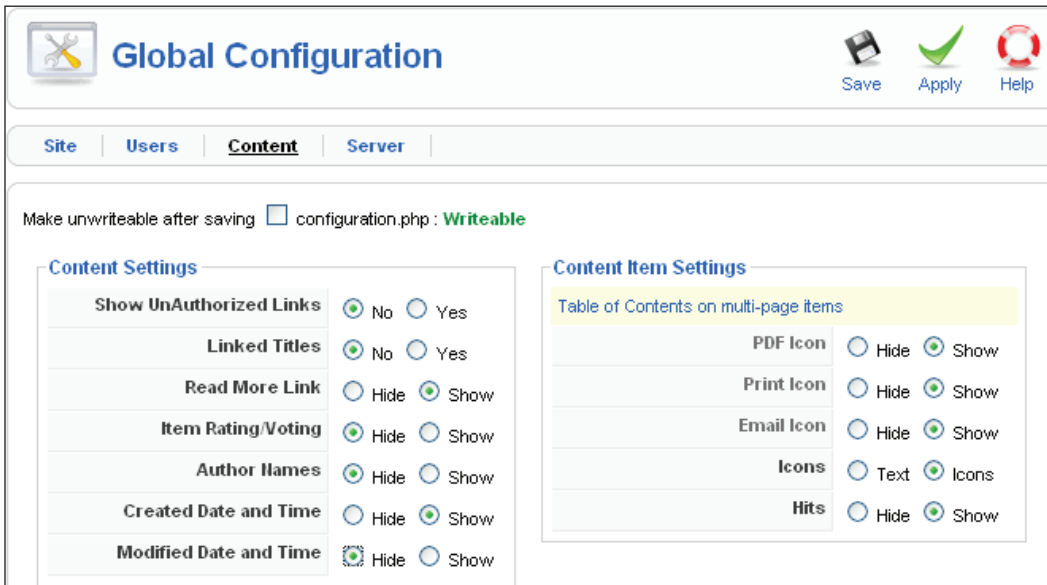


Figure 8.8: Configuration of the Content

The front page now looks the way he wants. Joomla! automatically creates an RSS Newsfeed for these news items and displays it in the Firefox browser, among others, with an orange colored symbol in the URL field. When the user clicks on this symbol, Firefox adds a dynamic bookmark:

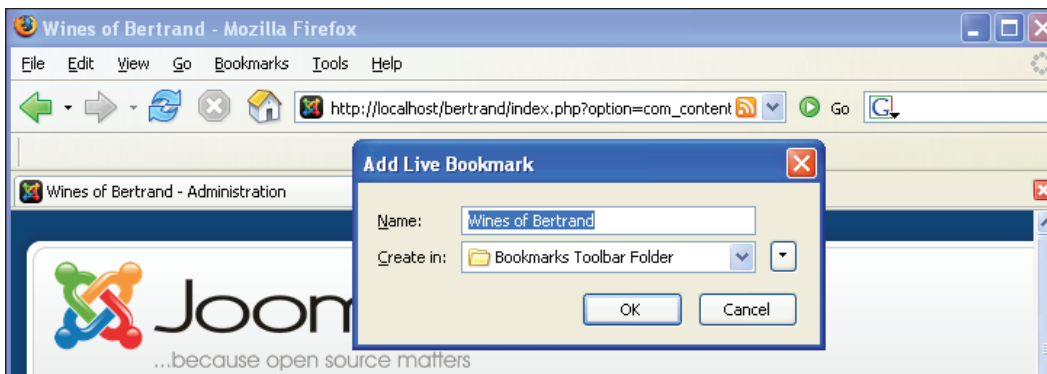


Figure 8.9: Integration of a Newsfeed in Firefox

As soon as M. Bertrand writes a new article for the front page, it is immediately displayed in his browser. In addition to Firefox, any other feed-reader programs can be used to read these feeds. In the meantime, this concept has become even more popular with mobile terminals such as telephones and PDAs. M. Bertrand hopes that a lot of customers will subscribe to his news service and will therefore become familiar with his product offering.

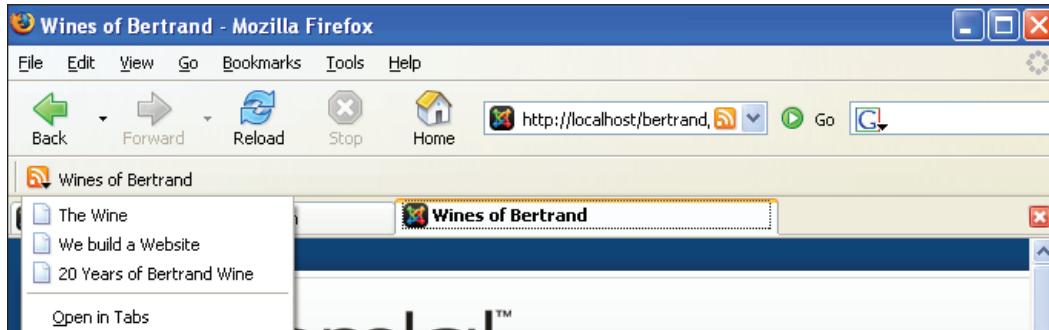


Figure 8.10: The Articles as Feed in the Bookmark Symbol Bar

Masthead

A masthead contains various pieces of information: a visitor to a website must be able to see who is responsible for the content of a website.

In M. Bertrand's case, his masthead looks like this:

Responsibility for the content of this site:

Fa. Bertrand

Musterstraße 1

12345 Musterhausen

Musterland

E-Mail: bertrand@cocoate.com

Telephone: 0123 4567-89

Since the site is being used for commerce, he also has to enter the sales tax identification number.

He sets up another non-categorized content item from the **Content | Articles | New** menu. This time he does not publish it on the front page. The link to the masthead has to be put into the **Main Menu** now. M. Bertrand goes to the **Menus | Main Menu | New** menu. The **Add New Item** starts up as described in Chapter 4. He selects **Articles** in the first window and then selects **Standard Article Layout**.

A Website with Joomla!

Now he gets the **New Menu Item** screen:

Menu Item Type
Standard Article Layout Change Type
The standard article layout displays a single article.

Menu Item Details

Name:

Link:

Display in:

Parent Item:

Published: No Yes

Menu Item Parameters

Select Article:

Unique Itemid: No Yes

Advanced Parameters

Figure 8.11: Setting Up the Masthead Link

M. Bertrand enters the name of the link (**Masthead**) and now has to select the desired article in **Menu Item Parameters**.

After clicking on the **Select** button, another selection window appears with all of the articles:

Article Selection

Filter:

#	Title	Access	ID	Section	Category	Date
1	Masthead	Public	5			07.02.07
2	We build a Website	Public	4			07.02.07
3	The Wine	Public	2			07.02.07
4	20 Years of Bertrand Wine	Public	3			07.02.07

Display # page 1 of 1

Figure 8.12: Selection of the Article for the Menu Link

M. Bertrand clicks on the **Masthead** article, the selection window closes, and the menu entry is completed. A final click on the **Save** icon makes sure that the link does indeed wind up in the menu:

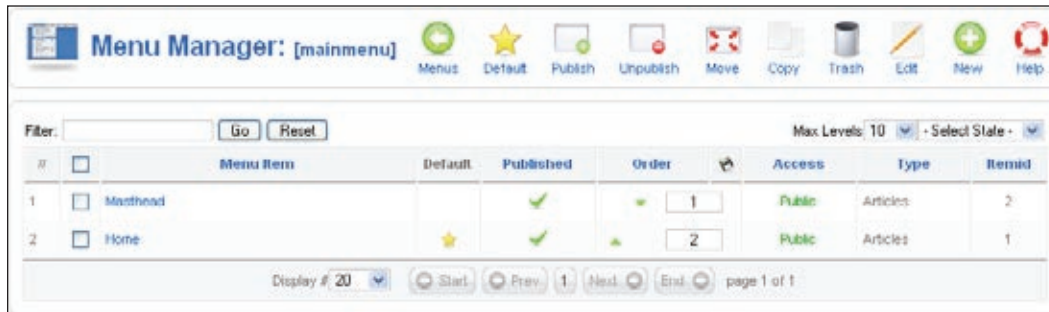


Figure 8.13: The Masthead in the Menu Section

Now it still has to be published by clicking on the red cross. Publishing here refers purely to the menu item, not the article. The link can now be positioned with the green arrow buttons. It now looks like the following figure on the website:

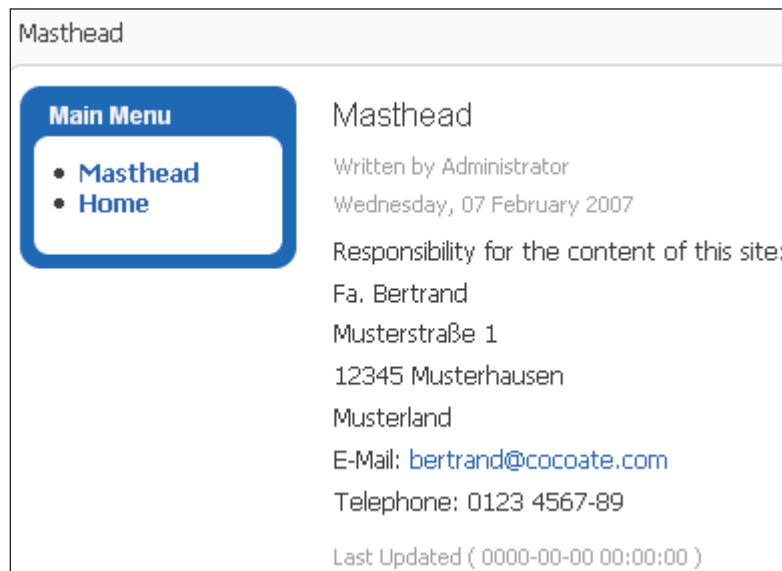


Figure 8.14: The Masthead on the Website

A Website with Joomla!

Now M. Bertrand still wants to hide the creation date and the print, PDF, and email icons. To do that, M. Bertrand calls up the masthead article once more using **Content | Articles Manager | Masthead** and changes the values for PDF, email, and print icons in Advanced Parameters to **Hide**:

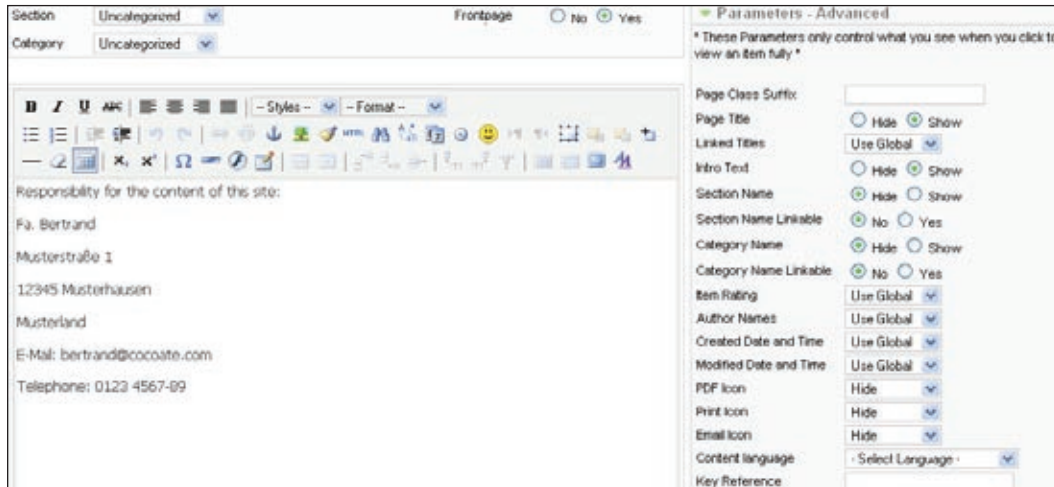


Figure 8.15: Refinements in the Masthead

The masthead is now complete and M. Bertrand starts working on the structure of the menu links for the rest of the pages.

Menu Structure

M. Bertrand wants to have three menus: A main menu, a horizontal top menu, and a user menu for registered users.

Structure of the Main Menu

The entire structure of the website should be shown in the main menu:

Start Page (News)

The wines (general text)

Wine 1 (Description of the product with purchase option)

Wine 2 (Description of the product with purchase option)

Wine 3 (Description of the product with purchase option)

The vineyard (general text)

Vineyard 1 (Description with link to the wine)

Vineyard 2 (Description with link to the wine)

Vineyard 3 (Description with link to the wine)

Impressions (Gallery Component with pictures of the grape harvest)

Offers (special offers in the user section)

Contacts

Masthead

Structure of the Top Menu

The top menu is to give quick access to four core topics.

The wines (general text)

The vineyard (general text)

Impressions (Gallery Component with pictures of the grape harvest)

Contacts

If someone clicks on a link in the top menu, the links that belong to it should also open in the **Main Menu**.

Structure of the User Menu

After registering and successfully logging on via the login module, registered users should have the option to take advantage of special offers. The goal of registration is to build close long-term relationships with customers. A regularly updated page should be displayed with an attractive wine offer and a way to make the purchase. M. Bertrand will make the scanned documents available to registered users within the **Download** link.

User Menu will contain:

Offers

Download Section

Log Out

Setting Up the Texts and the Menu Links in the Main Menu

All of the texts that M. Bertrand sets up in the **Content | Articles Manager** menu, except for the contact form and the gallery, are uncategorized content for the time being. He makes sure that the content is not displayed on the front page. All of the articles should display the PDF, email, and print icons, but not the creation date

A Website with Joomla!

After he has set up all of the articles, he switches to menu link creation. So that **Wine 1**, **Wine 2**, and **Wine 3** end up on the second menu level, the superordinated item **The Wine** must be selected.

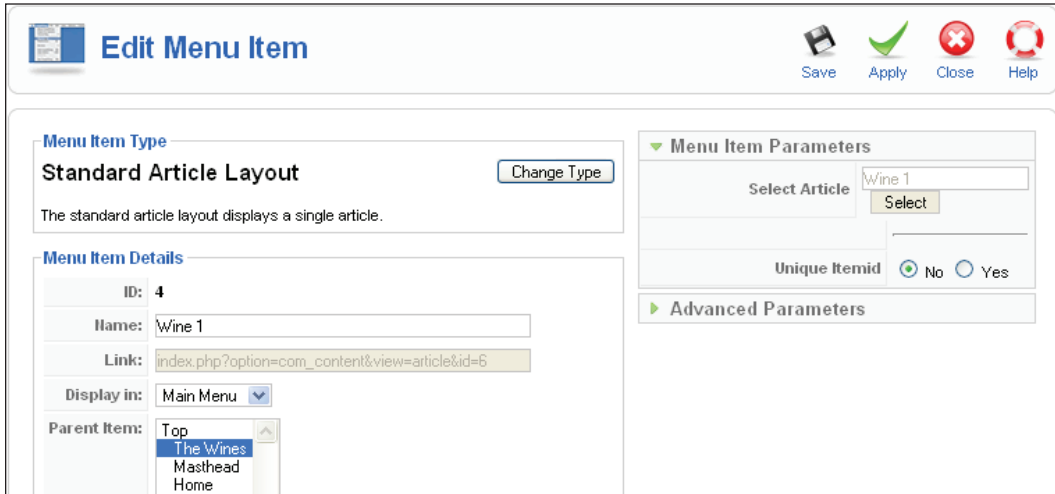


Figure 8.16: Links on the Second Menu Level

There is a tree-like menu in **Menus | Main Menu** that looks like the following screenshot:

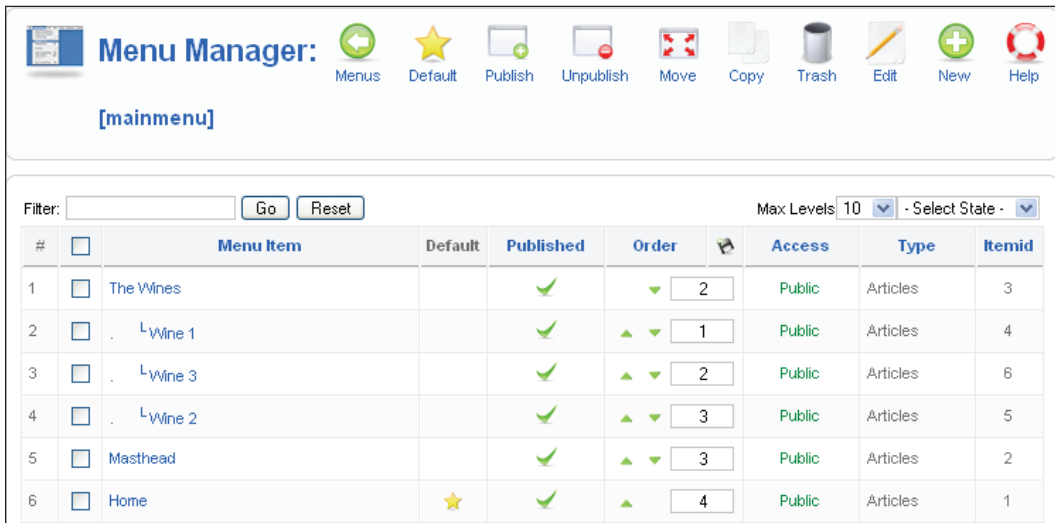


Figure 8.17: Tree-Like Structure of the Menu

M. Bertrand repeats the setting up of the articles for the three vineyards.

Contact Link

M. Bertrand wants to use the contact component that came with Joomla! for his contact form. First he needs a category by the name of *Fa. Bertrand*, which he sets up from the **Components | Contacts | Categories** menu item:

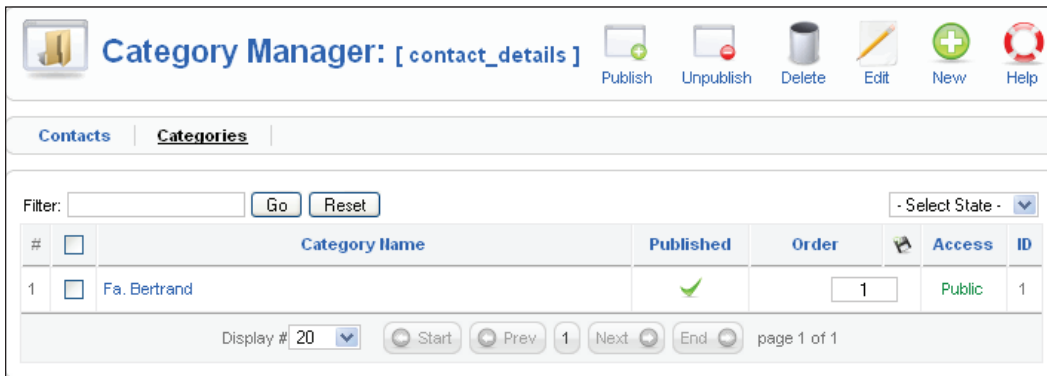


Figure 8.18: New Contact Category

He enters himself as the contact in the contact tab. Over the long term, more contacts will probably be entered here. At the moment, a central form should be sufficient.

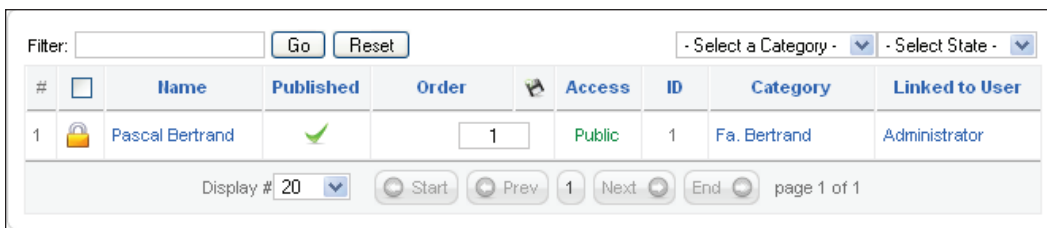


Figure 8.19: New Contact

Now he adds a contact link to the main menu. To do that he clicks on **Menus | Main Menu | New**. The **Add New Menu** starts up. He selects the `contacts` component and clicks on the **Next** button. He also selects the style. Later, if more contacts are added, he can change that assignment to **Category**. After clicking on the category, he lands in the edit template for new menu items. The available contacts are now shown in the parameter section. At the moment, of course, he is the only contact.

A Website with Joomla!

He checks his name and clicks on the **Save** button:

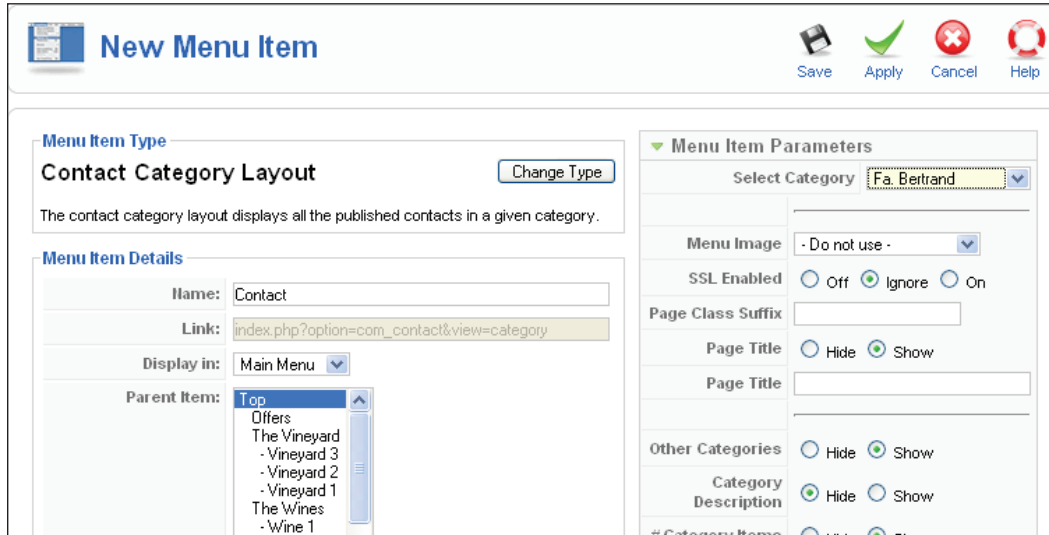


Figure 8.20: Contact Menu Link

M. Bertrand's information, as well as a form that a user can use to type in a question, is now in the contact link on the website.

Top Menu

Except for the integration of the gallery component, the **Main Menu** is complete. Now M. Bertrand creates a **Top Menu**. The top menu already exists in the module section under **Extensions | Module Manager**. It is positioned at user3. M. Bertrand does not even have to change the title, since it isn't usually displayed with a horizontal menu.

He now needs to put a few desired links from the **Main Menu** into the **Top Menu**. For that he creates similar menu items in the Top Menu: The links **The Wine**, **The Vineyards**, and **Contact**.

Instead of creating new links he can also copy the links from Main Menu into the Top Menu using the **Copy** icon:

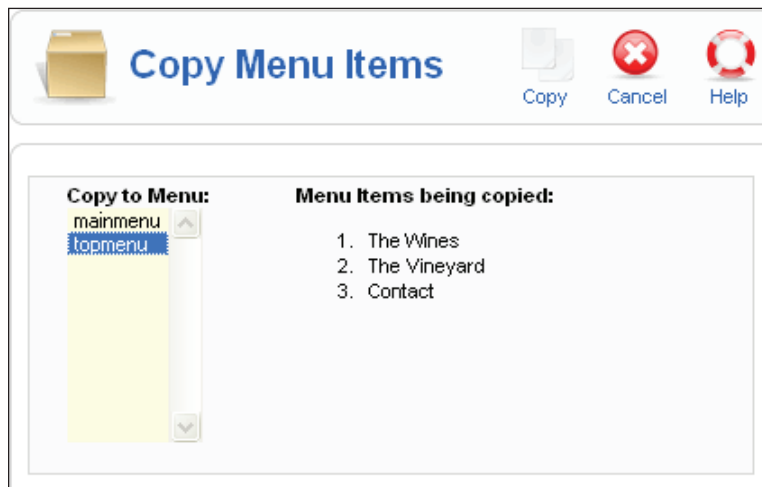


Figure 8.21: Copying Menu Items

He quickly checks the website to see whether the results are there.

Shop

A shopping cart system has to be installed on the website so that the customers have a way to pay online. There is no shop component yet that currently works with Joomla! and M. Bertrand would like to install as few additional components as possible in order to avoid maintenance complications.

Didier suggests integrating a **PayPal** shopping basket. PayPal is the world's biggest online payment system and is owned by eBay. It offers numerous merchant tools, among them a shopping basket system. M. Bertrand signs up for a PayPal user account. Other systems, for instance WorldPay and Google Checkout, offer similar features.

At the moment he is offering three products:

- A white wine at 6 Euro per bottle
- A rosé wine at 8 Euro per bottle
- A red wine at 10 Euro per bottle

PayPal has an option of creating buttons that go with the specific products. When you click on these buttons, the respective item is put into the PayPal shopping basket. Shipping costs and taxes can also be added.

A Website with Joomla!

The following screenshot is of PayPal where a **Buy Now** button has been added:

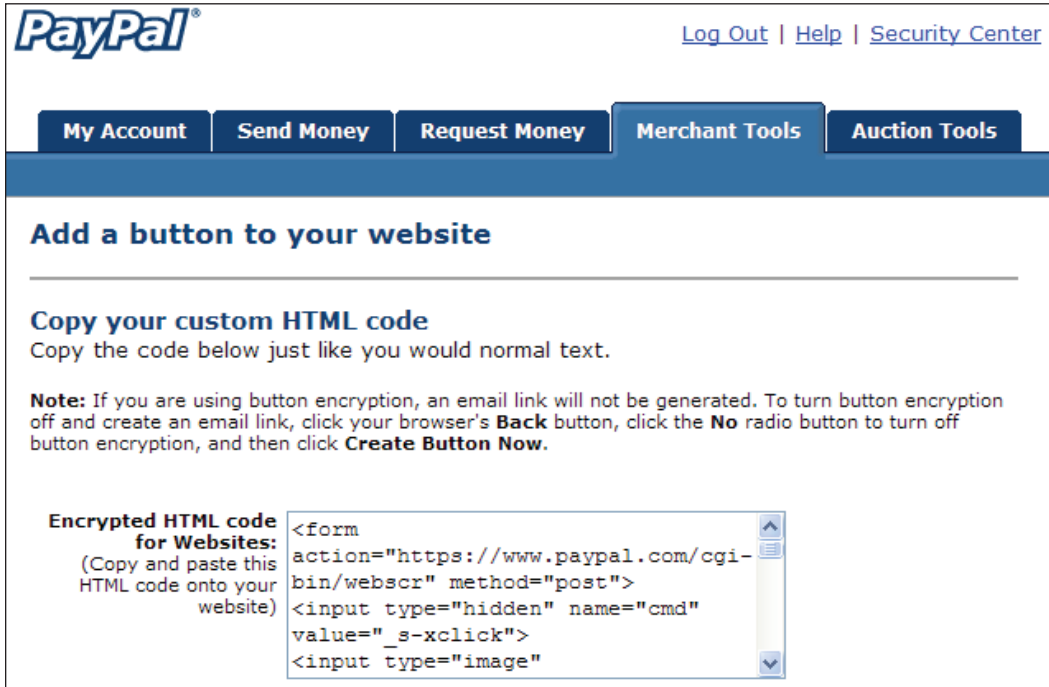


Figure 8.22: Button Code for the Button

M. Bertrand builds this button into the articles **Wine 1**, **Wine 2**, and **Wine 3**. To do that, he opens the article in the edit mode and inserts the PayPal source text into the HTML view of the editor (**Edit HTML Source** button). To make this job a little bit easier, he can disable the editor in the **Site | Configuration** menu. In PayPal's case, M. Bertrand's email address must be put into the source code. The Email Cloaking plug-in that Joomla! installs by default masks this email address so that PayPal cannot find the correct shopping basket. He disables this in the **Extensions | Plugins** menu. The website now looks like the following:



Figure 8.23: Buy Now Button on the Website

If a user now clicks on one of the buttons, a PayPal shopping basket pops up. He or she can put several items into the basket and change the quantities.

Payment For	Quantity	Price
White Wine Item #12	1	\$6.00 USD
Subtotal:		\$6.00 USD
Shipping & Handling:		\$0.00 USD
Total Amount:		\$6.00 USD

Figure 8.24: PayPal Shopping Basket

If the customer has a PayPal account, he or she can use it to pay for his purchase immediately. There is a good chance that people that buy wine online have a PayPal account as the trend is increasing.

The Impressions

M. Bertrand would like to integrate pictures and videos into the website to give his visitors a feeling for the countryside. He installs the Exposé Gallery as described in Chapter 5. He sets up a **Grape Harvest** album with the sub-albums **2005** and **2006**, a **Wine Festival** album with sub-albums **2005** and **2006**, and a **Production** album with a few pictures. His son Didier will add pictures bit by bit.

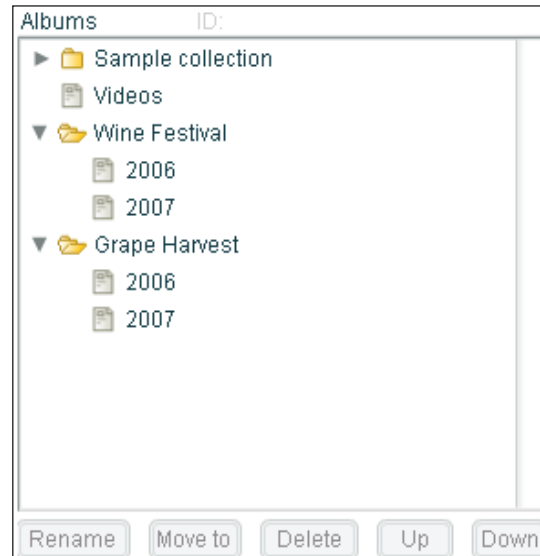


Figure 8.25: Album Manager

A Website with Joomla!

From **Menus | Main Menu | New**, he inserts the link to the Exposé component to the menu. He names it **Impressions** and clicks on the **Apply** button. The impressions appear on the website in the following form:

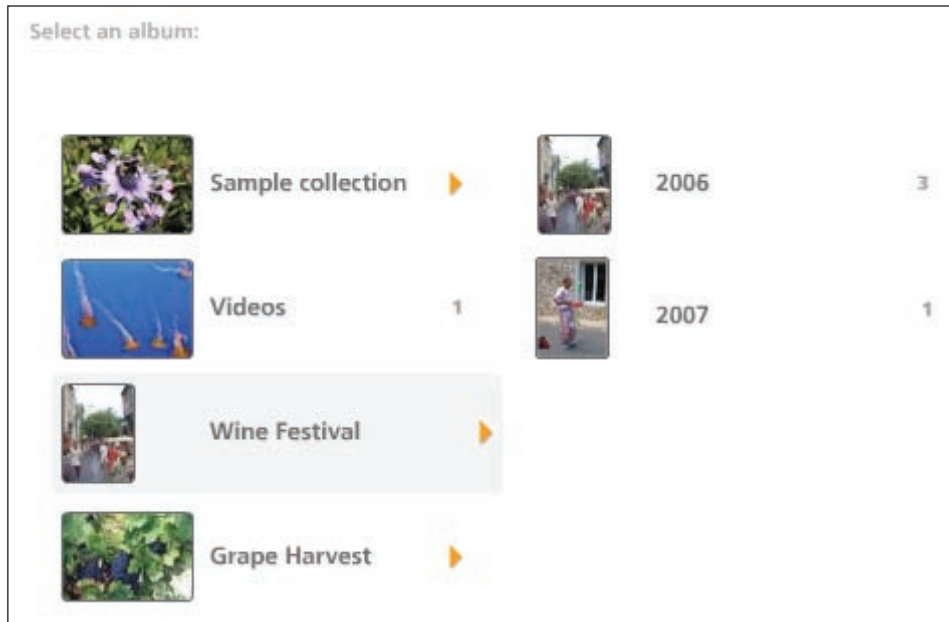


Figure 8.26: Impressions on the Website

The User Section

A page with special offers is to be set up in the section for registered users as well as a download area with PDF articles, certificates, ground surveys, and the like.

The User Menu

M. Bertrand sets up an uncategorized article with the heading of **Offers**. The article is published, not shown on the start page, and only accessible by registered users.

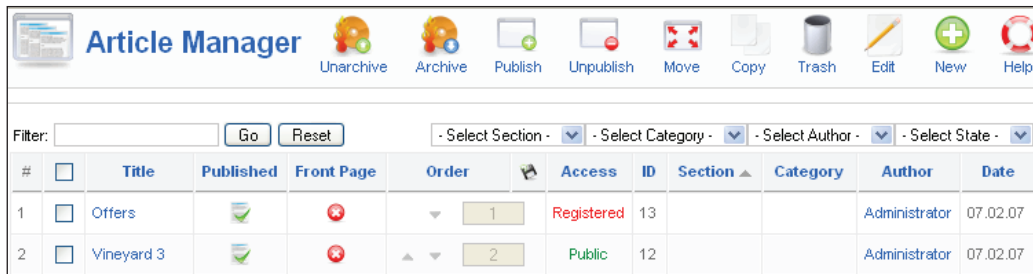


Figure 8.27: Offers Article

He adds the article to the user menu with the **Menus | User Menu | New** menu. He clicks on the **New** icon and selects the **Articles** component. He still has to block the link to the article from the general public by clicking on **Access** in the menu-link edit template.



Figure 8.28: User Menu in the Administration Section

M. Bertrand can now log on to the website; he uses his admin user account and is delighted to see that in addition to the newly created user menu, there is a pencil icon in front of every heading.

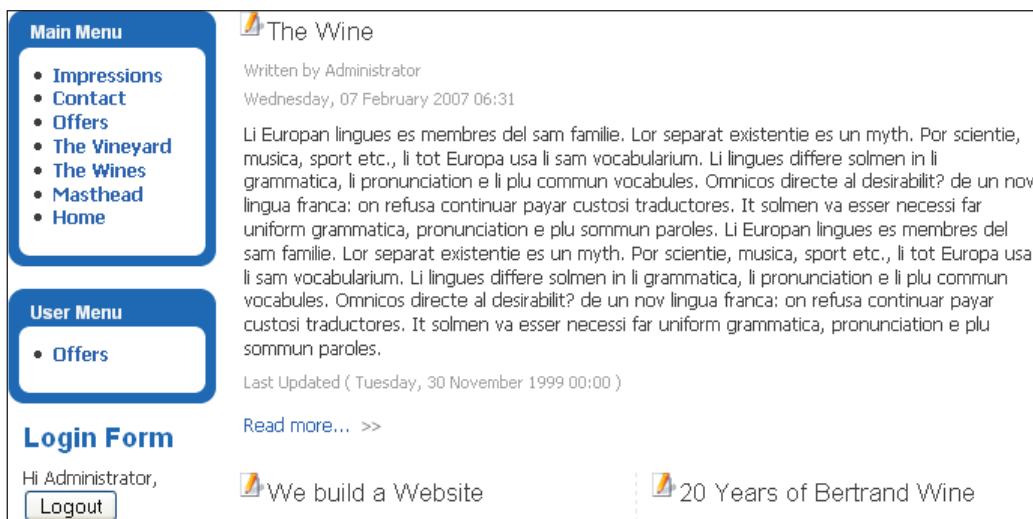


Figure 8.29: User Menu on the Website

A Website with Joomla!

By clicking on one of the pencil icons, he winds up in the edit template of one of the articles:

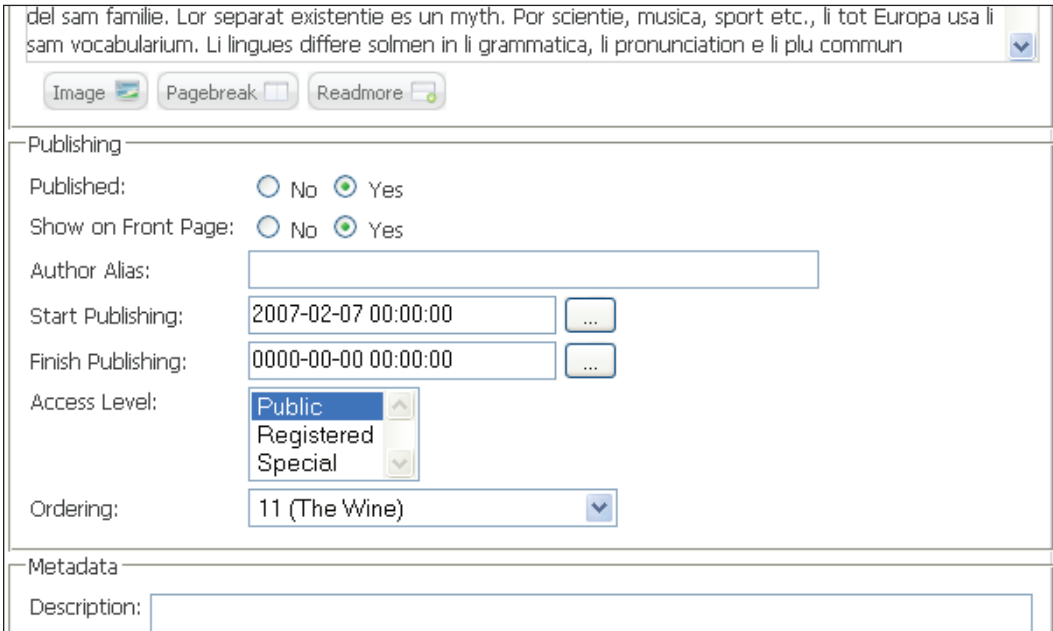


Figure 8.30: Editing an Article on the Website

Here he can change the text of the article, the parameters, and even the metadata. While working on an article, he notices that the **Logout** button is only displayed on the start page. He goes to the **Extensions | Module Manager** menu and changes the view to **All** in the **Pages/Articles** section of the **Login** module:

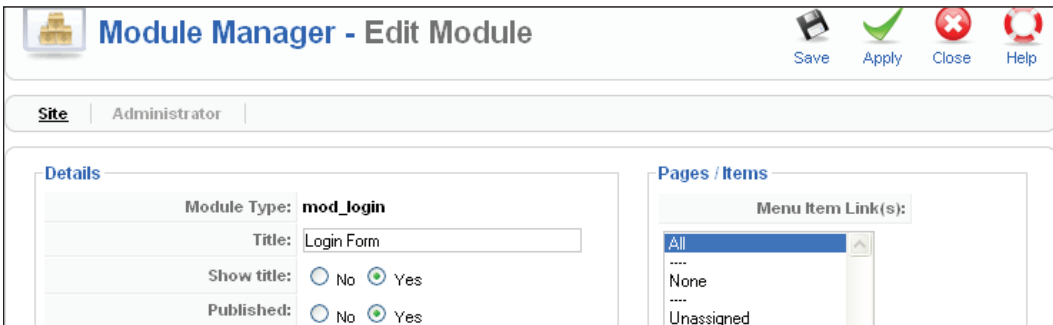


Figure 8.31: Login Module

The Download Section

Finally he would also like to have a download section for his scanned documents. He would like to display small graphics of the certificates in the vineyard descriptions and the comprehensive documents in the registered users section. For that he uses Joomla!'s weblinks component.

He switches over to the **Components | Weblinks | Categories** section and sets up three categories:

- Announcements
- Certificates
- Ground surveys

He adds a small description to each category and later he will also insert a picture. These categories will only be visible to registered users:

The screenshot shows the Joomla! Category Manager interface for the Weblinks component. It features a toolbar with icons for Publish, Unpublish, Delete, Edit, New, and Help. Below the toolbar are tabs for Links and Categories. A filter section includes a text input, Go and Reset buttons, and a state selection dropdown. The main area contains a table with columns for #, Category Name, Published, Order, Access, and ID. The table lists three categories: Ground surveys (ID 4), Certificates (ID 3), and Announcements (ID 2). At the bottom, there are pagination controls showing 'page 1 of 1'.

#	Category Name	Published	Order	Access	ID
1	Ground surveys	✓	1	Registered	4
2	Certificates	✓	2	Registered	3
3	Announcements	✓	3	Registered	2

Figure 8.32: Link Categories

A Website with Joomla!

The PDFs are to be stored in a special directory in the media section (**Site | Media Manager**). He types the name of the directory pdf into the field next to the path and clicks on the **Create Folder** button:

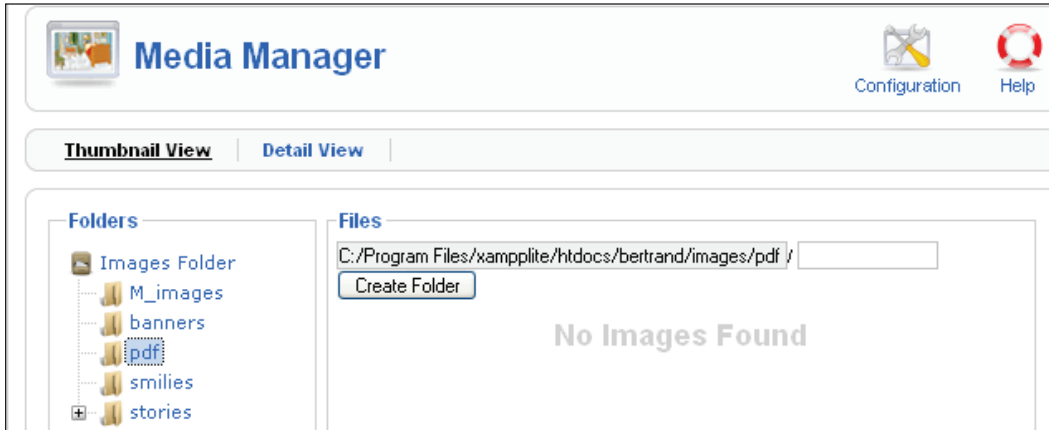


Figure 8.33: PDF Directory

When he attempts to upload his first PDF, he gets the message: **This filetype is not supported**. He clicks on the **Configuration** icon and adds the pdf suffix to the permitted file endings:

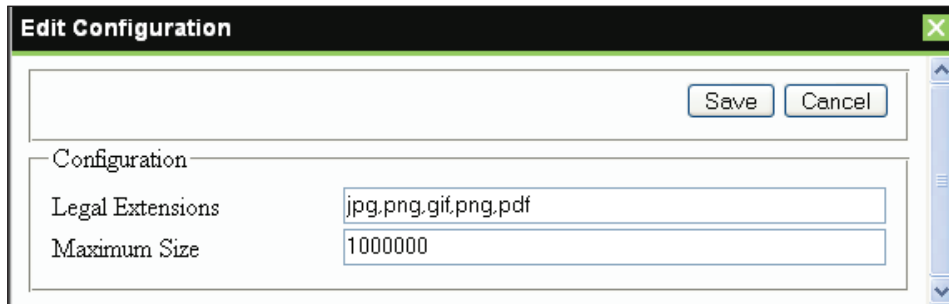


Figure 8.34: Media Configuration

Now he can load all of the PDFs into Joomla! in one go. He could also load all of the files into the local `C:/xampplite/htdocs/bertrand/images/pdf` xampplite directory by means of a file manager or upload them later via FTP to the server.

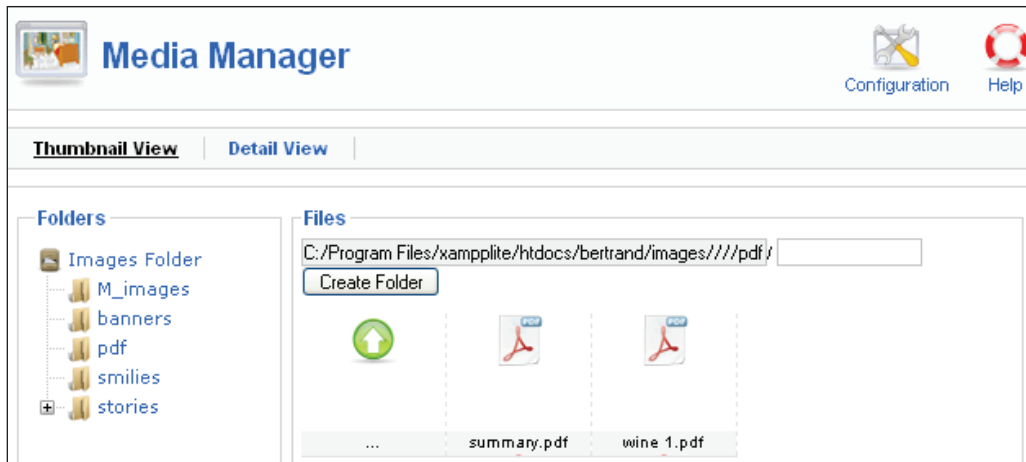


Figure 8.35: Uploaded PDFs

The URLs of the PDFs still have to be entered as links in the **Components | Weblinks | Links** menu. Complete URLs that are valid on the Internet have to be entered in this template. He selects the category as well and with one click defines the target for the PDF to be a **New Window Without Browser Navigation**. Since M. Bertrand is working locally at this time, the URLs refer to `http://localhost/bertrand/`. When M. Bertrand moves this website to his server he will have to modify this for his future domain name; or he could enter the correct domain name at this time.

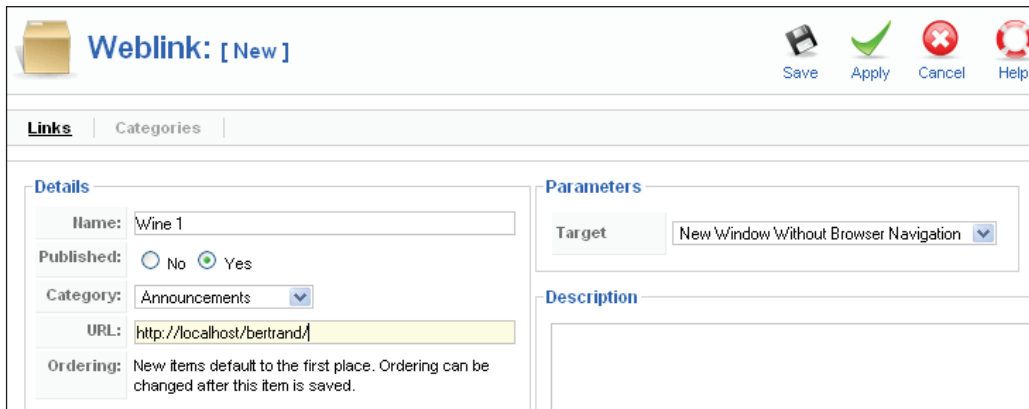


Figure 8.36: Inserting a Web Link

A Website with Joomla!

After a bit of typing, M. Bertrand has added a few links to various PDFs:

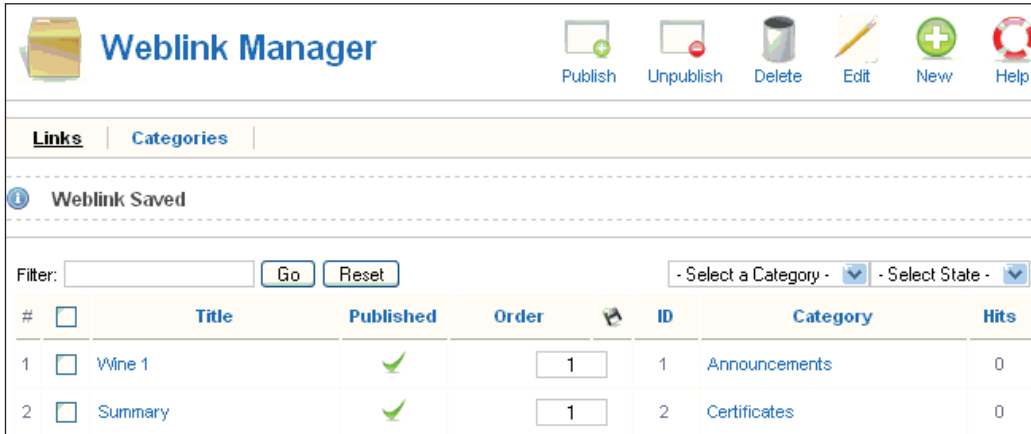


Figure 8.37: Weblinks | Links

This collection of links still has to be put into the user menu. M. Bertrand calls up **Menus | User Menu | New** and selects the **weblinks** component in the **Add New Menu Item**. He still enters a heading for the section (**Download Section**) in the subsequent editing template. Only registered users are to have access. He saves the link and places it below the Offers link. The link appears on the website as planned. The three categories are displayed:



Figure 8.38: Download Section on the Website

After clicking on one of the categories, a table with the PDFs is displayed.

M. Bertrand is satisfied. Even though it isn't a real document administration, it is totally sufficient for his purposes. Even the frequency of the clicks on the individual documents is displayed in the table. The display of the descriptions and the look of the table can also be defined in **Menus | User Menu | Download Section | Parameters**.

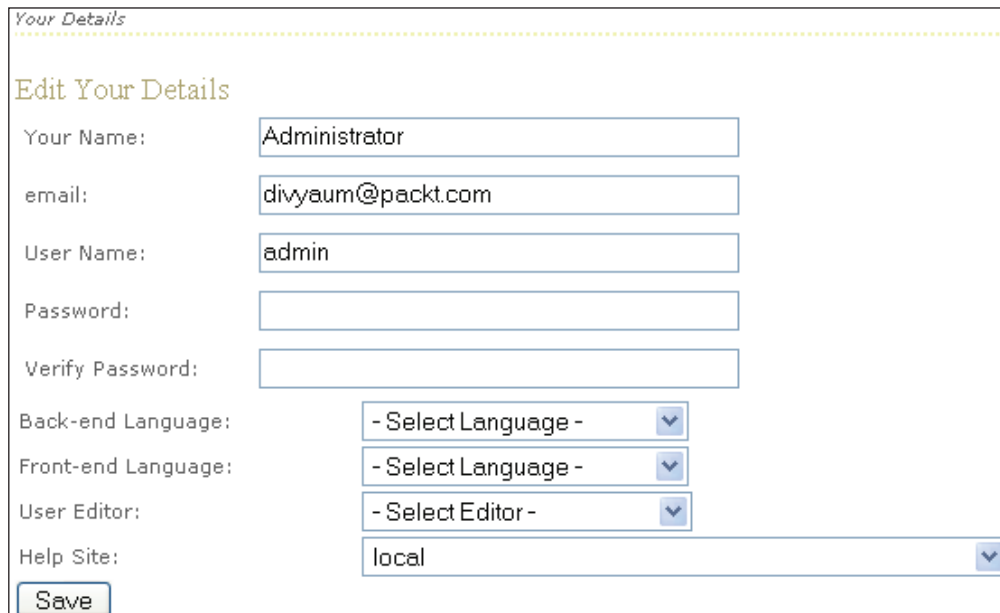
User Details

M. Bertrand remembers that he saw a link in the **Main Menu** in the sample data with which the user could edit his or her information (email address, password, etc.). He also wants that type of link in the **User Menu** of his website.

He creates a new menu item in the top menu named **Your Details**, clicks on **External Links** and puts in the following URL in the **Link** option:

```
index.php?option=com_user&task=edit
```

Now the user section contains the appropriate link and every user can edit his or her data:



Your Details

Edit Your Details

Your Name:

email:

User Name:

Password:

Verify Password:

Back-end Language:

Front-end Language:

User Editor:

Help Site:

Figure 8.39: Editing the User Account

A Website with Joomla!

First Results

M. Bertrand is positively surprised by how quickly it all went. He calls Ruth and tells her about the progress of his work. Ruth sounds a little distraught since the templates she had built until now were all for Joomla! 1.0.x and a lot has probably changed. But she is confident that she can show them the template the next day. She already sends a screenshot of her work via email:

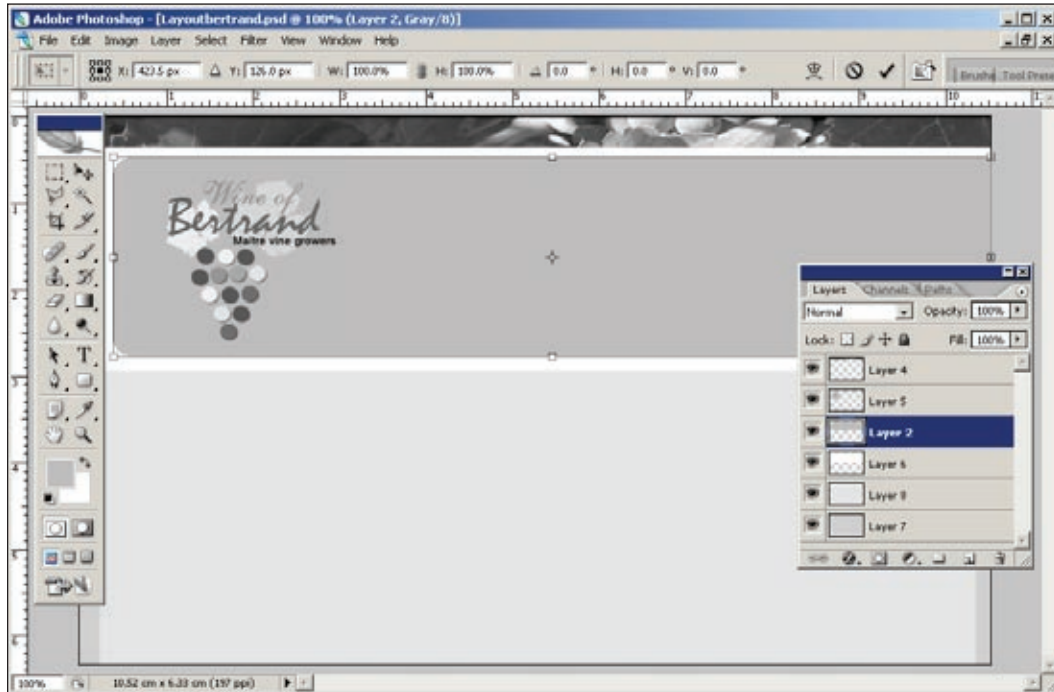


Figure 8.40: First Draft of the Website

When she integrates Joomla! 1.5., Ruth quickly discovers that a lot of the CSS classes have changed in name. To get a good overview of the CSS classes and the rest of the parameters, Ruth uses the Firefox Web Developer (the `web_developer-1.0.2-fx+fl.xpi` file). If you already have the Firefox browser installed on your system, just double-click on the file name to install it.

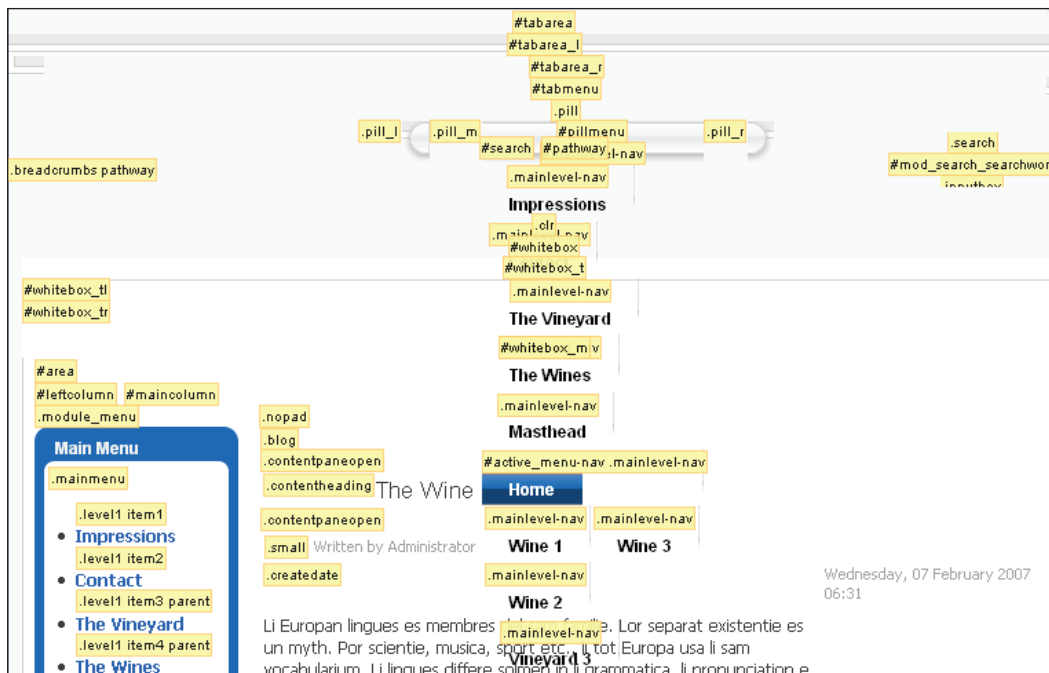


Figure 8.41: CSS Classes Info for Firefox Web Developer

Ruth and M. Bertrand agree that M. Bertrand can already install the website on his server. She will add the template later.

M. Bertrand Learns CSS

M. Bertrand is already pretty good at HTML and would love to know how his template works.

An extended conversation ...

Ruth explains to him that the look of the template is determined by two CSS files, which tell the browser how to display the website. For instance, whether the background is to be red or brown, whether an image is to be inserted, and many other things.

These instructions are written with cascading stylesheet commands and deposited in the two CSS files, `template.css` and `joomla_stuff.css`. Ruth has optimized these files for the most common browsers Internet Explorer 6.0 and Firefox 1.5 for Windows as well as Firefox 1.06 and Safari 2.03 for Mac OS X 10.4.

A Website with Joomla!

M. Bertrand wants to learn to understand CSS so that he can change or create a template himself. Ruth explains the principle of the Cascading Stylesheets (CSS): The term Cascading means that the instructions of the CSS formatting are read by the browsers from top to bottom, just like a cascade. If one instruction is written twice, the lower one overwrites the one above, in other words the latest one overwrites the earlier ones. CSS instructions can be inserted at the following places: an external CSS file, the head of a file, or directly in the respective HTML tag.

CSS instructions that are issued directly in the respective HTML tag overwrite the formatting from the external CSS files. It is important to know that, since Joomla!'s elements are sometimes issued with CSS formatting and are integrated directly into the HTML tag.

To keep the stylesheet of M. Bertrand's template simple, Ruth has used the CSS hierarchy. For instance, she has defined the font information in the general HTML tags: `body`, `table`, `div`, `p`, and `form`. With this she has defined that all of the content of these elements is to be displayed in the *Verdana* typeface, in a certain font size and in gray.

If this is to be changed at a particular position on the website, then this HTML tag has to be assigned to a class; the `greytable` class for instance, with the properties:

```
{ background-color: #333; }
```

This is integrated into the HTML framework with:

```
<table class="greytable">.
```

That way, individual elements can be controlled with CSS classes.

There are numerous such predefined classes in Joomla! by means of which someone can easily program their own template. Let us, for example, take a look at the site navigation:

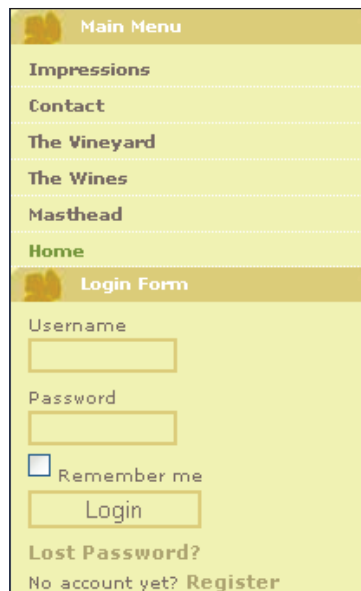


Figure 8.42: Site Navigation View

The site navigation consists of three Joomla! modules: Main Menu, User Menu, and User Login.

Ruth defines the layout of the site navigation with these classes:

Listing 8.1: HTML code of the site navigation

```
...
<table cellpadding="0" cellspacing="0" class="moduletable">
  <tr>
    <th valign="top">Mainmenu</th>
  </tr>
  <tr>
    <td>
      <ul class="mainmenu">
        <li class="level1 item1">
          <a href="http://...">Home</a>
        </li>
        <li class="level1 item2 parent">
          <a href="http://xy">The Wines</a>
        </li>
        <li class="level1 item3 parent">
          <a href="http://xy">The vineyards</a>
        </li>
        <li class="level1 item4">
          <a href="http://xy">Impressions</a>
        </li>
      </ul>
    </td>
  </tr>
</table>
```


A Website with Joomla!

```

        </li>
        <li class="level1 item5">
            <a href="http://xy">Contact</a>
        </li>
        <li class="level1 item6">
            <a href="http://xy ">Masthead</a>
        </li>
    </ul>
</td>
</tr>
</table>
<table cellpadding="0" cellspacing="0" class="moduletable_menu">
    <tr>
        <th valign="top">User Menu</th>
    </tr>
    <tr>
        <td>
            <ul class="usermenu">
                <li class="level1 item1">
                    <a href="http://xy ">Offers</a>
                </li>
            </ul>
        </td>
    </tr>
</table>
<table cellpadding="0" cellspacing="0" class="moduletable">
    <tr>
        <th valign="top">Login</th>
    </tr>
    <tr>
        <td>
            <form>Form fields </form>
        </td>
    </tr>
</table>
...

```

Formatting Step by Step

Schema

By looking at the HTML code, a pattern can easily be recognized.

All of the modules used here are laid out in tables. The tables contain lists and the links are in the lists. This gives us numerous design options because everything can be programmed either directly or with the resident Joomla! classes.

First Ruth assigns properties to the tables that are valid for all menus: Width, distance of the tables from one another, and a color. She only needs to use the two classes `moduletable` and/or `.moduletable_menu` for this.

`.naviside` addresses the position and/or the tag in the HTML framework into which the module is imported. `.naviside` doesn't come with Joomla!, Ruth set it up.

Listing 8.2: CSS Code `.naviside`

```
.naviside table.moduletable, .moduletable_menu {
  width: 210px;           /*width*/
  background-color: #EFF3AA; /*background color*/
  margin-top: 18px;      /*spacing to the top*/
  text-align: left;      /*left-justified*/
}
```

Title of the Module

It is easy to create a title that looks the same for all modules by addressing the headline of the tables directly with the `th` tag.

First you define the class again in the CSS file, here `.menutable`, and then you add the `th`. Ruth set up the title with a background image in order to have more variety in the look of the menus.

Listing 8.3: Module title

```
/*Top Section Module*/
.moduletable th, .moduletable_menu th {
  /*font size/-type*/
  font: 10px/18px Verdana, Arial, sans-serif;
  font-weight: bold; /*weight of the typeface*/
  color: #fff;       /*color of the typeface*/
  /*background image*/
  background: url(..images/page.jpg) top left;
  /*Repetition of background image*/
  background-repeat: no-repeat;
  padding-left: 40px; /*inside spacing left side*/

  height: 19px; ;    /*line height*/
  width: 210px; ;    /*width*/
}
```

Menu Items

After the title is done, the links that are set up in the HTML lists have to be formatted.



Figure 8.43: Menu Items

Ruth did not want to have any bullets before the links and wanted to lose the usual indentation of lists. She used the Joomla! predefined `.mainmenu` and/or `.usermenu` CSS classes to address the list.

```
/*List formatting of the module*/
ul.mainmenu, ul.mainmenu ul, ul.usermenu, ul.usermenu ul{
    list-style: none; /* List without the dot*/
```

Listing 8.4: List formatting of the module

```
display: block; /* block element, */
/* formattable as box */
width: 210px; /* width */
margin: 0px; /* external spaces */
border: 0px; /* border */
padding: 0px; /* internal spaces */
}
```

The subheadings of the links can be addressed separately, in this example they are drawn in on the left border.

Listing 8.5: Subheadings

```
/*List formatting of the subheadings*/
ul.mainmenu ul, ul.usermenu ul {
    /* The external space acts like a drawing in of the text */
    margin-left: 10px;
    background-color: #E7DC9E; /* other background color */
    width: 200px; /* smaller width */
}
```

Divider lines

Ruth wanted to show the individual links separated by lines. For this, she addresses the `li` tags of the lists in a specific way:

```
ul.mainmenu li, ul.usermenu li {
    border-bottom: 1px dotted #fff; /* Bottom border */
}
```

She could have formatted every individual list item separately, since every `li` tag has its own class: `level1 item1`, `.level1 item2 parent`, `.level1 item3 parent`, `.level1 item4`.

The actual links

Now coming to the links themselves. The formatting addresses the `a` tag, which is in the `ul` list with the `.mainmenu` class.

All links should look like this:

```
ul.mainmenu a, ul.usermenu a {
    text-decoration: none; /* Link without underline */
    color: #6B5E58; /* color of the font */
    font-size: 10px; /* font size */
    padding-left: 10px; /* distance of the text to the border */
}
```

The `hover` effect changes the background or the font color of the link when the mouse cursor rolls over it. Ruth has set it up in the following way:

```
ul.mainmenu a:hover, ul.usermenu a:hover {
    color: #6A8A0A;
    text-decoration: none;
}
```

And finally, the link of the currently active page is specially highlighted with the help of the following class:

```
.mainmenu li#current a, .usermenu li#current a {
    color: #6A8A0A; /* font color */
}
```

M. Bertrand is delighted and now finds formatting with CSS a lot easier after this explanation. His head is still spinning a little, but Ruth also gives him the www.w3schools.com/css/ link. That site has a good description of all of the attributes that can be assigned with CSS.

Changes for Joomla! Version 1.5

If you have already built templates with Joomla! 1.0.x, here is the good news:
Joomla! 1.5 is backward compatible!

There are, however, a few pitfalls you should be aware of. The previously used `mosload()` commands in the PHP as well as the old CSS classes of the `index.php` of a template don't work at certain places without modification, since formatting has changed.

Changes to the Menus

In the Main Menu and/or Top Menu (horizontal), the `mainlevel-nav` CSS identifier for the formatting of the links in the top menu is still there, but it is now programmed as a list (`ul`), while previously it was output as table span tags. An existing CSS file has to be rewritten to allow for this.

In the Site Menus, what was valid for the top menus is also valid here: If you are using an older template, you have to rewrite the formatting of the individual table cells for the new list items (`li`).

Floats

Some elements are output with `Float`, and therefore `Clear` has to be inserted at the appropriate places.

Installation on the Webserver

M. Bertrand's local website is now approximately what he had imagined. He has been looking for a provider for his website in the last few weeks and after a long search has become a customer of a small company in the neighborhood. This company offered M. Bertrand an SSL-protected administration interface by the name of Plesk for the setup of his web server, the database, and the email configurations, etc. PHP runs in Safe Mode, and the database can be administered with an installed phpMyAdmin interface.

M. Bertrand wanted to have web-based administration of his server, because he did not want to have to learn tedious configuration files.



There are numerous other interfaces, such as Confixx, Visas, Webmin. Many providers, like 1&1, Strato, Hetzner, and others use their own developments. We will discuss Plesk briefly as an example; other interfaces work similarly.

How to Do the Installation?

M. Bertrand used the Joomla! web installer to install the program in his local environment. Subsequently he spent a morning entering data, and he doesn't want to lose these with the live installation. There are at least two options:

- He could load all of the Joomla! files from FTP to the server at his provider, change the configuration file manually, and import the database with phpMyAdmin.
- He could load a Joomla! file to the server at his provider from FTP, install it with the Joomla! installer, and then import the data of the database with phpMyAdmin.

M. Bertrand decides on the second option since he doesn't trust himself to change things in the configuration file. In order to be able to start with the installation, he needs access data for FTP, MySQL, and the domain name of course. He finds the following access data in his administration interface.

FTP

Host: `bertrand.cocoate.com`

User: `fusfusfus`

Password: `pwpppw`

MySQL

Host: `localhost`

User: `dbusdbus`

Password: `pwppwppw`

Database: `dbdbdb`

Installation of Joomla! at the Provider

M. Bertrand starts his FTP program and enters the necessary information. He received the FTP program from his provider free of charge (you can find a free-ware FTP program at www.filezilla.sourceforge.net/).

He transfers his existing installation from the `c:\xampplite\htdocs\bertrand` subdirectory using FTP into the `htdocs` subdirectory on his provider's server. He then deletes the `configuration.php` file on the server. He creates an empty `configuration.php` file on his Windows computer somewhere else than in his

A Website with Joomla!

Joomla! directory. He uploads this file using FTP into the `/httpdocs` on the server and gives it `chmod 0777` rights.

If the `configuration.php` file is empty, Joomla! assumes that the installer is needed and starts it automatically. The installation works exactly as described in Chapter 2.

But M. Bertrand is in a Linux environment this time and he has to pay attention to such things as file and directory rights. The `chmod` command is there to set the rights. FTP programs can execute that command and therefore assign access rights to a directory and to files. At minimum, `chmod 755` has to be used for the directories; `chmod 644` is sufficient for the files.

Importing the Data

In order to be able to import the data to the MySQL database of the provider, they first have to be exported from the local xampplite version. M. Bertrand calls `http://localhost/phpmyadmin` from the browser, selects the `bertrand` database, and clicks on the **Export** tab. He clicks on **Check all** to pick all of the tables for export.

In the **Structure** section he selects **Structure** and **DROP TABLE**. **Drop TABLE** inserts an additional SQL command into the export file that makes sure that with future imports of data, existing tables that may possibly have the same names are deleted. This is important in M. Bertrand's case, as all of the tables are already there from the web-installer's installation.

In the data section he selects **Data** and **Extended inserts**. He chooses **Save as file** to get the data in file form.

Subsequently he clicks on the **Go** button and a `bertrand.sql` file is being offered for download.

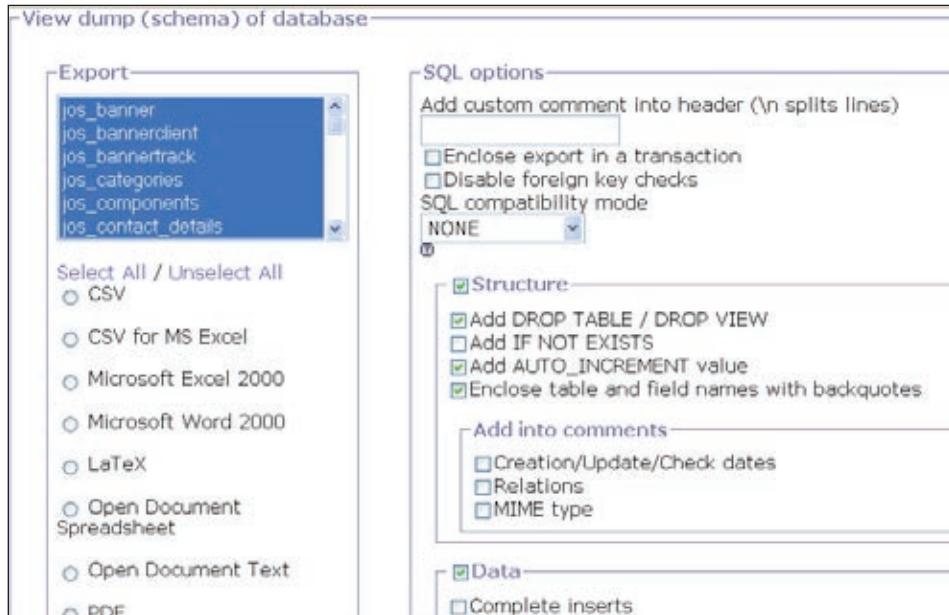


Figure 8.44: Exporting the Data

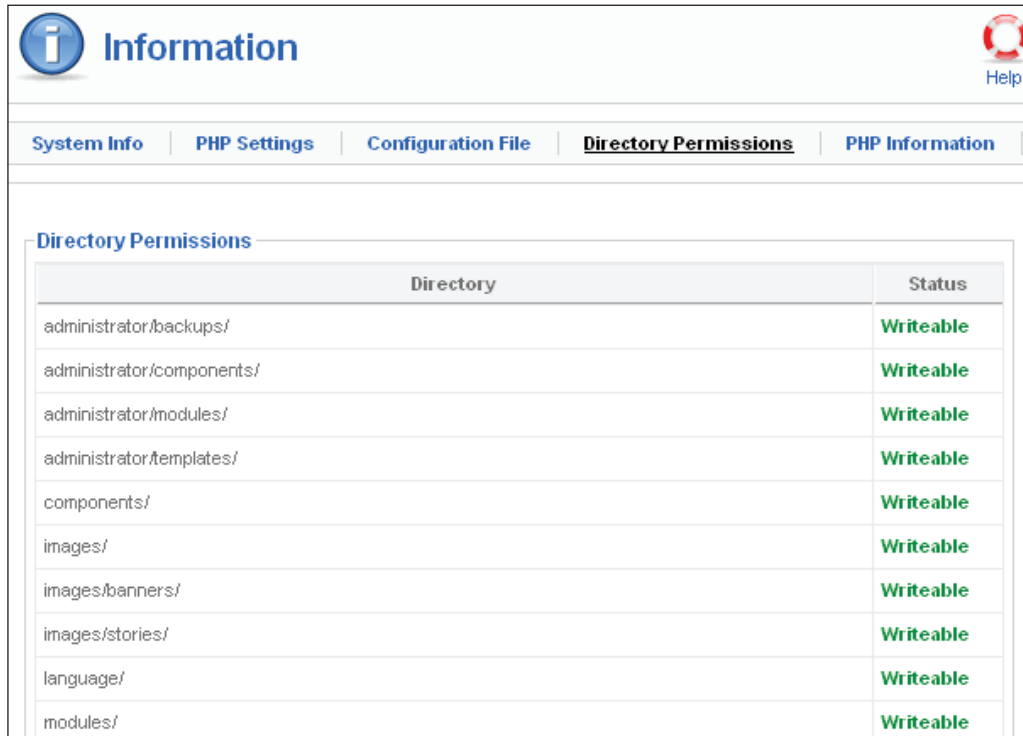
All of the content is in this file. He saves the file somewhere outside the Joomla! directory. The file contains SQL commands that exactly represent the status of the local installation, and when executed sequentially will recreate this exact status on another MySQL system.

This is exactly what M. Bertrand wants, so he opens phpMyAdmin on his provider's server. There he clicks on the SQL tab in his database (which is also called `bertrand`). In the template he clicks on the **Search** button and selects the local `bertrand.sql` file. One click on the **Go** button and the file is uploaded, the SQL commands are executed sequentially, and his website on the server is now a copy of the local version (as covered in Appendix).

A Website with Joomla!

File and Directory Rights

To ensure a normal operation of Joomla!, the directory rights have to be set. M. Bertrand checks the settings in the **Help | System Info** menu.



The screenshot shows the Joomla! administration interface. At the top, there is an 'Information' header with a help icon and a 'Help' link. Below this is a navigation menu with tabs for 'System Info', 'PHP Settings', 'Configuration File', 'Directory Permissions', and 'PHP Information'. The 'Directory Permissions' tab is selected. The main content area displays a table titled 'Directory Permissions' with two columns: 'Directory' and 'Status'. All listed directories have a 'Writeable' status.

Directory	Status
administrator/backups/	Writeable
administrator/components/	Writeable
administrator/modules/	Writeable
administrator/templates/	Writeable
components/	Writeable
images/	Writeable
images/banners/	Writeable
images/stories/	Writeable
language/	Writeable
modules/	Writeable

Figure 8.45: Directory Rights

Linux has users and user groups. M. Bertrand has a user and the Apache web server also has a user. Depending on configuration, they can both be in one user group or in separate user groups. If they are in different user groups, the directory rights have to be set to `chmod 0777`. (M. Bertrand also changes the password for the Exposé Gallery in the administration. It was still set to manager.)

Search Engines

There are two approaches for M. Bertrand's website to be listed in search engines.

Search Engine Friendly URLs

Dependent on the provider, search-engine-friendly URLs are one option. M. Bertrand's provider allows these URLs and M. Bertrand activates this feature in the **Site | Configuration | SEO settings** menu. Then he renames the `htaccess.txt` file to `.htaccess`. The Exposé Gallery, for example, now is accessed via the URL `http://bertrand.cocoate.com/component?option,com_expose/Itemid,16/` (see Chapter 5 for more details).

This type of renaming of URLs only works with the Apache web server.

Metatags

M. Bertrand diligently enters a short description and the keywords into his articles (see Chapter 4).

Design

The next morning has come, the wine festival is over, Didier and Marlene have taken a lot of photographs. M. Bertrand has created the website to his satisfaction and has even integrated the PDFs. The PayPal connection works. They all meet for breakfast and Ruth has, in the meantime, finished the template. She brings the finished components on a USB stick and talks about her experiences with the template for an hour.

Installation of the Template

M. Bertrand copies the template files to the `/httpddocs/templates/tmpl_bertrand` and `/httpddocs/templates/tmpl_bertrand_shop` directories using FTP. In the **Extensions | Template Manager** menu he defines the new `tmpl_bertrand` template as the default:

A Website with Joomla!

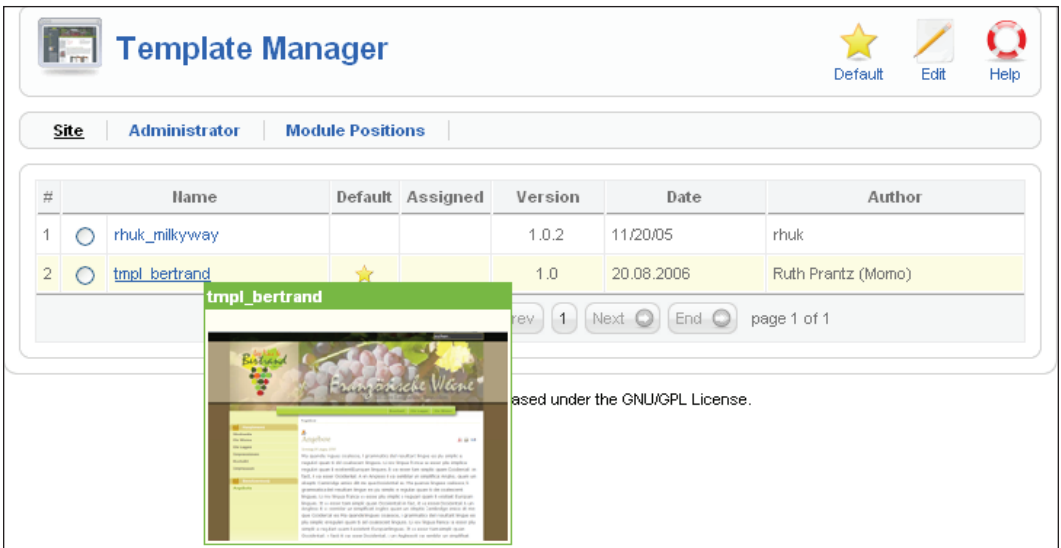


Figure 8.46: New Template in the Template Manager

He assigns the `tmpl_bertrand_shop` template to the wine pages in edit mode. That way the customer will know that he or she is in the shop section.

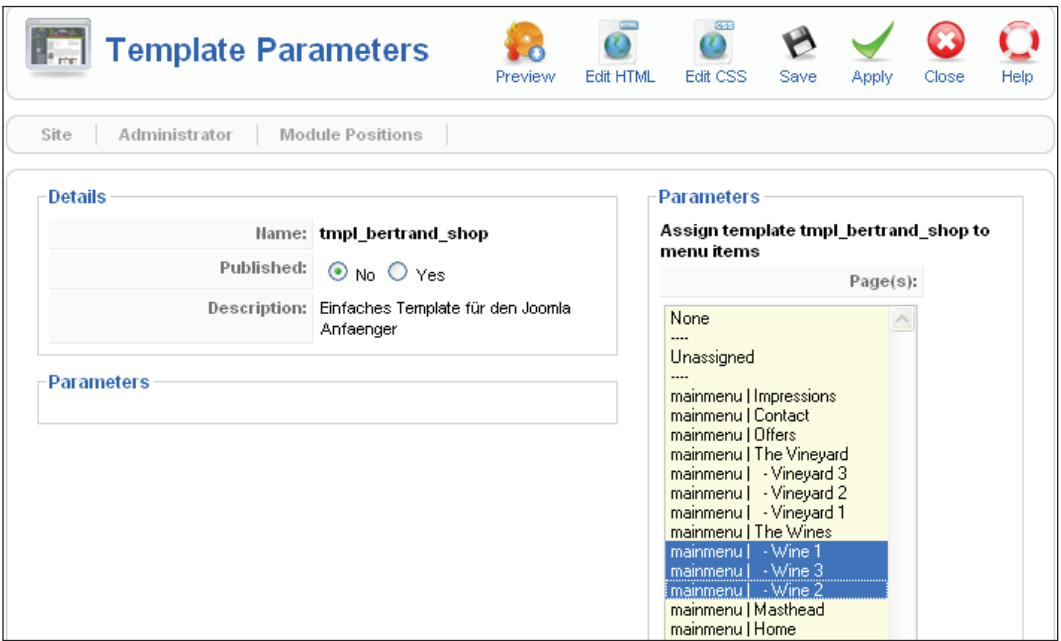


Figure 8.47: Special Template for the Shop Section

The Template

Ruth created the template according to the following considerations.

Maintenance

Since M. Bertrand will be maintaining the site himself, she used as simply and clearly arranged a basic grid pattern as possible. M. Bertrand can easily replace the images in the top section with new ones whenever he wants.

Layout

Ruth explains her effort like this: "There are many ways to creatively implement a website."

M. Bertrand wants to sell wine. This means that his website needs the presentation of a traditional, quality-conscious vintner where the customer would gladly purchase good wine.

A high-tech presentation with a metallic look would be inappropriate even if it were to be modern. No, the wine, the grapes, and the earth should determine the look and feel of the website and this is interpreted with the color choices and the photographs.

A visual appearance is created from several important elements. These include: logo, colors, the style of the photographs, and the graphics (by graphics she means all image elements that are not photographs, including the buttons and symbols).

The division of the space and the proportions also have an effect and all of these elements have to work together for the visual impression to be coherent. It is best to use only one typeface. Differences can easily be brought out with bold and italic styles of the same typeface.

In the Bertrand template, she did not use a very original, but rather a very classical typeface for the headings, one that leaves an impression of reliability. It is Times Classic, similar to what national newspapers use. Design does not always have to be particularly original; it suffices if it fits well.

A Website with Joomla!

A Portrait

During breakfast and during the telephone conversation when Ruth explained the CSS technology to M. Bertrand, she did a little sketching. She sends these sketches to M. Bertrand via email and he incorporates them into his image gallery.



Figure 8.48: M. Bertrand

Happy End

The site is online. It really only took two days.

M. Bertrand spends the next two weeks inserting pictures of the grape harvest and polishing the text. He contacts acquaintances and gets them to register as users and he even gets his first order over the new system.

The provider discussed Joomla!'s update problems with M. Bertrand for some time and M. Bertrand took out a service contract. The provider will take care of the update whenever Joomla! releases a new version.

It will be relatively simple to keep this site current, since it uses the standard version of Joomla! and since the Exposé Gallery runs in an HTML `iframe`, in other words similar to a wrapper component.

M. Bertrand is glad that he has outsourced the work with updates and the template.



If you need some design work done, why not contact
Ruth: Ruth Prantz, Certified Designer
(<http://www.ateliersite.de>).

How Does it Work in Reality?

A lot of small company websites are created just as in this story of M. Bertrand. I have been part of the process many times.

The bigger websites start like this too most of the time, the companies just don't like to admit it. With larger projects, as for example the website of Porsche in Brasil, the process is in principle the same. A graphic agency handles the design and a technical company does the programming and development. At Porsche there are departments handling text, corporate identity, and website security. Hosting also has to be approved. Since we are talking about Porsche: In 2002, Porsche Brasil had a website with thousands of HTML pages. In order to simplify the administration of the website, Porsche switched to Mambo in 2003. Lots of modifications were necessary to the Mambo source code in order to meet all of Porsche's demands.

At the moment, a project team is working on a Joomla! version of the Porsche website. It will also have a used vehicle component as well as a customized editor for the back end. Modifications to the source code only have to be made in the `Content` and in the `Search` components. The `Docman` document management component, the `Zoom` picture gallery, and the `Joomap` sitemap component, all from Joomla!'s extension cupboard, will be used. Take a look at their new website!

Maybe you are also looking for a new car.

Summary

Finally you are able to make your own website. You can download the `tmpl_bertrand` and `tmpl_bertrand_shop` from the Packt website and see the `template.css` and `joomla_stuff.css` files discussed in this chapter. In the next chapter we will discuss the three bonus templates provided with this book.



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

9

Bonus Templates

I bet that after M. Bertrand's experiences and the instructions on how to develop a template in Chapter 6, you are just itching to develop your own template. Thanks to Alex Kempkens, who organized the development of these templates, we can help you with a few ideas and several usable templates to start your adventure.

When we chose these templates, we wanted to present you with a variety of websites from different sectors and from different designers.

These are the templates that we have come up with and which you can freely use:

- A template from Tom Boháček for a non-governmental organization
- A template from Anne-Kathrin Merz for an online community website
- A template from Andy Miller for a venture

During the course of writing this book, the chapter about the wine merchant Bertrand and his e-commerce ambitions grew and grew, and there are now two additional templates that were not in the plans in the beginning:

- Two templates from Ruth Prantz for a middle-sized wine merchant in Languedoc, France

The story about how these templates were created and how the website was actualized is in Chapter 8.

NGO

The term *non-governmental organization* principally refers to every organized consortium of people who decide to start activities on a voluntary basis for a certain duration of time without a profit motive and without being organized by or dependent on some governmental body. Employers' associations, trade unions, sports clubs, and animal breeding associations are all examples of non-governmental

Bonus Templates

organizations. Every association that has been founded by the law of its country in the common public interest is potentially a non-governmental organization.

Tom Boháček

With his agency, Tom Boháček (Certified Computer Scientist (FH), <http://www.bohacek.de>) supports other agencies and enterprises with the development of their communication solutions. He works with Joomla! for an easily understandable reason: "It allows me the biggest freedom in implementing my ideas."

Read what Tom has to say about his template!

The World Knowledge Template

The idea was to develop a simple template that could satisfy the changing information status and the multitudinous sources of information of an NGO.

The example of an imaginary NGO **World Knowledge** was devised, which committed itself to the preservation of cultural stores of knowledge and their safekeeping.



Figure 9.1: World Knowledge Template

Creative Approach

"Gray, my dear friend, is every theory." With this Goethe quote as support, a friendly color scheme was developed that reminds one of parchment and libraries, associated with knowledge in other words. A light turquoise from the color palette was added for important elements and orientation help. This color that stands out from the layout symbolizes knowledge as such and is taken up symbolically in the header.

Since NGOs often operate interculturally, the layout is kept neutral and clear. This ensures that people of different countries quickly get their orientation.

The Helvetica family, which is very popular after the Web 2.0 Hype, was selected as the typeface, not just for its popularity but also because it represents competence and clarity and because it is installed on just about all systems.

Structural Approach

The layout of the website is divided into four parts – the header section, the navigation bar, and the two-part content section. The content section contains the articles and an additional section that contains pictures and other information such as links that are related to the content.

The navigation on the left does *not* contain a second navigation level. It can, however, be assembled from several navigation sections (for instance *Present* and *Future*). In order to communicate the request of the organization directly, a page is directly linked that delves deeper into the menu item as an article and/or blog. All deeper information is shown either next to the article as context-dependent links or as links directly in the article. This ensures that the navigation exhibits a consistent structure, while the individual articles can be exposed to strong dynamics.

Online Community

An online community is an association of people that meet and exchange views with each other on the Internet. An online community has to be built, maintained, and serviced. Members of the online community are often recruited for the tasks. Online Community is often misused as being synonymous with the technology that is used by the online community. An association of people is, however, not defined by the technology, but by the content that brings them together. It is a social phenomenon.

That is how the conversation with Anne-Kathrin Merz started. But before I write about it, why not read Anne's thoughts about this template.

Anne-Kathrin Merz

"It should become a theme for an Online Community website..."

– Anne-Kathrin Merz (<http://www.medanova.de>)

That is how Corali come to be, with the idea of a universal, modern lightweight theme for every occasion. It is a counter-piece to the typical three-column versions with colorful, glittering layout effects that can only be adapted to your own ideas with great difficulty.

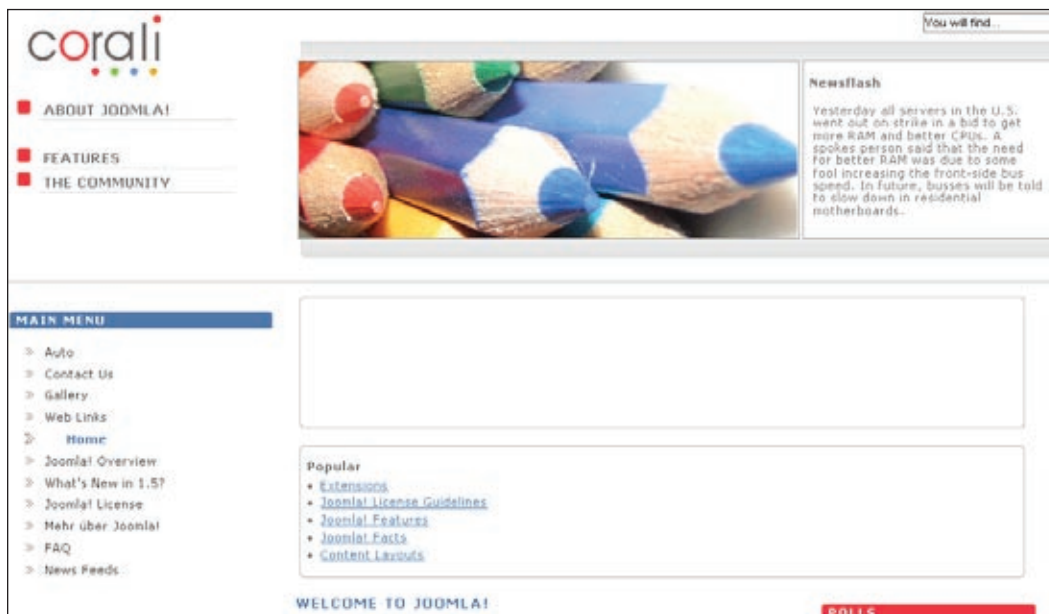


Figure 9.2: The Corali Theme with Two Columns



Figure 9.3: The Corali Theme with Three Columns

The biggest deliberation was: What is the capability of such a theme and what differentiates it from a business theme?

A Joomla! theme meets several challenges simultaneously: It presents the website with its own face, for example appropriate for the look and feel of an establishment, a club, an organization, or the corporate identity of an enterprise. But it also has to provide the proper view of all of the important information and, depending on the context, it has to blend into the page contents with a template that is laid on top of the contents and the modules. A "theme for every opportunity" thus also has to have as much flexibility as possible.

A Question of Orientation

Every website has important information and functionalities that should be available immediately and accessible from every page and which the user expects to be at particular positions. Among these are the link to the start page, to contacts, the masthead, a search field, and a main navigation that links to the items of greatest interest with one click.

This is why Corali has a small top menu in the upper section for the basic links as well as an optional module for searching; the logo also links to the start page, as you will find on many websites. If you don't want it to do that, you can integrate the logo as a background picture with CSS.

Bonus Templates

Corali also has a top menu that strictly enables quick access to the site and links the highlights of the site with a single click. For me this would be general links to site sections or important components such as a forum or the newsletter.

The main navigation breaches the widely used principle of the horizontal menu and with it the principle of a more or less symmetrical structure. Why? I wanted to try it: It should simply look a little different from the majority of Joomla! templates.

Disadvantage: The vertical structure makes the integration of submenus more difficult. If necessary, this can be solved very elegantly with CSS or specific Joomla! extensions.

The Header

The header, in many ways, is the eye-catcher, and it brings color and life to the website. But you lose valuable space if the image is too large. The newflash (as top module in the header section) should catch one's eye every bit as much as a lively image.

Since rounded corners give a soft and congenial impression, I decided to use them for the header. An image with rounded edges, however, makes the whole theme a little inflexible. It makes it more difficult to switch images.

A grouping of an image together with a module position for the newflash does both for me: round edges as a frame and easily switched graphics. It also brings those two elements together that should get the website visitors interest.

Left, Right, and the Contents

I believe that a community has a lot to say, about itself, about its members, information, the exchange of ideas amongst like-minded people, calendar, announcements, news, and pictures.

Modules that can be customized are therefore a lot more important here than on a classical business site. If you operate a picture gallery or a forum, then you know how important it is to have enough space available. So if the space on the left is not needed for a menu, then it is available for various contents.

Corali is a pure `<div>` theme without tables. On the one hand, I would be very happy to see code that conforms to W3C, but on the other hand, the `<div>` tag gives the developer a lot more freedom in terms of arranging the space on the page. It allows you to have fun!

And that is the strength of this theme: Corali "plays" with the ability to publish and hide modules in the context of the page and to dynamically adjust the width of individual elements.

In the main section of the page, in addition to the actual content, there are three additional sections: left, right, and above the page content. Each one of these sections is sort of a container for at least one module position. Containers form the basic framework for the layout and allow me to design the displayed modules in a more subtle fashion.

On the right and on the left I would typically see menus or modules such as the login to a newsletter, a mini-calendar, or a poll. If there is no menu or module on the right and/or on the left, the available space for content expands.

The top, taken up by `user1` and `user2`, can also be used for **Latest News** or for **Popular** and of course any other module. For flexible display of information, put it where it makes sense.

A forum, for example, can use the entire width; articles remain legible and clear and there are fewer conflicts with the layout of individual components. But most of all: a theme that adjusts itself to the scenario of the application.

But there are still a few issues to be deliberated:

- How much content can an individual module handle?
- What happens if the text is longer than the theme-designer has planned?
- Does the theme require more or fewer modules?
- What problems arise when `index.php` is modified?

Does this Theme Work with My Website?

Easy: Just try it out!

Coral is an attempt to break free from rigid, typical structures and to give the theme a character that is infused with content. End users should be able to modify their site according to their own imagination without extensive experience in the use of cascading stylesheets and the editing of graphics.

Graphics and colors can be quickly and easily modified in the CSS file, without having to make big changes in the heart of the template, the index file. The rest of the look and feel is affected by changes in the back end, by activating and hiding modules.

The numerous module positions and the flexible layout make Coral a community-oriented layout, that can, however, also be used for "conventional" purposes and it is therefore ready for "every opportunity".

Business Establishment

A business establishment is a legal, economically and financially independent unit with its own business management. The goal of a business establishment is monetary profit. Therefore a lot of value is put on respectability in order to gain the customers' trust. Dependent on the product, this respectability has to be reflected when a customer views the establishment's website and register in the customers' brains. The conversion of this respectability to websites keeps countless agencies busy day in day out and probably for another hundred years.

You can find one approach to this theme in Andy Miller's template!

Andy Miller

Andy Miller is a programmer and web developer with more than 10 years of experience in professional web application development. He is a member of the Joomla! Development team and was earlier on the Mambo team. In Joomla!'s development he is primarily responsible for the user interface, design, and web accessibility.

Andy has extensive experience with Joomla!, in particular in the area of template development. All of the administrator and front-end templates that have been shipped by both Mambo and Joomla! were (and are) designed by him (user name rhuk: http://dev.joomla.org/component/option,com_jd-wp/Itemid,33/cat,8/).

Andy is also the founder of *RocketTheme* (<http://www.rockettheme.com/>). RocketTheme offers new **monthly templates and a community for the maintenance** and continued development of the existing templates in a subscription-based business model. The forums can also be read by non-subscribers.

The **Versatility II template is a continued development of the Versatility template** and is technically very advanced. It, for instance, offers 15 module positions, each of which is naturally only displayed on the website if the respective module is available. This partitioning makes the template extremely flexible.

The template has four menu options:

1. SPLITMENU
2. SUCKERFISH
3. DROP-DOWN MENU
4. SUPERSUCKER MENU

In addition, your own or third-party menu systems can be integrated by means of the mechanism. You can see an example with the standard Joomla! menu system in the following figure:

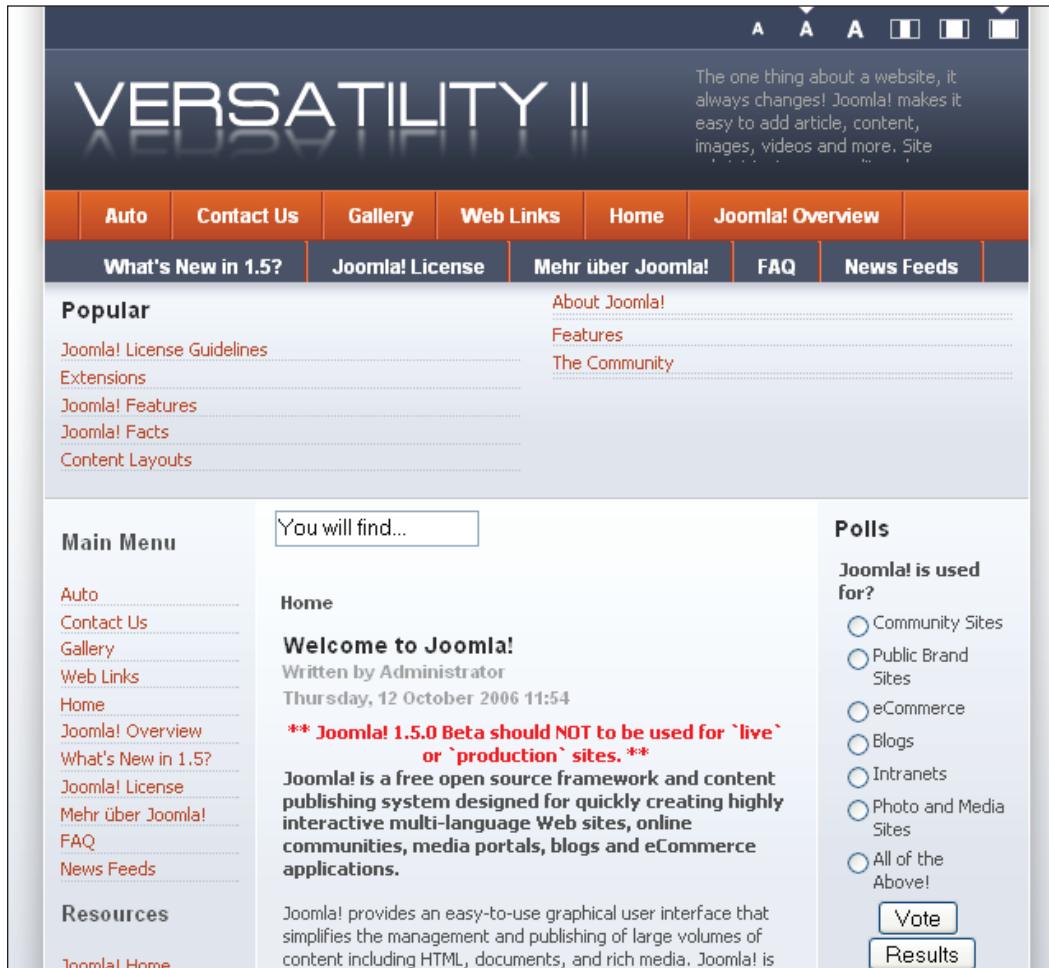


Figure 9.4: Versatility II by Andy Miller

The width and font size of Versatility II can be dynamically changed by guests to the website in three steps. The template is completely compatible with Joomla! 1.5.

Summary

This final chapter provided you with 3 bonus templates that can help you use templates in a similar way and develop your own website.



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386



Online Resources

Joomla! is an open-source project that collects the ideas and experiences of hundreds of thousands of individuals, merges them, develops them further, and offers them online. You will *always* find the most up-to-date information online.

In the English-speaking world, your best place to start is the **joomla.org** website at <http://www.joomla.org/>.

It is divided into the following sections:

- <http://help.joomla.org>
- <http://forum.joomla.org>
- <http://dev.joomla.org>
- <http://extensions.joomla.org>

Here you will really find everything you need to successfully use Joomla!.

For security you will get a well-maintained overview:

<http://forum.joomla.org/index.php/topic,81058.0.html>

Content for Download

You can download the necessary software packages from their respective project sites or from a summary on my website (<http://www.cocoate.com/>). The file packages that are offered here are well-suited for a local installation, since you will be able to duplicate the examples in the book with them.

Windows

Development environment **XAMPP lite**: xampplite-win32-1.5.3a.zip.

Linux

An Apache, PHP, and MySQL environment is contained in every popular Linux distribution and often is already pre-installed.

Independent of the Operating System



Please keep in mind that more current versions are always available on the appropriate sites on the Internet.

Chapter File Description

Chapter	File	Description
2	Joomla! 1.5 beta.zip	Joomla!-source code, Version 1.5.0 Pre-Beta
5	Joomla! 1.5 frontend language pack Joomla! 1.5 admin language pack	German language file
5	x-lite.exe	Editor XStandardLite
5	expose_php5_200608.zip	Exposé-Gallery
5	AlbumManager-2.7-SetupWin.zip	Album Manager for Exposé-Gallery
6	dreamweaver8-de.exe	Dreamweaver 8 (Trial-Version f. Windows)
6	dreamweaver8-de.dmg	Dreamweaver 8 (Trial-Version f. Mac OS X)
6	joomla150_book.zip	Your Own Template
6	joomlasolutions1.0.mxp	Dreamweaver-Extension
7	com_auto.zip	Your Own Component
7	mod_auto.zip	Your Own Module
7	plu_auto.zip	Your Own Plug-in
8	web_developer-1.0.2-fx+fl.xpi	Firefox-Extension Web Developer
8	FileZilla_2_2_25_setup.exe	Filezilla FTP
8	tmpl_bertrand.zip	Template Bertrand
8	tmpl_bertrand_shop.zip	Template Bertrand Shop
9	ngo.zip	Template NGO

Chapter	File	Description
9	corali.zip	Template Community
9	rt_versatility_ii_sienna.zip	Template Business

Table A.1: Book Content Files

Template Structure

The `<jdoc>` element gives you the option to address dynamic sections in your template without using a single PHP command.

Template Code	Effect
<code></jdoc:comment></code>	Comment line
<code><?xml version="1.0" encoding="utf-8"?></code> <code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"></code> <code><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="{LANG_TAG}" lang="{LANG_TAG}" dir="{LANG_DIR}" ></code>	Document type declaration
HEAD <code><jdoc:include type="head" /></code>	Creation of the metatags – defined as in the Global Configuration
<code><meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" /></code>	Declaration of the correct content type
<code><link rel="stylesheet" href="templates/_system/css/general.css" type="text/css" /></code> <code><link rel="stylesheet" href="templates/{TEMPLATE}/css/template.css" type="text/css" /></code>	Setting up the CSS file

Online Resources

Template Code	Effect
<pre><jdoc:tmpl name="loadcss" varscope="document" type="condition" conditionvar="LANG_DIR"> <jdoc:sub condition="rtl"> <link href="templates/{TEMPLATE}/css/ template_rtl.css" rel="stylesheet" type="text/css" /> </jdoc:sub> </jdoc:tmpl></pre>	Setting up the CSS support for "right to left" languages
<p>BODY</p> <pre><jdoc:include type="message" /></pre>	Joomla!-System message display
<pre><jdoc:include type="modules" name=" [position] "style=" [Style] " /></pre>	<p>Loading the module at position [position].</p> <p>[Style] is:</p> <p>0 = Module is displayed in a table column (<td>...</td>)</p> <p>1 = Horizontal Menu</p> <p>-1 = Raw - without surrounding layout elements</p> <p>-2 = XHTML (nearly barrier-free)-conforming output (<div>...</div>)</p> <p>-3 = Option for formatting "round corners".</p>
<pre><jdoc:exists type="modules" condition="left" > <jdoc:include type="modules" name="left" style="-3" /> </jdoc:exists></pre>	Condition instruction: If there is a left-module at the position, load it into the template.
<pre><jdoc:include type="component" /></pre>	Display of the component. The name of the component derives from the URL.

Table A.2: <jdoc:...> Insertion into the index.php of a Template

How do I Exchange a Picture (Logo) in the Template?

To switch a logo, you first have to examine the source code of the respective template. Graphics and images can be defined in the HTML and in the CSS structure. The size of the graphics is usually harmonized for the template.

If you want to exchange one of the images, you can do so by various methods.

Method 1

1. Create a graphic in a resolution and a file size that fits the space.
2. Load the graphic into Joomla!'s **Media Manager**.
3. Click on the graphic to get a link to it.
4. Change the respective `` tag in the source code of your template to the new image.

Method 2

1. Create the new graphic and give it the same name as the graphic in your template.
2. Simply overwrite the old graphic with the new one.

Joomla! API

The **API (Application Program Interface)** defines what functions/methods are at what places in the source code and what they do. It consists of statements that are produced automatically from the source code, each with an example. The Joomla! team makes this information available to you in English at http://api.joomla.org/li_Joomla-Framework.html.

Online Resources

Joomla! 1.5 Documentation

- [Home](#)
- [API reference wiki](#)
- [Browse SVN repository](#)

Packages

Joomla-Framework ▾

Package: Joomla-Framework

- Joomla-Framework
 - Class trees
 - Element index
 - Classes
 - Files
 - Abstract
 - Application
 - Cache
 - Client
 - Compatibility
 - Database
 - Document
 - Environment
 - FileSystem
 - Filter

Joomla! 1.5 Documentation

These pages are automatically generated by the [phpDocumentor](#) package from an analysis of the Joomla! source code and making use of specially formatted comments embedded in the source code itself.

The complete Joomla! CMS can be divided into a number of packages:

- The Joomla! API, called [Joomla-Framework](#).
- A package of legacy classes and functions, only used if legacy mode is turned on, called [Joomla-Legacy](#).
- The Joomla! CMS, which is an application built on the Joomla! Framework, called simply [Joomla](#).
- A number of third-party libraries used by Joomla! and/or available for use by third-party Joomla! extension developers. Please note that the Joomla! Developer Documentation Team is not responsible for maintaining documentation, including this automatically generated documentation, for third-party libraries.

You can select which package to view using the drop-down list in the menu on the left of this screen.

Additional documentation is available on the [Joomla! Developer Network](#) website and in particular there is a manually produced [API reference wiki](#) which has additional information and example code for classes and methods in the Joomla! Framework.

These pages were generated from Joomla! 1.5 SVN 6418 (Wed 31 Jan 2007)


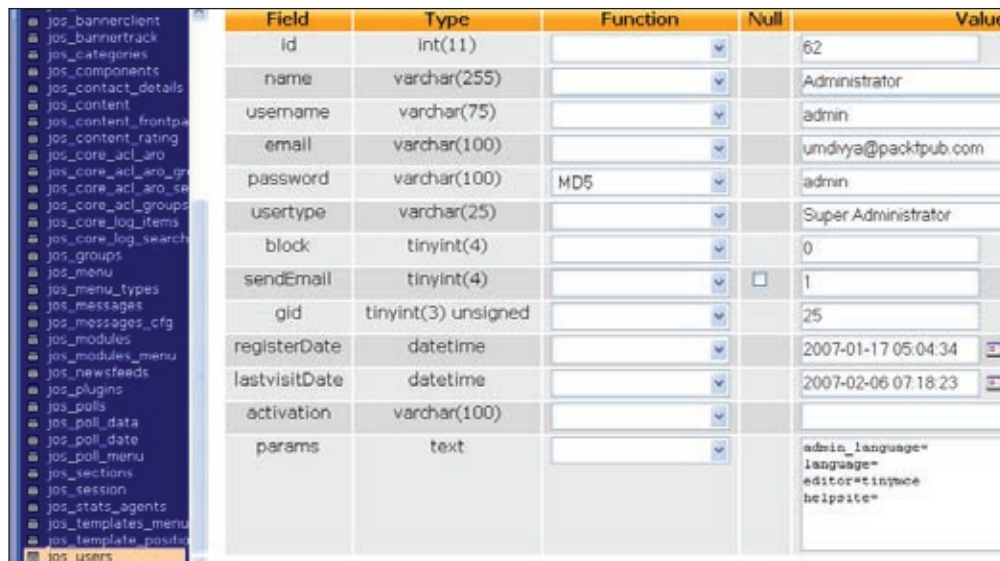


Figure A.1: Joomla! API

Forgot the Admin Password

If you have forgotten your admin password, you can change it directly in the database. You need a tool like [phpMyAdmin](#) to do this. In the xampplite environment, you call this up from your browser at <http://localhost/phpmyadmin/>.

Select the database you are using. There is a `joomla_users` table in this database. In it search for the user `admin`. The password is encrypted with an MD5 procedure. You can change it by selecting the MD5 entry in the options list on the left, next to the field, and entering the password in plain text.



Field	Type	Function	Null	Value
id	int(11)			62
name	varchar(255)			Administrator
username	varchar(75)			admin
email	varchar(100)			umdivya@packtpub.com
password	varchar(100)	MD5		admin
usertype	varchar(25)			Super Administrator
block	tinyint(4)			0
sendEmail	tinyint(4)		<input type="checkbox"/>	1
gid	tinyint(3) unsigned			25
registerDate	datetime			2007-01-17 05:04:34
lastvisitDate	datetime			2007-02-06 07:18:23
activation	varchar(100)			
params	text			admin_language= language= editor=tinymce helpsite=

Figure A.2: Changing the Admin Password

Many providers supply you with phpMyAdmin to maintain your database.

Migration from Joomla! 1.0.x to Joomla! 1.5

A lot of you are probably already operating a Joomla! 1.0.x or a Mambo 4.5.x website and now want to migrate it to Joomla!1.5.

You may have a number of basic problems:

- You need to back up the entire website.
- The general source-code files need to be updated.
- Additional components, modules, and mambots/plugin-ins have to be updated.
- Any changes that you yourself have made to the programs need to be updated.
- Your contents have to survive this update procedure without damage.
- Contents with special characters have to be converted according to UTF-8.

You can see from this list that an update can become quite complicated. The more you have worked with standard components, the easier it is; and the more individualized your website is, the more difficult it becomes. There are no general valid rules for updates at this time. This means that you have to first collect information to be able to establish an individual update plan for your site.

Backing Up Your Data

Before you make any plans and try out any migration components, back up your database and your files. Effect a MySQL dump with your provider's appropriate tool. In most cases this will be the phpMyAdmin program, which is also used in the xampplite environment. Click on the **Export** tab, mark all of the tables, and check all of the fields in the **Structure** checkbox. In the **Data** checkbox, select **Complete inserts**. You have to mark the **Save as** checkbox in the lower section and also check off the desired format. If the provider permits, use the zipped version. Compressed database files can be up to 95% smaller than normal database files. Confirm your selections by clicking the **Go** button.

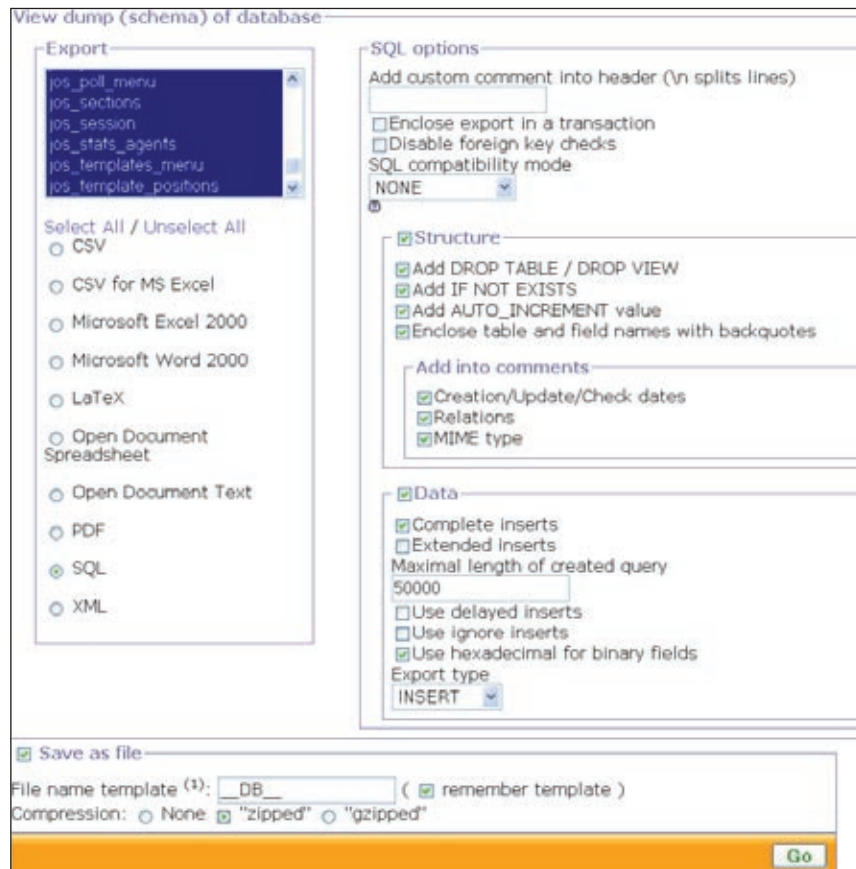


Figure A.3: Backing Up Your Database with phpMyAdmin

The contents of the database are now extracted and made ready for download. The file contains all of the SQL commands needed to create the tables in another database with your content. This is the optimal way of backing up your files.

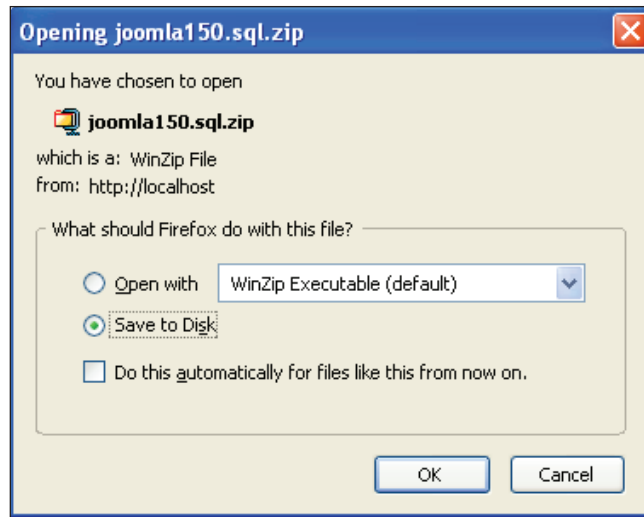


Figure A.4: Download of the Backed-Up DB

Now you can use phpMyAdmin to restore the data. Click on the **Import** link, select the file with the backed-up data, and click on the **Go** button. All of the SQL commands in the file are now executed and your data is restored.

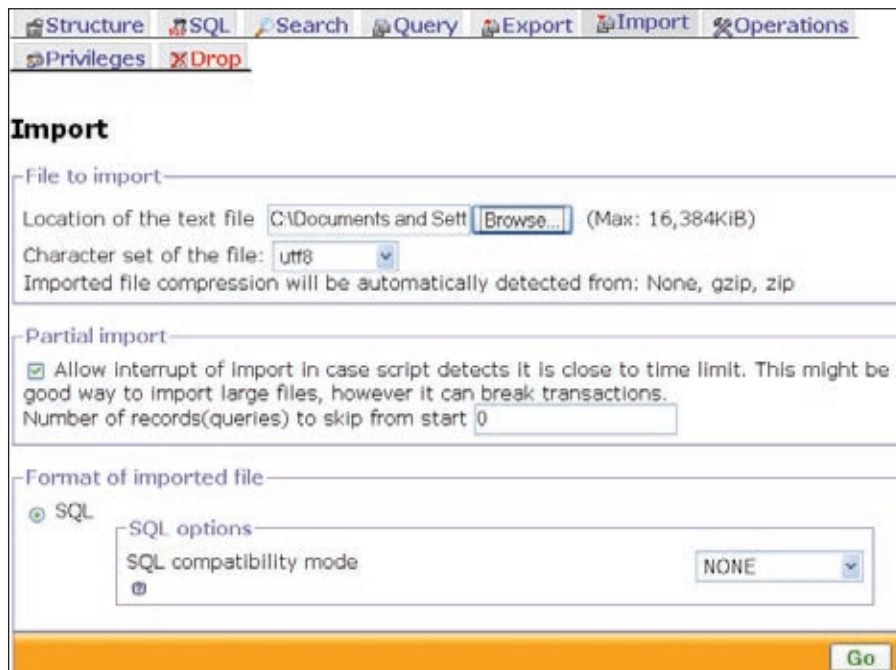


Figure A.5: Restoring your Backed-Up Data

Backing Up the Files

In addition to the data in the database, the source code should, of course, also be backed up. Use your favorite FTP program and copy the entire Joomla! folder.

The Migration Script

There is a migration component that has to be installed into the old Joomla! version 1.0.x. This component creates three ZIP files by clicking on the **Dump It** icon:

- A complete backup of the entire website
- A backup of the core components
- A backup of the third-party components

Save these files in a safe place. These files contain the original, unchanged data of your Joomla! 1.0.x website. This applies in particular to your character set!

New Installation of Joomla! 1.5.0

Set up a new folder and copy all of the Joomla! 1.5.0 files into this folder. Install Joomla! 1.5.0 up to step 6 (as discussed in Chapter 2).

The screenshot shows the Joomla! Webinstaller Migration interface. It features three radio button options, each with a descriptive text block and a set of controls.

- Install default sample data:** The text says, "Installing this is strongly recommended for beginners. It will install default sample content that is included in the Joomla! installation package." Below this is a single "Install sample data" button.
- Load local Joomla! 1.5 SQL script:** The text says, "The SQL scripts need to be Joomla! 1.5 compatible and should have the appropriate table prefix." Below this is a text input field, a "Browse..." button, and an "Upload and execute" button.
- Load migration script:** The text says, "The migration script needs to be created on the old site by the com_migrator tool or created manually to conform. Enter the table prefix of the old site and enter the encoding used in old site (_ISO setting in language file or as seen in browser info/encoding/source)." Below this are three fields: "Old table prefix" with a text input, "Old site encoding" with a dropdown menu showing "iso-8859-1", and "Migration Script" with a text input, a "Browse..." button, and an "Upload and execute" button.

Figure A.6: Webinstaller Migration

Now select the **Load Migration Script** radio button. Enter the old prefix (mostly `mos_` or `jos_`) and select the old encoding. If you are not certain, take a look in the old language files and search for the `_ISO` variable.

Now load the Core export file of the migration component and click on the **Upload and execute** button.

What happens next depends on your data. Normally you would get a success message and your data are transformed. But if you get database errors, this doesn't have to be a big problem either. Try to figure out what the error messages say and fix the errors manually in phpMyAdmin.



I know that this may not sound terribly helpful, but maybe it will set your mind at ease that the update of our website went flawless on the first try. I am going to spare you the story of the other website with all of the additional components and put my trust into the continued development of the installer.

No matter what happens, finish the installation as planned with the name of the site and the admin password.

What Happens Now?

All of the core components, like `com_content`, `com_weblinks`, `com_banner`, `com_contact`, `com_newsfeed`, `com_poll`, and `com_users` have been updated.

The menu links of the core components have been updated. The modules are all set to unpublished. The display positions of the modules and their parameters have to be reworked. No third-party components have been transferred and nor has any data (banner, image, PDF files, etc.).

Manual Method

First the bad news. Changing the tables does not work properly with phpMyAdmin because the extended characters are not interpreted correctly! The data are all imported, but the extended ASCII character set (like `Ä`, `Ö`, `Ü`, `ß`, `ä`, `ö`, `ü`, ... and many others) are not displayed correctly.

First you have to export all of the data and bring them into the UTF-8 format, and then you have to make a few changes to the table structures. The export with phpMyAdmin works without problem. (Make sure you only export the data, *not* the table structures.)

Online Resources

Check the **Complete inserts** checkbox. Now you have to save the exported data with an appropriate editor in UTF-8 format. (This even works with Windows Notepad, as long as the dump is not too large.)

Modifying the Joomla! 1.5 Database Scheme

There are only two fields that have to be renamed!

In the `jos_core_acl_aro` table, the field `aro_id` is renamed to `id`. In the `jos_core_acl_aro_groups` table, the field `group_id` is renamed to `id`.

Importing the Tables

These tables can be imported:

- `jos_banner`
- `jos_bannerclient`
- `jos_bannerfinish`
- `jos_categories`
- `jos_contact_details`
- `jos_content`
- `jos_content_frontpage`
- `jos_content_rating`
- `jos_core_acl_aro`
- `jos_core_acl_groups_aro_map`
- `jos_core_log_items`
- `jos_core_log_searches`
- `jos_messages`
- `jos_messages_cfg`
- `jos_newsfeeds`
- `jos_poll_data`
- `jos_poll_date`
- `jos_poll_menu`
- `jos_polls`
- `jos_sections`
- `jos_users`
- `jos_weblinks`

The `jos_usertypes` is no longer used in Joomla! 1.5.

Menus and modules can be either newly configured with phpMyAdmin or with the admin interface. And you are done.



If you have some experience with phpMyAdmin and MySQL and/or the SQL language, you will be able to upgrade the core components with no problem.

The migration of third-party components can be more problematic. I am positive that migration scripts will be made available soon that will migrate, for example, a Joomla!board or a Docman to Joomla! 1.5.0.

And as a final note, not much has changed with the tables except for the character set. The bulk of the component developers' work was done in the source code that will be applied to the new version.

Security without Global Variables

Mambo and Joomla! demand a default PHP setting with the option `register_globals = on`. This switch handles visibility in global variables that can, among other ways, wind up in external programs by the use of forms or get-strings. This setting is fraught with basic security problems and it is better to use the `register_globals = off` setting.

Even though this switch, of course, does not protect you from all security problems, it is a tremendous help in raising security. To prepare Joomla! accordingly, rename the file `[PathtoJoomla!]/globals.php` to `[PathtoJoomla!]/globals.php-on` and rename the file `[PathtoJoomla!]/globals.php-off` to `[PathtoJoomla!]/globals.php`.

Joomla!'s core will definitely work. Give it a try to see whether all of your components still work. If you have problems, change the settings back to `globals on` for the time-being and look for updates of the affected components.



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386

Index

A

- album manager**
 - about 200-202
 - as a stand-alone program 203, 204
 - integrating 204
 - uninstalling 203
- article manager**
 - about 114-116
 - archiving 117
- article statistics** 133
- article trash** 127

B

- back end, Joomla! website**
 - administrating 61-64
- banners, Joomla! components**
 - about 139, 140
 - banner clients 140
 - graphic banners 142-144
 - managing 141-145
 - modules 144
 - text links 144

C

- cascading style sheets.** *See* CSS
- category manager** 130
- changes, in Joomla! 1.5**
 - database support 18
 - framework 19
 - FTP system 19
 - Google projects 19
 - internationalization 18
 - search engine friendliness 19

- user plug-ins 18
 - web accessibility 19
 - XML-RPC support 18
- CMS.** *See* **content management system**
- component, creating**
 - about 250, 251
 - auto.html.php, front end 255-257
 - auto.php, front end 254, 255
 - front end 253
 - installation package, creating 269-271
 - Joomla!, administrating 258
 - main menu, integrating into 257
 - MySQL table 252, 253
 - SQL commands, MySQL table 252
 - testing 269
- components, Joomla!**
 - banners 139
 - components menu 139
 - contacts 145
 - creating 250
 - example 250
 - helloworld 245
 - news feeds 150
 - polls 152
 - sample component 245
 - web links 154
- contacts, Joomla! components**
 - about 145
 - category manager 149
 - contact manager 145-149
 - details, contact manager 146, 147
 - information, contact manager 147
 - parameters, contact manager 147-149
- content management system**
 - about 7

- CRM 7
- definition by Wikipedia 7
- DMS 7
- ECMS 7
- ERP 7
- history 8-10
- HRM 7
- HTML, disadvantage 9
- Joomla!, WCMS 8
- LAMP 10
- PHP, advantage 11
- WCMS 8
- content menu**
 - about 113
 - article manager 114-116
 - article statistics 133
 - article trash 127
 - category manager 130
 - content, editing from front end 134
 - frontpage manager 132
 - new article, creating 118-127
 - section manager 128
 - sections, editing 129
- Coral**
 - about 330
 - advantages 333
 - customizing 332
 - disadvantage 332
 - header 332
 - structure 331
- Corporate Identity**
 - about 207
 - components 207
- CSS**
 - about 208
 - HTML, integrating into 208
- E**
- Exposé flash gallery**
 - about 194
 - features 194
 - installing 195
 - integrating 196-199
 - videos 199
- extensions, Joomla!**
 - album manager 200
 - Exposé flash gallery 194
 - extensions menu 157
 - install/uninstall 157
 - language manager 187
 - module manager 158
 - plugin manager 175
 - template manager 181
- extensions, WCMS**
 - components 14
 - module 14
 - plug-ins 14
 - templates 14
- F**
- fixed size templates**
 - about 210
 - advantages 210
 - disadvantages 210
- formatting, Joomla!**
 - CSS code 313
 - divider lines 315, 316
 - menu item 314
 - module title 313
 - schema 312
- front end, Joomla! website**
 - advertising 57
 - banner, advertising 58
 - content, about 55, 56
 - content, announcing 56, 57
 - decorative elements 61
 - feeds, functions 60
 - functions 58
 - login area, functions 58
 - main menu, menus 54
 - menus 53, 54
 - modules 58
 - other menu, menus 54
 - overview 51-53
 - polling, functions 59
 - search field, functions 60
 - top menu, menus 53
 - who is online, functions 60
- frontpage manager 132**

H

helloworld, components

- functionality, adding 248, 249
- helloworld.php, home directory 247
- helloworld.php, Joomla! framework 247, 248
- home directory 246
- HTML output, separating 249, 250
- Joomla! framework 247
- logic, separating 249, 250

help menu

- Joomla! Help 67
- system info 68, 69

HTML/XHTML

- about 208
- tags 208

I

install/uninstall, Joomla! extensions 157

J

Joomla!

- Ada Gaffney Shaff, examples 22
- administration, configuring 65-67
- admin password, recovering 342, 343
- album manager, extensions 200
- API 341
- as real estate 15
- awards 12
- back end, website 61
- changes from 1.0 18-20, 316
- client-server-system 27
- commercial extensions 21
- components 139, 245
- content for download 337
- Corporate Identity, templates 207
- CSS, internet technologies 208, 209
- customizing 65
- examples 21-25
- Exposé flash gallery, extensions 194
- extensions 157
- features 20, 21
- for business establishment 334
- for NGO 327

- for online community 329
- fresh installation 346, 347
- front end, website 51
- functionality, extending 192, 193
- history 11, 12
- HTML/XHTML, internet technologies 208
- installing 27, 38
- installing, on virtual server 36-38
- internet technologies 208
- Janette Norton, examples 23
- Joomla! 1.5 db scheme, modifying 348
- Joomla.org, examples 21
- local environment, setting up 31-36
- migrating 343
- modules 272
- online resources 337
- picture (logo), exchanging in template 341
- plug-ins 276
- prerequisites 28
- requirements, technical 28
- roadmap 17
- tables, importing 348
- templates 327
- templates, creating with Dreamweaver 224
- templates, installing 321
- templates, writing 207
- third party components 205
- TinyMCE editor 74
- unric.org, examples 25
- Urth.tv, examples 24
- users, adding 76
- users, editing 72
- versions 16
- versions, numbering system for 16, 17
- web accessibility 240
- web installer 41
- website, accessing 27
- website, building 281
- XAMPP for Linux 338
- XAMPP for Windows 338
- XML, internet technologies 209

Joomla!, administrating

- admin.auto.html.php 265-267
- admin.auto.php 260-263
- article manager, content menu 114
- article statistics, content menu 133

- article trash, content menu 127
- category manager, content menu 130
- component table 258-260
- configuring 65-67, 80-96
- content, editing from front end 134, 135
- content menu 113
- control panel, site menu 70
- frontpage manager, content menu 132
- help menu 67
- Joomla! Help 67
- main menu, menus menu 98
- media manager, site menu 77
- menu manager, menus menu 97
- menus menu 96
- new article, creating 118
- new menu, creating 107
- section manager, content menu 128
- site menu 69
- system info 68, 69
- toolbar.auto.html.php 268
- toolbar.auto.php 267
- tools 135
- trash manager, menus menu 106
- user manager, site menu 70

Joomla!, installing

- directory, selecting 38-41
- web installer 41

Joomla!, migrating from 1.0.x to 1.5

- data, backing up 344, 345
- files, backing up 346
- migration script 346

Joomla! web installer

- configuring 46-48
- database, configuring 44
- finishing 48, 49
- FTP, configuring 46
- language, choosing 41
- licence 43
- pre-installation check 42, 43

L

language manager, Joomla! extensions

- about 187
- different language file, installing 187-189
- menu entry, translating 189-191
- menu name, modifying 192

local environment, Joomla!

- Debian, Linux 36
- Linux 35
- own server, Linux 36
- SUSE > 9.1, Linux 35
- Ubuntu, Linux 36
- Windows 31-33
- XAMPP, Windows 32-34

M

main menu

- about 98
- copy icon 99
- default icon 98
- edit icon 100
- menu item details, edit icon 101-103
- menu item type, edit icon 101
- move icon 99
- new icon 105
- parameter, edit icon 103-105
- publish/unpublish icon 99
- trash icon 100

media manager

- about 77
- configuring 80
- directory, creating 78
- file, uploading 79

menu manager 97

menus menu

- about 96
- main menu 98
- menu manager 97
- new menu, creating 107-113

module manager, Joomla! extensions

- about 158
- administrator module 173
- admin menu 174
- admin submenu 175
- advertisement 169
- archive 165
- banner 160, 161
- breadcrumbs 161
- Cpanel shell, toolbar 175
- feed display 167
- footer 162, 174
- Joomla! book 167

- latest news 170
- logged in users 173
- login 174
- login form 164, 165
- main menu 162-164
- menu stats 174
- module, copying 172
- newsflash 169
- online users 174
- polls 168
- popular 170, 174
- quick icons 174
- random image 168, 169
- recent added articles 174
- related items 165
- search 171
- sections 165
- site modules 158-160
- statistics 165
- syndication 169
- title 175
- toolbar 175
- unread messages 174
- user status 175
- who's online 168
- wrapper 166

modules

- about 272
- installing 274, 275
- source code 272-274
- viewing 276

N

news feeds, Joomla! components

- about 150
- categories 152
- feeds 151

NGO

- about 327
- creative approach, world knowledge template 329
- structural approach, world knowledge template 329
- world knowledge template 328

non-governmental organization. *See* NGO

O

online community

- about 329
- Coralii 330

P

plug-ins

- source code 276-279
- viewing 279

plugin manager, Joomla! extensions

- about 175
- authentication plug-ins 176
- code highlighter (GeSHi), content plug-ins 179
- code highlighter (Joomla!), content plug-ins 178
- content plug-ins 178
- editors-xtd plug-ins 180
- editors plug-ins 180
- email cloaking, content plug-ins 178
- Gmail, authentication plug-ins 176
- image, content plug-ins 178
- Joomla!, authentication plug-ins 176
- Joomla! SEF urls, system plug-ins 180
- LDAP, authentication plug-ins 176, 177
- loud module, content plug-ins 179
- no editor, editors plug-ins 180
- pagebreak, content plug-ins 179
- page navigation, content plug-ins 178
- rating, content plug-ins 178
- search plug-ins 180
- SEF, content plug-ins 178
- system plug-ins 180
- TinyMCE 2.0, editors plug-ins 180
- visitor statistics, system plug-ins 181
- XMLRPC plug-ins 181
- XStandard Lite 1.7, editors plug-ins 180

polls, Joomla! components

- about 152
- previewing 153

prerequisites, Joomla!

- Linux, local test environment 29
- local test environment 29
- Mac OS X, local test environment 29

- own server, productive environment 30
- productive environment 30
- rented virtual server, productive environment 30
- Windows, local test environment 29

S

section manager

- about 128
- sections, editing 129

SGML 209

site menu

- control panel 70
- directory, creating 78
- file, uploading 79, 80
- logging out 96
- media manager 77
- new users, adding 76, 77
- user manger 70
- users, editing 72-76

site menu, configuring

- cache settings 93, 94
- content 88
- content item settings 90, 91
- content settings 88, 89
- database settings 94, 95
- feed settings 90
- FTP settings 94
- local settings 93
- mail settings 95, 96
- metadata settings 83
- SEO settings 85, 86
- server 91
- server settings 92, 93
- site 81
- site settings 82
- statistics settings 85
- users 86, 87

Standardized Generalized Markup Language 209

T

template manager, Joomla! extensions

- administrator-templates 182
- module position 182
- new template, adding 182-184
- site-templates 182
- template, modifying 185, 186

templates

- <div> tags 236-240
- building 207
- Coralii 330
- Corporate Identity 207
- creating 210
- creating, Dreamweaver extension used 224
- directory structure 213-215
- files, directory structure 213-215
- fixed size, types 210
- Joomla! module, integrating 217-221
- package, creating 222, 223
- package, installing 223, 224
- structure 211, 229
- template file, creating 226-228
- trial run 215, 216
- types 210
- variable size, types 210
- Versatility II 334
- world knowledge template 328

templates, creating

- concept 210, 211
- concept, converting into HTML 212, 213
- directory structure 213, 214
- Dreamweaver extension, installing 225
- Dreamweaver extension, using 224
- package, creating 222, 223
- package, installing 223, 224
- structure 211
- template file, creating 226
- trial run 215, 216
- types 210

template structure

- table structure, creating 229-232
- with Dreamweaver 229

third party components, Joomla!

- security problems 205
- update problems 205

tools

- global check-in 137
- mass mails 136
- private messaging 135

trash manager 106

U

user manager

- about 70
- new users, adding 76
- special users 74
- user groups 73
- users, editing 72

users

- editing 72-75
- managing 70
- new users, adding 76
- special users 74
- TinyMCE editor 74
- user account, settings 86
- user manager 70

V

variable size templates

- about 210
- advantages 210
- disadvantages 210

Versatility II

- about 334
- menus 334

W

WCMS

- access rights, structure 13
- back end, structure 13
- configuration settings, structure 15
- content, structure 13
- extensions, structure 13, 14
- front end, structure 13
- structure 13-15
- workflow, structure 14

web accessibility, Joomla!

- about 240
- accessible sites 244
- criteria 240
- technology 242

web content management system. *See*

WCMS

web links, Joomla! components

- about 154
- categories 156
- links 155, 156

website with Joomla!

- appearance 282-284
- articles 286-289
- changes from 1.0 316
- contact link, setting up 295, 296
- CSS, working of 309-311
- designing 321, 322
- download section 303-307
- formatting 312
- idea behind website 281
- impressions, setting up 299, 300
- installing, locally 284-286
- installing, on webserver 316-321
- logo 282-284
- main menu, menu structure 292
- masthead 289-292
- menu links, setting up 293
- menu structure 292
- navigating code 311, 312
- results, checking 308, 309
- shopping facility, setting up 297-299
- technical conversion 284
- text, setting up 293
- top menu, menu structure 293
- top menu, setting up 296
- user details 307
- user menu 300-302
- user menu, menu structure 293

website with Joomla!, designing

- about 321
- template, installing 321-323

website with Joomla!, installing on

webserver

- about 316
- data, importing 318, 319

- directory rights 320
- file rights 320
- friendly urls, search engines 321
- installing, at provider 317
- metabags, search engines 321
- search engines 320
- world knowledge template**
 - about 328
 - creative approach 329
 - structural approach 329

X

XML

- about 209
- SGML 209



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386



This material is copyright and is licensed for the sole use by Don Fountain on 7th April 2007
24800 I-45, Suite #140, , The Woodlands, , 77386



**Thank you for buying
Building Websites with Joomla! 1.5
Beta 1**

Packt Open Source Project Royalties

When we sell a book written on an Open Source project, we pay a royalty directly to that project. Therefore by purchasing a paper copy of Building Websites with Joomla! 1.5 Beta 1, Packt will have given some of the money received to the Joomla! project.

In the long term, we see ourselves and you – customers and readers of our books – as part of the Open Source ecosystem, providing sustainable revenue for the projects we publish on. Our aim at Packt is to establish publishing royalties as an essential part of the service and support a business model that sustains Open Source.

If you're working with an Open Source project that you would like us to publish on, and subsequently pay royalties to, please get in touch with us.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to authors@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

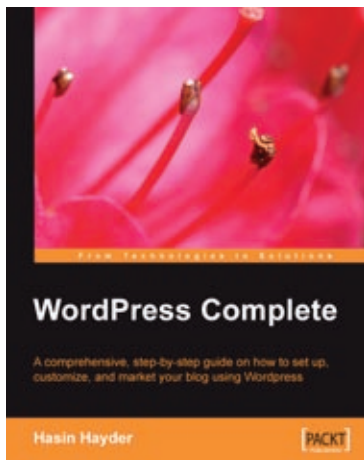
We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

About Packt Publishing

Packt, pronounced 'packed', published its first book "Mastering phpMyAdmin for Effective MySQL Management" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.PacktPub.com.



WordPress Complete

ISBN: 1-904811-89-2 Paperback: 272 pages

A comprehensive, step-by-step guide on how to set up, customize, and market your blog using WordPress

1. Clear practical coverage of all aspects of WordPress
2. Concise, clear, and easy to follow, rich with examples
3. In-depth coverage of installation, themes, syndication, and podcasting



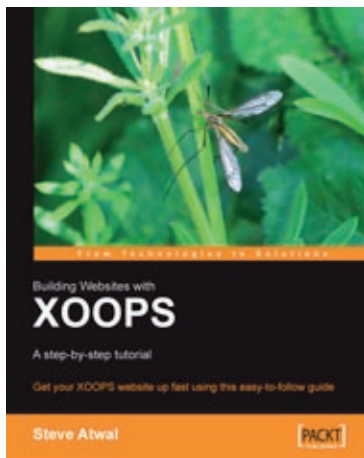
Alfresco Enterprise Content Management Implementation

ISBN: 1-904811-11-6 Paperback: 350 pages

How to Install, use, and customize this powerful, free, Open Source Java-based Enterprise CMS

1. **Manage your business documents:** version control, library services, content organization, and search
2. **Workflows and business rules:** move and manipulate content automatically when events occur
3. **Maintain, extend, and customize Alfresco:** backups and other admin tasks, customizing and extending the content model, creating your own look and feel

Please visit www.PacktPub.com for information on our titles



Building Websites with XOOPS

ISBN: 1-904811-28-2 Paperback: 180 pages

Get your XOOPS website up fast using this easy-to-follow guide

1. Simple and practical guide to XOOPS
2. Manage blocks, modules, users, and themes
3. Case study reinforces effective learning



Deep Inside osCommerce: The Cookbook

ISBN: 1-847190-90-1 Paperback: 400 pages

Ready-to-use recipes to customize and extend your e-commerce website

1. osCommerce expert "Monika in Germany" lets you in on her secrets on how to hack your way to that perfect osCommerce site
2. Create new modules and custom-code your default osCommerce installation
3. Add extensions and features like category driven designs and individual shipping modules

Please visit www.PacktPub.com for information on our titles