



От технологий к решениям

Создание веб-сайтов с помощью **Joomla! 1.5**

Популярное учебное пособие по Joomla!, обновленное
с учетом последней версии системы

Хаген Граф



Издательский дом "Вильямс"
www.williamspublishing.com

[РАСКТ]
PUBLISHING

О т т е х н о л о г и й к р е ш е н и я м

Создание веб-сайтов
с помощью
Joomla! 1.5

Хаген Граф



Издательский дом "Вильямс"
www.williamspublishing.com

From Technologies to Solutions

Building Websites
with
Joomla! 1.5

Hagen Graf

[PACKT]
PUBLISHING

Оглавление

Глава 1. Термины, концепции и соображения	17
Глава 2. Инсталляция	35
Глава 3. Тур по вашему новому веб-сайту	53
Глава 4. Настройка Joomla! — языки и шаблоны	63
Глава 5. Конфигурация администрирования Joomla!	67
Глава 6. Меню Site	73
Глава 7. Меню Menus	95
Глава 8. Меню Content	111
Глава 9. Меню Components	129
Глава 10. Меню Extension	143
Глава 11. Меню Tools	165
Глава 12. Дополнительные сведения о расширениях	169
Глава 13. Написание собственных шаблонов Joomla!	191
Глава 14. Свобода от барьеров и Joomla!	205
Глава 15. Собственные компоненты, модули и подключаемые модули	225
Глава 16. Построение веб-сайта с помощью Joomla!	251
Глава 17. Шаблоны-бонусы	277
Приложение А. Онлайн-ресурсы	281
Приложение Б. Модули шаблонов	282
Приложение В. Как изменить изображение (логотип) в шаблоне?	283
Приложение Г. Joomla! API	284
Приложение Д. Забыли пароль администратора?	285
Приложение Е. Переход от Joomla! 1.0.x к Joomla! 1.5	286
Приложение Ж. Безопасность без глобальных переменных	291
Предметный указатель	292

Содержание

Об авторе	13
Предисловие	14
О чем эта книга	15
Соглашения	15
Коды примеров	16
От издательства	16
Глава 1. Термины, концепции и соображения	17
Системы управления содержимым	17
Краткий курс в историю	18
История разработки Joomla!	20
Структура системы управления веб-содержимым	22
Клиентская и серверная части	22
Права доступа	22
Контент	22
Расширения	23
Рабочий поток	24
Конфигурационные установки	24
API	24
Является ли Joomla! частью “недвижимости”?	24
Версии Joomla!	25
Система нумерации версий Joomla!	25
Дорожная карта	26
Изменение в деталях	27
Интернационализация	27
Пользовательские подключаемые модули	27
Поддержка удаленных вызовов процедур XML	28
Поддержка нескольких баз данных	28
Система FTP	28
Пересмотр каркаса Joomla!	28
Свобода от барьеров	28
Удобство использования механизма поиска	29
Проекты Google Summer of Code	29
Средства Joomla!	29
Примеры страниц Joomla!	30
Joomla.org	30
Travel Shop, Ирландия	30
Frank Lüdtkе, Германия	30
Отель Schönruh, Австрия	32
Urth.tv , США	32
unric.org , Европа	33
porsche.com.br, Бразилия	33
Резюме	34

Глава 2. Инсталляция	35
Система клиент-сервер	35
Обращение к веб-сайту Joomla! в Интернете	35
Технические требования для Joomla!	36
Необходимые элементы для инсталляции системы Joomla!	36
Локальная тестовая среда	37
Рабочая среда	37
Установка локальной серверной среды	38
Windows	38
Linux	41
Ваш собственный сервер у провайдера (root-сервер)	42
Инсталляция Joomla! на виртуальном сервере в сети	42
Инсталляция Joomla!	43
Выбор каталога для инсталляции	43
Веб-инсталлятор Joomla!	45
Резюме	51
Глава 3. Тур по вашему новому веб-сайту	53
Интерфейсная часть	53
Меню	55
Контент	56
Реклама	58
Функции	59
Декоративные элементы	60
Перспектива	61
Серверная часть	61
Резюме	62
Глава 4. Настройка Joomla! — языки и шаблоны	63
Изменение языка веб-сайта и области администрирования	63
Инсталляция другого файла языка	63
Перевод пунктов меню	65
Модификация имени меню	65
Изменение шаблона для веб-сайта	66
Изменение цветов шаблона	66
Резюме	66
Глава 5. Конфигурация администрирования Joomla!	67
Компоновка экрана	67
Панель меню	67
Панель инструментов	68
Подменю	69
Фильтрующие элементы	69
Область контента	69
Списки	69
Диалоги	70
Прочее	70
Справка	70
Пиктограмма вызова справки	70
Справка всплывающими подсказками	71
Меню справки	71

Пункт Joomla! Help	71
Пункт System Info	71
Резюме	72
Глава 6. Меню Site	73
Пункт Control Panel	73
Пункт User Manager	74
Пиктограмма Logout	75
Пиктограмма Delete	75
Пиктограмма Edit	75
Пиктограмма New	78
Область Media Manager	79
Создание каталогов	80
Загрузка файла	80
Раздел Global Configuration	81
Настройки Site	82
Настройки System	85
Настройки Server	90
Пункт Logout	93
Резюме	93
Глава 7. Меню Menus	95
Пункт Menus	96
Настройка существующего меню	96
Пиктограмма Menus	97
Пиктограмма Default	97
Пиктограмма Publish и Unpublish	97
Пиктограмма Move	97
Пиктограмма Copy	98
Пиктограмма Trash	98
Пиктограмма Edit	98
Пиктограмма New	104
Меню Trash	105
Создание нового меню	105
Резюме	109
Глава 8. Меню Content	111
Пункт Article Manager	112
Описание списка статей	113
Пиктограммы редактирование статьи	114
Создание новой статьи	116
Параметры	117
Изображения	120
Корзина для статей	123
Диспетчер разделов	123
Редактирование разделов	123
Диспетчер категорий	126
Диспетчер титульных страниц	127
Редактирование контента из интерфейсной части	128
Резюме	128

Глава 9. Меню Components	129
Меню Banners	129
Баннеры	129
Клиенты	130
Управление баннерами	130
Графические баннеры	131
Текстовые ссылки	133
Контакты	133
Диспетчер контактов	133
Создание ссылки меню для веб-сайта	135
Категории	137
Ленты новостей	137
Наполнения	138
Категории	139
Опросы	139
Статистика поиска	140
Веб-ссылки	141
Ссылки	141
Категории	142
Резюме	142
Глава 10. Меню Extension	143
Пункт Install/Uninstall	143
Вкладки Install, Components, Modules, Plugins, Languages и Templates	144
Диспетчер модулей	144
Фильтры модулей	146
Модули сайта	146
Копирование модуля	155
Модули администратора	155
Диспетчер подключаемых модулей	157
Подключаемые модули аутентификации	157
Модуль Authentication – Joomla!	157
Модуль Authentication – LDAP	158
Модуль Authentication – OpenID	158
Модуль Authentication – GMail	158
Подключаемые модули контента	159
Подключаемые модули редакторов	160
Подключаемые модули Editors-xtd	160
Подключаемые модули поиска	160
Системные подключаемые модули	161
Подключаемые модули пользователя	161
Подключаемые модули XML-RPC	162
Модуль XML-RPC – Joomla!	162
Диспетчер шаблонов	162
Шаблоны сайта	162
Шаблоны администратора	164
Диспетчер языков	164
Резюме	164

Глава 11. Меню Tools	165
Система обмена приватными сообщениями	165
Массовая рассылка	166
Глобальная регистрация контента	167
Резюме	167
Глава 12. Дополнительные сведения о расширениях	169
Fireboard Forum	170
Инсталляция	170
Конфигурирование	171
Создание форума	172
Интеграция в веб-сайт	173
DOCman (раздел Download, Document Manager)	174
Инсталляция	174
Конфигурирование	175
Язык	176
Интеграция в веб-сайт	177
Подготовка контента	178
Подключаемый модуль поиска	179
Дополнительный модуль	180
Галерея Exposé	180
Инсталляция	181
Интеграция в веб-сайт	183
Диспетчер альбомов в административном разделе	185
Деинсталляция	187
Диспетчер альбомов как автономная программа	187
Интеграция в каркас Joomla!	187
Проблемы с компонентами от независимых поставщиков	188
Обновления	188
Безопасность	188
Что делать?	188
Резюме	189
Глава 13. Написание собственных шаблонов Joomla!	191
Корпоративная идентичность	191
HTML/ XHTML, CSS и XML	191
HTML/ XHTML	192
CSS	192
XML	193
Создание выших собственных шаблонов	193
Концепция	193
Преобразование в HTML	195
Структура каталога шаблона	197
Первый пробный запуск	198
Интеграция модуля Joomla!	199
Создание пакета шаблона	201
Какой исходный код поступает от Joomla!	201
Резюме	204

Глава 14. Свобода от барьеров и Joomla!	205
Что такое свобода от барьеров?	205
Общие положения о критериях доступных сайтов	208
Отделение контента от компоновки	208
Требования к дизайну контента	209
Визуальная и соответствующая тексту организация контента	210
Выбор цветов	210
Контрасты	211
Изменяемые размеры шрифтов	211
Масштабируемые компоновки	212
Графика	212
Достаточно большие навигационные элементы	213
Формы	213
Свобода от барьеров в Joomla! 1.5 возможна с помощью Beez	215
HTML	215
Метки для прыжков	217
index.php	217
Ссылки пропуска в формах	217
Beez и модули	218
Компонент com_content	218
Формы	219
Таблицы данных	219
Дизайн и CSS	219
Внутренние CSS-файлы Beez	220
Позиционирование	220
Компоновка	220
Прочее	221
Внутренние средства доступа Joomla!	222
Дополнительная информация	224
Вспомогательные технологии	224
Инструменты проверки CheckerTools	224
Рекомендуемые книги	224
Резюме	224
Глава 15. Собственные компоненты, модули и подключаемые модули	225
Концепция "модель-представление-контроллер"	225
Преимущества концепции MVC	226
Пример компонента helloworld	227
Домашний каталог helloworld	227
Пример компонента	232
Таблица MySQL	233
Интерфейсная часть	233
Администрирование com_auto	236
Тестирование	244
Создание инсталляционного пакета	244
Модули	245
Исходный код	245
Инсталляция	247
Представление на веб-сайте	247
Подключаемые модули	248
Резюме	250

Глава 16. Построение вуб-сайта с помощью Joomla!	251
Идея	251
Подготовка	252
Логотип и внешний вид	252
Технические преобразования	253
Структура меню	257
Пользовательский раздел	261
Первые результаты	263
Мосье Бертран изучает CSS	264
Форматирование шаг за шагом	266
Изменения для Joomla! версии 1.5	269
Инсталляция на веб-сервере	269
Как выполняется инсталляция	270
Инсталляция Joomla! у провайдера	270
Импортирование данных	271
Права доступа к файлам и каталогам	272
Механизмы поиска	273
SEF URL	273
Мета-дескрипторы	273
Инсталляция шаблона	273
Благополучное завершение	275
Как это работает в действительности?	275
Резюме	275
Глава 17. Шаблоны-бонусы	277
NGO	277
Том Бохачек	278
Шаблон World Knowledge	278
Творческий подход	278
Структурный подход	279
Деловое предприятие	279
Энди Миллер	279
Резюме	280
Приложение А. Онлайн-ресурсы	281
Приложение Б. Модули шаблонов	282
Приложение В. Как изменить изображение (логотип) в шаблоне?	283
Приложение Г. Joomla! API	284
Приложение Д. Забыли пароль администратора?	285
Приложение Е. Переход от Joomla! 1.0.x к Joomla! 1.5	286
Приложение Ж. Безопасность без глобальных переменных	291
Предметный указатель	292

Об авторе

Хаген Граф (Hagen Graf) родился в июле 1964 г. в Нижней Саксонии, Германия. Первое знакомство с компьютерами у него произошло в конце семидесятых, с Radioshack TRS 80. Как специалист по продажам, он организовал свои данные о клиентах, самостоятельно разработав соответствующие приложения. Это обеспечило ему значительные преимущества перед другими продавцами. Намереваясь повысить свою квалификацию, он записался на вечерние курсы программирования и, таким образом, стал программистом. В настоящее время он работает в консалтинговой компании своей жены в качестве инструктора, консультанта и программиста (<http://www.cocoate.com>). Хаген Граф опубликовал в Германии ряд книг, посвященных веб-серверу Apache, проблемам безопасности Windows, продуктам Mambo и Drupal. С 2001 г. он активно участвует в работе некоммерческого учебного сообщества под названием “machmit.org e.V.”, а также в нескольких национальных и международных проектах. Все проекты связаны с управлением содержимым, построением сообществ и использованием социального программного обеспечения вроде wiki и weblog. Он выбрал Joomla! CMS из-за простоты и легкости администрирования этого продукта. Вы можете ознакомиться и оставить комментарии в его блоге по адресу <http://www.bloghouse.org/en/hagen>.

Предисловие

Joomla! — это целый мир!

Joomla! — проект с открытым исходным кодом, который находится в постоянном движении. Это движение непредсказуемо, иногда неопишимо, отчасти противоречиво, время от времени несколько сонное и “провинциальное”. Несмотря на это, а может быть, именно потому данный проект стал исключительно успешным за последние два года и завоевал популярность у миллионов пользователей по всему миру.

Существует стабильная, широко используемая и популярная версия 1.0x. За последние два года в результате усилий разработчиков она претерпела существенные изменения.

Два впечатляющих года минуло с момента основания Joomla! 2005.

Команда Joomla! сформировалась за эти годы, расширив Joomla! 1.0 до 1.0.13, и осуществив теперь значительный шаг вперед с версией 1.5.

Пользователи системы также не оставались в стороне. Многие из них перевели свои веб-сайты с Mambo на Joomla! Однако, хотя к сообществу Joomla! за это время присоединилось множество новых пользователей, в мире все еще остается много людей, которые ничего не знают об этой системе.

Joomla! — наиболее широко распространенная система управления веб-содержимым (Web Content Management System — WCMS).

Спустя год после основания проекта — осенью 2006 г. — команде Joomla! было известно примерно о 5 000 000 инсталляций Joomla! на общедоступных веб-серверах, используемых более или менее постоянно. На тот момент существовало 45 000 зарегистрированных разработчиков с 1100 проектами, расширяющими Joomla! дополнительной функциональностью. На форуме *joomla.org* насчитывалось около 450 000 сообщений от 50 000 пользователей.

А вот статистика, полученная годом позже, в ноябре 2007 года:

- более 20 000 000 инсталляций;
- всего 28 членов команды разработчиков и 16 членов команды ядра;
- более 2000 проектов по расширению Joomla!;
- более миллиона сообщений от 100 000 пользователей в форуме Joomla!.

То есть наблюдается стопроцентный рост всего за один год!

Диапазон веб-сайтов, построенных на основе этой технологии, простирается от очень простых домашних страниц до сложных бизнес-приложений. В этой книге я объясняю, почему Joomla! настолько успешна, и как вы можете использовать ее.

Слово Joomla! происходит от “Jumla”, что на суахили означает “все вместе”.

Joomla! — результат серьезных противоречий между фондом Mambo Foundation, который был основан в августе 2005 г., и его командой разработчиков. Joomla! продолжила разработку успешной системы Mambo и, подобно Mambo, представляет собой программное обеспечение, которое обеспечивает простое администрирование веб-сайтов из веб-браузера.

Согласно ее собственному описанию, Joomla! — это “эксклюзивная система управления содержимым и одна из наиболее мощных систем управления содержимым с открытым кодом в мире. Она используется по всему миру для управления от простых домашних страниц до сложнейших корпоративных веб-сайтов. Система проста в инсталляции, управлении и очень надежна”.

О чем эта книга

Прежде всего, эта книга, естественно, о Joomla! и о том, как ее использовать. Joomla! представляет собой инструмент, обладающий мириадами опций, и в зависимости от вашего воображения и существующих потребностей, вы можете использовать его самыми разнообразными способами. Для того чтобы вы почувствовали себя уверенно с этим инструментом, я разделил книгу на следующие главы.

- В главе 1 описаны термины и соглашения, которые облегчают работу с Joomla!.
- Глава 2 посвящена установке Joomla! в разных средах.
- В главе 3 предлагается обзор, состоящий из экскурсии по структуре примера данных, доступных после установки Joomla!.
- В главе 4 рассматривается настройка языка и шаблонов Joomla!.
- В главе 5 рассказывается об операциях административной области, о ее конфигурировании и об администрировании контента. В ней также описаны элементы панели меню, панели инструментов и меню Help (Справка).
- Глава 6 посвящена настройке меню Site.
- В главе 7 рассматривается настройка меню Menus.
- Глава 8 посвящена настройке меню Content.
- В главе 9 рассказывается о настройке меню Components.
- В главе 10 описано меню Extensions.
- Глава 11 посвящена меню Tools, которое содержит административные инструменты: систему обмена приватными сообщениями, функцию массовой рассылки, а также глобальные регистрации элементов контента.
- В главе 12 представлены примеры возможностей расширений Joomla!.
- Глава 13 посвящена дизайну веб-сайта и созданию ваших собственных шаблонов.
- Глава 14 написана Энджи Радтке (Angie Radtke). Энджи – признанный эксперт в области свободы от барьеров Joomla!, к тому же она участвовала в создании свободного от барьеров шаблона Beez вместе с Робертом Деуцем (Robert Deutz).
- Глава 15 научит вас разрабатывать свои собственные расширения.
- Глава 16 поможет создать практическое приложение с помощью Joomla! – от концепции до реализации веб-сайта.
- В главе 17 представлены два шаблона, которые вы можете использовать для построения собственных веб-сайтов.

В приложениях вы найдете полезные детали по обновлениям, системе безопасности и другим важным вещам.

Соглашения

В этой книге вы встретите много стилей текста, которые подчеркивают отличия между разными типами информации. Ниже приведены некоторые примеры этих стилей вместе с объяснением их смысла.

Для кода используются три стиля. Кодовые слова в тексте выделены следующим образом: “Если вы заглянете в базу данных, то обнаружите множество компонентов в подкаталоге [Путь_к_Joomla!]/components, и среди них – компонент com_contact”.

Блок кода будет выглядеть так:

```
main .leading h2,#main2 .leading h2 {
background:#EFDEEA;
border-bottom:solid 0 #333;
color:#93246F;
font-family:trebuchet MS, sans-serif;
font-size:1.4em;
font-weight:normal;
```

Любой ввод и вывод командной строки будет записан следующим образом:

```
/etc/init.d/mysql start
```

Новые термины и важные слова представлены шрифтом с полужирным начертанием. Слова, которые вы видите на экране, в меню или диалоговых окнах примера, появляются в тексте так: “Щелкните на кнопке Next (Далее) и переходите к шагу лицензирования”.



В таких врезках даются важные замечания.



В таких врезках представлены советы и подсказки.

Коды примеров

Коды примеров, рассмотренных в книге, доступны для загрузки как на странице, посвященной книге (http://www.packtpub.com/files/code/5302_Code.zip), так и на Web-сайте издательства.

От издательства

Вы, читатель этой книги, и есть главный ее критик и комментатор. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо, либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится или нет вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг. Наши координаты:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Информация для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152

1

Термины, концепции и соображения

Прежде чем погрузиться в Joomla!, позвольте мне объяснить несколько терминов и концепций, а также изложить некоторый основополагающий материал.

Системы управления содержимым

Система управления содержимым (Content Management System – CMS) – понятие, которое содержит термины "содержимое" и "управление", что однозначно указывает на некий комплекс, управляющий контентом. Такой комплекс может быть классной доской и кусочком мела, или же некоторой общедоступной онлайн-энциклопедией вроде Wikipedia, либо онлайн-аукционом наподобие eBay. Во всех этих примерах присутствует административное управление контентом, а в последнем случае – с многочисленными участниками.

Эти участники играют главную роль в системах управления содержимым, с одной стороны как администраторы, а с другой – как пользователи и редакторы. Помимо CMS, вам встретятся такие термины, как **ERP** (Enterprise Resource Planning System – Система планирования бизнес-ресурсов), **CRM** (Customer Relationship Management System – Система управления взаимоотношениями с клиентами), **DMS** (Document Management System – Система управления документооборотом), **HRM** (Human Resource Management System – Система управления персоналом) и многие другие.

Определить термин CMS довольно трудно из-за его обобщающей природы и разнообразия функций. Определение, приведенное в Wikipedia – мое любимое:

Система управления содержимым, или CMS– это компьютерная программная система, используемая для обеспечения и организации совместного процесса создания и редактирования текстовых и мультимедийных документов (контента).

Аббревиатура **ECMS** расшифровывается как Enterprise Content Management System (Система управления содержимым масштаба предприятия). Другие перечисленные выше аббревиатуры являются подмножествами ECMS.

Joomla! относится к категории **WCMS** (Web Content Management Systems – система управления веб-содержимым), поскольку ее функциональность администрируется из браузера в Web.

Вообще термин "управление содержимым" используется в сочетании с веб-страницами, которые могут поддерживаться браузером. Пожалуй, нет необходимости еще более упрощать определение.

Краткий экскурс в историю

В то время как Sun Microsystems предложила в 90-х гг. свою концепцию "сеть — это компьютер", Microsoft не собиралась успокоиться до тех пор, пока компьютер под управлением Windows не окажется на каждом рабочем столе. Теперь, когда все это стало реальностью, Microsoft смогла отдохнуть, и теперь в действительности ищет новые рынки и новые продукты.

Компьютер рассматривался Microsoft как смесь файлов данных и исполняемых двоичных файлов. Файлы с исполняемым двоичным контентом называются **программами**; они покупались и устанавливались потребителями для манипулирования данными. Microsoft Office стал несомненным победителем в большинстве офисных продуктов по всему миру.

Компьютер, над которым работала Sun, был дешевым, неинтеллектуальным терминалом с экраном, клавиатурой, мышью и доступом в Интернет. Программы и данные не хранились на таком компьютере, а находились где-то в сети.

Практика Microsoft руководствовалась философией "мое", в то время как Sun взяла на вооружение философию "наше".

Главной мотивацией выбора философии в обеих компаниях был коммерческий интерес. Microsoft, прежде всего, продвигала программное обеспечение для ПК на потребительский рынок. Sun же предлагала серверное оборудование и программы для рынка крупных предприятий.

Интернет, изобретенный в шестидесятых годах, претерпел взрывоподобный рост в середине 90-х. Помимо прочего, его популярности способствовал язык гипертекстовой разметки **Hyper Text Markup Language (HTML)** — язык, используемый для написания веб-страниц, — а также разработка веб-серверов и веб-клиентов (браузеров).

Сам Интернет — это просто набор правил, которые должны понимать различные устройства, чтобы взаимодействовать друг с другом осмысленным образом по всему миру, причем практически мгновенно.

Человек, не имеющий адреса электронной почты, отныне не может быть доступен, а компания без веб-сайта не только старомодна, а просто не существует в глазах многих клиентов. Весь мир за короткое время ринулся в Интернет, чтобы стать его частью. Такие фильмы, как "Матрица", стали всемирными хитами, вытеснив "1984" — книгу Джорджа Оруэлла о фантастическом тоталитарном обществе.

Те, кто привык покупать программы, приобретают редакторы HTML и создают с их помощью собственные Интернет-страницы. Другие предпочитают писать свой собственный код HTML в любом текстовом редакторе, который у них есть. И как грибы стали появляться веб-студии, где любой может заказать себе веб-страницу.

Обе группы столкнулись с проблемой, связанной с тем, что страницы HTML были статичными. Чтобы изменить контент страницы, ее необходимо было модифицировать на локальном ПК и затем копировать на сервер. Это было не только затруднительно и дорого, но делало невозможным представление в Web таких вещей, как eBay или Amazon.

Обе группы выработали собственные решения, которые более или менее решали эту проблему.

Фракция "мое" разработала быстрые двоичные, программы, с помощью которых можно быстро генерировать HTML-страницы и загружать их посредством автоматизированных процедур на сервер. Интерактивные элементы, такие как счетчики посетителей, среди прочих, встраивались в такие страницы.

Фракция "наше" разработала Java-апплеты и с ними — возможность написания программ, располагаемых централизованно на сервере и управляемых через браузер. На основе этого решения были реализованы такие бизнес-идеи, как онлайн-системы заказов и системы резервирования билетов.

Обе группы старались расширить свое присутствие на рынке различными способами. В результате для обеих сформировались более-менее стабильные сегменты рынка, между которыми идут постоянные битвы вокруг вопросов выбора правильной операционной системы (Windows, Linux или Mac OS X), постоянно стимулируя рост номеров версий. Пользователи же осознали тот факт, что все не так просто.

Но в подобных ситуациях бывает и третий выбор. В нашем случае, помимо прочих, он проявился в появлении языков сценариев с открытым кодом, таких как PHP (<http://www.php.net>). Расмус Лердорф (Rasmus Lerdorf) хотел предоставить некоторые интерактивные элементы на своей домашней странице, а из этого родился новый язык программирования. С самого начала PHP был оптимизирован для взаимодействия с базой данных MySQL, которая также распространялась на основе лицензии GNU/GPL.

К счастью, уже существовала операционная система Linux и веб-сервер Apache, которые обеспечили необходимую инфраструктуру на сервере. Устройством отображения на стороне клиента был браузер. Аббревиатура **LAMP** (Linux, Apache, MySQL и PHP) стала синонимом интерактивного, поддерживаемого базой данных, представления информации в Интернете.

Самые разнообразные системы, такие как форумы, сообщества, онлайн-магазины, страницы опросов и другие подобные вещи, обеспечивающие возможность организации контента с помощью браузера, были разработаны сообществом творческих энтузиастов.

Вскоре после создания "трудных" частей вроде Linux и Apache были разработаны и "мягкие" продукты.

Девяностые годы подходили к концу, "мыльный пузырь" Интернет-бизнеса лопнул, и все стали возвращаться к традиционной классической бизнес-модели.

Там, где никак не удается экономить, особое внимание обращается на поиск снижения издержек. И здесь, как и бывало ранее, оказалось немало вариантов! PHP-приложения распространились в миллионах экземпляров. В качестве примеров стоит лишь взглянуть на проекты phpBB (<http://www.phpbb.com/>) и phpMyAdmin (<http://www.phpmyadmin.net/>). Один из них стал квазистандартом для программного обеспечения форумов, а второй — стандартом для манипуляций базами данных MySQL через веб-интерфейсы.

Исходный код языка PHP и приложений становились все лучше и лучше с ростом количества пользователей и разработчиков. Чем больше запускалось проектов, тем более успешными они становились. Индивидуальные гуру смогли экономить предприятиям огромные суммы денег, почти без затрат времени.

Статические HTML-страницы теперь рассматриваются как дорогие и устаревшие, и подвергаются пересмотру. Они обязаны стать динамическими! После десятилетия напряженной работы разработчиков Linux, Apache, MySQL и PHP были охотно приняты индустрией. Начался поиск профессионально используемых приложений PHP.

К этим приложениям предъявляются следующие требования.

- Простой процесс инсталляции.
- Легкость обслуживания исходного кода.
- Безопасность исходного кода.
- Дружественность к пользователю.
- Простая расширяемость.
- Простота разработки.
- Легкость обучения новых разработчиков.
- Стандартизированный интерфейс с другими программами.
- Низкая стоимость.
- Независимость от поставщика.

Особым преимуществом приложений PHP является независимость от аппаратного обеспечения и операционной системы. LAMP также существует в виде WAMP (Windows, Apache, MySQL и PHP) для Windows, MAMP (Mac, Apache, MySQL и PHP) для Apple, а также для ряда других платформ. И, наконец, картину дополняет Joomla!.

История разработки Joomla!

Австралийская компания Miro (<http://www.miro.com.au>) в 2001 г. разработала CMS под названием Mambo. Она сделала эту систему доступной как программное обеспечение с открытым кодом — для тестирования и обеспечения более широкого распространения. В 2002 г. компания разделила свой продукт Mambo на коммерческую версию и версию с открытым кодом. Коммерческий вариант был назван Mambo CMS, а версия с открытым кодом называлась **Mambo Open Source**, или сокращенно — **MOS**. К концу 2004 г. все участники пришли к соглашению, что версия MOS должна официально называться **Mambo**, и с этого момента успешное будущее для самой быстро разрабатываемой CMS было обеспечено.

Преимущества коммерческой версии в основном касались повышенной безопасности для поддержки разработки со стороны компании Miro.

Версия с открытым исходным кодом имела преимущество бесплатного распространения и огромное сообщество пользователей и разработчиков, которые осуществляли постоянное ее расширение и совершенствование. Вдобавок у предприятий появилась возможность взять систему Mambo в качестве основы и строить собственные решения поверх нее.

Чтобы обеспечить существование и постоянное развитие Mambo, в течение 2005 г. велись дискуссии об основании фонда для версии Mambo с открытым кодом. Осенью 2005 г. руководство Mambo Foundation анонсировало страницу проекта Mambo. После получения положительной реакции в течение первых нескольких часов стало очевидным, что фонд основала компания Miro из Австралии, и что она не планирует включать в дальнейшую работу прежнюю команду разработчиков. На форумах сообщества развернулись горячие дискуссии, а команда разработчиков замолчала на несколько дней.

Вскоре команда выработала совместную позицию и опубликовала ее на opensource.matters.org, сообщив, что после консультаций с независимой адвокатской конторой Software Freedom Law Center (<http://softwarefreedom.org/>) они ре-

шили продолжить разработку Mambo под собственную ответственность. Информация об усовершенствованной системе Mambo, основанной на новом исходном коде, медленно появилась на форумах.

Вскоре "война Алой и Белой розы" между контролируемым Migo фондом Mambo Foundation, который вдруг лишился команды разработчиков, и самой командой разработчиков, которая, конечно же, нуждалась в новом имени для своего продукта, затронула международное сообщество из сотен и тысяч пользователей. Противоположные стороны клеймили друг друга в блогах, форумах и на соответствующих страницах проектов.

А между тем разработка обоих проектов продолжалась. Ответвление продукта получило название Joomla!. Команда разработчиков придавала большое значение соблюдению принципов демократии. Новый проект нуждался в логотипе, и был объявлен конкурс в "новом" сообществе. К этому моменту на новом форуме зарегистрировалось около 8000 пользователей.

Mambo Foundation вскоре после этого представила свою новую команду разработчиков.

Версия 1.0 продукта Joomla! была опубликована 17 сентября 2005 г.

Довольно скоро многие независимые разработчики, а также группы, которые занимались разработкой компонентов Mambo, перевели проекты Simpleboard (сегодня – FireBoard), DOCman и многие другие на платформу Joomla!, тем самым выразив доверие новому проекту. Префикс `mos`, который использовался во многих переменных и терминах, быстро превратился в `jos`.



Детальное изложение этих событий доступно в Интернете по адресу:

<http://www.devshed.com/c/a/BrainDump/Joomla-is-the-New-Mambo/>

Через два года после своего основания Joomla! стал одним из наиболее популярных проектов с открытым кодом в мире. Однако за эти два года много времени было потрачено на организационные пертурбации и преодоление разногласий из-за реструктуризации.

Web 2.0 отпраздновал свою победу. Создаваемый пользователем контент становится все более и более важным. Язык программирования Ruby и, в частности, Ruby on Rails, все больше и больше использовались для разработки веб-сайтов. Программные интерфейсы играли все более важную роль.

Система Joomla 1.0x стала выглядеть несколько устаревшей и серой в такой компании. Она не предоставляла широких возможностей разработки, а пользователи жизненно нуждались в системах вроде Plone, ТурoЗ, Drupal и многих других, которые не имели проблем с размерами и гибкостью, и позволяли постоянно включать новые технологии в свои проекты.

За последние два года Joomla! версии 1.0x уже не расширялась далее, хотя обновления, связанные с безопасностью, с минимальными изменениями кода периодически выходили. Двухлетнее ожидание новой версии – не слишком хорошая реклама проекта. Уже раздавались голоса, что Joomla! почилa, она подвергалась насмешкам, появлялись статьи под заголовками вроде "Как разрушить невероятный программный проект".

Однако команда разработчиков выпустила, наконец, Joomla! версии 1.5, вызвав оживление на форумах, списках рассылки и личных дискуссиях. Теперь в моих глазах будущее Joomla! выглядит более радужным, чем когда-либо ранее.

С одной стороны, переход от версии 1.0x к 1.5x требует приложения некоторых усилий, поскольку не обеспечена полная обратная совместимость из-за изменений в исходном коде. Однако, с другой стороны, новая версия определенно переводит Joomla! в более высокую лигу промышленных систем управления содержимым. Умелая стратегия создания строительных блоков с помощью Joomla! (каркас Joomla!), которая позволяет более широкому кругу разработчиков участвовать в разработке, принесла свои дивиденды.

Наконец-то стало возможным выбирать различные методы аутентификации для создания безбарьерных (barrier-free) веб-сайтов на различных языках и включиться в основной поток Web 2.0, который захлестнул большинство предприятий. Теперь открыта дорога к разработке компонентов в таких средах, как Eclipse.

С существующим сообществом, разработчиками и плотностью инсталляций Joomla! легко покрывает многие секторы рынка. Многие хост-провайдеры уже предоставляют в распоряжение своих клиентов предварительно инсталлированную систему Joomla!, чтобы они не имели серьезных проблем с переходом от более простых веб-сайтов.

Структура системы управления веб-содержимым

Используя Joomla! в качестве примера, я кратко объясню структуру WCMS.

Клиентская и серверная часть

WCMS состоит из клиентской и серверной частей. Клиентская часть — это веб-сайт, который видят посетители и зарегистрированные пользователи.

С другой стороны, серверная часть содержит слой администрирования веб-сайта, с которым имеет дело администратор. Конфигурация, обслуживание, очистка, генерация статистики и создание нового контента — все это делается в серверной части авторизованным персоналом. Серверная часть доступна через другой URL, отличный от того, по которому доступен сам веб-сайт.

Права доступа

Всегда, когда речь идет об управлении, мы говорим о интеллектуальном администрировании существующих ресурсов. В WCMS имена пользователей и групп присваиваются людям, вовлеченным в этот процесс, и каждому из них выдаются определенные различные права доступа. Они простираются от прав простого зарегистрированного пользователя, через права "автора" и "редактора", вплоть до "супер-администратора", который имеет полный контроль над доменом. В зависимости от этих прав, веб-сайт отображает для каждого из этих категорий пользователей разный контент; открывается доступ к редактированию контента непосредственно на стороне клиента, или же пользователю выдается право работать на стороне сервера.

Контент

Контент (content), или содержимое, может принимать всевозможные формы; в простейшем случае — это текст. Однако контент также может быть изображением или ссылкой, музыкальным фрагментом, фрагментом приложения вроде Google Map или комбинацией всего перечисленного. Чтобы сохранять контроль над контентом, он

структурируется; например, текст разбивается на разные категории. Разумеется, категории также являются контентом, которое нуждается в администрировании. Очень популярными стали новостные ленты (newsfeeds), с тех пор, как Explorer 7 получил возможность чтения RSS-каналов, учитывая, что этот браузер применяется на более чем 80% компьютеров. Ленты новостей дают возможность интегрировать информацию из других источников. Интеграция, категоризация, рейтинги, перемешивание и обновление контента лент новостей становятся все более и более важными.

Расширения

Компоненты, модули, шаблоны и подключаемые модули — все это известно под общим термином расширений (Extensions). Они предоставляют дополнительные функции, которых нет в ядре Joomla!

Компоненты

Joomla! должна была быть расширяемой системой, способной расти вместе с предъявляемыми требованиями. Расширения, обеспечивающие дополнительную функциональность и имеющие собственные области в администрировании Joomla!, называются **компонентами**. Например, типичными компонентами последних лет стали онлайн-магазин, галерея картин, рассылка новостей или система форумов. Сегодня такие вещи, как оптимизация механизма, права доступа пользователей, многостраничные формы и переменная структура контента становятся все более и более важными.

Компоненты содержат бизнес-логику своих сайтов и отображают контент в "основном теле" веб-сайта.

Шаблоны

Шаблон (template) — это тип визуального редактирования, который накладывается на контент. Шаблон определяет цвета, шрифты символов, их размеры, фоновые изображения, интервалы и разбиение страницы на части — другими словами, все, что касается внешнего вида страницы. Шаблон состоит, по крайней мере, из одного файла HTML для структуры страницы и одного файла CSS для дизайна. Он может также иметь намного более сложную структуру, чтобы подготовить контент Joomla! для свободы от барьеров либо для совершенно других целей.

Подключаемые модули

Подключаемый модуль (plug-in) — это кусок программного кода, который добавляется в определенные места каркаса Joomla!, чтобы изменить его функциональность. Такой подключаемый модуль может, например, использоваться внутри текста контента, чтобы загружать в него контент программного модуля. Подключаемые модули также применяются для комплексного поиска на веб-сайтах, который задействует дополнительные компоненты. Иногда подключаемые модули используются в Joomla! в качестве макроязыка. Специальные модули, принадлежащие компонентам, используются для интеграции контента в шаблоны в желаемой форме. Модуль *Recent News* (Последние новости), например, доставляет в шаблон заголовки последних пяти статей, которые были помещены *компонентом контента* (Content component). Другой модуль может определять количество пользователей, находящихся на связи в данный момент, и отображать полученный результат.

Рабочий поток

Под рабочим потоком (workflow) понимается последовательность операций. Бюрократические процедуры (подпись, печать, картотека) — пример рабочего потока. Рецепт приготовления пирожного — тоже рабочий поток. Поскольку несколько людей обычно работают с контентом CMS, хорошо организованный рабочий поток чрезвычайно полезен.

В прошлом это называлось просто планами работ, которые имелись у определенных пользователей. Например, редактор видит список неопубликованных фрагментов новостей, которые он должен проверить. После их проверки редактор помечает эти фрагменты как корректные, и они появляются в плане работ публикатора. Публикатор затем решает, нужно ли публиковать каждый фрагмент на лицевой странице.

Конфигурационные установки

Настройки, касающиеся всего веб-сайта, специфицируются в конфигурационных установках. Сюда относится текст заголовка в окне браузера, ключевые слова для поисковых механизмов, переключатели, разрешающие или запрещающие протоколирование на сайте, либо включающие и выключающие доступ ко всей странице, и множество других функций.

API

В наше время, чтобы идти в ногу со временем, любая система должна предоставлять интерфейс прикладного программирования (**Application Programming Interface — API**). Должна быть возможность доступа к Joomla! из других программ, чтобы до определенной степени обращаться к ней удаленно. Это открывает возможность создания совершенно новых приложений, которые ранее были невозможны с Joomla!. API — наиболее важная связь между независимыми разработчиками расширений и ядром Joomla!.

Является ли Joomla! частью “недвижимости”?

Joomla! — разновидность конструкторского инструментария, который, будучи однажды инсталлированным на сервере, позволяет создавать и сопровождать веб-сайт. Joomla! подобна дому, который вы строите на своем участке, и который впоследствии можете постепенно отделять по своему вкусу. То есть, в определенном смысле это — “недвижимость”.

Стоп! Но я же постоянно говорил о мобильности, а теперь — о постройке недвижимости? Не волнуйтесь, эта недвижимость, которую вы строите в одном месте (на вашем сервере), доступна отовсюду. Чтобы заселить часть вашей недвижимости, понадобятся такие службы, как отопление, электричество и водоснабжение. Вот почему важно обеспечивать максимальную безопасность Joomla! на сервере, чтобы “электро-снабжение” в вашем доме никогда не отключалось. То есть необходимо обеспечить режим работы 24/7 (24 часа в день, 7 дней в неделю).

Как и реальный дом, ваш экземпляр Joomla! имеет определенную планировку комнат. Имеется гостиная (для показа контента), кухня и комната переговоров (форум), рабочий кабинет (административная область) и личные апартаменты, которые вы показываете только близким друзьям (членская область). Возможно, также есть некоторая большая комната, которая объединяется все это.

Неважно, какую именно планировку вы выберете — в любом случае вы должны обставить свой дом, настелить красивый пол, наклеить обои, развесить несколько картин по стенам (шаблонам) и, конечно, регулярно проводить уборку. Это потому, что многие гости оставляют следы, и не всегда желательные.

Посетители должны иметь возможность найти ваш дом. Поэтому как можно больше людей должны знать его адрес. Поскольку в Интернете нет всеобщей регистрационной службы, вам следует самому позаботиться об этом, задав себе вопрос: "Как меня можно будет найти?".

Возможно, у вас также есть сад, окружающий дом, в который ведут несколько ворот. Есть официальный главный вход, задняя дверь и, возможно, другая садовая калитка для близких друзей.

А может быть, вам вообще не нравятся такие дома, и вы предпочитаете трейлеры, палатки, передвижные домики, отели или же жизнь в коммунах — вы согласны платить ренту и не беспокоиться о деталях.

Если вы примените последние несколько предложений к вашему веб-сайту, то увидите, насколько важно знать, чего именно вы хотите, кто вы есть по натуре, и как вы хотите выглядеть в глазах вашего сообщества. Никто не может обойтись без общения! Однако любой может оказаться неправильно понятым. Так что ответственно подходите к планированию собственного дома в Интернете. Продумайте текст, интерактивные элементы вроде календаря или формы и, конечно, область, доступную только зарегистрированным пользователям. Подумайте об указателях, которые помогают пользователям, но не навязывают им ничего, и, наконец, посмотрите, как все это делают другие. Поговорите с людьми, которым вы хотите адресовать свой веб-сайт. Им будет приятно высказать свои пожелания заранее, и они, возможно, дадут ряд полезных советов, которые совсем не очевидны при рассмотрении темы с вашей точки зрения. Это экономичный и очень эффективный способ накопить идеи и избежать ошибок. Вложите душу и сердце в то, что совершенно необходимо для успеха будущего сайта.

Версии Joomla!

Как любое программное обеспечение, Joomla! имеет свою историю разработки. 1 сентября 2005 г. команда Joomla! опубликовала "дорожную карту", начинающуюся с Joomla! версии 1.0.

Первая версия Joomla! естественно, получила номер 1.0, чтобы не было никакой путаницы с существующими версиями Mambo. Версия 1.0— это пересмотренная последняя версия Mambo под номером 4.5.2.3. Пересмотру подверглись: название, известные ошибки и заплатки безопасности. А тем временем, вышла версия Mambo под номером 4.6.2.

За последние два года вышло 13 версий Joomla! 1.0.x, которые совершенствовали и исправляли некоторые детали в коде. Если вы проследите за разработкой, то отметите, что от версии к версии Joomla! становилась все более надежной.

Система нумерации версий Joomla!

Нумерация версий Joomla! состоит из трех частей.

- **Шаг 1 — старший номер выпуска.** Увеличивается при внесении фундаментальных изменений на уровне исходного кода. Версии с более высоким номером иногда не совместимы с более ранними версиями.

- **Шаг 2 — младший номер выпуска.** Увеличивается при внесении существенных изменений в функциональность. Версия с более высоким номером обычно совместима (с небольшой подгонкой) с более ранними версиями.
- **Шаг 3 — номер выпуска сопровождения.** Увеличивается при исправлении ошибок и брешей в безопасности. Увеличение этого номера свидетельствует лишь о небольших изменениях и очень незначительных новых средствах. Эти версии полностью совместимы с версиями, имеющими тот же номер на шагах 1 и 2.
- **Полный выпуск.** Выпускается при изменении номеров на шагах 1 и 2. При этом имеется определенный период альфа- и бета-тестирования. Длительность периода тестирования не фиксирована и определяется командой разработчиков. Бета-версии должны быть доступны для тестирования в течение не менее трех недель, чтобы дать время разработчикам компонентов подстроить свои компоненты.
- **Выпуск поддержки.** Этот выпуск может быть использован немедленно.

Версия 1.5.0 предоставила первый полный новый выпуск за два года. У него были альфа и бета-версии, а также выпуски-кандидаты. Независимые разработчики адаптируют свои компоненты для новой версии. Сообщество тестирует это программное обеспечение на предмет несовместимостей с предыдущей версией.

Вы можете отслеживать сообщения об ошибках и их исправлениях на веб-сайте Joomla!, и если сами найдете ошибку, то можете сообщить о ней там. Там же вы найдете список желательных усовершенствований (<http://joomlancode.org/gf/project/joomla/tracker/>). Процесс разработки Joomla! 1.5 в значительной мере проходил незаметно, начиная с осени 2005 г. Концепции, положенные в основу этой версии, уже в полной мере выкристаллизовались к февралю 2006 г., когда вышла первая бета-версия этого выпуска.

Дорожная карта

Дорожная карта, конечно, может измениться в любое время, но, тем не менее, она позволит вам сориентироваться. В табл. 1.1 дается список версий Joomla! (по состоянию на ноябрь 2007 г.).

Таблица 1.1. Дорожная карта Joomla!

Версия	Дата выпуска	Комментарий
Mambo 4.5.2	17 февраля 2005 г.	Последняя стабильная версия Mambo.
Joomla! 1.0.x	С сентября 2005 г.	Переход от Mambo версии 4.5.2.3. Исправление ошибок и заплатки безопасности. Последняя стабильная версия — Joomla! 1.0.13.
Joomla! 1.5 Beta 1	Октябрь 2006 г.	Интернационализация (полная поддержка UTF-8). Интерфейс администрирования для каждого языка. Пользовательские подключаемые модули. База данных: поддержка серверов баз данных MySQL и MySQLi. Система FTP, для связи безопасного режима PHP с поставщиками. Фундаментальные изменения и пересмотр структуры, каркаса и самой Joomla! для подготовки создания безбарьерных веб-сайтов с помощью Joomla!. Отделение программной логики от компоновки. Повышение удобства использования механизма поиска (search engine friendliness — SEF). Переработанный механизм кэширования.

Версия	Дата выпуска	Комментарий
Joomla! 1.5 Beta 2	Май 2007 г.	Документация разработчика (API, How-to), совершенствование API. Переработанное кэширование. Переработанная поддержка дружественных URL механизма поиска (SEF URL). Введение в ядро шаблона MooTool Javascript Framework Barrier-free <i>Beez</i> .
Joomla! 1.5 RC1-4	Июль 2007 г. — декабрь 2007 г.	Пользовательская документация. Исправления. Усовершенствования безопасности и производительности. Тестирование на различных платформах и браузерах.
Joomla! 1.5 stable	Январь 2008 г.	Стабильная версия.
Будущие версии Joomla!	Дата не указана	Новая система управления пользовательским доступом. Joomla! Контроль версий контента. Многосайтная инсталляция (много сайтов Joomla в одной инсталляции Joomla). Механизм обновления. Виртуальная файловая система. Поддержка других баз данных.

Изменения в деталях

Как видно из приведенной выше таблицы, версия 1.5 — это первая настоящая версия Joomla!. Команда разработчиков Joomla! потратила первый год на приведение в порядок наследия Mambo под новым именем Joomla! и выбор своего собственного направления дальнейшего развития. Изменения, произошедшие в Joomla!, ясно указывают направление будущей разработки.

Интернационализация

- Каждая порция статического текста теперь может быть перенесена в языковые файлы. Это особенно важно для административной области, которая до настоящего времени была доступна только на английском языке.
- Поддержка сценариев с записью справа налево (т.е. RTL, арабский, иврит, фарси и урду).
- Полный переход на набор символов UTF-8 для кодирования и отображения символов в Unicode.

Пользовательские подключаемые модули

То, что раньше называлось Mambots (мамботами), теперь называется подключаемыми модулями (plug-in). Пользовательские подключаемые модули, подключаемые модули аутентификации, подключаемые модули xmlrpc и системные подключаемые модули теперь объединяют контент, редактор и поисковые подключаемые модули.

С помощью подключаемых модулей, помимо прочего, можно использовать альтернативные механизмы регистрации от внешних программ.

Поддержка удаленных вызовов процедур XML

Удаленные вызовы процедур XML (**XML Remote Procedure Call – XML-RPC**) – это спецификация, которая позволяет программному обеспечению, работающему на разных системах и в разных средах, взаимодействовать между собой. Все важнейшие языки программирования поддерживают этот интерфейс, и существуют библиотеки, которые преобразуют код в XML-RPC (<http://en.wikipedia.org/wiki/XML-RPC>). Joomla! также предоставляет этот интерфейс. С его помощью, например, можно передать изображение из Flickr или написать статью в OpenOffice и затем опубликовать ее в Joomla!. Это открывает впечатляющие возможности для разработчиков. Например, они могут теперь обращаться к Joomla! из Java-программ.

Поддержка нескольких баз данных

Joomla! 1.5 содержит слой абстракции, который позволяет запускать Joomla! с разными версиями систем управления базами данных (СУБД). Однако в каждой конкретной инсталляции Joomla! может использоваться только одна из этих СУБД. На данный момент поддерживаются MySQL версий 4.x и 5.x. В будущем будут поддерживаться и другие СУБД.

Система FTP

Слой FTP был добавлен для того, чтобы избежать проблем с правами доступа. Таким образом, инсталляция новых компонентов и прочие загрузки могут выполняться через загрузку PHP и через FTP. Ограниченный (вполне разумно) подход провайдеров услуг, касающийся языка PHP, затруднял инсталляцию расширений и загрузки файлов.

Пересмотр каркаса Joomla!

До сих пор не существовало такого понятия, как каркас (framework), в пакетированном наборе функциональности Joomla!. Однако с отделением от исходного кода Mambo стало ясно, что в усовершенствовании нуждается буквально все. Нужно было переписать и ясно закодировать всю функциональность Joomla!. Каркас должен был быть гибким, масштабируемым, независимым от вывода и, прежде всего, понятным, чтобы независимые разработчики могли писать качественные компоненты за разумное время. Патентованный API-интерфейс для этого был незаменим.

Свобода от барьеров

Свобода от барьеров (barrier-free) – важная тема, к тому же она является обязательным требованием в Германии для государственных веб-сайтов, с тех пор, как в январе 2006 г. W3C издала стандарт для нее. Joomla! уже обладает полноценным шаблоном, свободным от барьеров (Beez), который соответствует этим стандартам.

Свобода от барьеров достигается соответствием стандартам (корректного HTML/ XHTML), а также полным отделением контента (текста, изображений и т.п.) от компоновки за счет применения каскадных таблиц стилей (CSS). Это утверждение в данный момент на 100% применимо к клиентской части. В последующих версиях запланирован перевод в режим, свободный от барьеров, также и административной области.

Удобство использования механизма поиска

Поддержка SEF URL была исключена из ядра Joomla! и перенесена в подключаемый модуль. Это позволило добавлять функциональность компонентов от независимых поставщиков, что было очень трудно делать ранее.

Проекты Google Summer of Code

Начиная с 2005 г., Google поддерживает талантливых студентов с их новыми идеями в проекте Summer of Code Project (<http://code.google.com/soc/2007/>), выделяя по \$4500 на каждый. Вместо поиска работы на лето, чтобы заработать денег, студенты могут заниматься своим хобби для общей пользы, и, разумеется, для выгоды Google. Ежегодно проекты Summer of Code приносят Google ошеломляющий объем рекламы, удачных идей и хороших программистов. Проекты с открытым кодом также выигрывают от стратегии привлечения новых талантов, и, конечно, от результатов их труда. В каждом случае один член соответствующего сообщества проектов становится наставником для одного студента.

Результаты этих проектов будут постепенно интегрированы в Joomla!. В прошлом (2007) году в разработке Joomla! участвовали программисты-студенты, которым платили в Google.

Ниже перечислены некоторые проекты.

- *Расширенная модель вложенных множеств с "жесткими вложенными множествами";* Энно Класинг (Enno Klasing), наставник — Луис Бентон Ландри (Louis Benton Landry). Работа касается популярных глубоко вложенных категорий.
- *Интерфейс электронной почты для публикации;* Нур Аини Рахмавати (Nur Aini Rakhmawati), наставник — Матеуш Кржешович (Mateusz Krzeszowiec). Создание контента Joomla! посредством электронной почты.
- *Семантическая веб-интеграция;* Майкл Мейсон (Mickael Maison), наставник — Эндрю Эдди (Andrew Eddie). Касается интеграции географических стандартов, таких как KML и GeoRSS, с целью манипуляции данными в Joomla!, например, для показа родного города пользователя в виде карты.
- *Подключаемый модуль для Eclipse, предназначенный для разработки компонентов/модулей Joomla!;* Мухаммед Фуад Дви Ризки (Muhammad Fuad Dwi Rizki), наставник — Лоуренс Вандепут (Laurens Vanderput). Работа посвящена созданию подключаемого модуля Joomla! к популярной среде разработки Eclipse, для облегчения создания компонентов Joomla!.
- *Компонент Joomla! общих рекомендаций по контенту;* Фаолан Чеслак-Постава (Faolan Cheslack-Postava), наставник — Сэмюэль Александр Моффатт (Samuel Alexander Moffatt). Касается автоматических рекомендаций по контенту в определенном контексте в определенное время.

Средства Joomla!

Ниже представлен список основных характеристик Joomla!.

- Свободно доступный исходный код.
- Огромное заинтересованное сообщество пользователей и разработчиков.
- Простая система рабочего потока.

- Система публикации контента.
- Файловый диспетчер для загрузки и администрирования файлов.
- Суммирование контента в формате RSS.
- Корзина для удаленного контента.
- Дружественные URL механизма поиска (SEF URL).
- Управление баннерами.
- Многоязычность веб-сайтов и административного интерфейса.
- Административный интерфейс, отдельный от веб-сайта.
- Макроязык для контента (подключаемые модули).
- Механизм кэширования для защиты быстрого создания страниц с популярными сайтами.
- Простая инсталляция дополнительных расширений.
- Мощная система шаблонов (HTML, CSS, PHP).
- Иерархические группы пользователей.
- Простая статистика посетителей.
- Редактор WYSIWYG для контента.
- Простое голосование.
- Система рейтингов для контента.

Существует также множество бесплатных и коммерческих расширений, которые можно найти на <http://extensions.joomla.org/>, например:

- форумы;
- галереи изображений;
- системы управления документами;
- календари;
- более 2000 других.

Примеры страниц Joomla!

Чтобы получить представление о существовании и внешнем виде страниц Joomla!, взгляните на следующие несколько страниц.

Joomla.org

Это, конечно же, один из крупнейших веб-сайтов Joomla! (рис. 1.1).

Travel Shop, Ирландия

Веб-сайт туристического агентства с интересной системой меню от joomlart.com (рис. 1.2).

Frank Lüdtkе, Германия

Успешная комбинация Joomla! и фото-галереи Corpermine (рис. 1.3).

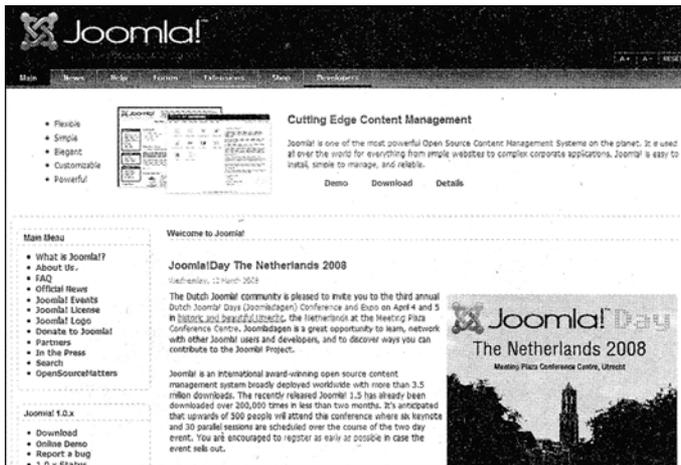


Рис. 1.1. Веб-сайт Joomla.org



Рис. 1.2. Веб-сайт Travel Shop



Рис. 1.3. Frank Lüdtke

Отель Schönruh, Австрия

Простой веб-сайт отеля в деревне Циллер (рис. 1.4).



Рис. 1.4. Веб-сайт отеля Schönruh

Urth.tv, США

Социальная платформа для граждан мира (рис. 1.5)!



Рис. 1.5. Веб-сайт отеля Urth.tv

unric.org, Европа

Организация Объединенных Наций также работает с Joomla! (рис. 1.6). В настоящее время веб-сайт представлен на тринадцати языках.



Рис. 1.6. Веб-сайт unric.org

porsche.com.br, Бразилия

И последний по порядку, но не по значению — сайт Stuttgart SP Veiculos Ltd. из Сан-Паулу (рис. 1.7). Эта компания принимает участие с самого начала разработки от Мамбо до Joomla!, и теперь уже перешла на Joomla! 1.5.



Рис. 1.33. Веб-сайт porsche.com.br



Найти список веб-сайтов на Joomla! можно в разделе Site Showcase (Выставка сайтов) форума ПО адресу <http://forum.joomla.org/index.php/board,58.0.html>.

Резюме

В этой главе был приведен краткий обзор истории Joomla! и структуры WCMS. Вы ознакомились с версиями Joomla! и ее возможностями, а также увидели несколько оснащенных Joomla! веб-страниц.

2

Инсталляция

Инсталляция Joomla! — пятиминутное дело. Если вы уже инсталировали ее раньше, то справитесь и за три минуты. В главе 16 вы сможете в этом убедиться. Чтобы иметь возможность начать инсталляцию, ваша среда разработки должна включать веб-сервер, поддерживающий PHP, и СУБД, поддерживаемую Joomla!.

Файлы Joomla! копируются в систему и затем инсталируются веб-инсталлятором. Такого рода среда также называется системой клиент-сервер.

Система клиент-сервер

Система клиент-сервер — это сетевая структура, в которой неограниченное количество рабочих станций (клиентов) обращается к службам, предоставленным центральным сервером. Сервер отвечает за доставку служб. Клиент может взаимодействовать с сервером и предоставляет действующий пользовательский интерфейс. Клиентом выступает Интернет-браузер. Серверами в нашем случае являются Apache и MySQL.

Обращение к веб-сайту Joomla! в Интернете

Итак, чтобы вы имели приблизительное представление о том, что происходит за кулисами, ниже приведен список событий, которые должны произойти, чтобы компьютер имел возможность обратиться к веб-сайту Joomla!.

- Установить соединение с Интернет через провайдера.
- Обратиться к необходимому веб-адресу (URL) из браузера.
- Браузер устанавливает связь с веб-сервером.
- Веб-сервер посылает запрос интерпретатору языка PHP, инсталированному на сервере. Интерпретатор языка PHP вызывает PHP-файлы Joomla! и интерпретирует их.
- Интерпретатор PHP запрашивает необходимую информацию из базы данных.
- Интерпретатор PHP создает код HTML или XHTML — в зависимости от используемого веб-сервера.
- Веб-сервер доставляет "страницу" или сгенерированный код клиентскому браузеру.

- Браузер "обнаруживает" указатели на CSS и другие файлы (изображения, Flash-элементы и т.п.) на только что полученной странице и запрашивает их у веб-сервера отдельно.
- При загрузке всех необходимых ресурсов браузер пытается "разобрать" (parse) и "визуализировать" (render) страницу; другими словами, разобрать ее на части и отобразить.

Технические требования для Joomla!

Joomla! требует наличия следующих компонентов.

- Инсталлированный и функционирующий веб-сервер, например, Apache версии 1.3.19 или выше, или же Microsoft IIS.
- Язык сценариев PHP версии 4.3 или выше, а также поддержка MySQL и Zlib, скомпилированная в PHP. **Zlib** — это библиотека, позволяющая PHP читать файловые пакеты, упакованные процедурой ZIP.
- Система управления базами данных MySQL, начиная с версии 3.23.x, или с набором символов Unicode, начиная с версии MySQL 4.1.x.

Необходимые элементы для инсталляции системы Joomla!

Все перечисленные выше компоненты понадобятся для того, чтобы установить у себя систему Joomla!. ПК, браузер и Интернет-соединение обычно доступны. Но существует много вариантов для веб-сервера, интерпретатора PHP и СУБД.

Вы можете:

- установить систему локально на вашем ПК;
- установить систему на сервере, обслуживающем внутреннюю сеть вашей компании;
- арендовать виртуальный сервер у провайдера;
- арендовать или приобрести физический сервер у провайдера (с административным доступом).

Вы можете также:

- разместить веб-сервер и СУБД на компьютерах, которые физически отделены друг от друга.

Разумеется, вы можете также использовать разные:

- марки веб-серверов;
- версии интерпретаторов PHP;
- версии СУБД MySQL.

И вдобавок ко всему этому вы можете инсталлировать и работать со всеми упомянутыми компонентами под управлением различных операционных систем.

Такая свобода выбора иногда смущает новичков, поэтому ниже обсуждается несколько типичных сценариев.

Локальная тестовая среда

В сценарии, когда вы находитесь дома или в своем офисе, и хотите установить веб-сайт Joomla!, можете использовать любую из перечисленных ниже операционных систем.

Операционная система Windows

В среде операционной системы Windows можно пользоваться одним из следующих двух веб-серверов.

- Windows XP Professional и Windows Vista Ultimate поставляются с веб-сервером Internet Information Server. Вы должны дополнительно установить PHP и СУБД, после чего все готово к работе.
- Можно взять предварительно сконфигурированный пакет (XAMPP), распаковать его и получить все, что требуется.

Операционная система Linux

Здесь все зависит от версии дистрибутива. Все дистрибутивы предусматривают простую инсталляцию (щелчком кнопкой мыши) пакетов Apache, PHP и MySQL. Иногда, в зависимости от конкретного дистрибутива, они могут быть уже предварительно инсталлированы. Поэтому вы можете использовать:

- программы, содержащиеся в дистрибутиве;
- предварительно сконфигурированный пакет (XAMPP) – распакуйте его и сразу получите все необходимое.

Операционная система Mac OS X

В операционной системе Mac OS X имеется веб-сервер по умолчанию (Apache), который следует лишь активизировать, но, к сожалению, в ней нет PHP. Не существует официальной версии PHP для Mac OS X, однако доступен модуль Apache, который можно установить (<http://www.entropy.ch/software/macosx/php/>). Существуют исполняемые версии MySQL для Mac OS X, которые можно установить после их загрузки (<http://dev.mysql.com/downloads/mysql/5.0.html>).

Таким образом, вы можете использовать:

- инсталлированный веб-сервер Apache, добавив к нему недостающее программное обеспечение;
- предварительно сконфигурированный пакет XAMPP для Mac OS X – распакуйте его, и вы получите все необходимое;
- другой готовый пакет под названием MAMP, который также очень популярный в среде Mac OS X (<http://www.mamp.info>).

Рабочая среда

Здесь также имеется несколько вариантов выбора.

Арендванный виртуальный сервер

Вы арендуете у провайдера пакет веб-пространства с СУБД, поддержкой PHP и часто также вашим собственным доменным именем. В этом случае у вас есть функциональная среда, и вы можете установить в ней Joomla!. Проконсультируйтесь

с провайдером касательно версий (PHP, MySQL). Иногда провайдеры предоставляют предварительно инсталлированную систему Joomla! с различными шаблонами. Если это так, то все, что вам остается сделать — это активизировать Joomla! одним щелчком кнопкой мыши — и все готово.

Ваш собственный сервер

Вы арендуете сервер у провайдера и инсталлируете операционную системы по своему выбору. Вы — администратор системы, и можете работать на ней, как на своем домашнем ПК.

Прежде чем выйти на просторы Интернета, вы должны сначала попрактиковаться на локальном компьютере. Преимущество этого подхода состоит в том, что не нужно платить за соединение, работает оно очень быстро, и можно не спеша практиковаться шаг за шагом.

Вы можете даже организовать у себя дома маленькую локальную сеть, в которой инсталлировать Joomla! на одном компьютере, а обращаться к ней с другого.

Напомню, что, возможно, вы найдете более современные версии на сайтах соответствующих проектов в Интернете.



Если вы инсталлируете Joomla! на рабочем сервере в Интернете, то всегда должны использовать наиболее сведую стабильную версию. Никогда не инсталлируйте бета-версию для реальной работы.

Установка локальной серверной среды

Чтобы инсталлировать Joomla! локально, вы должны настроить соответствующую серверную среду, как было описано выше.

Windows

Windows — чрезвычайно дружелюбная к пользователю и очень популярная система. Более 90% всех ПК работают под управлением этой операционной системы. Windows XP и Windows Vista — преобладающие версии. К сожалению, веб-сервер Apache, СУБД MySQL и PHP не входят в стандартную поставку Windows.

Вы можете инсталлировать каждую из этих программ отдельно или же установить предварительно сконфигурированный пакет (это очень практично и сэкономит немало времени).

В Windows XP и Windows Vista вы регистрируетесь в системе в режиме администратора. Если вы не знаете, имеете ли права администратора, проверьте тип вашей пользовательской учетной записи: выберите в меню **Starts⇒Control Panel⇒User Accounts** (Пуск⇒Панель управления⇒Учетные записи пользователей) и при необходимости измените соответствующим образом права (рис. 2.1).

XAMPP для Windows

XAMPP — проект Кая "Освальда" Сейдлера (Kai "Osvald" Seidler) и Кэя Фогельгесанга (Kay Vogelgesang). Несколько лет они занимались комплектацией полной среды разработки, включающей следующие ингредиенты: Apache, MySQL, PHP, Perl и различные соответствующие расширения. XAMPP можно загрузить с сайта <http://www.apachefriends.org/> в виде ZIP-архива для различных операционных систем. Этот пакет чрезвычайно удобен для таких людей, как вы и я, кого, прежде все-

го, интересует сама Joomla!, и не особенно интересует, как все это работает. Полная инсталляция также может быть удалена без следа с компьютера единственным щелчком кнопкой мыши.

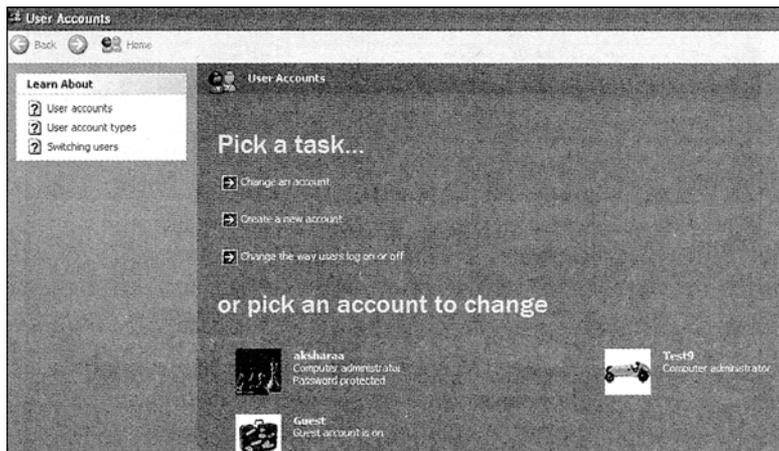


Рис. 2.1. Диалоговое окно *User Accounts* (Учетные записи пользователей) в Windows

Чтобы загрузить и инсталлировать XAMPP, необходимо выполнить перечисленные ниже шаги.

1. Загрузить файл `xampplite-win32-1.6.5.exe` с веб-сайта <http://www.apachefriends.org/en/xampp.html>.
2. Выполнить двойной щелчок на файле, чтобы распаковать все файлы, содержащиеся в архиве. Вы можете распаковать их на любой диск (рис. 2.2).

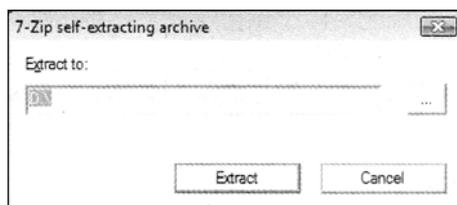


Рис. 2.2. Запуск самораспаковывающегося архива

Каталог `xampplite` создается автоматически, а его содержимое показано на рис. 2.3.

3. Когда вы дважды щелкнете на файле `xampp-control.exe` в каталоге `xampplite`, запустится Apache и MySQL (рис. 2.4). PHP стартует автоматически как модуль Apache. XAMPP не создает никаких новых элементов в реестре Windows и не устанавливает системных переменных.
4. Может случиться так, что при этом запустится брандмауэр Windows, который с негодованием сообщит, что вы хотите запустить два сервера; он затребует подтверждения ваших намерений. Щелкните на `Do Not Block Anymore` (Не блокировать в дальнейшем). После вы должны явно щелкнуть на `Continue` (Продолжить) в Windows Vista.

5. Откройте Интернет-браузер и введите в строке адреса `http://127.0.0.1/` или `http://localhost/`. Вы должны увидеть стартовую страницу XAMPP. Щелкните на ссылке English (Английский), и вы увидите страницу, показанную на рис. 2.5.

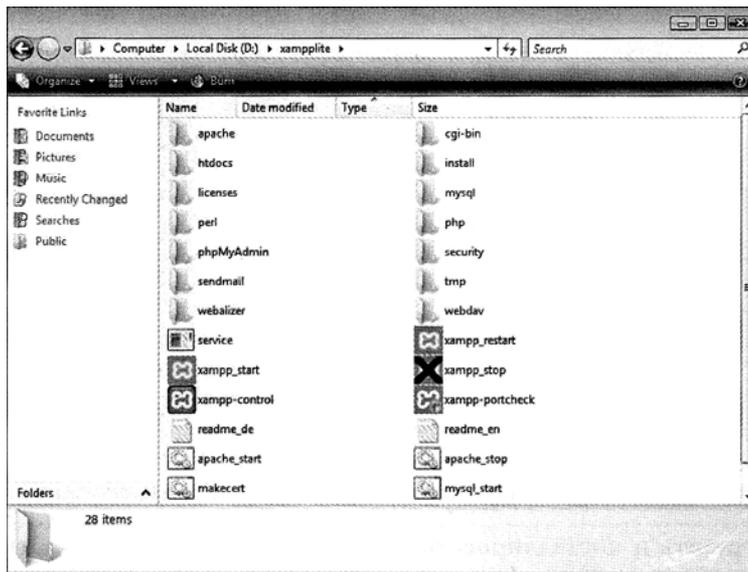


Рис. 2.3. Содержимое каталога `xampplite` после распаковки

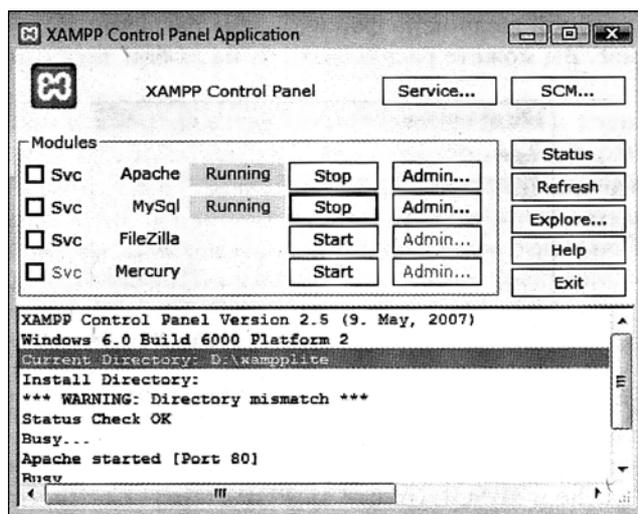


Рис. 2.4. Панель управления приложениями XAMPP

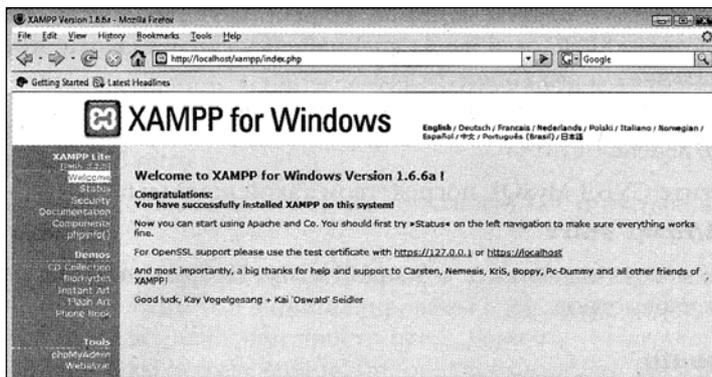


Рис. 2.5. Стартовая страница XAMPP

Каталог документов вашего веб-сайта будет выглядеть следующим образом:

```
[диск]:\xampplite\htdocs
```

Этот каталог содержит все страницы, доступные удаленному компьютеру в Интернете. Для вашей собственной безопасности внимательно прочтите включенный файл `readme_en.txt`. Он также содержит дополнительную информацию о необходимых паролях.

Чтобы деинсталлировать пакет, остановите все работающие серверы и просто удалите каталог `xampplite`.



Если веб-сервер Apache не стартует при запуске `xampplite`, это может быть связано с тем, что другая служба в данный момент занимает порт 80 на вашем компьютере. Одной из наиболее популярных программ, занимающих по умолчанию порт 80 в своих старых версиях, и тем самым предотвращающих использование его другими программами, является Skype. Вы можете либо изменить номер порта в настройках Skype, либо запустить `xampplite` сначала, а Skype — потом.

Linux

В среде Linux все проще. Доступны различные дистрибутивы с разными стандартными конфигурациями. Наша "команда мечты" уже предварительно инсталлирована, и ее нужно просто запустить. Версия XAMPP также может быть инсталлирована в Linux. По моему мнению, однако, лучше воспользоваться оригинальными программами. Инсталляция выполняется системным диспетчером пакетов и очень проста.

Версии openSUSE старше 10.x

Вы можете проверить, инсталлированы ли уже Apache, MySQL и PHP с помощью конфигурационной программы YaST2. Если они не инсталлированы, выберите соответствующие пакеты для инсталляции и позвольте YaST2 установить их.

Вот полный список пакетов:

- `apache2`
- `apache2-devel`
- `apache2-mod_php5mysql`
- `php5-mysql`

Вы можете найти эти пакеты через интерфейс YaST2 на дисках, поставляемых с вашим дистрибутивом SUSE, или же на специальных страницах пакетов в Интернете (<http://www.rpmseek.com/index.html>).

Запустите веб-сервер Apache с помощью следующей команды:

```
/etc/init.d/apache2 start
```

Затем запустите СУБД MySQL посредством такой команды:

```
/etc/init.d/mysql start
```

Оба сервера можно остановить командой `stop`. Набрав `help`, вы получите обзор всех доступных параметров.

Debian/Ubuntu

В Debian и Ubuntu вашим агентом для инсталляции будет `apt`. С помощью этой программы можно инсталлировать Apache, MySQL и PHP.

```
apt-get install [имя_пакета]
```

Ниже представлен детальный список пакетов:

- `apache-common` – поддерживает файлы для веб-серверов Apache;
- `php5` – встраиваемый в HTML язык сценариев стороны сервера;
- `mysql-common` – общие файлы СУБД MySQL (`/etc/mysql/my.cnf`);
- `mysql-server` – бинарные файлы СУБД MySQL.

Затем запустите Apache командой `/etc/init.d/apache2 start` и MySQL – командой `/etc/init.d/mysql start`.

Ваш собственный сервер у провайдера (root-сервер)

Если вы арендуете у провайдера целый сервер, то обычно получаете к нему доступ через командную оболочку и вольны выбирать дистрибутив Linux, который хотите использовать. Вдобавок, вы получаете предварительно сконфигурированную систему, содержащую все необходимые файлы пакетов и конфигураций. Обычно для конфигурирования таких серверов применяется специальный административный интерфейс, такой как SiteBuilder (<http://www.swsoft.com/en/products/sitebuilder/>) или Plesk (<http://www.swsoft.com/en/products/plesk/>). С помощью этого инструмента вы можете удобно стартовать, останавливать и конфигурировать сервер и службы Apache и MySQL из интерфейса браузера.

Инсталляция Joomla! на виртуальном сервере в сети

Эта тема очень сложна, поскольку существует немалое количество провайдеров и еще более немалое количество комбинаций инсталлированных версий Apache, PHP и MySQL, а также административных инструментов Webspaces.

Ниже перечислены "подводные камни", с которыми вы можете столкнуться во время инсталляции.

- Безопасный режим (Safe Mode) PHP который может быть активизирован при необходимости.

- Запрет преобразования URL-адресов в Apache из-за отсутствия активизации так называемого Apache Rewrite Engine (`mod_rewrite`).
- Права доступа к каталогам в Linux, которые отличаются от Windows.

В принципе простейший подход, который подходит для всех целей и случаев, выглядит так, как описано ниже.

- Загрузить файл Joomla! 1.5.zip на локальный ПК и распаковать его во временный каталог.
- Загрузить только что распакованные файлы программой FTP на арендованный сервер. Файлы должны быть установлены в каталоге, обычно именованном как `htdocs`, `public_html` или просто `html`. Если в этом каталоге уже есть другие инсталляции, вы можете специфицировать подкаталог внутри каталога, куда инсталировали свой экземпляр Joomla!. Многие веб-хосты позволяют связать арендованное доменное имя с каталогом. Это имя понадобится для обращения к вашему веб-сайту из браузера.
- Вы должны узнать, как называется ваша база данных. Обычно в ваш пакет веб-хостинга включена одна или несколько баз данных. Иногда имя пользователя, имя базы данных и пароль фиксированы; иногда их необходимо установить явно. Обычно в ваше распоряжение предоставляется браузерный интерфейс конфигурирования. Вы можете увидеть пример такого интерфейса на рис. 2.6. Эти данные доступа понадобятся для веб-инсталлятора Joomla!.

Вы можете проявить настойчивость, чтобы после загрузки данных на сервер сразу же получить у провайдера доступ к информации в базе данных.

Инсталляция Joomla!

Для инсталляции Joomla! понадобится исходный код. Загрузите пакет Joomla_1.5.zip и сохраните его в системе.

Выбор каталога для инсталляции

Вы должны решить, куда должна Joomla! инсталироваться — непосредственно в каталог документов или же в его подкаталог. Это важно, поскольку многие пользователи предпочитают короткие URL для своих домашних страниц.

Пример

Если Joomla! распаковывается непосредственно в `/htdocs`, то веб-страница стартует при обращении к ней по доменному имени `http://www.myhomepage.com` из локального компьютера `http://localhost` и/или с сервера в Интернете. Если вы создали подкаталог внутри `/htdocs/`, например, `/htdocs/Joomla150/`, и распаковали пакет туда, то в браузере придется вводить `http://localhost/Joomla150/`. При локальной работе это не представляет собой проблемы, но выглядит не очень хорошо для рабочего Интернет-сайта.

Однако некоторые HTML-файлы и подкаталоги уже присутствуют в каталоге `/htdocs` локальной среды XAMPP Lite под Windows, среди которых, например, страница приглашения XAMPP Lite. В зависимости от дистрибутива и веб-сервера, начальная страница также отображается в локальной среде Linux.

Каталог

Я рекомендую создать подкаталог по имени Joomla150 внутри каталога документов в Windows, используя проводник Windows. (В среде Linux используйте средства Shell, KDE Konqueror или Midnight Commander.)

[home]/htdocs/Joomla150/

Дерево каталогов в Windows Explorer теперь должно выглядеть так, как показано на рис. 2.6.

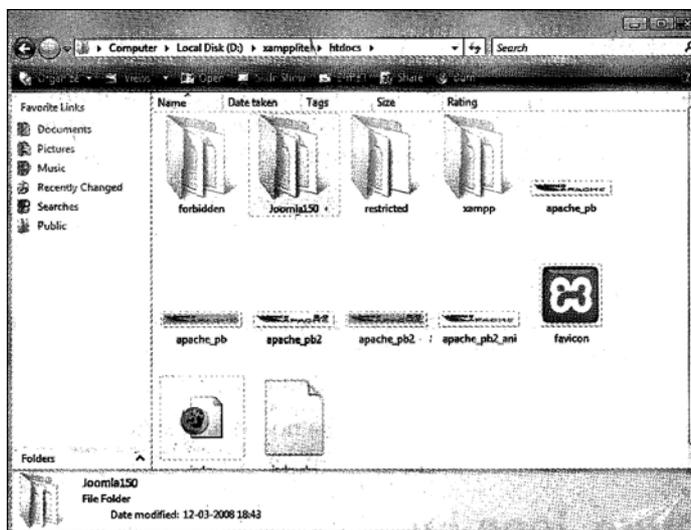


Рис. 2.6. Дерево каталогов в Windows Explorer

Страница с пустым индексом появляется в версии XAMPP Lite, когда вы вводите в браузере URL-адрес <http://localhost/Joomla150> (рис. 2.7).

В Linux или другой конфигурации вы можете получить сообщение об отсутствии доступа к данному каталогу. Это зависит от конфигурации веб-сервера. По причинам безопасности автоматическое отображение каталога часто отключается в конфигурации Apache. Потенциальный хакер может сделать много интересных заключений о структуре каталогов и файлах на вашей домашней странице и предпринять атаку на компьютер.

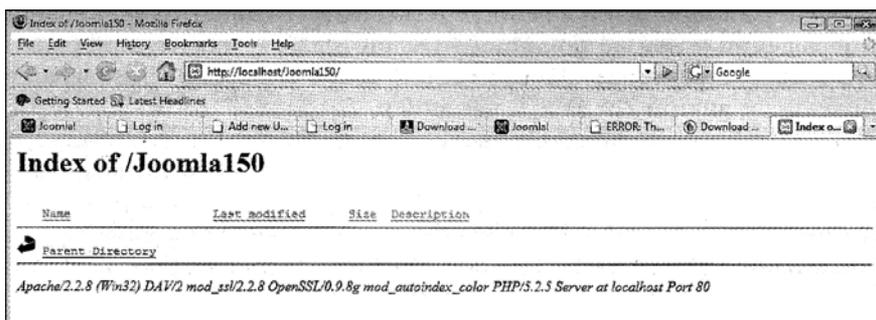


Рис. 2.7. Страница <http://localhost/Joomla150> в браузере

Из тех же соображений безопасности обычно не разрешен доступ к соответствующему конфигурационному файлу веб-сервера Apache. Чтобы иметь такой доступ, вы должны оставить каталоги контента отключенными и/или активизировать только те каталоги, которые содержат файлы для загрузки.

Распаковка

Теперь, наконец, можно распаковать файл пакета Joomla-1.5.zip в предназначенный для этого каталог. В Windows XP и Windows Vista распаковать этот пакет можно непосредственно в проводнике. В других версиях Windows может потребоваться специальная программа распаковки – вроде TUGZip. Эта структура одинакова на всех операционных системах; отличается только представление.

Веб-инсталлятор Joomla!

С этого момента все пойдет быстрее, потому что команды будет принимать и выполнять веб-инсталлятор Joomla!. Перейдите по адресу <http://localhost/Joomla150/>.

Шаг 1: выбор языка

Choose Language (Выберите язык) – первый из семи шагов инсталляции (рис. 2.8). Выберите желаемый язык и щелкните на кнопке Next (Далее).



Рис. 2.8. Выбор языка

Шаг 2: предустановочная проверка

Затем вы увидите страницу Pre-Installation Check (Предустановочная проверка), которая показана на рис. 2.9. Это проверка должна помочь определить, подходит ли ваша среда сервера для инсталляции Joomla!.

Если вы преимущественно видите результаты тестов, выделенные зеленым цветом, – это хороший знак. В зависимости от вашей конфигурации, могут быть различия. Инсталлятор Joomla! принимает во внимание конфигурационные установки веб-сервера (в нашем случае – Apache), PHP и операционной системы. На Unix-образных системах (Linux, Mac OS X) вы должны уделить внимание правам на запись. Это особенно важно для файла `configuration.php`. Этот файл создается в конце инсталляции со специально настроенными значениями. Если инсталлятор не будет иметь

права на запись в каталог, то Joomla! не сможет создать этот файл, и инсталляция закончится неудачей. Если такое случится, попробуйте сконфигурировать права доступа соответствующим образом, и щелкните на кнопке Check Again (Повторить проверку).

Если вы работаете с решением XAMPP Lite под Windows, экран должен быть похожим на то, что показано на рис. 2.9. Щелкните на кнопке Next и переходите к шагу ознакомления с лицензией.

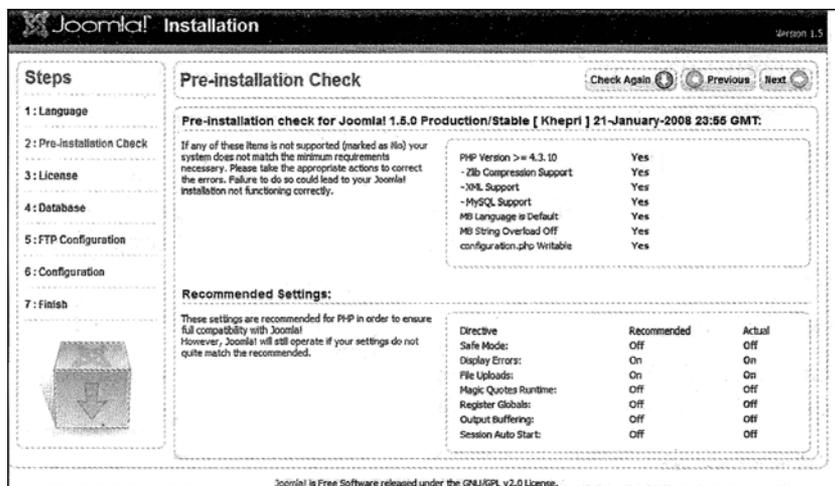


Рис. 2.9. Предустановочная проверка

Шаг 3: лицензия

Каждая часть программного обеспечения лицензируется в соответствии с определенными условиями. Joomla! использует лицензию GNU/GPL версии 2.0. На рис. 2.10 показана страница License (Лицензия). После прочтения лицензии щелкните на кнопке Next.

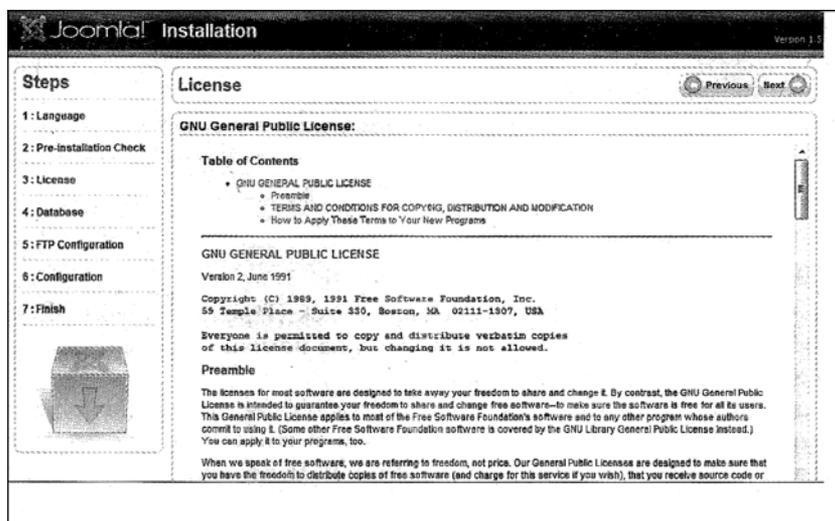


Рис. 2.10. Страница лицензии

Шаг 4: конфигурация базы данных

На четвертом шаге – Database Configuration (Конфигурация базы данных) – запрашиваются параметры вашей базы данных (рис. 2.11). Вы можете настроить столько баз данных, сколько потребуется для серверной среды XAMPP Lite. Вам доступен пользователь MySQL по имени `root` (без пароля). Пользователь `root` – это администратор MySQL, который может делать все в системе MySQL. Инсталляция без пароля представляет серьезный риск для безопасности. Поначалу, когда вы настраиваете систему локально для быстреего запуска ее в работу, это не является проблемой. Но в долговременной перспективе вы определенно должны снабдить вашу инсталляцию XAMPP Lite паролями на странице проверки безопасности (**security check**; <http://localhost/security/>).

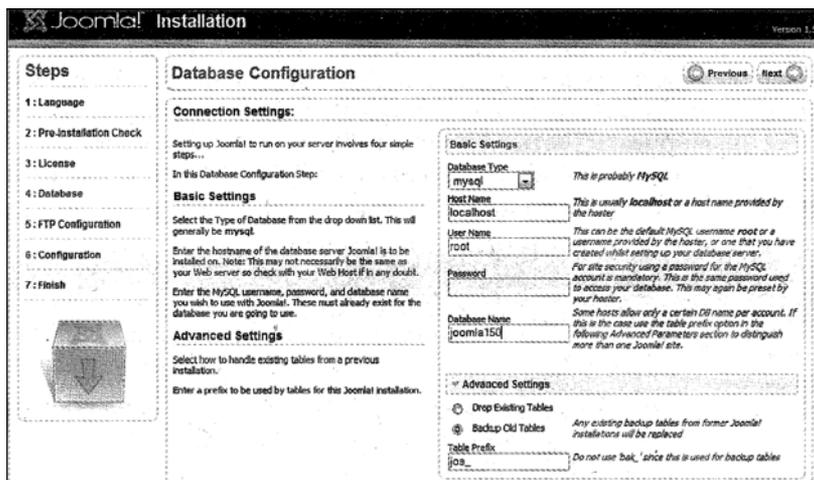


Рис. 2.11. Страница конфигурации базы данных

Введите следующие параметры в инсталляции XAMPP Lite:

- Host Name (Имя хоста): `localhost`
- User Name (Имя пользователя): `root`
- Password (Пароль): оставить пустым, но помнить о риске нарушения безопасности!

Теперь заполните поле Database Name (Имя базы данных). В локальной среде XAMPP Lite может быть настроено неограниченное количество баз данных, поскольку пользователю `root` это позволено. В активной среде, предоставленной провайдером, вы, возможно, будете иметь заранее настроенный перечень баз данных.

По этой причине я могу также ввести имя базы данных, которая еще не существует в инсталляторе. Здесь используется имя базы данных `joomla150`.

Щелкнув на треугольнике зеленого цвета перед Advanced Settings (Дополнительные настройки), вы можете активизировать дополнительные опции. Здесь можно выбрать, следует ли удалять таблицы базы данных существующей инсталляции Joomla!, или же их сохранить, а также, нужно ли снабдить их префиксом `bak_`.

Поле Table Prefix (Префикс таблиц) – очень практичная вещь. Текст, который вводится в этом поле, добавляется в начало имени каждой таблицы, созданной веб-инсталлятором. По умолчанию веб-инсталлятор рекомендует опцию `jos_`.

На то имеется простая причина. Иногда вы получаете только одну базу данных MySQL от Интернет-провайдера. Если необходимо управлять двумя сайтами Joomla!, в этом случае возникает проблема, поскольку вы не сможете отличать одну таблицу от другой. И тогда можно различить таблицы из разных инсталляций Joomla! (например, jos_smith_ и jos_jones_) посредством Table Prefix. В данном случае примите префикс по умолчанию – jos_. Префикс также используется для пометки сохраненных данных (bak_).

Шаг 5: конфигурация FTP

Чтобы решить проблемы с правами доступа и, возможно, активизированным режимом PHP Safe Mode, начиная с Joomla! 1.5, у вас есть возможность выбора использования FTP функций PHP для загрузки и обработки файлов. В этом нет необходимости при локальной инсталляции XAMPP Lite. Но если вы устанавливаете Joomla! на виртуальном веб-сервере провайдера, укажите здесь параметры FTP, которые сообщит провайдер. Если ваш провайдер разрешит эти функции, то с точки зрения безопасности желательно установить разные учетные записи FTP для вас как пользователя и Joomla!, и активизировать учетную запись FTP, относящуюся к Joomla!, только для соответствующего каталога Joomla!. Страница FTP Configuration (Конфигурация FTP) показана на рис. 2.12.

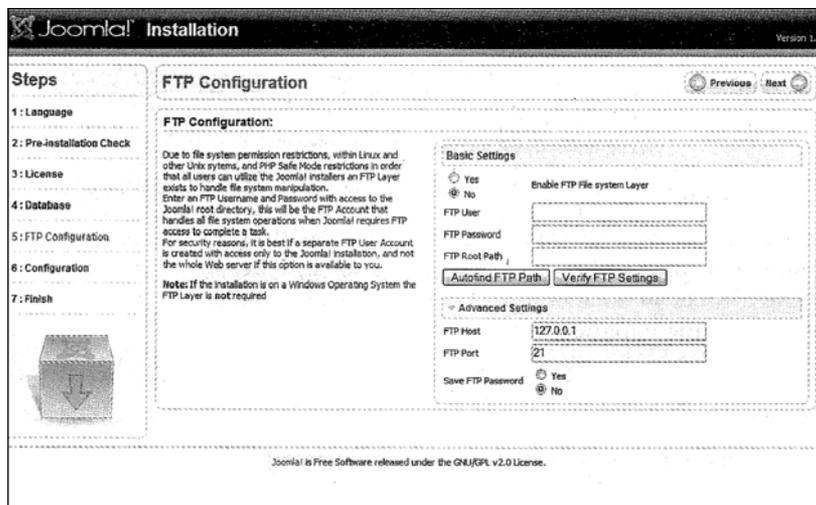


Рис. 2.12. Конфигурация FTP

Шаг 6: главная конфигурация

Этот главный шаг конфигурации делится на три части.

Первая часть конфигурации имеет дело с именем вашего веб-сайта. Это имя появляется в заголовке окна браузера, когда кто-то обращается к веб-сайту. Это имя используется также и в других местах, например, в конфигурировании электронной почты для зарегистрированных пользователей. Для сайта-примера выбрано имя Joomla! 1.5.0.

Во второй части запрашиваются пути веб-сайта, адрес электронной почты администратора и его пароль. Запишите пароль на бумажке (но не приклеивайте ее на монитор или клавиатуру!). Страница Main Configuration (Главная конфигурация) показана на рис. 2.13.



Рис. 2.13. Страница главной конфигурации

В третьей части вы определяете тип данных, которые будет содержать ваша инсталляция Joomla! (рис. 2.14).

- Install Default Sample Data (Инсталлировать стандартные данные примеров). Данные — это наиболее важная часть вашей инсталляции. Joomla! позволяет и явно рекомендует инсталлировать данные примеров, чтобы вы могли без проблем поэкспериментировать с ними.
- Load Local Joomla! 1.5 SQL Script (Загрузить локальный SQL-сценарий Joomla! 1.5). У вас уже может быть создан локальный веб-сайт Joomla!, и вы, возможно, хотите иметь именно эти данные в онлайн-овой базе данных. В этом случае вы можете специфицировать файл .sql, содержащий данные Joomla!, которые вы экспортировали из локальной версии. Убедитесь, что этот файл содержит именно дизайн базы данных Joomla! 1.5 в кодировке UTF-8.
- Load Migration Script (Загрузить сценарий миграции). Эта опция переносит инсталляцию Joomla! 1.0.x в версию Joomla! 1.5.

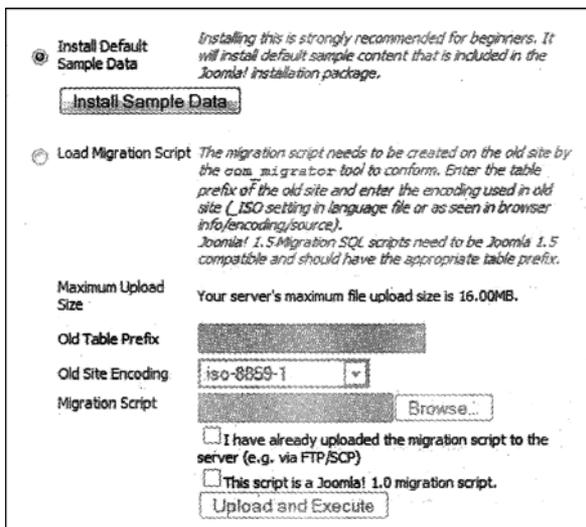


Рис. 2.14. Третья часть главной конфигурации

Флажок Install Default Sample Data выбран по умолчанию. На протяжении этой книги мы рассмотрим и другие опции.

Щелкните на кнопке Install Sample Data (Инсталлировать данные примеров), как показано на рис. 2.15. Инсталлятор загрузит данные в вашу базу и изменит отображение. Процесс несколько невыразительный, но необходимый для установки данных примеров. Затем кнопка исчезнет и появится маленькое текстовое окно.

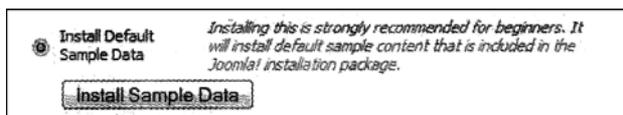


Рис. 2.15. Установка данных примеров

Теперь щелкните на кнопке Next и ваши данные сохранятся.

Шаг 7: завершение

На седьмом, заключительном, шаге вы получите поздравления с успешной инсталляцией. Позвольте и мне также поздравить вас. Вы увидите примечание в виде выделенного полужирным текста с предложением удалить каталог installation. Отнеситесь с должным вниманием к этому примечанию, поскольку ваш веб-сайт Joomla! просто не запустится без этого шага. Страница Finish (Завершение) показана на рис. 2.16.

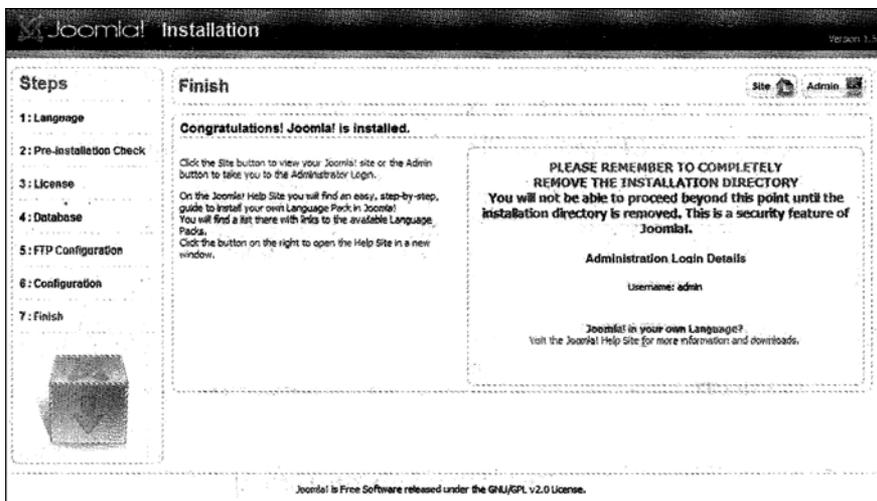


Рис. 2.16. Завершающая страница инсталляции

Читатели, которые хотят исследовать конкретный пример построения веб-сайта с нуля, могут обратиться к главе 16.



Файл по имени `configuration.php` создается в вашем каталоге документов. Если вы хотите повторить инсталляцию, то должны удалить файл `configuration.php` перед тем, как удалить инсталляционный каталог. После этого зайдите в браузере на URL, содержащий файлы Joomla!, и инсталлятор Joomla! запустится снова.

Имя пользователя для администратора – всегда `admin`.



В случае если вы забудете или потеряете пароль администратора, ищите решение этой проблемы в приложении.

На этом инсталляция завершена, и вы должны выбрать между кнопками Site (Сайт) для просмотра вашей домашней страницы и Admin (Администрирование) для отображения административного интерфейса. Чтобы взглянуть на только что созданную домашнюю страницу, щелкните на кнопке Site. Результат показан на рис. 2.17. Если вы еще не удалили инсталляционный каталог, то получите дружеское напоминание о том, что это нужно сделать, прежде чем можно будет увидеть вашу страницу.

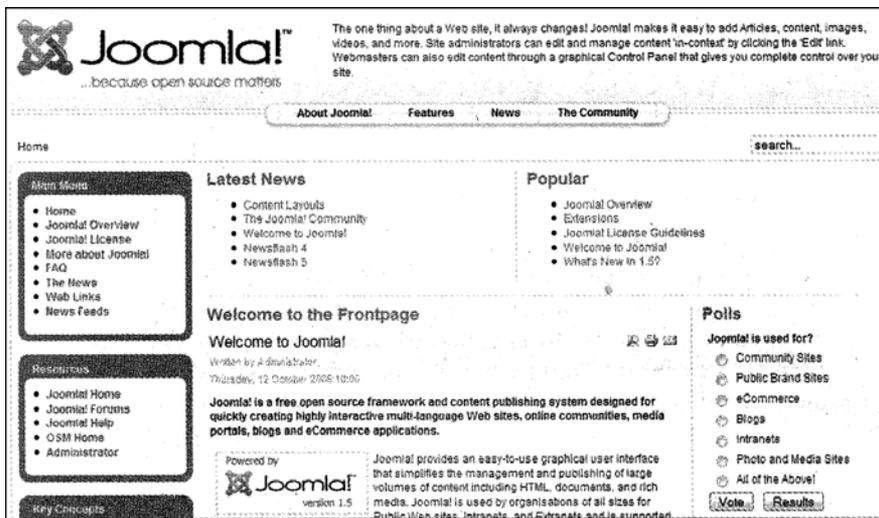


Рис. 2.17. Внешний вид домашней страницы Joomla!

Резюме

Результат выглядит достаточно впечатляюще. Переведите дух, осмотритесь, пощелкайте на различных опциях и попытайтесь сориентироваться. На этой домашней странице, которая загружается с данными примеров, используется значительная часть функциональности Joomla!. Мы рассмотрим ее в последующих главах.

3

Тур по вашему новому веб-сайту

Теперь, когда вы установили веб-сайт и хорошенько рассмотрели его, мы можем вместе исследовать результат. На первый взгляд, страницы выглядят несколько запутано. В принципе, они делятся на интерфейсную часть (собственно, ваш веб-сайт) и серверную часть (административное представление вашего веб-сайта). Клиенты и веб-серверы видят интерфейсную часть, а серверная часть доступна только сотрудникам и/или администраторам.

Интерфейсная часть

Вы сразу можете заметить, что веб-сайт примера иллюстрирует массу функциональности, предоставляемой Joomla!, показывает хороший обзор его возможностей.

Для ясности различные области на рис. 3.1 снабжены подписями.

Искусство дизайна теперь состоит в распознавании элементов, которые важны для вашего веб-сайта, пропуске неважных и представлении их пользователю в логичном, понятном и привлекательном формате. Полученный результат всегда представляет собой компромисс между функциональностью и организацией.

По своей конфигурации эта структура напоминает одну из ежедневных газет вроде “Нью-Йорк Таймс” или портал наподобие MSN или Yahoo!. Слева и справа находятся рамки с четко определенным контентом. В центре (главного тела) — элементы новостей (категоризированный контент).

Так называемый шаблон определяет компоновку страницы. Шаблоны заменяемы и модифицируемы, в том смысле, что один и тот же контент может отображаться в разных компоновках. Любая ежедневная газета может позавидовать такой функциональности.

Давайте внимательней рассмотрим компоновку примера. На странице присутствуют области пяти разных категорий:

- меню;
- контент;
- реклама;
- дополнительные функции;
- декоративные элементы.



Joomla!™

...because open source matters

Место под логотип

Joomla! 1.5 - "Experience the Freedom" It has never been easier to create your own dynamic site. Manage all your content from the back end and in virtually any language you speak.

Экстренные сообщения

About Joomla!

Features

The Community

Верхнее меню

Поле поиска

Main Menu

- Home
- Joomla! Overview
- Joomla! License
- More about Joomla!
- FAQ
- The News
- Web Links
- News Feeds

Главное меню

Latest News

- Content Layouts
- The Joomla Community
- Welcome to Joomla!
- Newflash 4
- Newflash 5

Последние новости

Popular

- Joomla! Overview
- Extensions
- Joomla! Features
- Joomla! Facts
- Content Layouts

Welcome to Joomla!

Written by Administrator

Thursday, 12 October 2006 11:59

**** 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.

Первая страница

Титульная страница

Стартовая страница

Last Updated (Wednesday, 13 October 2006 16:37)

Read more... >>

Example News Item 1

Written by Administrator

Wednesday, 07 July 2004 11:54



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 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.



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 nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Last Updated (Wednesday, 13 October 2006 17:40)

Read more... >>

Read more...

- How do I upgrade to Joomla! 1.5 ?

« Start Prev 1 2 Next End »

Page 1 of 2

Polls **Опрос**

Joomla! is used for?

- Community Sites
- Public Brand Sites
- eCommerce
- Blogs
- Intranets
- Photo and Media Sites
- All of the Above!

Кто в онлайн?

Who's Online

We have 1 guest online

Advertisement

Featured Links:

Joomla!

Joomla! The most popular and widely used Open Source CMS Project in the world.

JoomlaCode

JoomlaCode, development and distribution made easy.

Joomla! Extensions

Joomla! Components, Modules, Plugins and Languages by the bucket load.

Joomla! Shop

For all your Joomla! merchandise.

Ads by Joomla!

Место под рекламу

Key Concepts

- Extensions
- Content Layouts
- Example Pages

Resources

- Joomla! Home
- Joomla! Forums
- Joomla! Help
- OSM Home
- Administrator

Дополнительные новости

Login Form

Username

Password

Remember Me

- Forgot your password?
- Forgot your username?
- Create an account

Регистрация пользователя

Feed Entries **Лента новостей**

Нижний колонтитул

Powered by Joomla!, Valid XHTML and CSS.

Рис. 3.1 Области веб-сайта Joomla!

Меню

Меню максимально облегчают навигацию для пользователя. Существуют разные меню для разных задач. Joomla! имеет шесть predefined меню в данных примеров. Вы можете добавить столько дополнительных меню, сколько хотите, или, естественно, исключить их. Кстати, два из доступных типов меню в нашем примере не показаны. Это так называемое пользовательское меню — User Menu, которое зарегистрированный пользователь видит после того, как входит в систему, и меню ключевых концепций — Key Concepts, которое показывается только тогда, когда вы щелкаете на ссылке Key Concepts.

Верхнее меню

Верхнее меню (рис. 3.2) располагается на странице как можно выше. Оно предназначено для того, чтобы обеспечить пользователю быстрый доступ к наиболее важному контенту. Такое меню часто отображает концепции, товары, контакты, адрес компании и язык.



Рис. 3.2. Верхнее меню

Главное меню/указания

Главное меню (рис. 3.3) — центральная навигационная область страницы. Здесь всегда должна быть ссылка для возврата пользователя на первую страницу. Это меню должно появляться точно в одной и той же позиции на каждой странице веб-сайта. Главное меню — важная точка, к которой обращается пользователь. То же верно и для Breadcrumbs (Указания), также называемое меню типа “хлебных крошек”. Эта навигация с помощью указаний должна облегчить перемещение внутри ветвей дерева контента. Название для него происходит из сказки “Гензель и Гретель” (Hansel and Gretel). Указания (“хлебные крошки”) — это ссылки на ранее посещенный, вышестоящий или ассоциированный контент.



Рис. 3.3. Главное меню

Другие меню

Другие меню (рис. 3.4) могут появляться в любых местах (позициях модулей). В зависимости от контента и контекста страницы может иметь смысл предоставлять дополнительные опции меню вроде компонента магазина или, как здесь, например — ключевых концепций Joomla! (Key Concepts).

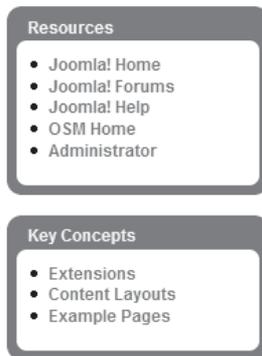


Рис. 3.4. Меню Resources (Ресурсы) и Key Concepts (Ключевые концепции)

Контент

И, наконец, мы добрались до контента, которым хотим управлять!

Что такое контент?

Контентом (content), или содержимым, может быть сообщение, редакторская статья или статическая страница с объяснениями. Контентом может быть также каталог динамических ссылок (<http://www.google.com/>), магазин (<http://dell.com/>) или “блошинный рынок” (<http://www.ebay.com/>).

Контент также может быть чем-то полностью динамическим и открытым для всех. Общедоступная энциклопедия Wikipedia, например, использует системы администрирования контента, которая позволяет любому изменять контент. Эта специальная форма администрирования контента получила название wiki. Каждый может изменить или даже удалить контент. До сих пор эта концепция работает исключительно хорошо с Wikipedia, несмотря на постоянные дискуссии о структуре и содержимом статей.

Противоположность wiki — статический контент, который, будучи однажды написанным, остается таковым в течение длительного периода времени. Возьмем, например, настоящую книгу. Она устареет с выходом новых версий программного обеспечения, хотя имеет преимущество в том, что излагает тему всесторонне и согласовано. Я произвожу контент определенного типа и потому оперирую разновидностью управления содержимым — и не только в Joomla!, но и в OpenOffice.

Папки, флайеры, наклейки, бизнес-отчеты и руководства также обычно являются статичными по природе. Созданные по определенному событию, они устаревают либо просто со временем становятся неверными.

Многие старые веб-страницы состоят исключительно из статических элементов. В Интернете, однако, время идет немного быстрее. То, что вполне нормально для книг, папок и прочих печатных материалов (в конце концов, я также могу почитать книгу на пляже и в метро), было бы серьезным недостатком для посетителей веб-сайтов. Нет ничего хуже для имиджа вашей компании, чем устаревший статический веб-сайт с кнопкой, помеченной “Powered by...”, указывающей на безнадежно устаревшее программное обеспечение.

Возможности представления контента неисчерпаемы. Они зависят от доступного терминала, пропускной способности сети и многих других вещей, которые, в свою очередь, зависят от пользователя и его творческих способностей. Пользователь сообщения играет чрезвычайно важную роль. Это напомнило мне вопрос: какова в действительности ваша целевая группа?

Банально, но верно:

Контент – глава всему!

Все зависит от контента. Каждая веб-студия теперь может, засучив рукава, приступить к ежедневной работе по созданию очередного веб-сайта. Миллионы долларов, выделенные на рекламу таких продуктов, как консервированный шпинат или пиво, доказывают тот факт, что успешные коммуникации возможны и без уникального контента.

Тем не менее, утверждение о важности контента остается фундаментально верным. Если вам нечего сказать или нечего предложить, то вас никто не станет слушать. Поскольку у вас, наверное, нет миллиона долларов на рекламу, вы не заставите людей читать ваш контент. Неважно, насколько прекрасно может выглядеть веб-сайт без контента, или сколько терминалов могут отображать его — никто по своей воле не посетит его.

Первая страница/титовая страница

Контент анонсируется на первой странице веб-сайта, как показано на рис. 3.5.

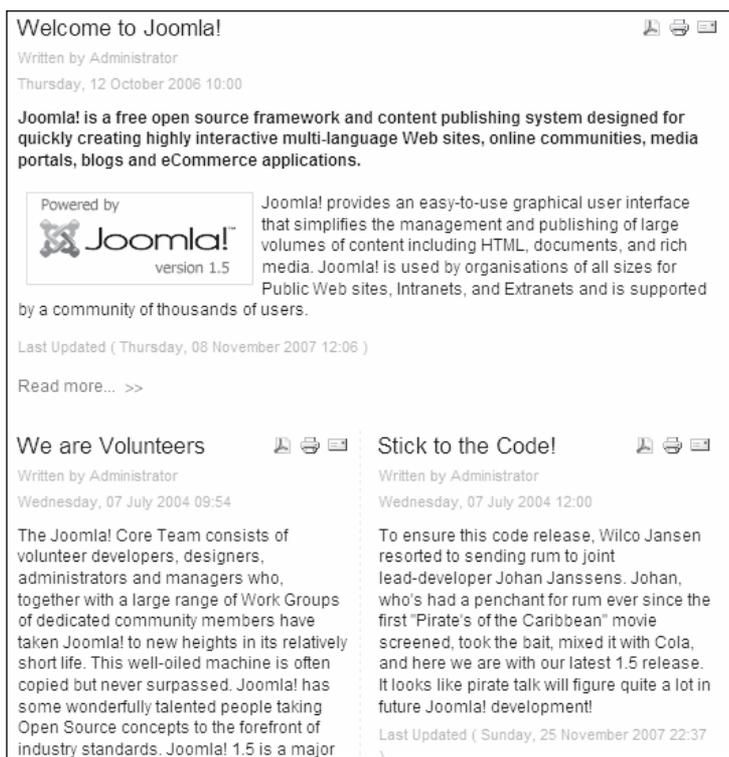


Рис. 3.5. Первая страница

Контент имеет автора – в данном случае, администратора – дату создания, заголовок, приманку и, возможно, изображение. Приманка призвана вызвать у посетителя любопытство и вынудить его щелкнуть на ссылке *Read more* (Читать дальше), чтобы прочесть все сообщение. В качестве вариантов могут быть предоставлены ссылки (пиктограммы) для чтения контента в виде PDF, для его распечатки или отправки по электронной почте кому-нибудь еще.

Последние новости/наиболее популярные статьи

Сообщение может быть отображено в разных форматах. Обычно людей интересуют все самое *новое*, а также то, что читают *другие*.

По этой причине компоновка нашего примера содержит соответствующую область, в которой отображаются последние пять статей, и еще одну область со ссылками на наиболее читаемые материалы (рис. 3.6). Автоматическое формирование этой второй области возможно благодаря тому, что Joomla! протоколирует каждое открытие статьи в базе данных и отслеживает количество обращений к ним.



Рис. 3.6. Последние новости/наиболее популярные статьи

Реклама

Когда ваш сайт достигает популярности и его контент верен, вы можете продавать рекламное пространство (рис. 3.7). Под рекламным пространством обычно понимаются баннерные ссылки. Баннеры – это небольшие графические изображения (в форматах *.gif*, *.jpg*, *.png* или *.swf*), которые предлагают посетителю перейти на веб-сайт рекламодателя единственным щелчком. Если вы действительно хотите этого, найдите место в своей компоновке и выделите его под рекламу.

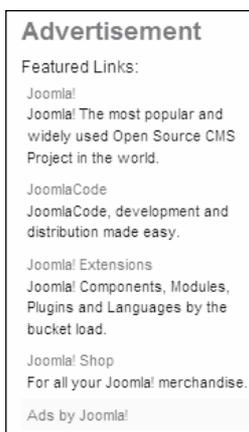


Рис. 3.7. Область рекламы

Область баннеров

Область баннеров может включать текстовые и графические ссылки. На рис. 3.8 показан пример типичного баннера размером 468×60 пикселей.

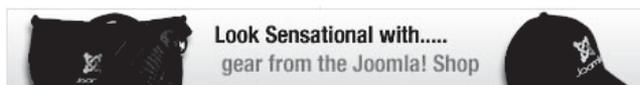


Рис. 3.8. Пример баннера

Функции

Функции — это элементы сайта, которые необходимы для обеспечения интерактивности. В Joomla! эти функции встроены в модули.

Область входа

Модуль входа (login module) важен, если вы хотите разбить веб-сайт на общедоступную и приватную области. В этом случае пользователь должен иметь возможность зарегистрироваться и войти. Возможно, он иногда даже забудет свой пароль. Модуль входа должен принимать во внимание такую ситуацию, как показано на рис. 3.9.

Опросы

Поскольку наш контент ориентирован на определенные целевые группы, стоит периодически проводить опрос мнения в группах, которые в действительности посещают сайт. Это простейший способ получения полезных откликов о вашем сайте.

Joomla! включает интегрированный компонент опроса (polling component). Вы можете видеть соответствующий модуль отображения на примере страницы, показанной на рис. 3.10.

Кто в онлайн?

Этот модуль касается коммуникаций и сообщества. После того, как пользователь увидит новые и наиболее популярные статьи, естественно, он захочет узнать, кто просматривает сайт прямо сейчас. Между гостями и зарегистрированными пользователями делается различие, как видно на рис. 3.11.

A screenshot of a Joomla! login form. It has a title "Login Form" and two input fields for "Username" and "Password". Below the password field is a "Remember Me" checkbox. There is a "Login" button. At the bottom, there are three links: "Forgot your password?", "Forgot your username?", and "Create an account".

Рис. 3.9. Пример формы входа/регистрации

A screenshot of a Joomla! poll form. The title is "Polls". The question is "Joomla! is used for?". There are seven radio button options: "Community Sites", "Public Brand Sites", "eCommerce", "Blogs", "Intranets", "Photo and Media Sites", and "All of the Above!". At the bottom, there are "Vote" and "Results" buttons.

Рис. 3.10. Форма опроса

A screenshot of a Joomla! "Who's Online" module. It displays the text "Who's Online" and "We have 1 guest online".

Рис. 3.11. Кто в онлайн?



Тщательно обдумайте, действительно ли вы хотите представлять это средство на своем веб-сайте. Если вы сообщаете в контенте, что вы — крупнейший, скажем, в США сайт по загрузке рингтонов, а на нем окажется единственный посетитель, это подорвет доверие к вам. Однако если вы в действительности постоянно имеете 10–20 посетителей и множество зарегистрированных пользователей, то это хороший способ продемонстрировать динамику посещений.

Лента новостей

Ленты новостей (news feeds) становятся все более и более популярными. Это стандартизованные, машинно-читаемые коллекции контента, которое может передаваться на дальнейшую обработку для добавления к контенту вашего сайта, без нарушения шаблона и компоновки. Модуль **Syndication** (Синдикация) предоставляет ленту новостей для веб-сайта (рис. 3.12). Далее в книге вы узнаете больше об этой технологии.



Рис. 3.12. Лента новостей

Поле поиска

Функциональность, которая положена в основу поля поиска (рис. 3.13), имеет большое значение для дружелюбности к пользователям веб-сайта. Многие сайты имеют поле поиска. Однако часто они обеспечивают поиск только по части сайта. Что касается Joomla!, то это не так — здесь все страницы сканируются в обязательном порядке. Если добавляются новые расширения, поиск выполняется и в их контенте.

Вы можете ввести поисковый фрагмент и нажать клавишу <Enter> на клавиатуре. В результате появится список вхождений с визуальным выделенным поисковым фрагментом.



Рис. 3.13. Поле поиска

Декоративные элементы

Учитывая такое богатство функций, модулей и контента, возникает проблема дизайна, корпоративной идентичности, внешнего вида и поведения веб-сайта.

Шаблон представляет компоновку страницы и накладывается на контент подобно экрану. Поскольку о вкусах не спорят, у вас есть возможность использовать разные шаблоны для представления одного и того же контента. Например, внешний вид вашего веб-сайта может отличаться зимой и летом, или же он может иметь олимпийскую символику в период проведения Олимпийских Игр. (В главе 13 описано создание ваших собственных шаблонов.)

В принципе, шаблон состоит из логотипа, определенной комбинации цветов, выбранных шрифтов и размеров символов, а также, насколько возможно, разумного размещения доступного контента. Пример шаблона, конечно же, включает логотип Joomla!, показанный на рис. 3.14.



Рис. 3.14. Логотип Joomla!

Перспектива

Я уверен, что после этого краткого тура вы поймете, что администрирование контента может оказаться весьма непростой задачей. Помимо прочего, важно не потерять ваше собственное видение.

Серверная часть

Администрирование веб-сайта происходит в серверной части, в Joomla! Administration (рис. 3.15). Вы можете обратиться к разделу администрирования по следующему URL:

[Доменное_имя]/administrator/

Если вы также работаете с локальной инсталляцией, то URL будет таким:

http://localhost/joomla150/administrator/



Рис. 3.15. Форма административного входа Joomla!

Войдите под своим идентификатором admin. Вы должны были специфицировать пользовательские данные в веб-инсталляторе при установке Joomla!.

После этого вы увидите интерфейс с меню, пиктограммами и вкладками, похожий на интерфейс вашей операционной системы (рис. 3.16).

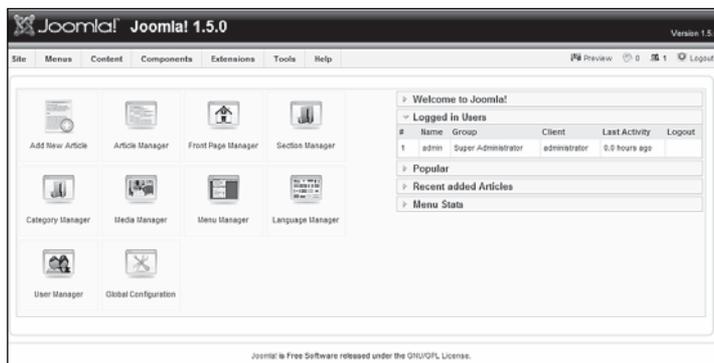


Рис. 3.16. Административный интерфейс Joomla!



Из соображений безопасности вы должны защитить от доступа каталог [путь_к_Joomla]/administrator/ через файл .htaccess. Из-за популярности Joomla! хакеры постоянно пытаются прорваться в область администрирования.

Резюме

Эта глава помогла вам “прочувствовать” веб-сайт Joomla!. Теперь вы знакомы с интерфейсной и серверной частями Joomla!. В следующей главе речь пойдет о настройке Joomla! в соответствии с существующими требованиями.

4

Настройка Joomla – языки и шаблоны

В предыдущей главе я упомянул, что вы можете настраивать свой сайт, вносить изменения и наполнять контентом – и все это в административной области Joomla!.

Первые две вещи, которые обычно хотят сделать владельцы сайтов – это изменить язык по умолчанию своим родным языком, а также изменить цвета и компоновку.

Итак, обсудим эти две вещи.

Изменение языка веб-сайта и области администрирования

В том случае, если вы ориентируете свой сайт на определенную национальную целевую группу, вы должны понимать язык этой группы и строить сайт на этом языке. Если же вы хотите ориентировать сайт на интернациональную аудиторию, то используйте в качестве языка по умолчанию английский.

Независимо от того, какой язык вы выберете, вам придется настраивать языковой файл.

Инсталляция другого файла языка

Если вы хотите обратиться, например, к немецкой целевой группе, загрузите на свой ПК языковые файлы с веб-сайта команды переводчиков на немецкий. Есть языковой файл для интерфейсной части и языковой файл для административной области. Эти файлы созданы командой переводчиков на немецкий (German Translation Team) в составе Николая Плата (Nikolai Plath), Ахима Раджи (Achim Raji), Антонио Камбула (Antonio Cambule), Дэвида Ярдина (David Jardin), Ульриха Эйхенсеера (Ulrich Eichenseer) и Йорна Геркена (Joern Gerken).

Чтобы инсталлировать файлы, войдите на страницу Joomla! Administration, как было описано в главе 2, и щелкните на Extensions⇒Install/Uninstall (Расширения⇒Инсталлировать/деинсталлировать). Появится диспетчер расширений Joomla!, показанный на рис. 4.1.

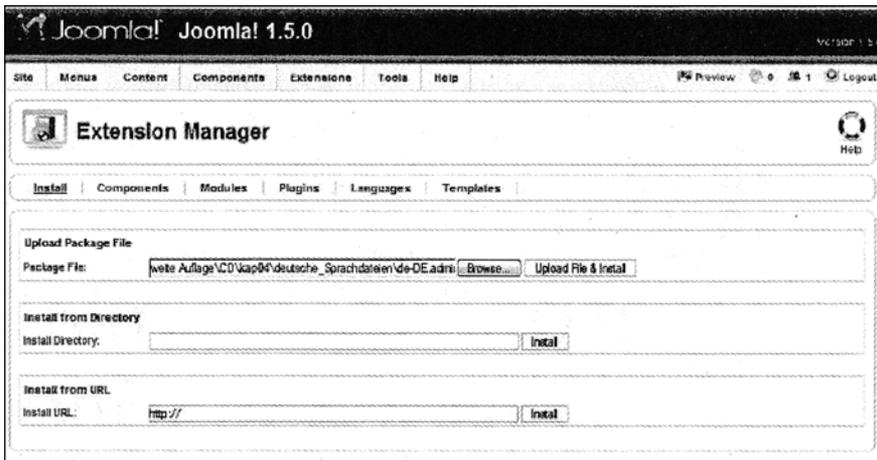


Рис. 4.1. Диспетчер расширений Joomla!

Теперь у вас есть три опции для импорта файлов немецкого языка:

- загрузкой файла с вашего домашнего ПК;
- из каталога в области документов вашего веб-сервера;
- из URL, указывающего на соответствующий языковой пакет.

Выберем первый вариант. Щелкните на кнопке **Browse** (Обзор) рядом с полем **Upload File & Install** (Загрузить файл и установить) и сначала выберите файл `de-DE.site.joomla15RC3.zip`. Затем щелкните на **Upload File & Install**. Если все права корректно настроены, вы получите следующее сообщение:

Install language success
Инсталляция языка прошла успешно

Повторите процедуру с другим файлом.

Теперь обратитесь к диспетчеру языков (рис. 4.2), щелкнув на **Extensions**⇒**Language Manager** (Расширения⇒Диспетчер языков). Вы увидите здесь доступные языки, разделенные на вкладки **Site** (Сайт) и **Administrator** (Администратор). Помимо **English (United Kingdom)** вы увидите теперь **German formal — Sie** и информацию о создателе языкового файла.

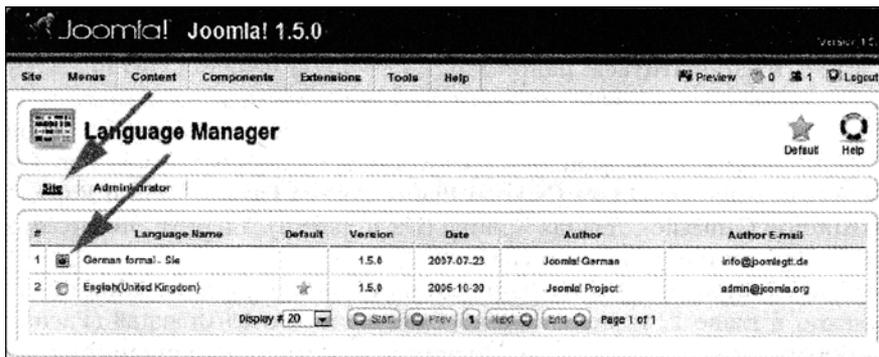


Рис. 4.2. Диспетчер языков

Однако метка зеленого цвета для стандартного языка по-прежнему стоит рядом с English. Выберите переключатель German formal — Sie (Немецкий формальный), а затем щелкните на пиктограмме Default (По умолчанию) справа сверху. Пиктограмма, завершающая диалоги и действия, всегда находится в этой позиции административного интерфейса.

Не забудьте также переключить на немецкий язык и административный интерфейс. После этого ваша Joomla! будет общаться с вами на немецком, как в интерфейсной, так и в серверной части. Такой функциональности серверной части пользователи ждали годами! Теперь, с появлением Joomla! 1.5, их желание наконец-то исполнилось.

Перевод пунктов меню

Теперь ваш сайт "говорит" по-немецки. Зайдите на <http://localhost/joomla150/> и посмотрите. Посмотрите на термин поля Browse.

Эта страница отображает умопомрачительную смесь немецкого и английского языков. Все, что запрограммировано для автоматической работы, вроде процедуры поиска или процедуры входа, будет на немецком, но остальной контент — нет. Почему?

Ответ довольно прост. Переведены могут быть только известные слова и предложения. Значительная часть страницы, однако, содержит введенный пользователем контент. Этот контент появляется на том языке, на котором он был введен. В нашем случае данные примера были представлены на английском.

Так что же можно сделать, чтобы переименовать, например, пункт меню Home (Домой) в Startseite или Main Menu (Главное меню) — в Hauptmenu?

В случае Home щелкните на Menus⇒Main Menu (Меню⇒Главное меню). Вы окажетесь в области Menus.



Joomla! версии 1.0.x имела опцию для изменения контента языковых пакетов посредством языкового редактора. Этот языковый редактор исчез из стандартной версии Joomla! 1.5. Теперь он представлен как устанавливаемое дополнение под названием *диспетчера перевода* (translation manager), доступного для загрузки по адресу: http://extensions.joomla.org/component/option,com_mtree/task,viewlink/link_id,1776/Itemid,35/

Не обращайте внимания на многочисленные опции конфигурации, просто замените Home на Startseite и щелкните на кнопке Apply (Применить). Теперь Startseite появится в вашем Main Menu на веб-сайте.



Joomla! различает действия Apply (Применить), Save (Сохранить) и Exit (Выход). Apply сохраняет изменения и оставляет вас в режиме редактирования. Save также сохраняет изменения, но закрывает режим редактирования и возвращается к отображению списка. Exit не сохраняет ничего и выходит из режима редактирования; другими словами, ведет себя подобно отмене.

Модификация имени меню

Для того чтобы заменить Main Menu на Hauptmenu, откройте диспетчер модуля, щелкнув на Extensions⇒Module Manager (Расширения⇒Диспетчер модулей). Щелкните на ссылке Main Menu для открытия формы — точно так же, как и в случае пункта меню.

Просто измените текст, щелкните на Apply и/или Save, и Main Menu превратится в Hauptmenu.

Изменение шаблона для веб-сайта

Теперь, когда все выглядит более знакомо, вы можете решить, что ваш сайт должен иметь совершенно другой дизайн. (В главе 13 вы узнаете больше о шаблонах.)

На данный момент версия Joomla! 1.5.0 по умолчанию поставляется со следующими шаблонами: ранее представленный шаблон под названием Kepri, затем rhuk_milkyway и свободный от барьеров шаблон Beez. Чтобы гарантировать свободу от барьеров, наилучшую часть исходного кода Joomla! 1.0 по существу пришлось переписать. Шаблон Beez был создан Энджи Радтке (Angie Radtke) и Робертом Деуцем (Robert Deutz) и потому вызвал длившуюся целый месяц дискуссия о смысле и абсурдности свободы от барьеров. В конечном итоге шаблон был включен в стандартную версию Joomla!, что, по моему мнению, добавит популярности Joomla! в Германии (подробнее о Beez читайте в главе 14).

В настоящее время на joomla.org происходит состязание шаблонов. Выигравший шаблон будет включен в качестве третьего шаблона в Joomla! стандартной версии.

Переключиться на шаблон Beez очень легко. Чтобы иметь более широкий выбор и увидеть, насколько это легко на самом деле, просто установите новый шаблон.

Зайдите в инсталлятор Joomla! через меню Extensions⇒Install/Uninstall. Щелкните на кнопке Browse (Обзор) и загрузите файл tmpl_bertrand.zip. Щелкните на Upload File & Install. Новый шаблон будет установлен и инсталлятор сообщит о результате.

Используйте Extensions⇒Templates Manager (Расширения⇒Диспетчер шаблонов) для перехода в область шаблонов. Текущий активный шаблон помечен звездочкой желтого цвета. Если вы наведете курсор мыши на ссылку с именем шаблона, то появится маленькая пиктограмма.

Чтобы назначить шаблон вашему сайту, выберите переключатель перед названием нужного шаблона и щелкните на пиктограмме Default справа вверху. Звездочка желтого цвета теперь окажется рядом с выбранным шаблоном. Переключитесь на вебсайт и щелкните на кнопке Refresh (Обновить) в браузере. Теперь вы получите другую компоновку и полностью иной внешний вид. Создание ваших собственных шаблонов рассматривается в главе 13, а в главе 16 будет создан веб-сайт для мосье Бертрана.

Изменение цветов шаблона

Шаблон по умолчанию rhuk_milkyway таит в себе еще несколько сюрпризов. Вы можете выбрать другую цветовую схему посредством установки параметров.

Выберите снова стандартный шаблон и перейдите к маске редактирования, выбрав переключатель перед шаблоном и щелкнув на пиктограмме Edit (Редактирование). Это переведет вас к конфигурированию шаблона. Существуют различные параметры, которыми можно изменять цвета и ширину шаблона, и даже назначать специальные пункты меню. Поэкспериментируйте с опциями, возвращаясь периодически к веб-сайту, чтобы наблюдать влияние изменений конфигурации.



Joomla! позволяет работать с вкладками в окне браузера. Internet Explorer, начиная с версии 7, также поддерживает просмотр сайтов с использованием вкладок. Благодаря этой технологии, вы можете оставлять веб-сайт и его область администрирования одновременно открытыми в окне браузера и переключаться между ними щелчками на вкладках. Другие браузеры, такие как Firefox, Opera, Safari и Konqueror, имеют эти средства уже не один год.

Резюме

В этой главе было показано, как настроить веб-сайт в части выбора языка и шаблонов.

5

Конфигурация администрирования Joomla!

Joomla! старается обеспечить такую же дружелюбность к пользователю, как у любой среды с графическим интерфейсом пользователя, такой как Windows, KDE, Gnome или Aqua (Mac OS X). Это необычно для веб-сайтов, и стало возможным благодаря обобщенному применению JavaScript и интеграции элементов AJAX. Код JavaScript выполняется локально на вашем компьютере, и его поддержка может быть в любой момент отключена в браузере. Если вы деактивизируете ее, то не сможете выполнять администрирование Joomla!.

Браузеры, однако, успешно работают с JavaScript уже не один год, и уже не представляют серьезных проблем безопасности. Поэтому вам стоит разрешить выполнение кода JavaScript. В этом контексте я настоятельно рекомендую для пользователей Windows два Интернет-браузера с открытым кодом – Mozilla или Firefox. Оба более безопасны и просты в применении, чем Internet Explorer. Главное отличие, однако, в том, что эти браузеры реализуют стандарт W3C.

Компоновка экрана

Панель меню

Страница администратора Joomla!, как и ваш сайт, состоит из разных элементов. Меню представлены в верхней панели меню, как показано на рис. 5.1.

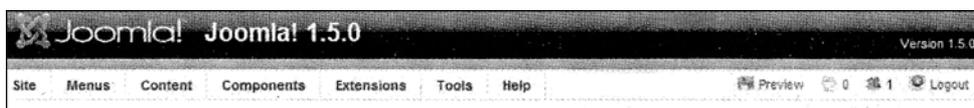


Рис. 5.1. Панель меню интерфейса администратора Joomla!

Здесь справа представлены четыре элемента:

- ссылка на веб-сайт (Preview (Предварительный просмотр));
- примечание о полученных сообщениях и их количестве;
- еще одно примечание о количестве пользователей на сайте в данный момент;
- кнопка Logout (Выход).

Панель инструментов

Ниже панели меню располагается панель инструментов. На рис. 5.2 представлен пример пунктов в главном меню (mainmenu).



Рис. 5.2. Главное меню в диспетчере пунктов меню

Слева находится имя текущей области редактирования с соответствующей пиктограммой (панели инструментов). Справа от него, в зависимости от конкретного диспетчера, расположены различные неактивные пиктограммы. Если вы наведете курсор мыши на одну из них, вокруг появится рамка, и вы сможете щелкнуть левой кнопкой мыши и реализовать соответствующую функцию. В табл. 5.1 перечислены наиболее популярные пиктограммы и их назначение.

Таблица 5.1. Элементы панели инструментов

Элемент панели инструментов	Назначение
Cancel	Редактирование отменяется без сохранения
Enable	Выбранный элемент включен и потому опубликован
Apply	Изменения сохраняются, диалог остается открытым
Edit	Выбранный элемент загружается в модуль редактирования
Disable	Выбранный элемент отключается (скрывается)
Apply	Контент включен и потому опубликован
Upload	Выбранный файл загружается на сервер
Help	Онлайновая справка Joomla!
Copy	Выбранный элемент копируется в другой раздел или категорию
Delete	Выбранный элемент удаляется
Trash	Выбранный элемент помещается в корзину
Save	Выбранный элемент сохраняется и диалог закрывается
Block	Контент отключен (скрыт)
Default	Выбранный элемент становится элементом по умолчанию
Move	Выбранный элемент перемещается в другой раздел или категорию
Preview	Выбранный элемент отображается в его собственном окне предварительного просмотра
Restore	Выбранный элемент извлекается из мусорной корзины

Подменю

Под панелью инструментов часто располагаются дополнительные меню, зависящие от текущего диспетчера. На рис. 5.3 показано подменю Banner Manager (Диспетчер баннеров).

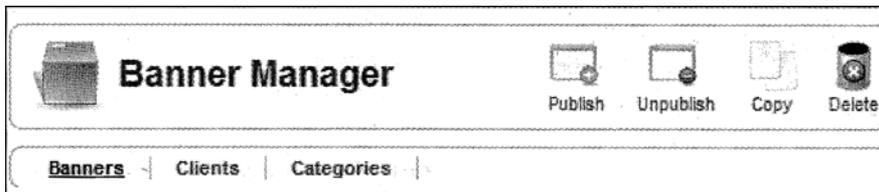


Рис. 5.3. Подменю диспетчера баннеров

Вы найдете здесь вкладки Banners (Баннеры), Clients (Клиенты) и Categories (Категории). Их называют также подменю или подкатегориями. В зависимости от контекста, здесь также могут появляться и системные сообщения.

Фильтрующие элементы

Ниже подменю находятся фильтрующие элементы. Они могут фильтровать информацию по разным критериям, в зависимости от отображаемого списка. Вы можете искать по заданной строке или, например, отображать только модули в определенных позициях, только модули определенного типа или, как сделано здесь (рис. 5.4), отображать только включенные (Enabled) модули.

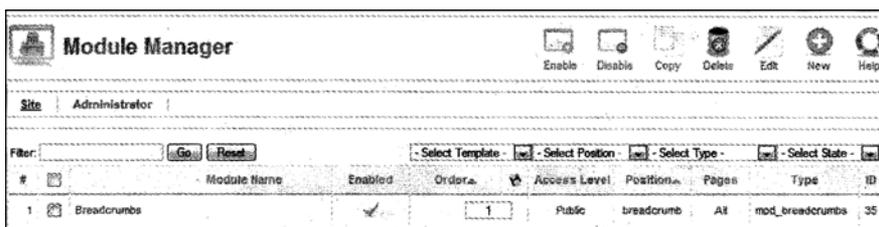


Рис. 5.4. Диспетчер модулей с элементами фильтрации

Область контента

Ниже фильтрующих элементов находится обширная область, в которой отображается действительный контент. Он бывает трех типов.

Списки

Список элементов, которые могут редактироваться после отметки соответствующего флажка и/или щелчка на имени. Если вы щелкнете на самом верхнем флажке, то все элементы будут выбраны для совместного редактирования. Также в вашем распоряжении функция сортировки списка вместе с опцией включения/выключения элемента единственным щелчком. Под списком находится элемент навигации. Вы можете изменить количество отображаемых элементов списка и перемещаться по страницам. На рис. 5.5 показан пример списка из меню ключевых концепций.

#	Menu Item	Default	Published	Order	Access Level	Type	ItemID
1	Extensions		✓	1	Public	Articles » Article	40
2	Content Layouts		✓	2	Public	Articles » Article	38
3	Example Pages		✓	3	Public	Articles » Article	43

Filter: [] [Go] [Reset] Max Levels: 10 [v] - Select State: [v]
Display # 20 [v]

Рис. 5.5. Пример списка

Диалоги

В зависимости от диалога, главный контент обычно отображается в левой области, а параметры для этого элемента – в правой. Выполнить семантическое разделение свойств и параметров элемента зачастую не так просто; поэтому иногда имеются и другие области, взятые в рамки, содержащие конфигурационные опции слева.

Прочее

Под этим понимается информация, например, в Control Panel (Панель управления), которую вы видите первым делом после входа в административную область (рис. 5.6). Наиболее часто используемые команды меню находятся в левой части в форме пиктограмм. Информация о вашем веб-сайте (модуль admin) расположена в правой части. Иногда контент Joomla! не вписывается в схему список/диалог; в этом случае вы встретите и другие области, структурированные иначе. Эта проблема отображения иногда сопровождается наиболее сложные дополнительные компоненты.

The screenshot shows the Joomla! Control Panel. On the left, there is a grid of icons for various management tasks: Add New Article, Article Manager, Front Page Manager, Section Manager, Category Manager, Media Manager, Menu Manager, Language Manager, User Manager, and Global Configuration. On the right, there is a sidebar with the following sections:

- Welcome to Joomla!
- Logged in Users
- Popular
- Recent added Articles
- Menu Stats

Menu	# Items
ExamplePages	4
keyconcepts	3
mainmenu	9
othermenu	5
topmenu	4
usermenu	4

Рис. 5.6. Панель управления

Справка

Справка очень важна, и по этой причине Joomla! пытается предоставить ее везде, где только возможно.

Пиктограмма вызова справки

В большинстве списков и диалогов в панели пиктограмм вдоль правой границы присутствует пиктограмма под названием Help (Справка). Если вы щелкнете на этой пиктограмме, будучи подключенными к Интернету, в окне браузера будет отображена

соответствующая страница справки с сервера joomla.org. Идея состоит в том, что существуют серверы справочной информации на разных языках, которые локализуют текст справки, и потому вы всегда можете рассчитывать на текущую помощь.

Справка всплывающими подсказками

Всплывающие подсказки (speech balloons) — очень удобное решение. Текст справки для них вводится в языковой файл и отображается в соответствии с текущими локальными установками, без подключения к Интернету.

Меню справки

Меню Help (Справка), которое находится в самом конце, предоставляет концентрированную информацию о Joomla! и ее окружении. Оно содержит два пункта: Joomla! Help (Справка Joomla!) и System Info (Информация о системе).

Пункт Joomla! Help

Рабочая область разделена на три части (рис. 5.7). В верхней части есть поле поиска и панель со ссылками.

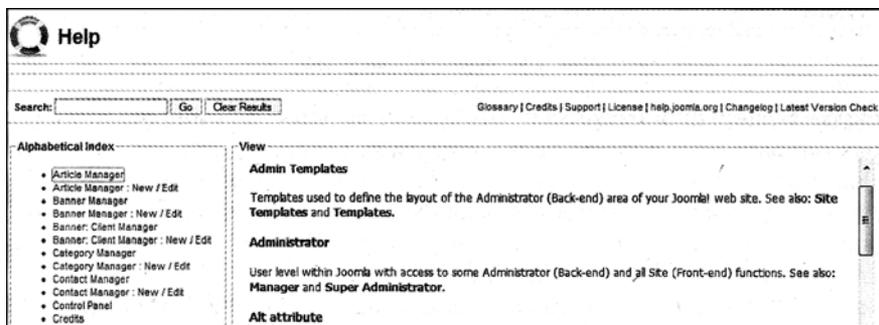


Рис. 5.7. Экран Joomla! Help

Ниже слева находится указатель доступных в онлайн текстов справки, а справа — соответствующая область отображения. По умолчанию вы увидите новые средства используемой вами версии Joomla!. Ссылки, ассоциированные с текстами справки, указывают на сервер <http://help.joomla.org/>. Вы можете конфигурировать этот сервер справки для применяемого языка (см. главу 6). Поиск также обращается к этому серверу, и, конечно, для его использования вы должны быть подключены к Интернету. Это необходимо в любом случае, если только вы не работаете с локальной версией без соединения с Интернетом.

Другие ссылки ведут к тексту лицензии GNU/Public License и журналу изменений, который документирует модификации индивидуальных разработчиков.

Пункт System Info

Системную информацию вы можете найти в пункте меню Help⇒System Info (Справка⇒Информация о системе). Эта информация представлена на пяти вкладках (рис. 5.8):

- System Info (Информация о системе)
- PHP Settings (Настройки PHP)
- Configuration File (Конфигурационный файл)
- Directory Permissions (Права доступа к каталогам)
- PHP Information (Информация PHP)

Setting	Value
PHP Built on:	Windows NT TEST56-PC 6.0 build 6000
Database Version:	5.0.51a
Database Collation:	utf8_general_ci
PHP Version:	5.2.5
Web Server:	Apache/2.2.8 (Win32) DAV/2 mod_ssl/2.2.8 OpenSSL/0.9.8g mod_autoindex_color PHP/5.2.5
Web Server to PHP interface:	apache2handler
Joomla! Version:	Joomla! 1.5.0 Production/Stable [Khepri] 21-January-2008 23:55 GMT
User Agent:	Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.8.1.12) Gecko/20080201 Firefox/2.0.0.12

Рис. 5.8. Экран System Info

На вкладке System Info, например, отображается итоговая информация о главных параметрах операционной системы и настройках сервера, которые наиболее важны для Joomla!. В данный момент я работаю локально в среде Windows и XAMPP Lite (за описанием процедуры инсталляции обращайтесь в главу 2), и потому у меня функционируют более современные версии (Apache 2.2.6 и PHP 5.2.6) по сравнению с серверами, которые можно арендовать в Интернете.

На вкладке PHP Settings представлена важнейшая информация из файла `php.ini`. Этот файл, например, определяет, включен или выключен безопасный режим PHP. В большинстве арендованных серверов вы не можете вносить изменения в файл `php.ini`.

На вкладке Configuration File отображаются значения переменных из файла `configuration.php`. Этот файл создается во время инсталляции. Из соображений безопасности параметры базы данных маскируются.

На вкладке Directory Permissions показаны права доступа к вашим подкаталогам. Чтобы Joomla! работала без ошибок, все каталоги должны быть открыты для записи (а не защищены от записи). Вы можете изменить права доступа к каталогам в программе FTP с помощью команды `chmod`.

На вкладке PHP Information отображается результат вызова PHP-функции `phpinfo()`. Это касается общей конфигурации интерпретатора PHP. В зависимости от провайдера, в этом файле можно, например, увеличить доступное пространство хранилища.

Резюме

В этой главе были описаны операции административной области, ее конфигурация и администрирование контента. Вы узнали об элементах меню, панели инструментов и т.п., а также об их назначении. Было также рассмотрено меню получения справки.

6

Меню Site

Существуют глобальные установки, которые касаются всех индивидуальных страниц и вашего сервера. Все эти установки собраны в меню Site (Сайт).

Здесь вы найдете пять пунктов меню (рис. 6.1):

- Control Panel (Панель управления)
- User Manager (Диспетчер пользователей)
- Media Manager (Диспетчер медиа)
- Configuration (Конфигурация)
- Logout (Выход)

Пункт Control Panel

Первый пункт меню – Control Panel (рис. 6.1) – переносит вас на стартовую страницу административного раздела.

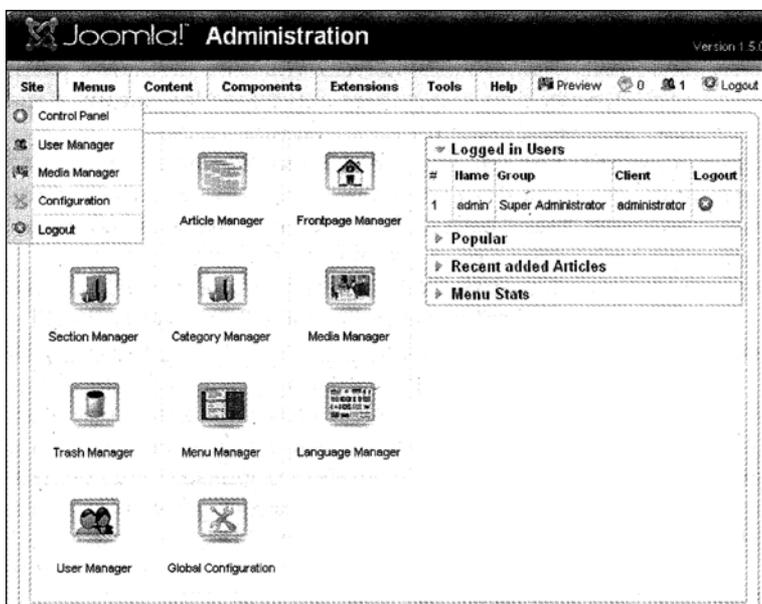


Рис. 6.1. Пункты меню Site

Вы можете использовать Control Panel для перехода к различным разделам администрирования — либо выбором пункта меню, либо щелчком на отображаемых пиктограммах. Приобретя некоторый опыт в администрировании, вы предпочтете работать с пиктограммами быстрого доступа.

Пункт User Manager

Пользователи играют особую роль на вашем сайте Joomla!. Вы можете зарегистрировать столько пользователей, сколько хотите, и в зависимости от их прав, они могут создавать собственный контент и/или просматривать контент, предназначенный для определенных пользователей.

На данный момент вы — единственный пользователь, которого распознает администратор Joomla!. Ваше имя — admin. Вы можете работать со всей инсталляцией Joomla!. Если вы разрешите регистрацию пользователей на сайте (см. раздел "Раздел Global Configuration"), то очень скоро у вас заведется множество новых пользователей.

В рабочей области пользователя (рис. 6.2) вы можете редактировать и удалять пользователей, выдавать им различные права и, разумеется, также настраивать новые пользовательские учетные записи.

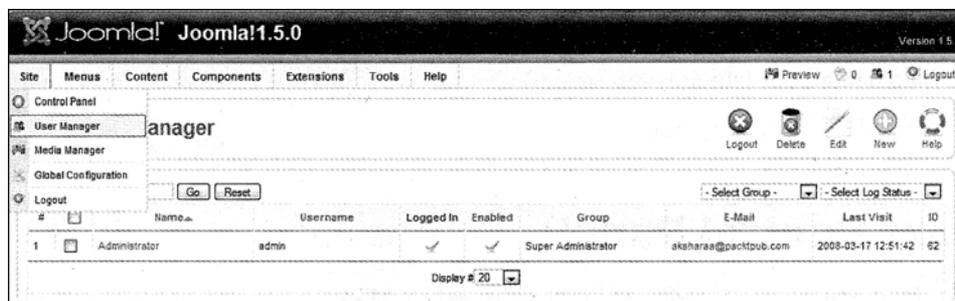


Рис. 6.2. Пункты меню User Manager

В обзорном списке вы можете видеть наиболее важную информацию о пользователе—действительное имя (Name) пользователя (Administrator), регистрационное (User name) имя (admin), вошел ли пользователь (Logged In) на сайт в данный момент (отмечено галочкой), активизирован ли пользователь (Enabled), его участие (Group) в группах (Super Administrator), его адрес электронной почты (E-Mail) и дата последнего посещения сайта (Last Visit). Это касается регистрации как в интерфейсной, так и в административной части сайта. Вы можете также узнать номер ID, который присвоен данному пользователю в базе данных (62).

Вы можете отфильтровать список по группам пользователей и состоянию регистрации. Поле поиска слева позволяет искать пользовательские учетные записи, вводя частичные имена.

В панели инструментов области пользователей можно выбрать пиктограммы действий Logout (Выход), Delete (Удалить), Edit (Редактировать), New (Создать) и Help (Справка). Некоторые действия могут быть применимы к нескольким пользователям сразу, например, Delete. Достаточно только отметить флажки слева или в самом верху, чтобы выбрать всех пользователей в списке. Затем для удаления нужно щелкнуть на пиктограмме Delete.

Пиктограмма Logout

Эта функция удобна, если вы хотите отключить зарегистрированного в данный момент пользователя. Подобное может понадобиться в, качестве крайней меры.

Пиктограмма Delete

Посредством этой функции вы можете удалять одну или несколько пользовательских учетных записей, отметив их флажки.

Пиктограмма Edit

Это позволяет редактировать индивидуального пользователя. Возьмем, к примеру, учетную запись администратора. Щелкните на ссылке Administrator или отметьте флажок и щелкните на пиктограмме Edit. Появится форма редактирования информации о пользователе, показанная на рис. 6.3.

Рис. 6.3. Форма редактирования информации о пользователе

На этом экране вы видите три следующих раздела:

- User Details (Детальные сведения)
- Parameters (Параметры)
- Contact Information (Контактная информация)

Раздел User Details включает перечисленные ниже поля.

- Name (Имя). Реальное имя пользователя (здесь – Administrator).
- Username (Регистрационное имя). Регистрационное имя пользователя служит для идентификации при процедуре входа и не зависит от реального имени. Это имя должно быть кратким и легко запоминающимся. Например, в качестве регистрационного имени для Hagen Graf может использоваться hgraf. Joomla! допускает в регистрационном имени пробелы, специальные символы и действительные символы Unicode. Однако имейте в виду, что однажды вы можете оказаться за клавиатурой, не поддерживающей эти специальные символы. Как тогда вы регистрируетесь?

- Email (Адрес электронной почты). Действительный адрес электронной почты пользователя.
- New Password (Новый пароль) и Verify Password (Проверить пароль). Пароль пользователя должен быть введен два раза для верификации. Если вы оставите эти поля пустыми, останется в силе прежний пароль. Длина пароля должна составлять не менее шести символов.
- Group (Группа). Каждый пользователь включен в группу с соответствующими правами. Группы делятся на две большие части: пользователи, которым разрешено лишь посещать ваш веб-сайт (Public Front-end), и пользователи, которые могут заходить в область администрирования Joomla! (Public Back-end). Весь контент Joomla! может быть назначен этим группам. В табл. 6.1 перечислены пользовательские группы интерфейсной части сайта.

Таблица 6.1. Группы пользователей интерфейсной части

Группа	Права
Registered (Зарегистрированный)	Зарегистрированный пользователь может входить и видеть части сайта, не доступные незарегистрированным посетителям.
Author (Автор)	Автор может делать все, что и зарегистрированный пользователь. Автор может также записывать информацию и модифицировать свою собственную информацию. Обычно для этого в пользовательском меню появляется соответствующая ссылка.
Editor (Редактор)	Редактор может делать все то же, что и автор. Редактор также может записывать и редактировать всю информацию, появляющуюся в интерфейсной части сайта.
Publisher (Публикатор)	Публикатор может делать все то же, что и редактор. Публикатор может также записывать информацию и редактировать каждую часть информации, которая появляется в интерфейсной части сайта. Вдобавок он может решать, публиковать либо нет ту или иную информацию.

В табл. 6.2 показаны группы пользователей административной части.

Таблица 6.2. Группы пользователей административной части

Группа	Права
Manager (Менеджер)	<p>Менеджер может создавать контент и видеть различную информацию о системе. Ему не разрешено:</p> <ul style="list-style-type: none"> • администрировать пользователей; • устанавливать модули или компоненты; • повышать статус пользователя до супер-администратора или модифицировать супер-администратора; • работать с пунктом меню Site⇒Global Configuration (Сайт⇒Глобальная конфигурация); • осуществлять массовые рассылки электронной почты всем пользователям; • изменять и/или устанавливать шаблоны и языковые файлы.

Группа	Права
Administrator (Администратор)	Администратору не разрешено: <ul style="list-style-type: none"> • повышать статус пользователя до супер-администратора или модифицировать супер-администратора; • работать с пунктом меню Site⇒Global Configuration (Сайт⇒Глобальная конфигурация); • осуществлять массовые рассылки электронной почты всем пользователям; • изменять и/или устанавливать шаблоны и языковые файлы.
Super Administrator (Супер-администратор)	Супер-администратор может выполнять все функции в административной области Joomla!. Только супер-администратор может настроить другого супер-администратора.

В разделе User Details можно указывать, должен пользователь получать системную электронную почту или нет. Вы можете видеть дату регистрации пользователя и дату последнего успешного входа.



Special Users (Специальные пользователи). На протяжении этой книги мы еще вернемся к проблеме прав специальных пользователей. Специальным считается любой пользователь, который имеет больше прав, чем менеджер. На данный момент в Joomla! невозможно создавать ваши собственные группы пользователей. Поэтому группа специальных пользователей удобна для ограничения элементов контента для этой группы. Это может пригодиться, например, если кто-то желает предоставить ссылки для доступа к внутренним файлам справки только этим специальным пользователям. Модификация и расширение таких групп были изначально отличительной чертой Joomla! версии 1.5.0, но пока это включено в дорожную карту следующей версии и есть соответствующий проект Google Summer of Code. Это средство может быть реализовано к моменту выхода в свет настоящей книги.

Поле Parameters (Параметры) содержит переключатели для пользовательского языка. Вы можете выбрать редактор, разрешенный пользователю, а также сервер справки с нужным языком.

Поскольку оперировать интерфейсной и административной частями в Joomla! 1.5 можно на разных языках, теперь появилась возможность иметь административные учетные записи со смешанными языками. В полях Back-end Language (Язык административной части) и Front-end Language (Язык интерфейсной части) можно выбрать требуемые языки для пользовательской учетной записи из списка доступных.

То же самое можно сделать с User Editor (Пользовательский редактор). Для этих целей Joomla! 1.5 предлагает WYSIWIG-редактор TinyMCE, который показан на рис. 6.4.

На веб-сайте TinyMCE (<http://tinymce.moxiecode.com/>) можно приобрести программное обеспечение манипуляций файлами и изображениями. Доступно множество других редакторов, таких как JCE (<http://www.cellardoor.za.net/>) и FCKeditor (<http://www.fckeditor.net/>), которые могут применяться в Joomla!. Для поддержки чистоты кода XHTML можно пользоваться бесплатным редактором XStandard-lite (<http://www.xstandard.com/>). Это очень совершенный WYSIWYG-редактор, который генерирует корректный код XHTML и предоставляет богатые возможности для интеграции. Однако сам редактор не включен в поставку Joomla! и должен быть загружен отдельно с веб-сайта компании.

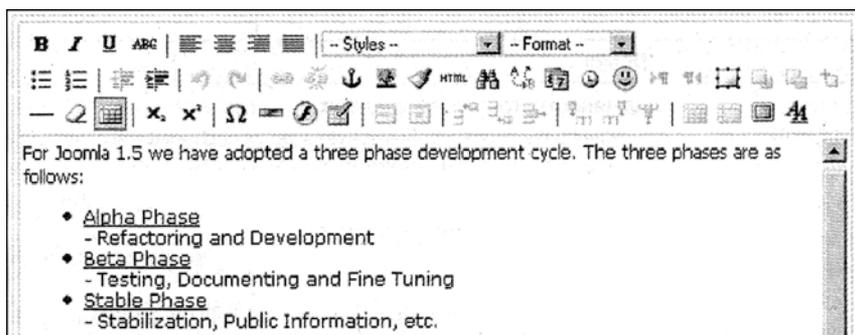


Рис. 6.4. Редактор TinyMCE

Существует бесплатная облегченная версия, а также профессиональная версия, которую можно приобрести. Firefox обнаружит это и при необходимости повторно установит редактор в качестве подключаемого модуля. После инсталляции редактора и выбора пользовательской учетной записи он будет показан в соответствующих полях редактирования.

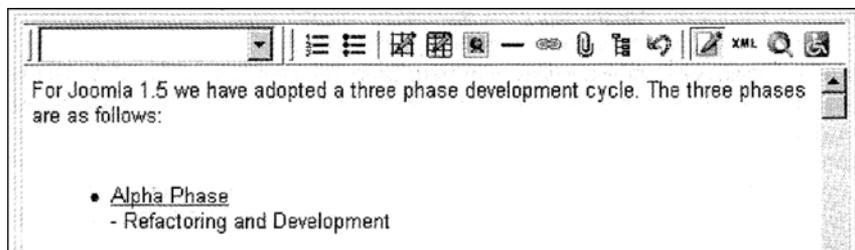


Рис. 6.5. Поля редактирования, связанные с TinyMCE

Из списка в области параметров можно также выбрать индивидуальный сайт справки (Help Site). Благодаря широкой популярности Joomla!, теперь имеются многочисленные локальные серверы справки.

Раздел Contact Information также предоставляет информацию о доступных формах контактов с данным пользователем. Joomla! содержит компонент, который позволяет создавать индивидуальные формы контактов (см. раздел "Контакты" в главе 9 или выберите Components⇒Contacts⇒Contacts (Компоненты⇒Контакты⇒Контакты)).

Сохранить изменения и выйти из диалога можно щелчком на пиктограмме Save (Сохранить). Щелчок на пиктограмме Apply (Применить) сохранит изменения и оставит вас в режиме редактирования шаблона.

Пиктограмма New

Теперь, когда вы внесли несколько начальных изменений, установите пользовательскую учетную запись с вашим собственным именем. Назначьте пользователя в группу или зарегистрируйте пользователей и задайте требуемые параметры на странице User Manager (Диспетчер пользователей), которая показана на рис. 6.6. Мой пользователь называется Hagen Graf, его регистрационное имя – hgraf, он не вошел в систему в данный момент (но уже активизирован), входит в группу пользователей под названием Registered и имеет ID, равный 63. Имеется сообщение об ошибке,

связанное с тем, что приветственное электронное письмо не может быть отправлено. Поскольку мы работаем локально и не сконфигурировали сервер электронной почты, интерпретатор PHP не может отправить почту.

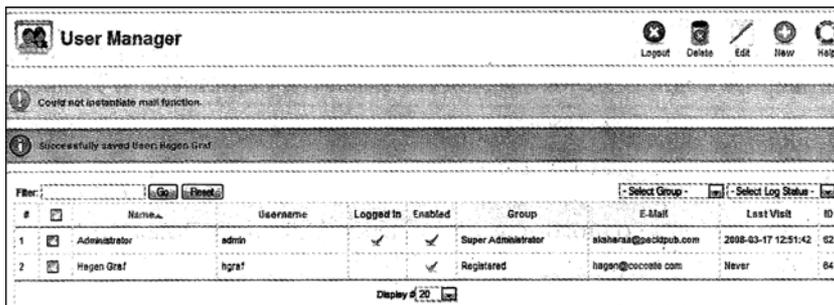


Рис. 6.6. Страница User Manager

Новый пользователь может теперь регистрироваться в интерфейсной части, используя свое регистрационное имя и пароль. Администратор имеет выбор, установить пользователя предварительно или позволить ему регистрироваться самостоятельно. Мы вернемся к этой теме в разделе "Раздел Global Configuration".

Область Media Manager

О рабочей области Media Manager (Диспетчер медиа) можно думать, как о проводнике файлов или программе FTP из операционной системы (рис. 6.7). Здесь можно загружать файлы с расширениями bmp, csv, doc, eps, gif, ico, jpg, odg, odp, ods, odt, pdf, png, ppt, swf, txt, xcf, xls, BMP, CSV, DOC, EPS, GIF, ICO, JPG, ODG, ODP, ODS, ODT, PDF, PNG, PPT, SWF, TXT, XCF и XLS и администрировать их в различных каталогах. Этот диспетчер особенно удобен, если у вас нет FTP-доступа, даже при наличии административных прав.

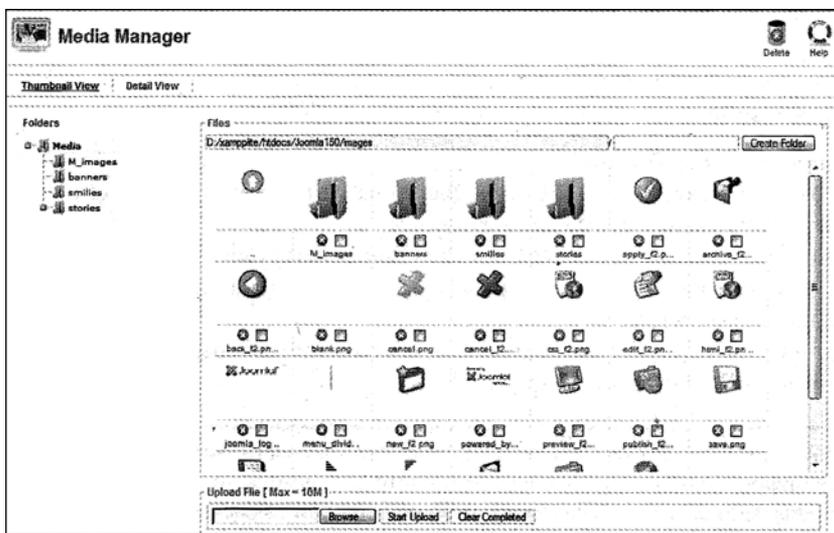


Рис. 6.7. Страница Media Manager

В медиа-области есть две вкладки: Thumbnail View (Миниатюры), которая показывает пиктограммы предварительных отображений файлов, и Detail View (Детальный вид), которая отображает имена и размеры файлов в списке.

Имеется также отдельная область для папок и файлов. Если вы щелкнете на папке слева, то увидите справа файлы, содержащиеся в этой папке.

Вся медиа-информация, к которой обычно обращается Joomla!, администрируется диспетчером медиа. Это, в частности, касается смайликов и графики в папке `M_images`.

В медиа-области доступны следующие функции:

- создание и удаление каталогов (в Detail View можно только удалять);
- загрузка и удаление одного или нескольких файлов (в Detail View можно только удалять).

Создание каталогов

В разделе Files (Файлы) показан путь к текущему каталогу. Он имеет поле ввода и кнопку Create Folder (Создать папку). Введите желаемое имя каталога в поле ввода (здесь введено `france`) и щелкните на кнопке Create Folder. Каталог создается и отображается с соответствующей пиктограммой каталога (рис. 6.8).

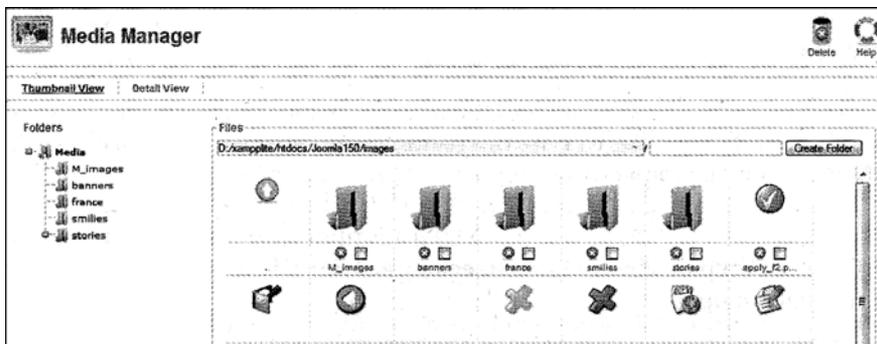


Рис. 6.8. Создание нового каталога

Загрузка файла

В нижней части страницы Media Manager есть кнопка Browse (Обзор). Щелчок на ней приводит к вызову диалога выбора файла вашей операционной системы, где можно выбрать необходимые файлы для загрузки.

После щелчка на кнопке Start Upload (Начать загрузку) выбранные файлы будут сохранены в новом каталоге и отображены в обзоре. Позднее они будут вставлены в различные части контента.

Щелчок на миниатюре приводит к тому, что изображение отображается в его исходном размере на затемненном фоне.



Имейте в виду, что будет плохой идеей отображать 5-мегабайтные изображения, полученные цифровой камерой, на вашем веб-сайте с их исходным разрешением. Изображения должны быть размером максимум от 50 до 150 Кбайт. Помните о людях, не имеющих высокоскоростного доступа в Интернет.

Принимайте во внимание основополагающее правило о длительности загрузки 100-килобайтового файла (примерно таков размер веб-страницы портала с картинками), описанное в табл. 6.3. Даже если у вас будет на странице одна картинка размером в 300 Кбайт, вы потеряете всех пользователей, имеющих ISDN-подключение и обычную телефонную линию для доступа в Интернет.

Таблица 6.3. Длительность загрузки 100 Кбайт для разных типов соединений

Тип соединения	Длительность загрузки 100 Кб данных
DSL	Зависит от провайдера DSL, но меньше секунды
ISDN	Около 15 секунд
Модем (56 К)	Около 25 секунд

Раздел Global Configuration

В разделе Global Configuration (Глобальная конфигурация) можно определять настройки, которые действительны для всего веб-сайта (рис. 6.9). Значения переменных сохраняются в файле `configuration.php`. В этом файле хранится важнейшая информация, такая как параметры доступа к серверу баз данных, и менее важные параметры, вроде предварительно определенной длины отображаемых списков. Рабочая область делится на три вкладки:

- настройки Site (Сайт)
- настройки System (Система)
- настройки Server (Сервер)

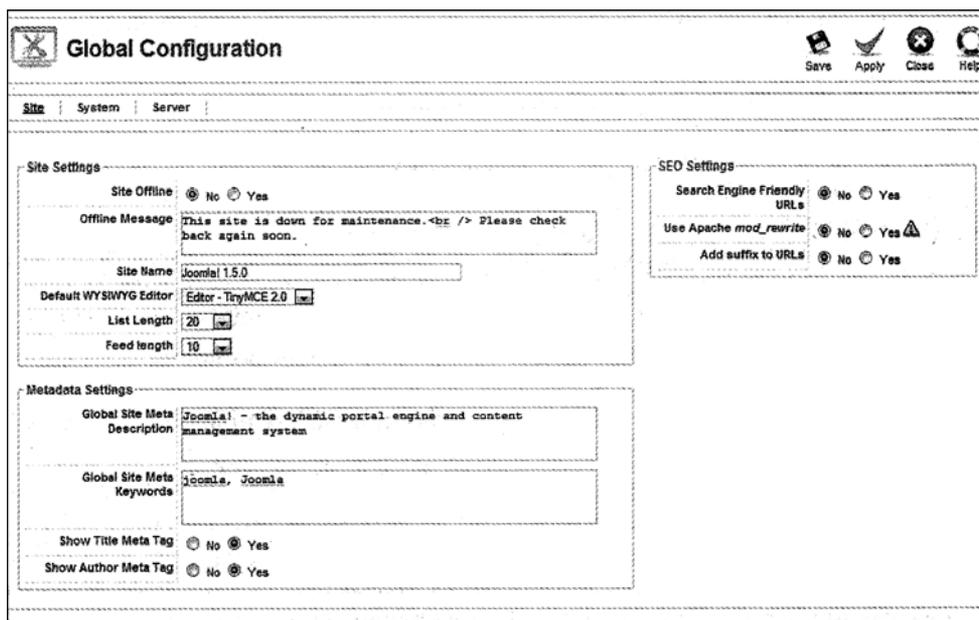


Рис. 6.9. Страница Global Configuration

Настройки Site

Раздел настроек Site, в свою очередь, делится на три визуальных области: Site Settings (Настройки сайта), Metadata Settings (Настройки метаданных) и SEO Settings (Настройки SEO).

Область Site Settings

Обсудим подробно все настройки из этой области.

- Site Offline (Сайт отключен). Если необходимо внести изменения в ваш веб-сайт и информировать посетителей об этом, щелкните на переключателе Yes (Да).
- Offline Message (Сообщение об отключении). Текст, который вы введете здесь, будет отображаться в верхней части вашего веб-сайта, когда он находится в отключенном состоянии. Попробуйте установить Site Offline в Yes, щелкнуть на пиктограмме Apply (Применить) и зайти на веб-сайт. Вы обнаружите, что он недоступен (рис. 6.10). Пользователи, имеющие право доступа к административной области (Rights⇒Manager (Права⇒Менеджер)), по-прежнему могут входить.

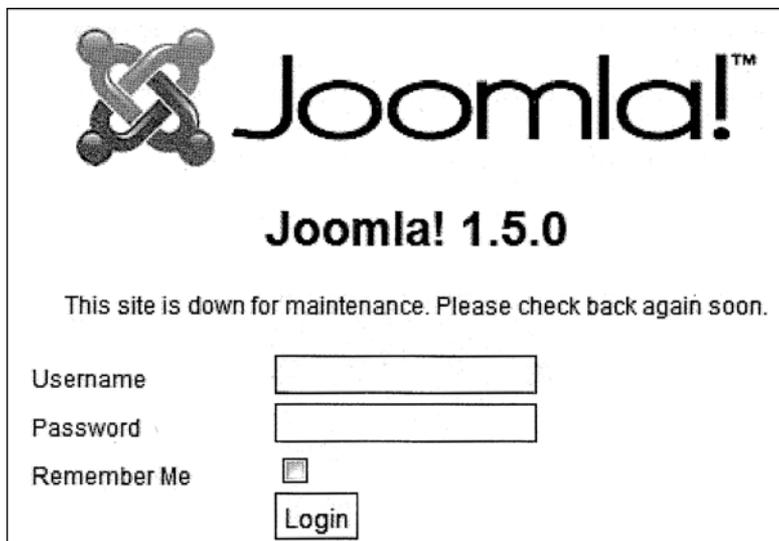


Рис. 6.10. Экран входа с сообщением о недоступности сайта

- Site Name (Имя сайта). Имя сайта, которое вы ввели во время инсталляции. Отображается, в частности, в заголовке браузера, а также в качестве отправителя системных сообщений электронной почты, в лентах новостей и других местах, что позволяет пользователям легко опознать ваш веб-сайт.
- Default WYSIWYG Editor (WYSIWYG-редактор по умолчанию). **WYSIWYG** — сокращение от "What you see is what you get" ("что видишь, то и получаешь"). Этот термин происходит от первых дней существования графических интерфейсов пользователя, когда впервые стало возможно видеть на экране текст в точности так, как он выглядит при печати. В наше время он меньше касается печати, а больше того, как текст выглядит на веб-сайте.

В Интернете вы обычно заполняете формы без каких-либо опций форматирования. Форматирование осуществляется посредством дескрипторов HTML или специфичных для программы мнемоник. Редактор WYSIWYG более дружелюбен к пользователю, поскольку вам достаточно щелкнуть на соответствующих пиктограммах, подобно редактированию текста. Редактор автоматически включается в текстовых полях, требующих форматирования. Он работает со всеми популярными браузерами. Joomla! также предлагает опцию для инсталляции других HTML-редакторов, таких как XStandard Lite. По умолчанию в настоящее время в Joomla! используется редактор TinyMCE.

- List Length (Длина списка). Списки новостей и ссылок разбросаны по всему сайту. С помощью этого параметра задается количество элементов по умолчанию, которое должен содержать список.

Область Metadata Settings

В этой области находятся настройки для метаданных (рис. 6.11).

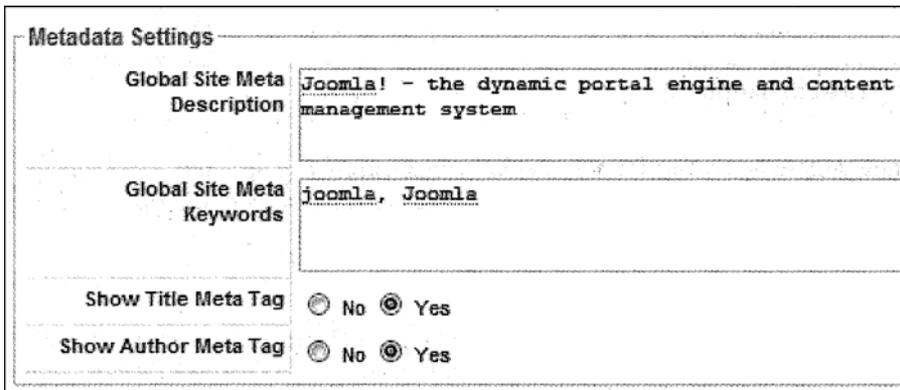


Рис. 6.11. Область Metadata Settings

Метаданные — это данные о данных, например, описание вашего веб-сайта. Метаданные важны для механизмов поиска и представляются на языке XHTML. Действительная важность метаданных проявляется в рейтинге, учитываемом поисковым механизмом. Однако метаданные предлагают хороший способ описания вашего веб-сайта — краткими и точными словами. Если вы просмотрите исходный код HTML сайта Joomla!, то обнаружите дескрипторы метаданных в его верхней части:

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

- Global Site Meta Description (Глобальное мета-описание сайта). Это описание контента сайта часто отображается как результат работы механизмов поиска. Вы должны уделить особое внимание этому дескриптору, поскольку пользователь, который ввел искомый фрагмент, часто использует эту информацию для при-

нятия решения относительно того, стоит ли посетить обнаруженный сайт. Вы можете добавить дополнительную информацию к глобальному описанию, с дополнительными описаниями для каждой индивидуальной страницы.

- **Global Site Meta Keywords** (Глобальные ключевые слова сайта). Ключевые слова составляют важнейшую часть документа. Здесь потребуется перечислить наиболее важные термины, используемые на вашем веб-сайте. Некоторые поисковые механизмы специально индексируют ключевые слова. Индивидуальные слова разделяются запятыми; несколько слов можно ввести между запятыми с обычными пробелами между ними. Ключевые слова должны быть ограничены максимум 1000 символами; больше просто не будет прочитано поисковым механизмом. Обратите внимание, что применение меньшего количества ключевых слов повышает достоверность поиска каждого для поискового механизма. Тщательно продумывайте наиболее часто используемые ключевые слова, и которые из них, скорее всего, будут искать чаще всего. На каждой индивидуальной странице веб-сайта можно вводить дополнительные ключевые слова в глобальный список ключевых слов.

- **Show Title Meta Tag** (Показывать мета-дескриптор заголовка). Заголовок контента для каждой индивидуальной страницы контента показывается в виде дескриптора `meta`, например:

```
<meta name="title" content="Welcome to Joomla!" />
```

- **Show Author Meta Tag** (Показывать мета-дескриптор автора). Имя автора для индивидуальных страниц контента отображается как дескриптор `meta`, например:

```
<meta name="author" content="Administrator" />
```

Область **SEO Settings**

Опции в области настроек SEO (Search Engine Optimization – поисковая оптимизация) касаются дружественных к механизму поиска URL, или SEF URL (рис. 6.12). Обычно URL Joomla! выглядят примерно так:

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

URL подобного типа обычно не сохраняются поисковым механизмом, поскольку этот механизм предполагает, что контент, сконструированный динамически, вскоре изменится.

- **Search Engine Friendly URLs** (Дружественные к механизму поиска URL). С помощью этого переключателя вы можете трансформировать динамические URL в дружественные для поискового механизма URL (SEF URL). Если вы установите этот переключатель в **Yes** (Да), ссылка будет выглядеть примерно так:

```
http://localhost/joomla150/contact
```

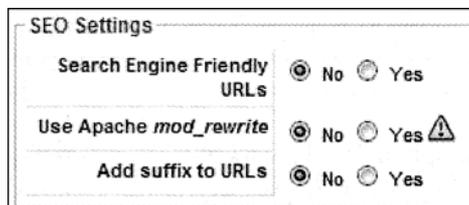


Рис. 6.12. Область **SEO Settings**

- Use Apache *mod_rewrite* (Использовать Apache-модуль *mod_rewrite*). Этот переключатель активизирует модуль *mod_rewrite*. Принцип основан на функции веб-сервера Apache, которая может манипулировать URL-адресами принудительно с помощью внутреннего механизма *перезаписи* (*rewrite engine*). Помимо выбора переключателя, вы также должны переименовать файл *htaccess.txt* в каталоге Joomla! в *.htaccess*. В Windows такое переименование возможно лишь в определенных программах, например, в редакторе Ultraedit (<http://www.ultraedit.com/>) или в командной строке посредством следующей команды:

```
rename htaccess.txt .htaccess
```

В Linux переименование работает без проблем; однако файл впоследствии не отображается в клиенте FTP (в зависимости от серверной конфигурации). В дополнение к этому, провайдер может не разрешать работать с файлами *.htaccess*, поскольку они представляют угрозу безопасности для веб-сервера.



Для того чтобы ваш сайт Windows/XAMPP также трансформировался в SEF URL, потребуется отредактировать следующий файл:

```
[диск]:\xampplite\apache\conf\httpd.conf
```

Откройте этот файл в программистском редакторе или в WordPad и найдите в нем следующую строку:

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

Удалите символ # из этой строки, чтобы она выглядела так:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

и сохраните файл. Затем запустите веб-сервер Apache снова через панель управления XAMPP Lite. После этого SEF URL должны заработать

- Add a suffix to URLs (Добавлять суффикс к URL). Это добавляет *.html* к URL после расширения имени файла, так что оно выглядит как сайт HTML. Тем не менее, влияние, которое это оказывает на механизмы поиска, спорно.

Настройки System

Область System Settings

Область System Settings (Настройки системы) показана на рис. 6.13.

System Settings	
Secret Word	Ej48y5X7XbaDVVwt
Path to Log folder	D:\xampplite\htdocs\joomla150\logs
Enable Web Services	<input type="radio"/> No <input checked="" type="radio"/> Yes
Help Server	English (GB) - help.joomla.org <input type="button" value="Refresh"/>

Рис. 6.13. Область System Settings

- Secret Word (Секретное слово). Когда вы устанавливаете Joomla!, создается случайная последовательность символов — некая разновидность печати. Это секретное слово используется для подключений через интерфейс XML/RPC. Это нужно для того, чтобы гарантировать, что в будущем применять этот интерфейс

для получения доступа смогут только зарегистрированные и/или авторизованные удаленные системы.

- Path to the Log folder (Путь к папке журналов). Если вы хотите использовать функцию протоколирования Joomla!, вы, естественно, должны ввести путь. Путь определен по умолчанию, и может быть изменен в этом поле. Этот путь должен быть вне общедоступного пути, то есть за пределами вашего каталога документов htdocs.
- Enable Web Services (Разрешить веб-службы). В Joomla! поддерживается интерфейс xmlrpc. Этот интерфейс обеспечивает доступ к веб-службам, которые могут быть включены или отключены. Данная настройка влияет только на входящие соединения XML-RPC, а не на исходящие.
- Help Server (Сервер справки). Здесь вы определяете сервер справки по умолчанию. Для каждой пользовательской учетной записи можно сконфигурировать отдельный сервер справки.



Также можно обратиться к локальному серверу справки. Текст справки должен быть загружен на соответствующем языке (в нашем случае — немецком de-DE) в каталог C:\xampplite\htdocs\Joomla150\administrator\help\de-DE. На данный момент автоматическая инсталляция этого текста справки недоступна. Тексты на языках, отличных от английского, созданы соответствующими командами переводчиков.

Область Users Settings

В области Users Settings (Настройки пользователей) вы найдете глобальные настройки для пользовательских учетных записей (рис. 6.14).

- Allow User Registration (Разрешить регистрацию пользователей). Здесь вы можете выбрать, хотите ли вы разрешить пользователям выполнять регистрацию самостоятельно. Если вы управляете сайтом компании, то сами должны заводить регистрационные записи для коллег, не разрешая им делать это самостоятельно.

Рис. 6.14. Область Users Settings

Если же речь идет об общедоступном сайте сообщества, то здесь желательно чтобы пользователи могли регистрироваться сами.

- New User Registration Type (Тип регистрации для новых пользователей). Здесь можно определить, к какой группе пользователей должны принадлежать вновь зарегистрированные пользователи.
- New User Account Activation (Активизация учетных записей новых пользователей). Для того чтобы защитить себя от автоматических программ, которые могут создавать, скажем, 20 000 пользователей на вашем сайте, можно потребовать отдельной активизации. После регистрации пользователь получает автоматически сформированное письмо электронной почты по адресу, введенному им при регистрации. Это письмо содержит ссылку, активизирующую учетную запись. После активизации новый пользователь сможет нормально заходить на сайт.

- Front-end User Parameters (Параметры интерфейса пользователя). Этим параметром вы определяете, может ли пользователь устанавливать свой собственный язык и параметры часового пояса (рис. 6.15).

Рис. 6.15. Параметры интерфейса пользователя

Область Media Settings

В области Media Settings (Настройки медиа) конфигурируются параметры для медиа (рис. 6.16). Здесь можно указывать расширения файлов, допустимые для загрузки, и пути, по которым эти файлы могут быть сохранены.

Рис. 6.16. Область Media Settings

Помимо расширений файлов вы можете также определить типы MIME (<http://ru.wikipedia.org/wiki/MIME>). Это помогает контролировать безопасность, проверяя содержимое файлов. Если необходимые для их обработки модули не включены на веб-сервере Apache, вы можете гарантировать, что пользователи интерфейсной части могут лишь загружать изображения, установив переключатель Restrict Uploads (Ограничивать загрузки). Можно также задать максимальный размер, вплоть до предопределенного PHP максимального размера (см. следующую врезку).



Максимальный размер загружаемого файла в конечном итоге определяется конфигурацией PHP на сервере провайдера. В данном случае он установлен равным 16 Мбайт. Более крупные файлы должны загружаться через FTP или же потребуются изменить значение `upload` в конфигурационном файле `php.ini`.

Область *Debug Settings*

Программисты говорят об отладке, когда они ищут ошибки. Этот термин с годами эволюционировал. Ошибка (bug) — это "жучок" ("баг"), и 50 лет назад это на самом деле были жучки, которые заводились по соседству с теплыми вакуумными лампами компьютеров, вызывая периодически короткие замыкания там и тут. Отладка в те времена буквально означала уничтожение жучков. В наши дни этими "жучками" являются ошибки в программах, например, нерабочие запросы к базе данных. Область *Debug Settings* (Настройки отладки) показана на рис. 6.17.

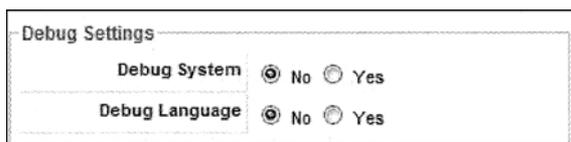


Рис. 6.17. Область *Debug Settings*

- **Debug System** (Система отладки). После активизации этой функции запросы базы данных отображаются под веб-сайтом. Например, как показано на рис. 6.18, для генерации одной страницы Joomla! необходимо выполнить 24 запроса к базе данных.

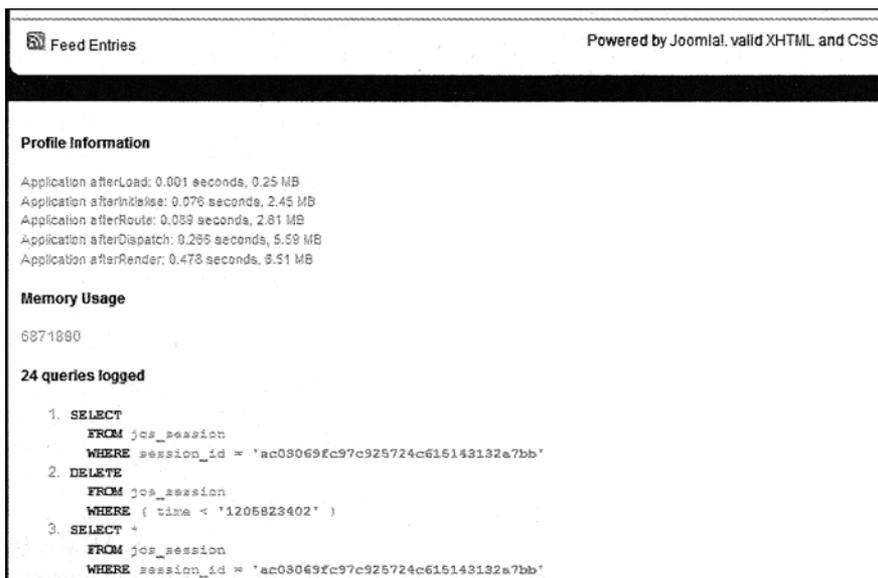


Рис. 6.18. Включенный режим отладки

- **Debug Language** (Язык отладки). Отладочные сообщения в этом случае ссылаются на языковые строки. Отображаются языковые файлы, к которым осуществляется обращение.

Перед и после всех идентификаторов устанавливаются толстые метки черного цвета, чтобы вы не забыли, что находитесь в режиме отладки (рис. 6.19).

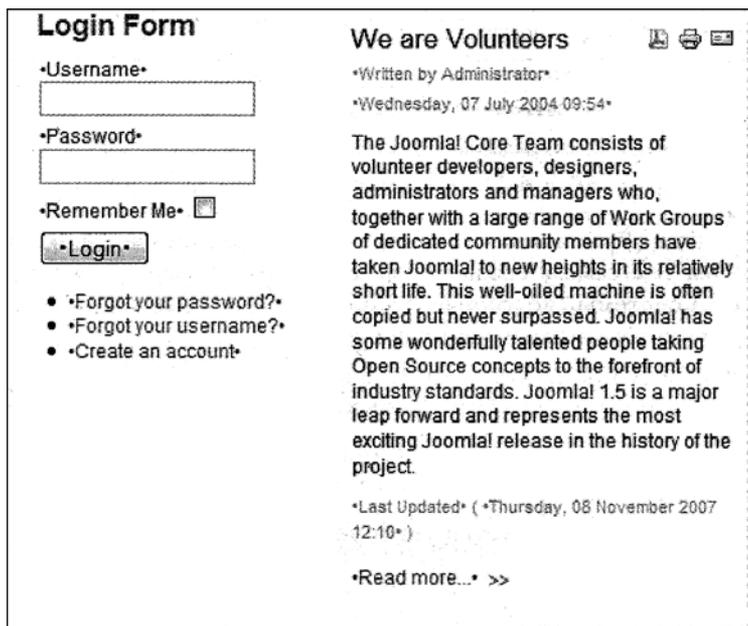


Рис. 6.19. Метки идентификаторов в режиме отладки

Область Cache Settings

Кэш (cache) — это средство временного хранения. Ваш браузер, например, имеет кэш изображений, который делает уже загруженные изображения быстро доступными при повторном к ним обращении. Joomla! использует аналогичный механизм на сервере для кэширования страниц, сгенерированных PHP. Эта возможность может значительно сократить время реакции для часто посещаемых страниц. Область Cache Settings (Настройки кэша) показана на рис. 6.20.

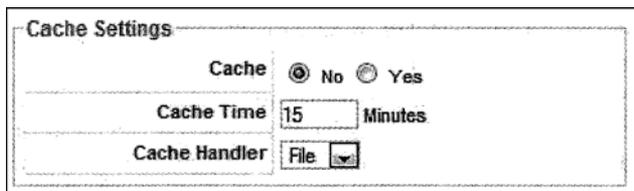


Рис. 6.20. Область Cache Settings

- Cache (Кэш). Здесь кэширование включается и отключается.
- Cache Time (Время кэширования). Здесь указывается время, которое должно пройти перед тем, как контент в кэше должен быть обновлен.
- Cache Handler (Обработчик кэша). Здесь определяется, должен кэш быть основан на файлах или на базе данных (на данный момент реализовано только файловое кэширование).

Область Session Settings

Область Session Settings (Настройки сеанса) показана на рис. 6.21.

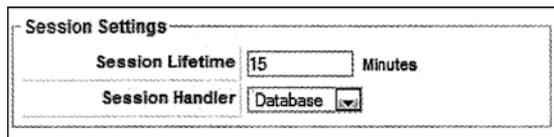


Рис. 6.21. Область Session Settings

- Session Lifetime (Время жизни сеанса). Всякий раз, когда вы входите как пользователь, вы тем самым создаете так называемый сеанс (session). Если вы не выходите из системы (не закрываете регистрацию), то сеанс будет прерван по истечении указанного здесь периода времени.
- Session Handler (Обработчик сеанса). Здесь вы определяете, должен сеанс основываться на файлах или на базе данных. На веб-сайтах с высокой загрузкой метод на основе базы данных существенно быстрее, чем метод на основе файлов.

Настройки Server

На этой вкладке конфигурируется техническая информация, касающаяся существующей серверной среды.

Область Server Settings

Область Server Settings (Настройки сервера) показана на рис. 6.22.

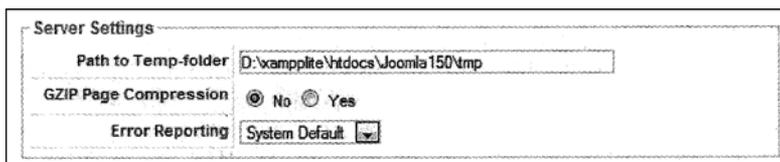


Рис. 6.22. Область Server Settings

- Path to Temp-folder (Путь к временной папке). Joomla! создает временные файлы, например, в период загрузки. Их нужно где-то хранить некоторое время, и здесь вы указываете, где именно.
- GZIP Page Compression (Сжатие страниц с помощью GZIP). Здесь вы можете включить сжатие страниц. Если и браузер, и веб-сервер поддерживают эту функцию, то страницы отправляются в формате ZIP и распаковываются в клиентском браузере. Это может существенно ускорить загрузку страниц, особенно при медленных Интернет-соединениях. Если сервер обрабатывает высокую нагрузку, вы должны использовать эту опцию, только если соответствующие библиотеки GZIP непосредственно установлены на уровне сервера. В зависимости от конфигурации сервера, может оказаться, что без этой опции сервер работает даже быстрее.
- Error Reporting (Отчеты об ошибках). С помощью этого переключателя активируется собственный механизм сообщений об ошибках PHP. Журналы ошибок не отправляются в файл для последующего изучения, а вместо этого сообщения отображаются непосредственно в браузере. В табл. 6.4 перечислены категории отчета об ошибках.

Таблица 6.4. Категории отчета об ошибках

Категория	Описание
System Default (Принятые в системе по умолчанию)	Принимаются установки из конфигурационного файла <code>php.ini</code> .
None (Нет)	Ошибки не протоколируются. Эта установка рекомендуется для рабочих сайтов, чтобы не предоставлять потенциальным хакерам информацию для атак.
Simple (Простые)	Протоколируются ошибки и предупреждения. Эта установка эквивалентна параметру <code>error_reporting (E_ERROR E_WARNING E_PARSE)</code> .
Maximum (Максимальные)	Протоколируются ошибки, предупреждающие сообщения и инструкции. Эта установка эквивалентна параметру <code>error_reporting (E_ALL)</code> .

Область *Locale Settings*

Область *Locale Settings* (Региональные настройки) имеет описанный ниже параметр.

- **Time Zone** (Часовой пояс). Эта установка используется для отображения часового пояса, в котором работает ваш сервер; например, может быть так, что сервер, на котором функционирует Joomla!, расположен в США, а предназначен для посетителей из Германии. Базой для вычисления локального времени является UTC (http://ru.wikipedia.org/wiki/Coordinated_Universal_Time).

Область *FTP Settings*

FTP (File Transfer Protocol – протокол передачи файлов) используется для обхода проблем, связанных с установками безопасности у провайдеров. Если запрещено загружать файлы на сервер через PHP, то вместо этого используется метод FTP. В области *FTP Settings* (Настройки FTP) необходимо указать параметры доступа к FTP-серверу вашего провайдера (рис. 6.23). Поле **FTP Root** (Корень FTP) служит специальной цели. В зависимости от провайдера, после входа через FTP вы попадаете в определенный каталог сервера. И здесь вы должны указать путь к вашей инсталляции Joomla! относительно точки FTP-соединения, например: `/htdocs/Joomla150`.

The image shows a screenshot of the 'FTP Settings' configuration form. At the top, it says 'FTP Settings'. Below that is the 'Enable FTP' section with two radio buttons: 'No' and 'Yes', with 'Yes' being selected. Underneath are five input fields: 'FTP Host' containing '127.0.0.1', 'FTP Port' containing '21', 'FTP Username', 'FTP Password', and 'FTP Root'.

Рис. 6.23. Область *FTP Settings*

Область *Database Settings*

В области *Database Settings* (Настройки базы данных) можно изменить параметры доступа к вашей базе данных (рис. 6.24). Имейте в виду, что опечатка здесь может навредить Joomla!. Это значит, что доступ к административной части сайта будет невозможен.

The screenshot shows the 'Database Settings' form with the following fields:

Database Type	mysql
Hostname	localhost
Username	root
Database	joomla150
Database Prefix	jos_

Рис. 6.24. Область Database Settings

Если подобное случится, вы можете вручную отредактировать конфигурационный файл `configuration.php` в текстовом редакторе. Все установки, которые вы выполняете через меню `Site⇒Global Configuration`, сохраняются в переменных в файле `configuration.php`:

```
<?php
class JConfig {
    var $dbtype = 'mysql';
    var $host = 'localhost';
    var $user = 'root1';
    var $password = '';
    var $db = 'Joomla150';
    var $dbprefix = 'jos_';
    // ...
    // другие переменные
    // ---
}
?>
```

Область Mail Settings

В области Mail Settings (Настройки почты) можно указать опции, касающиеся отправки электронной почты из Joomla! (рис. 6.25).

The screenshot shows the 'Mail Settings' form with the following fields:

Mailer	PHP Mail Function
Mail from	aksharaa@packtpub.com
From Name	Joomla! 1.5.0
Sendmail Path	/usr/sbin/sendmail
SMTP Authentication	<input checked="" type="radio"/> No <input type="radio"/> Yes
SMTP Username	
SMTP Password	
SMTP Host	localhost

Рис. 6.25. Область Mail Settings

- Mailer (Почтовая программа). Здесь вы решаете, что будет использоваться для отправки почты – встроенные почтовые функции PHP, программа Sendmail или другой агент электронной почты наподобие Yahoo!, Google или GMX.

- Mail from (Отправитель). Этот адрес электронной почты указывается в качестве адреса отправителя автоматически генерируемой почты Joomla!.
- From Name (Имя отправителя). Это имя также вводится в качестве имени отправителя автоматически генерируемой почты Joomla!.
- Sendmail Path (Путь к программе Sendmail). Если вы используете программу Sendmail, которая, скорее всего, установлена на серверах Linux, вместо функции почты PHP здесь следует ввести путь, указывающий на эту программу.
- SMTP Authentication (Аутентификация SMTP). Выберите Yes (Да), если хотите использовать внешний почтовый сервер (Yahoo!, GMX или ваш собственный).
- SMTP Username (Имя пользователя SMTP). Ваше имя пользователя у провайдера электронной почты.
- SMTP Password (Пароль пользователя SMTP). Ваш пароль у провайдера электронной почты.
- SMTP Host (Хост SMTP). Сервер SMTP провайдера электронной почты.

Пункт Logout

Щелчок на этом пункте меню приводит к выходу из административной части и автоматическому перенаправлению в интерфейсную часть.

Резюме

В этой главе было показано, как настраивается меню Site. Следующая глава посвящена меню Menus (Меню).

7

Меню Menus

В интерфейсной части присутствуют многочисленные меню. Часто они отображаются в отдельно стоящих рамках. Пункты меню обычно выровнены друг под другом, как показано на рис. 7.1.



Рис. 7.1. Главное меню

Меню также могут быть интегрированы в дизайн страницы горизонтально, так что на первый взгляд они даже не воспринимаются как взаимосвязанные пункты меню (рис. 7.2).



Рис. 7.2. Горизонтальное меню

Меню CSS, которые могут быть даже полупрозрачными, выглядят весьма впечатляюще. Примеры можно найти на joomla.com.

Эти меню и меню-ссылки динамически формируются Joomla! на основе содержимого базы данных в рабочей области Menus (Меню). Joomla! имеет шесть разных меню в данных примеров (Main Menu, Top Menu, Other Menu, User Menu, Example Pages и Key Concepts). Top Menu — горизонтальное меню, остальные меню вертикальные. Каждое меню связано с так называемым модулем, который администрируется диспетчером модулей.

Пункт Menus

Щелкнув на пункте меню Menus (Меню), можно получить представление обо всех доступных меню. К содержимому этих меню можно также обратиться через панель меню – Menus⇒Main Menu (Меню⇒Главное меню), Menus⇒Top Menu (Меню⇒Верхнее меню) и т.д. – или щелкнув на соответствующей ссылке меню в обзоре.

Диспетчер меню (Menu Manager), показанный на рис. 7.3, служит средством обзора и показывает множество пунктов меню Published (Опубликовано) и Unpublished (Не опубликовано), количество пунктов меню, находящихся в корзине (Trash), и соответствующие идентификаторы (ID) меню. В этом разделе можно, например, скопировать меню или создать новое.

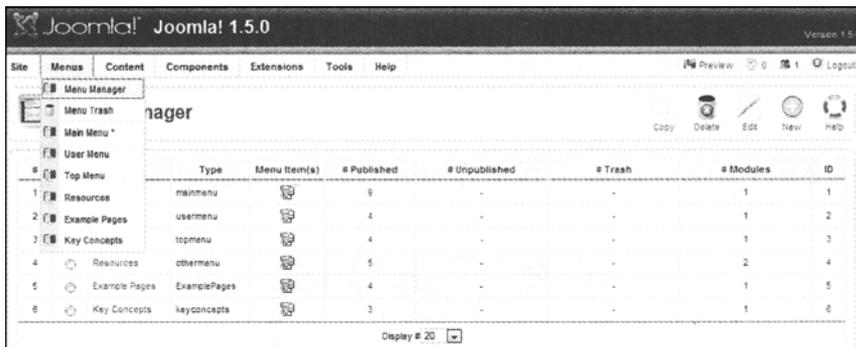


Рис. 7.3. Диспетчер меню

Настройка существующего меню

Поэкспериментируйте немного с меню, чтобы получить представление о предмете. Следующие шаги редактирования одинаковы для всех меню. Обратитесь к пункту меню Menus⇒Main Menu. Вы увидите список пунктов меню, которые появляются в mainmenu (рис. 7.4).

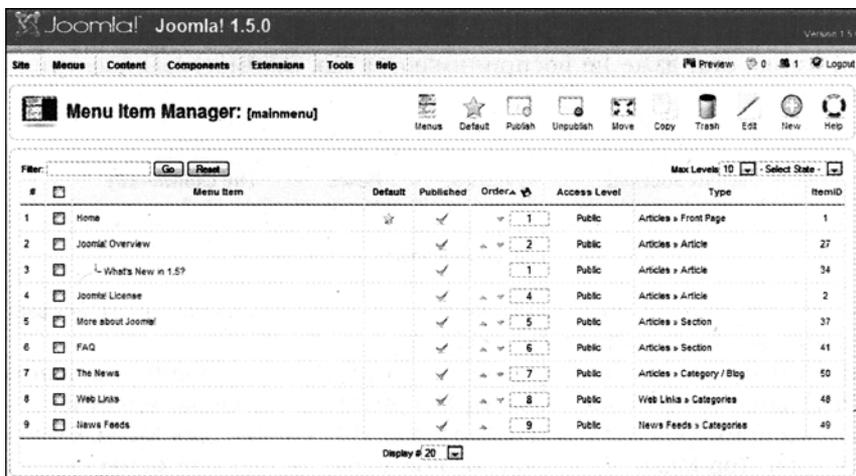


Рис. 7.4. Редактирование Main Menu

Многие функции могут быть запущены простым щелчком кнопкой мыши. Щелкая на флажке, вы можете включать или выключать ссылку меню. Порядок элементов можно изменить, щелкая на треугольниках или вводя номера в поле Order (Порядок). В случае использования метода с вводом номеров нужно будет щелкнуть на символе диска в заголовке, чтобы изменения вступили в силу. В столбце Access Level (Уровень доступа) щелчком кнопкой мыши можно устанавливать общий доступ всем пользователям к меню (Public), доступ только зарегистрированным пользователям (Registered) или только определенному кругу пользователей (Special). Пункт меню затем отображается или скрывается в зависимости от прав пользователя.

Пиктограмма Menu

Щелчок на этой пиктограмме приводит к переходу на обзорный экран меню.

Пиктограмма Default

Пункт меню, помеченный звездочкой как пункт по умолчанию, отображается на стартовой странице, когда кто-либо обращается по URL-адресу вашего веб-сайта. На данный момент это пункт Home (Домой), однако можно назначить пунктом по умолчанию любой другой элемент, который должен служить стартовой страницей. Для этого просто отметьте флажок и щелкните на пиктограмме Default (По умолчанию).

Пиктограммы Publish и Unpublish

Состояние элемента контента может быть опубликованным (активизированным) или не опубликованным (деактивизированным). Это состояние можно переключать индивидуально, щелкая на галочке зеленого цвета и/или на кресте красного цвета, либо отмечая флажок с последующим щелчком на пиктограмме Publish (Опубликовать) или Unpublish (Отменить публикацию). При последнем методе допускается переключение состояния сразу нескольких пунктов меню.

Пиктограмма Move

Щелчок на пиктограмме Move (Переместить) приводит к перемещению пунктов меню. Переместим для примера текст More about Joomla! в верхнее меню. Выберите соответствующий элемент меню или даже несколько элементов меню, и щелкните на пиктограмме Move. Это откроет форму, перечисляющую все доступные меню (рис. 7.5). Справа показаны элементы, которые будут перемещены.

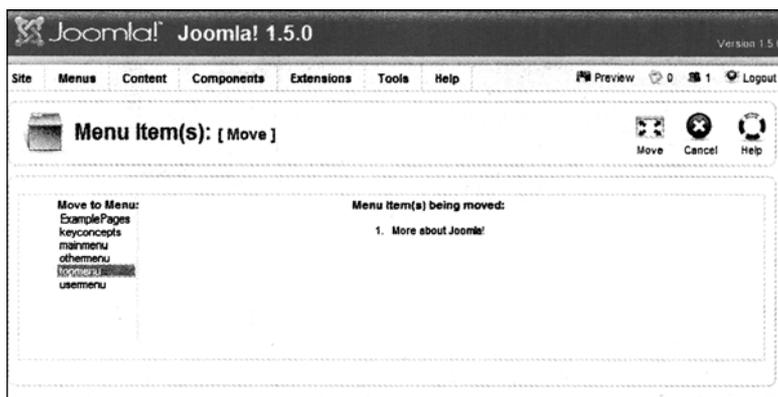


Рис. 7.5. Перемещение пунктов меню

Выберите меню, в которое хотите переместить помеченные пункты меню. Для примера перемещается пункт More about Joomla! из Main Menu в Top Menu. Увидеть результат можно в интерфейсной части сайта (рис. 7.6).



Рис. 7.6. Меню Top Menu после перемещения пункта

Пиктограмма Copy

Пункты меню можно также копировать. Для этого выберите один из пунктов меню и щелкните на пиктограмме Copy (Копировать). Точно так же, как и при перемещении, откроется форма с доступными меню. Выберите меню, куда нужно копировать отмеченные пункты.

Пиктограмма Trash

Для того чтобы защитить от нечаянного удаления элементов, при редактировании они не сразу удаляются окончательно, а только помещаются в корзину.

Чтобы перетащить элемент в корзину, выберите один или более элементов и щелкните на пиктограмме Trash (Корзина). После этого отмеченные пункты меню будут сброшены в корзину. Отобразить содержимое корзины можно, щелкнув на Menu^Menu Trash (Меню^Корзина меню).

Пиктограмма Edit

Щелкнув на пиктограмме Edit (Редактирование), можно модифицировать существующее меню, например, Web Links (Веб-ссылки). После щелчка на имени Web Links появляется форма редактирования элементов меню, показанная на рис. 7.7.

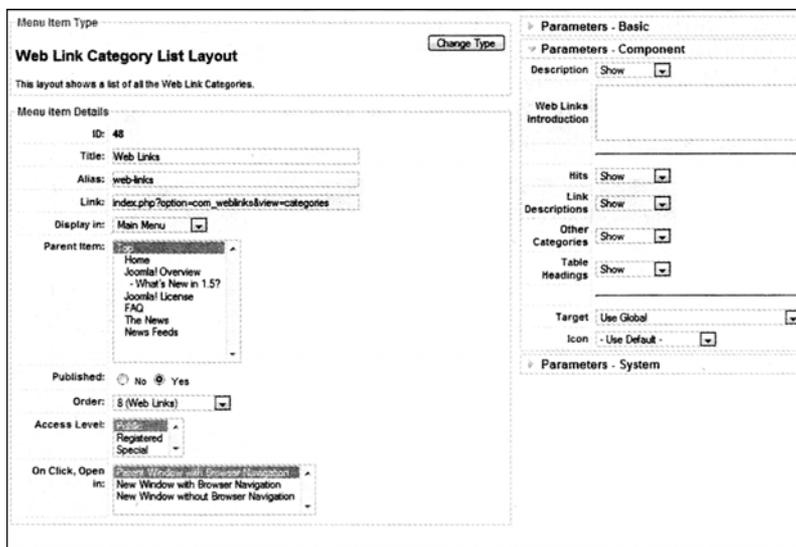


Рис. 7.7. Форма редактирования пункта меню

Форма редактирования разделена на три области:

- Menu Item Type (Тип пункта меню)
- Menu Item Details (Информация о пункте меню)
- Parameters (Параметры)

Область Menu Item Type

Каждый пункт меню относится к определенному типу. Подробнее мы рассмотрим то при создании новых меню. Например, пункт меню может ссылаться на установленный компонент Joomla!, элемент контента, ссылку на внешний веб-сайт или [ногие другие вещи. В этой области представлен тип ссылки; в данном случае это ссылка на компонент weblinks Joomla!. Также здесь присутствует кнопка с меткой Change Type (Изменить тип). Если вы щелкнете на этой кнопке, то увидите экран, показанный на рис. 7.8.

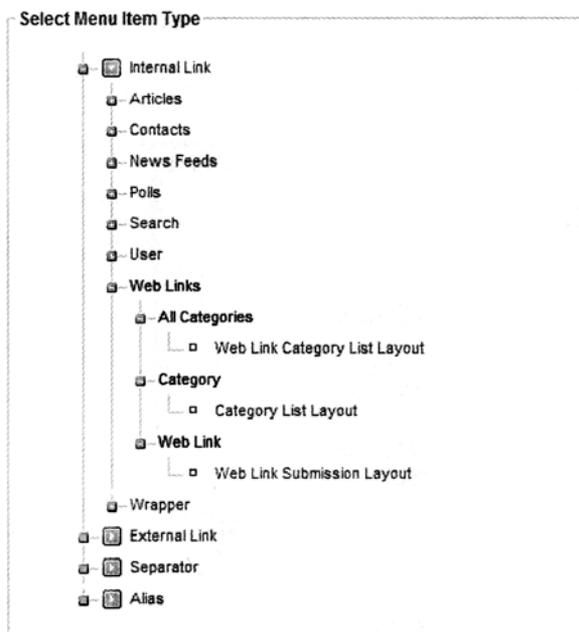


Рис. 7.8. Формы выбора типа меню

Этот диспетчер – новый в Joomla! версии 1.5, и он действительно удобен. В версии 1.0.x не было возможности изменять тип пункта меню. Там приходилось удалять старый пункт и создавать новый. Теперь можно изменять отображение до одиночной категории или пункта меню – рекомендуемой ссылки, с помощью чего вы приглашаете пользователя пройти по этой ссылке.

Закройте форму; мы вернемся к ней при создании нового меню.

Область Menu Item Details

Эта часть формы содержит перечисленные ниже опции.

- ID (Идентификатор). Все элементы при администрировании требуют идентификационные номера (ID); то же касается и пунктов меню. В данном случае пункт меню имеет номер ID, равный 48. Joomla! присваивает этот номер для внутренних потребностей администрирования в момент создания элемента. Этот номер изменяться не может.
- Title (Заголовок). Это название меню, которое будет отображаться на вашем веб-сайте.
- Alias (Псевдоним). Это название SEF URL после доменного имени. Когда этот параметр включен, URL меню будет выглядеть так:
<http://localhost/joomla150/web-links>
- Link (Ссылка). Это запрос компонента, другими словами – часть URL, следующая после доменного имени, по которому вызывается веб-сайт. В данном случае это `index.php?option=com_weblinks&view=categories` (рис. 7.9).

The screenshot shows the 'Menu Item Details' form in Joomla! with the following fields and values:

- ID:** 48
- Title:** Web Links
- Alias:** web-links
- Link:** index.php?option=com_weblinks&view=categories
- Display in:** Main Menu
- Parent Item:** Top (with a list of other items: Home, Joomla! Overview - What's New in 1.5?, Joomla! License, FAQ, The News, News Feeds)
- Published:** Yes
- Order:** 8 (Web Links)
- Access Level:** Public
- On Click, Open in:** Parent Window with Browser Navigation

Рис. 7.9. Область Menu Item Details

- Display in (Отображать в). Здесь можно изменить место, в котором отображается элемент; другими словами, его можно переместить в другое меню. В поле предоставляется список доступных меню.
- Parent Item (Родительский элемент). Разумеется, меню также может содержать вложенные элементы с древовидной структурой. Тор (Вверху) означает, что элемент находится на высшем уровне иерархии. Остальные элементы представляют существующие пункты меню. Если, например, вы классифицируете и сохраняете Web Links под The News, то отображение в списке элементов и отображение на вашем веб-сайте изменятся. На рис. 7.10 можно видеть изменения. Пункт меню Web Links должен быть перемещен в The News на веб-сайте. Поэтому вы сначала щелкаете на The News, чтобы увидеть элемент Web Links (рис. 7.11). Веб-сайт может быть легко и эффективно структурирован подобно дереву.

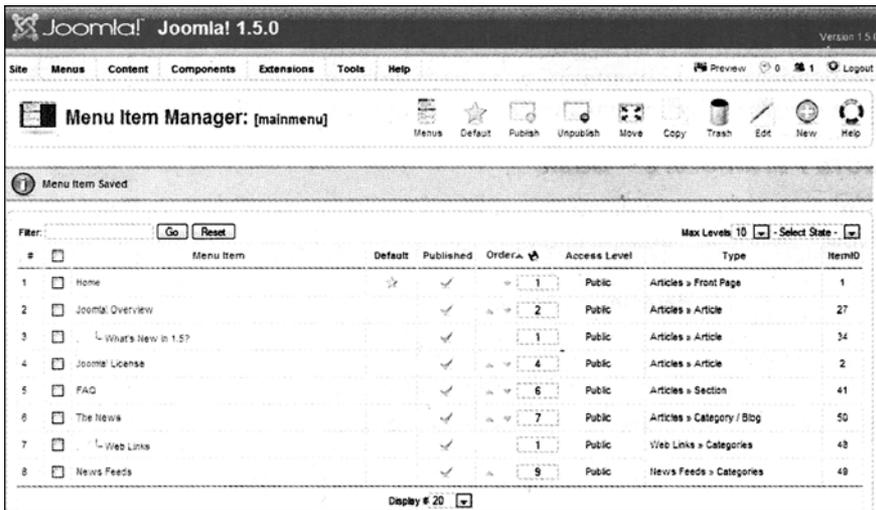


Рис. 7.10. Диспетчер пунктов меню после перемещения пункта



Рис. 7.11. Меню после перемещения пункта

- **Published (Опубликовано).** Выбрав переключатель Yes (Да), можно опубликовать (активизировать) пункт меню.
- **Order (Порядок).** В предлагаемом списке можно выбрать ссылку, после которой нужно установить редактируемую ссылку.
- **Access Level (Уровень доступа).** Можно ограничить перечень пользователей, которые будут видеть данный пункт.
- **On Click, Open in (При щелчке открыть в).** Очень удобная опция, влияющая на поведение ссылки. После щелчка страница открывается либо в существующем окне, либо в новом окне браузера. Вы можете также определить, должно ли новое окно отображаться с элементами навигации браузера или без них.

Параметры

Возможные параметры пункта меню зависят от его типа. Простая ссылка, конечно, имеет меньше параметров, чем конфигурируемый список или, например, ссылка на титульную страницу.

В этом случае мы имеем ссылку на категории (эти компоненты будут подробно описаны в главе 9). Количество и тип параметров зависит от типа пункта меню. Вы можете разворачивать и сворачивать поля параметров, щелкая на заголовке. Если поля параметров открыты, то стрелка рядом с заголовком указывает вниз.

Область *Parameters - Basic*

Базовые параметры одинаковы для всех ссылок меню. Область *Parameters – Basic* (Параметры – базовые) показана на рис. 7.12.

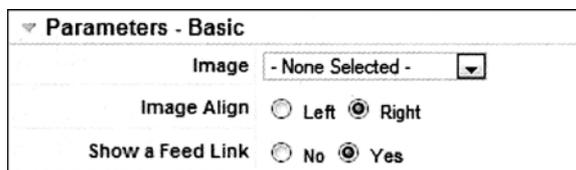


Рис. 7.12. Область *Parameters – Basic*

- **Image** (Изображение). Здесь можно специфицировать файл изображения, который должен находиться в корневом каталоге диспетчера медиа (/images/stories/). В зависимости от шаблона, картинка отображается слева, рядом с пунктом меню.
- **Image Align** (Выравнивание изображения). Здесь можно указать, где должна располагаться картинка – слева или справа.
- **Show a Feed Link** (Показать ссылку на ленту). Для каждого списка, отображаемого в Joomla! 1.5, есть возможность создать наполнение RSS. Это может быть желательно или нежелательно, в зависимости от содержимого списка. В данном случае при отображении списка ссылки RSS-наполнения, содержащие пункты списка, включены в браузер.

Область *Parameters – Component*

Область *Parameters – Component* (Параметры – компонент) имеет дело с компонентом, адресованным в ссылке, в данном случае – компонентом **Web Links**. Он состоит из нескольких уровней. При первом щелчке на нем отображаются доступные категории и текст по умолчанию (рис. 7.13).



Рис. 7.13. Вид формы *Weblinks*

Если затем щелкнуть на **Joomla! Specific Links** (Ссылки, специфические для Joomla!), отобразится таблица с соответствующими ссылками, которая показана на рис. 7.14. Эту конструкцию можно изменить в области *Parameters – Component* (рис. 7.15)

Joomla! Specific Links

A selection of links that are all related to the Joomla! Project.



Display #

#	Web Link	Hits
1	 Joomla! Home of Joomla!	3
2	 OpenSourceMatters Home of OSM	11
3	 php.net The language that Joomla! is developed in	6
4	 Joomla! - Forums Joomla! Forums	4
5	 MySQL The database that Joomla! uses	1
6	 Ohloh Tracking of Joomla! Objective reports from Ohloh about Joomla's development activity. Joomla! has some star developers with serious kudos.	1

Рис. 7.14. Таблица Joomla! Specific Links

Parameters - Component

Description

Web Links Introduction

Hits

Link Descriptions

Other Categories

Table Headings

Target

Icon

Рис. 7.15. Область Parameters – Component

- Description (Описание). С помощью этого можно отобразить или скрыть заголовок над категориями.
- Web Links Introduction (Введение для пункта Web Links). Здесь можно предусмотреть индивидуальный заголовок, как показано на рис. 7.16.

Main Menu

- Home
- Joomla! Overview
- Joomla! License

Weblinks

We are regularly out on the Web. When we find a great site we list it.

- Joomla! Specific Links (6)

Рис. 7.16. Установка индивидуального заголовка

- Hits (Попадания). Попадания (хиты) посетителей на соответствующие ссылки вашей страницы отслеживаются в столбце Hits. Здесь этот столбец можно отобразить или скрыть.
- Link Descriptions (Описания ссылок). Здесь можно показывать или скрывать описание, находящееся под ссылкой или списком ссылок.
- Other Categories (Другие категории). Когда вы отображаете таблицу ссылок, то видите либо текст по умолчанию, либо текст, который настроили в параметрах. Ниже находится список используемых категорий и разделов (если есть другие категории). Этот список может быть отображен или скрыт.
- Table Heading (Заголовок таблицы). С помощью этой опции можно отображать или скрывать заголовки над ссылками.

Область Parameters – System

В области Parameters – System (Параметры – система) расположены дополнительные параметры, которые влияют на внешний вид ссылки (рис. 7.17).

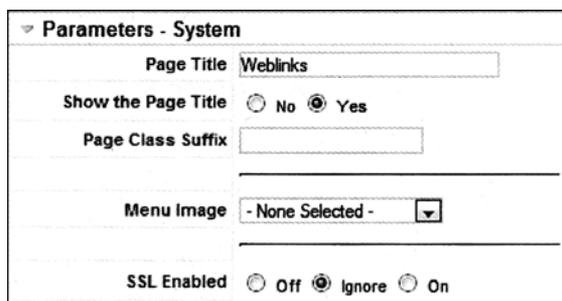


Рис. 7.17. Область Parameters – System

- Page Title (Заголовок страницы). Здесь можно вводить текст заголовка страницы. Он будет отображен в верхней панели веб-сайта и в браузере.
- Show the Page Title (Показать заголовок страницы). Здесь можно отобразить или скрыть заголовок страницы.
- Page Class Suffix (Суффикс класса страницы). Здесь можно ввести текстовую строку, например, `my_menuitems`. Она затем будет добавлена к имени класса в коде HTML. На дизайн влияет соответствующий класс в файле CSS, который можно настроить.
- SSL Enabled (Включенный SSL). Здесь можно выбрать, поддерживает ли ссылка безопасный протокол HTTPS. Этот выбор требует функционирующей на сервере среды SSL. Шифрование SSL работает без каких-либо проблем в локальном окружении XAMPP Lite.

Пиктограмма New

Эта пиктограмма используется для создания новых меню. Речь об этом пойдет в разделе "Создание нового меню".

Меню Trash

В корзине (trash) собираются удаленные пункты меню (рис. 7.18).

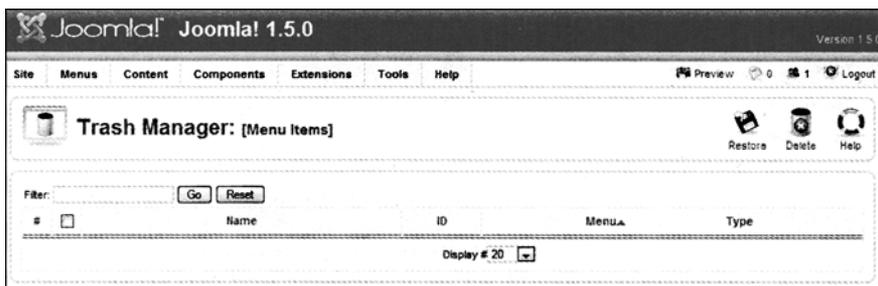


Рис. 7.18. Диспетчер корзины

Если выбрать удаленные элементы и щелкнуть на пиктограмме Restore (Восстановить), можно извлечь их из корзины (рис. 7.19).

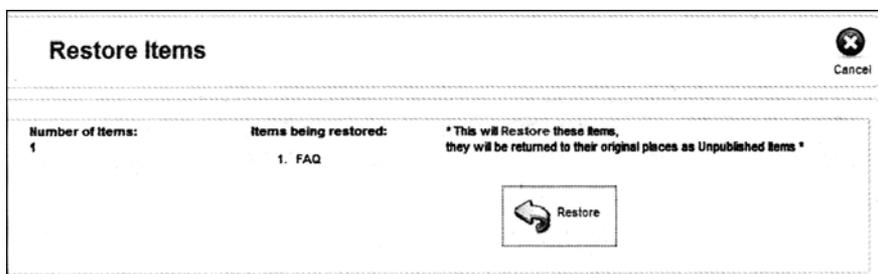


Рис. 7.19. Восстановление элементов

Если же щелкнуть на пиктограмме Delete (Удалить), они будут удалены безвозвратно.

Создание нового меню

В этом разделе мы создадим новое меню по имени Joomla! 1.5 Book со ссылкой на страницу <http://www.joomla.org/>, которая будет отображаться в новом окне. Мы хотим расположить его слева, под главным меню.

Перейдите к Menus⇒Menu Manager⇒New (Меню⇒Диспетчер меню⇒Создать) и введите внутреннее имя меню в поле Unique Name (Уникальное имя). Необходимо выбирать осмысленное имя без пробелов. В поле Title (Заголовок) ведите название, которое должно будет отображаться впоследствии на вашем веб-сайте (рис. 7.20). Меню состоит из внутреннего, действительного меню, куда можно добавлять ссылки меню, и соответствующего модуля, который может быть позиционирован позднее.

- Unique Name (Уникальное имя). Это имя, используемое Joomla! в коде. Пробелы в нем не допускаются. В данном случае выбрано имя `joomla_book`. Это имя не будет отображаться на веб-сайте; его предназначение — исключительно для связи между меню и модулем.
- Title (Заголовок). Название меню. Здесь — Joomla! 1.5 Book (с пробелами).

Menu: [New]

Save Cancel Help

Unique Name:

Title:

Description:

Module Title:

Рис. 7.20. Форма создания нового меню

- Description (Описание). Описание меню. Это описание отображается только внутренне, в административной части, например, при отображении списков.
- Module Title (Заголовок модуля). Имя модуля; здесь – также Joomla! 1.5 Book.

После щелчка на пиктограмме Save (Сохранить) Joomla! создает новый модуль со специфицированными параметрами. Происходит перенаправление к обзору меню, в котором появилось новое меню, пока не содержащее пунктов (рис. 7.21).

Menu Manager

Copy Delete Edit New Help

New Menu created [joomla_book]

#	Title	Type	Menu Item(s)	# Published	# Unpublished	# Trash	# Modules	ID
1	Main Menu	mainmenu		7	-	1	1	1
2	User Menu	usermenu		4	-	-	1	2
3	Top Menu	topmenu		5	-	-	1	3
4	Resources	othermenu		5	-	-	2	4
5	Example Pages	ExamplePages		4	-	-	1	5
6	Key Concepts	keyconcepts		1	-	-	1	6
7	Joomla! 1.5 Book	joomla_book		-	-	-	1	7

Display # 20

Рис. 7.21. Диспетчер меню после добавления нового меню

Если щелкнуть на пиктограмме с изображением карандаша справа, рядом со ссылкой Joomla! 1.5 Book, отобразится экран обзора содержимого меню Joomla! 1.5 Book (рис. 7.22).

Menu Item Manager: [joomla_book]

Menu Default Publish Unpublish Move Copy Trash Edit New Help

Filter: Go Reset

Max Levels: 10 - Select State:

#	Menu Item	Default	Published	Order	Access Level	Type	ItemID
Display # 20							

Рис. 7.22. Обзор нового меню

Поскольку здесь еще нет никакого содержимого, щелкните на пиктограмме New (Создать). После этого можно будет выбрать тип пункта меню из разных областей появившегося экрана выбора (рис. 7.23). Поскольку ваши познания в Joomla! еще находятся в зачаточном состоянии, давайте просто вставим простую внешнюю ссылку на внешний веб-сайт.

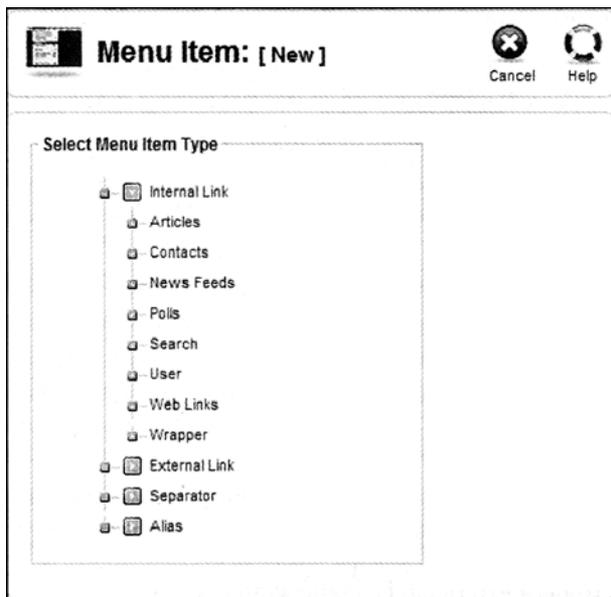


Рис. 7.23. Выбор типа меню

Чтобы сделать это, щелкните на External Link (Внешняя ссылка). После этого вы сможете определить детали и параметры ссылки (рис. 7.24).

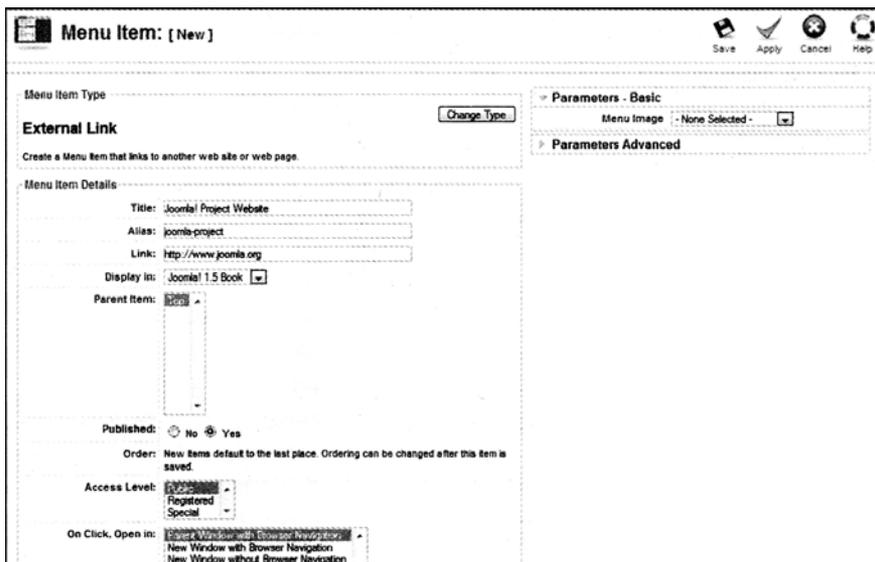


Рис. 7.24. Параметры пункта меню типа внешней ссылки

- Title (Заголовок). Название ссылки, которое появляется в меню (Joomla! Project Website).
- Parent Item (Родительский элемент). Поскольку это – первый элемент, родительских элементов у него нет.

- Alias (Псевдоним). Имя короткого URL (joomla-project).
- Link (Ссылка). Ссылка на страницу (http://www.joomla.org).
- Display in (Отображать в). Конечно, в нашем новом меню Joomla! 1.5 Book.
- Published (Опубликовано). Выберите Yes (Да) для публикации меню.
- Order (Порядок). Поскольку это первый элемент, порядок еще не указан. Новые элементы по умолчанию помещаются в конец. Порядок может быть изменен после первого сохранения.
- Access Level (Уровень доступа). Должен ли данный пункт меню быть видимым группам пользователей Public (посетители), Registered или Special?
- On Click, Open in (При щелчке открыть в). Что должно произойти, когда кто-то щелкнет на ссылке? Должна ли целевая страница отображаться в том же окне браузера, в новом окне браузера с кнопками навигации или в новом окне браузера без кнопок навигации?

После щелчка на пиктограмме Apply (Применить) введенная информация сохраняется. Если щелкнуть на пиктограмме Save (Сохранить), то информация сохранится, а диалог закроется.

Таким образом, только что было создано меню, а в него добавлена ссылка. Но прежде чем оно отобразится в интерфейсной части, потребуется опубликовать вновь созданный модуль. Выберите Extensions⇒Module Manager (Расширения⇒Диспетчер модулей) и затем установите галочку зеленого цвета в колонке Enabled (Включено), как показано на рис. 7.25. Новое меню можно разместить под главным меню, щелкая на треугольниках в колонке Order (Порядок).

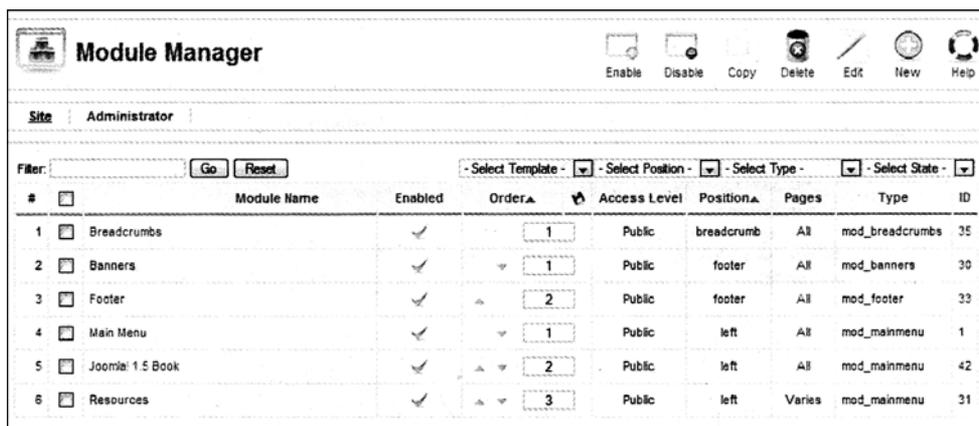


Рис. 7.25. Управление новым модулем

Теперь при обращении к веб-сайту должно появиться вновь созданное меню Joomla! 1.5 Book (рис. 7.26).

К сожалению, новое меню выглядит иначе, чем меню над ним. Отсутствует синяя рамка. Этот атрибут определен в CSS-файле шаблона и, разумеется, может быть модифицирован. Для этого модуль должен вызвать определенный класс CSS. В нашем случае класс, с которым отображается созданное меню, называется module.

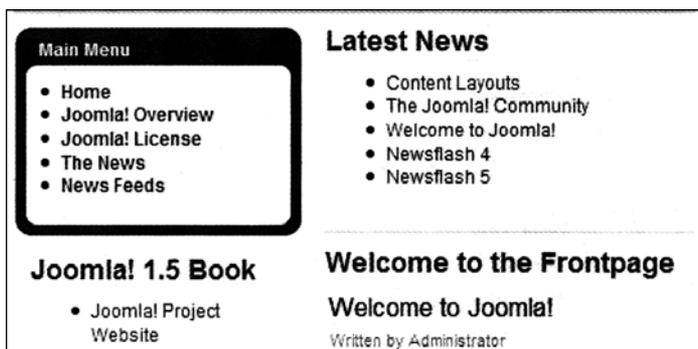


Рис. 7.26. Новое меню Joomla! 1.5 Book

Вы можете убедиться в этом, взглянув на исходный HTML-код веб-сайта:

```
<div class="module"><div><div><div>
<h3>Joomla! 1.5 Book</h3>
<ul class="menu">
<li class="item54">
<a href="http://www.joomla.org" target="_blank">
<span>Joomla! Project Website</span>
</a>
</li></ul></div></div></div></div>
```

Главное меню, с другой стороны, вызывает класс `module_menu`. Вот исходный код этого меню:

```
<div class = "module_menu"><div><div><div>
<h3>main menu</h3>
<ul class="menu">
<li id="current" class="active item1">
<a href="http://localhost/joomla150/">
<span>start page</span>
</a>
</li>
... дополнительные ссылки ...
</ul>
</div></div></div></div> </ul>
</div>
```

При редактировании модуля можно сделать так, что новое меню также будет вызывать этот класс. Выберите `Extensions⇒Module Manager⇒Joomla! 1.5 Book` (Расширения⇒Диспетчер модулей⇒Joomla! 1.5 Book). Затем просто введите в поле `Module Class Suffix` Суффикс класса модуля) текст `_menu`, как показано на рис. 7.27.

После этой модификации меню будет отображаться так, как планировалось (рис. 7.28). Если теперь щелкнуть на ссылке `Joomla! 1.5 Book`, должно открыться новое окно браузера с кнопками навигации, в котором отобразится веб-сайт проекта.

Резюме

В этой главе было показано, как настраивать меню `Menus`. Следующая глава посвящена меню `Content` (Контент).

▶ **Module Parameters**

▼ **Advanced Parameters**

Show Whitespace ▼

Caching ▼

Menu Tag ID

Menu Class Suffix

Module Class Suffix

Maximum Menu Depth

▶ **Legacy Parameters**

Рис. 7.27. Дополнительные параметры модуля

Main Menu

- Home
- Joomla! Overview
- Joomla! License
- The News
- News Feeds

Latest News

- Content Layouts
- The Joomla! Community
- Welcome to Joomla!
- Newsflash 4
- Newsflash 5

Joomla! 1.5 Book

- Joomla! Project Website

Welcome to the Frontpage

Welcome to Joomla!

Written by Administrator

Рис. 7.28. Внешний вид меню после модификации параметров

8

Меню Content

Меню Content (Контент) включает все, что связано с контентом веб-сайта. Контент в терминах текста, форматированного с помощью HTML, в Joomla! называется *статьей* (article). Статьи Joomla! организованы согласно следующей структуре:

```
Контент +- Раздел1 +- Категория1 +- Контент1
      |           |           +- Контент2
      |           |           +- Контент3
      |           +- Категория2 +- Контент1
      |           |           +- Контент2
+- Раздел2 +- Категория1 +- Контент1
      |           |           +- Контент2
      |           |           +- Контент3
      |           +- Категория2 +- Контент1
      |           |           +- Контент2
+- Статический контент +- Контент1
                        +- Контент2
                        +- Контент3
```

Эту структуру можно сравнить с деревом файлов на жестком диске. Можно создавать произвольное количество категорий и разделов. Такую структуру можно использовать со статьями, но делать этого не стоит. В этом случае структура перенесется и в архив.

Данная структура обладает преимуществами и недостатками. Самое большое преимущество в том, что поскольку есть структура, администратор должен подчиняться определенной иерархии. Это улучшает обзор и сохраняет веб-сайт в управляемом состоянии.

Недостатком является то, что двухступенчатая структура не может быть легко изменена, и потому некоторые администраторы чувствуют себя ограниченными в индивидуализации своего веб-сайта. Но это по большей части субъективное ощущение. Используя разумную комбинацию меню и контента, можно соорудить любую навигационную структуру, которую только можно себе вообразить.

Самую большую сложность представляет следующий вопрос: *что планируется представить, и в каком контексте?*

Меню Content предоставляет несколько рабочих пространств для работы с контентом и структурой, как показано на рис. 8.1.



Рис. 8.1. Меню Content

Пункт Article Manager

Пункт меню Article Manager (Диспетчер статей) перемещает на экран обзора существующих статей. Это центральный пункт управления организацией статей на веб-сайте.

Отображаемый контент можно фильтровать по разделу (Section), категории (Category), автору (Author) и состоянию (State) в списках опций внутри информационной области. Имеется также поле поиска, которое можно использовать для выполнения поиска в статьях.

В нижней области находится также панель навигации, с помощью которой можно перелистывать контент. Один из списков опций позволяет определить, сколько элементов отображать в списке. Установки, которые вы ввели ранее, являются стандартными.

Список отсортирован по разделу (Section), категории (Category), автору (Author) и заголовку (Title), как показано на рис. 8.2. Порядок сортировки можно изменить, щелкнув на соответствующей колонке.

#	Title	Published	Front Page	Order	Access Level	Section	Category	Author	Date	Hits	ID
1	Example Pages and Menu Links	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	Public			Administrator	12.10.06	42	43
2	What's New in 1.5?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	Public	About Joomla!	The CMS	Administrator	11.10.06	68	22
3	Joomla! Overview	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	Public	About Joomla!	The CMS	Administrator	09.10.06	148	19
4	Extensions	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	Public	About Joomla!	The CMS	Administrator	11.10.06	100	26
5	Joomla! Features	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4	Public	About Joomla!	The CMS	Administrator	08.10.06	59	18
6	Content Layouts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5	Public	About Joomla!	The CMS	Administrator	12.10.06	69	24
7	Joomla! Facts	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	Public	About Joomla!	The Community	Administrator	09.10.06	50	21
8	The Joomla! Community	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	Public	About Joomla!	The Community	Administrator	12.10.06	50	27
9	Support and Documentation	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	Public	About Joomla!	The Project	Administrator	09.10.06	5	20
10	Joomla! License Guidelines	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	Public	About Joomla!	The Project	Administrator	20.08.04	69	5
11	Platforms and Open Standards	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	Public	About Joomla!	The Project	Administrator	11.10.06	11	23
12	Where did the Installers go?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1	Public	FAQs	Current Users	Administrator	10.10.06	3	36
13	What happened to the locale setting?	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2	Public	FAQs	Current Users	Administrator	06.10.06	10	13

Рис. 8.2. Диспетчер статей

Описание списка статей

Перед каждым заголовком стоит флажок, который можно использовать для выбора элементов с целью редактирования. Если отметить флажок выше заголовков статей, будут выбраны все элементы списка.

Содержимое колонки Title (Заголовок) – это ссылки, переводящие статьи в режим редактирования, окно которого показано на рис. 8.3.

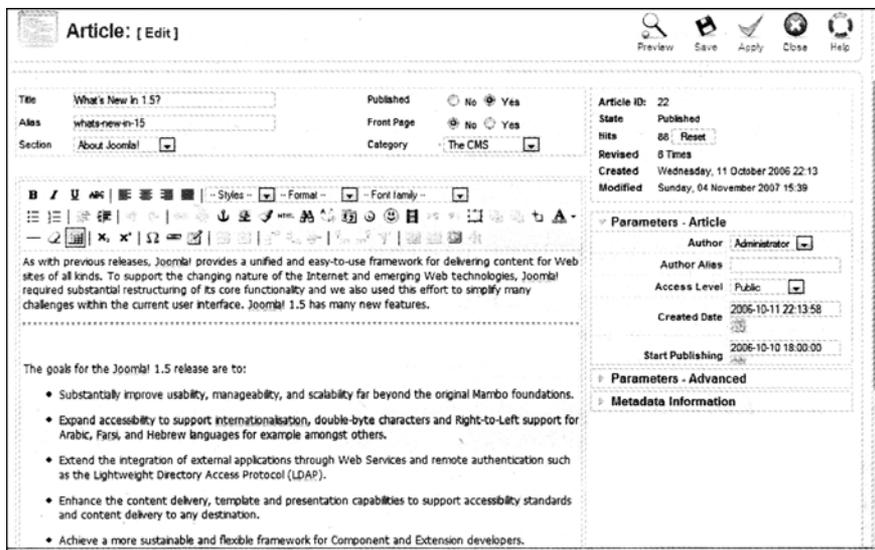


Рис. 8.3. Редактирование статьи

Колонка Published (Опубликовано) отображает, статья опубликована (галочка зеленого цвета) или нет (крестик красного цвета). Помимо этих двух символов существует также вероятность истечения срока актуальности статьи и возможность, что статья уже помещена в архив. В нижнем колонтитуле страницы представлены обозначения присутствующих здесь пиктограмм. Символы для состояния ожидания и истечения срока актуальности также объясняются здесь.

Колонка Front Page (Титульная страница) отображает, опубликована статья на стартовой странице (галочка зеленого цвета) или нет (крестик красного цвета).

В колонке Order (Порядок) можно перемещать элементы, щелкая на стрелках для каждой статьи. Элементы можно также сортировать, вводя номер и затем щелкая на пиктограмме с диском возле слова Order.

В колонке Access Level (Уровень доступа) могут отображаться ссылки Public (Общедоступно) зеленого цвета. Уровень доступа можно переключать, выбирая из трех групп – Public, Registered и Special, щелкая на одноименных ссылках.

В колонке Section (Раздел) представлен раздел, в который назначена статья. Если вы щелкнете на ссылке, то попадете на страницу Section: [Edit] (Раздел: [Редактирование]). Некатегоризированные статьи соответствуют статическому контенту в Joomla! 1.0.

В колонке Category (Категория) указана структура разделов, которым данный элемент назначен. Щелчок на такой ссылке перемещает на страницу Category:[Edit] (Категория: [Редактирование]).

Колонка Author (Автор) содержит авторов статей. Щелчок на такой ссылке переносит на страницу Author: [Edit] (Автор: [Редактирование]).

В колонке Date (Дата) указаны даты создания статей.

Колонка ID (Идентификатор) хранит номера наборов данных внутри таблицы базы данных. ID появляется в URL-адресе статьи.



В Joomla! 1.0.x было два текстовых поля. Одно предназначено для краткого рекламного описания, а другое — для остального текста. В Joomla! 1.5 это разделение было исключено, но функциональность осталась. Теперь имеется возможность включить соответствующие разделители (continue reading (продолжить чтение), page break (разрыв страницы)) щелчком кнопки мыши в единственном окне. Разделение на статический и динамический контент упразднено. Статические статьи теперь просто считаются некатегоризированными, и могут быть опубликованы на титульной странице.

Пиктограммы редактирования статьи

Для редактирования одной или более статей за один раз в вашем распоряжении имеется несколько функций, показанных на рис. 8.4.

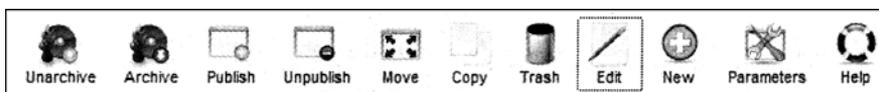


Рис. 8.4. Панель инструментов редактирования статей

Пиктограммы Archive и Unarchive

Идея, положенная в основу архива, состоит в том, что старый контент не удаляется, а сохраняется для более позднего использования. Статьи архивируются в диспетчере статей щелчком на пиктограмме Archive (Архивировать), а изымаются из архива щелчком на пиктограмме Unarchive (Изъять из архива). Архивированная статья отображается в обзоре серым цветом, как показано на рис. 8.5.

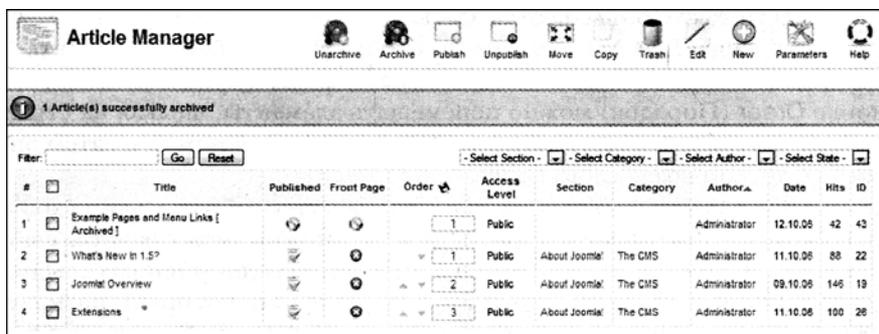


Рис. 8.5. Отображение архивированной статьи

В случае выбора Archived (Архивированные) в фильтре State (Состояние) отображение ограничивается только архивированными статьями. Архивированные статьи могут быть показаны в разных позициях на веб-сайте.

Пиктограммы *Publish* и *Unpublish*

С помощью пиктограмм *Publish* (Публиковать) и *Unpublish* (Отменить публикацию) статьи утверждаются для публикации или же их публикация отменяется.

Пиктограммы *Move* и *Copy*

С помощью пиктограмм *Move* (Переместить) и *Copy* (Копировать) статьи можно перемещать или копировать в другие категории.

Пиктограмма *Trash*

С помощью пиктограммы *Trash* (Корзина) одна или несколько статей помещаются в корзину.

Пиктограмма *Parameters*

Щелчок на пиктограмме *Parameters* (Параметры) приводит к отображению области параметров статьи, где можно определять установки по умолчанию для всех статей. Эти установки изначально применимы ко всем статьям. Если вы хотите задать индивидуальные установки для статьи, вы можете сделать это в управляемых параметрах статьи. Страница глобальной конфигурации (*Global Configuration*) показана на рис. 8.6.

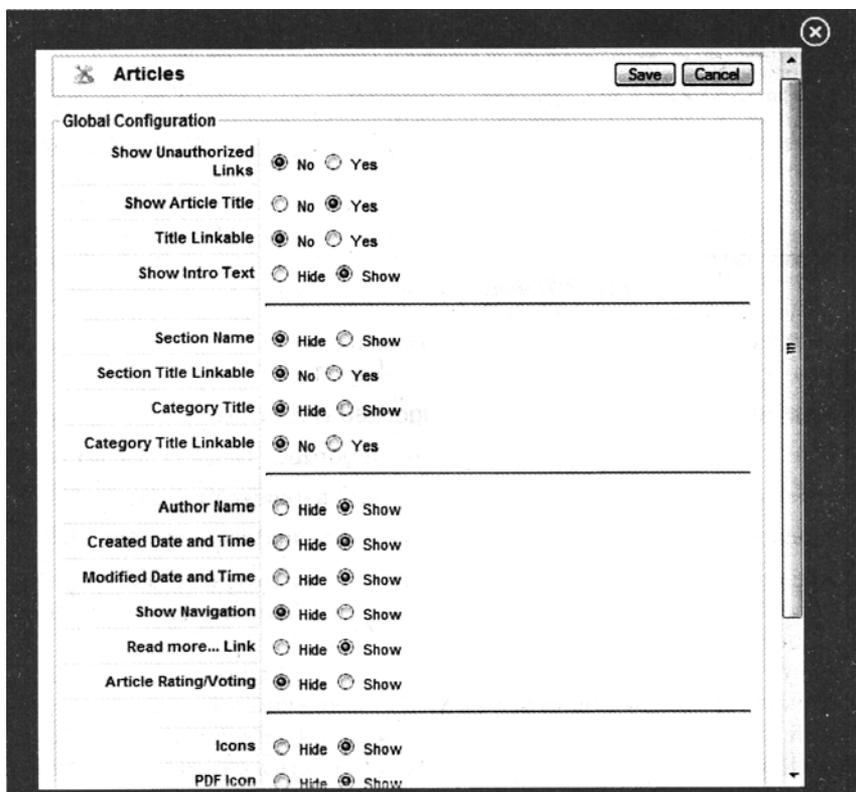


Рис. 8.6. Страница *Global Configuration*



В Joomla! 1.0 многие предварительные установки располагались на странице Global Configuration. В Joomla! 1.5 эти предварительные установки были перемещены в индивидуальные компоненты. Таким образом, теперь вы найдете предварительные установки для статей в центральной области управления статьями, а предварительные установки для, скажем, протоколирования поисковых строк — в компоненте поиска.

Создание новой статьи

Давайте создадим статью для фрагмента новостей, который должен быть отображен на титульной странице. Щелкните на пиктограмме New (Создать). Появится шаблон ввода с большим текстовым полем для редактирования, как показано на рис. 8.7.

The screenshot shows the Joomla! article creation interface. At the top, there's a title bar 'Article: [New]' and navigation buttons: Preview, Save, Apply, Cancel, Help. Below this are several sections: 'Title' and 'Alias' input fields; 'Section' dropdown menu (set to 'Uncategorized'); 'Published' and 'Front Page' radio buttons (both set to 'Yes'); 'Category' dropdown menu (set to 'Uncategorized'); 'State' dropdown menu (set to 'Draft Unpublished'); 'Hits' input field; 'Revised' radio buttons (set to 'Times'); 'Created' and 'Modified' date pickers (both set to 'Tuesday, 18 March 2008 14:20'). A rich text editor is in the center. On the right, the 'Parameters - Article' section includes 'Author' dropdown (set to 'Administrator'), 'Author Alias' input field, 'Access Level' dropdown (set to 'Public'), 'Created Date', 'Start Publishing', and 'Finish Publishing' date pickers (all set to '2008-03-18 14:20:41'). Below are expandable sections for 'Parameters - Advanced' and 'Metadata Information'.

Рис. 8. 7. Форма ввода новой статьи

В поле Title (Заголовок) введите заголовок и какой-нибудь текст в большой области редактирования внизу. Выберите:

- в списке Section (Раздел) пункт Uncategorized (Некатегоризированная);
- в списке Category (Категория) пункт Uncategorized (Некатегоризированная);
- для переключателя Published (Опубликована) вариант Yes (Да);
- для переключателя Front Page (Титульная страница) вариант Yes (Да).

Щелкните на пиктограмме Save (Сохранить) и посмотрите, как будет выглядеть веб-сайт. Только что созданный фрагмент новостей действительно появится на титульной странице (рис. 8.8).

Итак, текст на месте. Теперь необходимо позаботиться о мелких деталях. Прежде всего, весь текст отображать не нужно, а лишь маленький кусочек. Вернитесь к административной части, поместите курсор в то место, где должен заканчиваться рекламный фрагмент, и щелкните на кнопке Read More (Читать далее) ниже текстового поля. В окно будет вставлена линия красного цвета, символизирующая разделитель между двумя частями текста, как показано на рис. 8.9.

- Article ID (Идентификатор статьи). Номер набора данных статьи.
- State (Состояние). Текущее состояние (опубликована ли в данный момент).
- Hits (Попаданий). Количество обращений к статье. Щелчок на кнопке Reset (Сброс) приводит к сбросу счетчика в 0.
- Revised (Пересмотрена). В начале статья имеет номер версии, равный 1. Каждый раз, когда она сохраняется, номер версии увеличивается на 1 (это предоставляет фундамент для планового администрирования версий).
- Created (Создана). Дата создания.
- Modified (Изменена). Дата редактирования.

Область Parameters – Article

В области Parameters – Article (Параметры – статья) представлены базовые параметры (рис. 8.11).

Рис. 8.11. Область Parameters – Article

- Author (Автор). Можно выбрать автора материала из списка, в котором отображаются все существующие пользователи.
- Author Alias (Псевдоним автора). Можно ввести псевдоним автора, который будет отображен на веб-сайте.
- Access Level (Уровень доступа). Эта опция используется для обеспечения доступа к трем группам пользователей.
- Created Date (Дата создания). Здесь можно изменить дату создания статьи. Если щелкнуть на пиктограмме календаря, появится графический календарь, помогающий ввести дату.
- Start Publishing (Дата начала публикации). Позволяет установить дату публикации. По умолчанию контент публикуется немедленно. Если щелкнуть на пиктограмме календаря, появится графический календарь, помогающий ввести дату.
- End Publishing (Дата окончания публикации). Здесь можно ввести дату устаревания контента. По умолчанию контент не устаревает никогда. Если щелкнуть на пиктограмме календаря, появится графический календарь, помогающий ввести дату.

Область Parameters – Advanced

В области Parameters – Advanced (Параметры – дополнительно) находятся параметры, которые были определены для данного контента (рис. 8.12). Эти установки действительны только для запроса полной статьи в Site⇒Global Configuration (Сайт⇒Глобальная конфигурация). Эти установки также действительны для вызова полной статьи – в нашем случае по ссылке Read more.

Parameters - Advanced	
Show Title	Use Global ▾
Title Linkable	Use Global ▾
Intro Text	Use Global ▾
Section Name	Use Global ▾
Section Title Linkable	Use Global ▾
Category Title	Use Global ▾
Category Title Linkable	Use Global ▾
Article Rating	Use Global ▾
Author Name	Use Global ▾
Created Date and Time	Use Global ▾
Modified Date and Time	Use Global ▾
PDF Icon	Use Global ▾
Print Icon	Use Global ▾
E-mail Icon	Use Global ▾
Content Language	- Select Language - ▾
Key Reference	<input type="text"/>
Alternative Read more: text	<input type="text"/>

Рис. 8.12. Область Parameters – Advanced

- Show Title (Показывать заголовок). Должен ли заголовок страницы отображаться или его следует скрыть.
- Title Linkable (Ссылка на заголовок). Должен ли заголовок отображаться в виде ссылки.
- Intro Text (Вводный текст). Следует ли отображать рекламный текст.
- Section Name (Имя раздела). Должно ли отображаться наименование раздела.
- Section Title Linkable (Ссылка на заголовок раздела). Должно ли имя раздела отображаться как ссылка на все статьи этого раздела.
- Category Title (Заголовок категории). Должно ли отображаться наименование категории.
- Category Title Linkable (Ссылка на заголовок категории). Должно ли наименование категории отображаться как ссылка на все категории в данном разделе.
- Перечисленные выше семь параметров переопределяют глобальные установки для индивидуальной статьи.
- Content Language (Язык контента). Язык статьи. Это новое средство предлагает интересные варианты контроля языка в сочетании с Joom!Fish Алекса Кемпенса (<http://www.joomfish.net/>). В зависимости от контента, веб-сайт может отображаться на одном или другом языке.
- Key Reference (Ключевая ссылка). Здесь могут определяться ссылки для экспорта в формат DocBook (<http://ru.wikipedia.org/wiki/DocBook>). На данный момент это имеет смысл только для разработчиков и связано с системой справки.

- Alternative Read more text (Альтернативный текст для *Read more*). Для ссылки Read more можно задать альтернативный текст. Это важно при создании веб-сайта, свободного от барьеров.

Вкладка Metadata Information

На вкладке Metadata Information (Информация о метаданных) можно ввести специфическое описание и ключевые слова для каждого раздела страницы в виде метаданных. Текст, введенный здесь, затем вставляется с сообщением в дескрипторы meta исходного кода HTML веб-сайта в дополнение к метаданным, специфицированным в области Global Configuration (рис. 8.13).

The image shows a screenshot of the Joomla! administration interface, specifically the 'Metadata Information' tab. The tab is titled 'Metadata Information' with a dropdown arrow. Below the title, there are four main sections:

- Description:** A large, empty text area for entering a description.
- Keywords:** A large, empty text area for entering keywords.
- Robots:** A small text input field for entering search engine directives.
- Author:** A small text input field for entering the author's name.

 The form is presented in a clean, structured layout with labels on the left and input areas on the right.

Рис. 8.13. Вкладка Metadata Information

В поле Robots (Роботы) можно ввести поисковые слова для роботов. Затем формируется соответствующий дескриптор meta:

```
<meta name="robots" content="слово1 слово2" />
```

В поле Author (Автор) можно ввести специальное имя автора для дескрипторов meta.

Изображения

Каким образом изображения поместить в контент? Изображения – это постоянно повторяющаяся тема в мире CMS. Изображение должно быть создано (камерой), передано (по кабелю или WLAN), отредактировано (средствами улучшения изображений), загружено на сервер (через FTP или средствами PHP) и привязано к статье.

Joomla! по умолчанию позволяет связывать изображения в диспетчере медиа (Media Manager) со статьей через диалог выбора.

Поместите курсор в то место в тексте, куда необходимо вставить изображение. Затем щелкните на кнопке Image (Изображение) ниже текстового окна. После этого должен появиться экран, показанный на рис. 8.14.

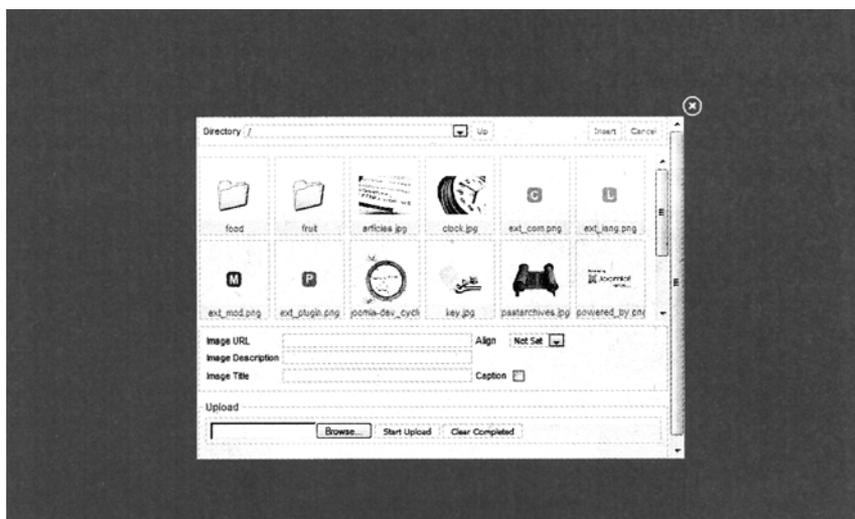


Рис. 8.14. Диалог выбора изображения

Можно связать любое изображение с любой частью контента. Если нужное изображение еще не находится в медиа-области, его следует загрузить щелчком на кнопке Start Upload (Начать загрузку).

Выберите подкаталог и щелкните на изображении, которое хотите вставить. Щелкните на кнопке Insert (Вставить). После этого можно добавить заголовок для изображения и выровнять его. Изображение теперь появится в текстовом окне, как показано на рис. 8.15.

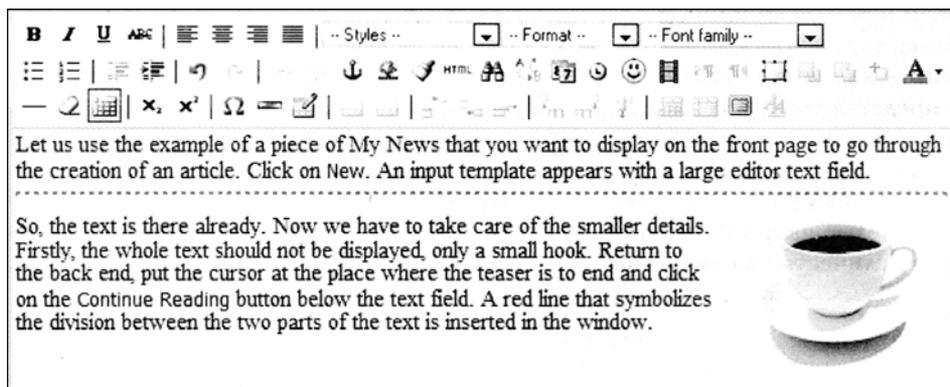


Рис. 8.15. Текст статьи с изображением

Параметры изображения можно отредактировать в редакторе TinyMCE. Отметьте изображение щелчком левой кнопкой мыши. Затем щелкните на пиктограмме с деревом над текстовым окном. Откроется диалоговое окно изображения редактора TinyMCE. Теперь можно определить дополнительные параметры, такие как ссылки, всплывающие окна, позиция и ряд других. На рис. 8.16 задано 15-пиксельное пространство вокруг изображения.

После щелчка на пиктограмме Apply (Применить) проверьте результат либо на веб-сайте, либо щелчком на пиктограмме Preview (Предварительный просмотр). Вы можете так же легко вставлять разрывы страниц, как и ссылки Read more или изображения. Установите курсор в позицию, где необходимо поместить разрыв страницы, и щелкните на кнопке Pagebreak (Разрыв страницы) под текстовым полем. Появится экран, показанный на рис. 8.16.

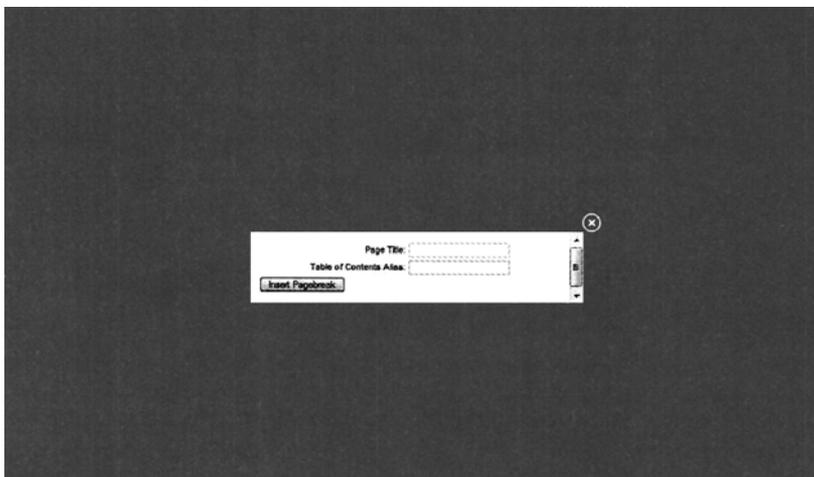


Рис. 8.16. Вставка разрыва страницы

В поле Page Title (Заголовок страницы) можно ввести заголовок для следующей страницы. Вдобавок в поле Table of Contents Alias (Псевдоним для содержания) также можно указать информацию для таблицы содержания, которую автоматически строит Joomla!. Например, заголовком статьи здесь является My News, а в таблице содержания будет присутствовать заголовок *Everything you ever wanted to know about coffee*. Щелкните на кнопке Insert Pagebreak (Вставить разрыв страницы). Разрыв страницы будет представлен серой линией в текстовом окне.

Как показано на рис. 8.17, над текстом статьи теперь имеется строка Page 1 of 2 (Страница 1 из 2), справа сверху появилась область Article Index (Индекс статьи) с текстом, введенным вами, а под текстом статьи расположились ссылки Prev (Предыдущая) и Next (Следующая) для навигации по страницам. В текст можно встраивать произвольное количество изображений и разрывов страниц.



Рис. 8.17. Контент с изображением и разрывом страницы

Корзина для статей

Точно так же, как и с пунктами меню, удалять статьи непосредственно нельзя. Щелчком на пиктограмме Trash (Корзина) в интерфейсе управления статьями статьи отправляются в корзину для мусора. Щелчком на пиктограмме Restore (Восстановить) на экране Trash Manager (Диспетчер корзины) удаленные статьи можно восстановить, а щелчком на пиктограмме Delete (Удалить) – удалить их окончательно (рис. 8.18).

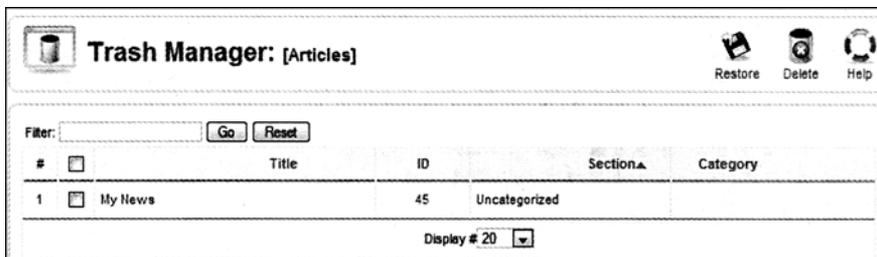


Рис. 8.18. Страница Trash Manager

Диспетчер разделов

Обзорная таблица на экране Section Manager (Диспетчер разделов) перечисляет в структурированном формате информацию, которую вы уже видели в других списках. В данном случае она включает категории, которые содержатся в разделе, вместе со счетчиками активных и находящихся в корзине разделов (рис. 8.19).

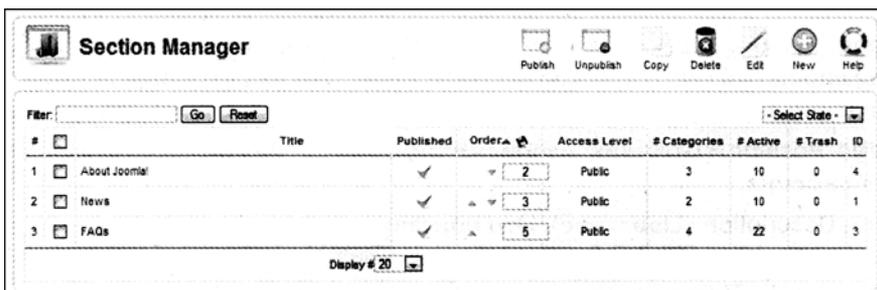


Рис. 8.19. Страница Section Manager

Здесь можно публиковать, копировать, удалять и редактировать существующие разделы, а также создавать новые.

Редактирование разделов

Для примера модифицируем существующий раздел News и свяжем его с ранее созданным меню Joomla! 1.5 Book. Отметьте флажок перед News в списке разделов и щелкните на пиктограмме Edit (Редактирование). Появится экран редактирования раздела (Section: [Edit]), показанный на рис. 8.20.

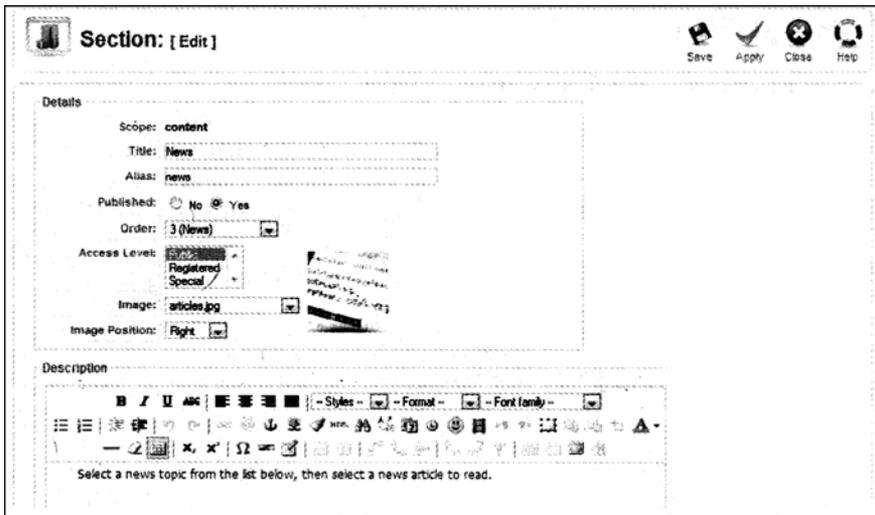


Рис. 8.20. Экран редактирования раздела

- Title (Заголовок). Заголовок раздела (также появляется в строке заголовка браузера).
- Alias (Псевдоним). Имя во внутреннем реестре Joomla!.
- Published (Опубликовано). Должен ли раздел быть немедленно опубликован.
- Order (Порядок). Здесь определяется порядок сортировки разделов.
- Access Level (Уровень доступа). Кто имеет доступ к этому элементу.
- Image (Изображение). Здесь можно выбрать изображение, которое будет отображено на веб-сайте при запросе категории. Это изображение должно храниться в корневом каталоге Media Manager – в данном случае это `articles.jpg`.
- Image Position (Позиция изображения). Здесь можно задать выравнивание изображения.
- Окно Description (Описание). Это описание раздела. Если вы выбрали редактор WYSIWYG в Site⇒Global Configuration, то здесь будет предоставлен базовый текстовый редактор.

Единственное, чего не хватает – это ссылки на меню Joomla! 1.5 Book.

Привязать этот раздел к меню можно, выбрав Menu⇒Main Menu (Меню⇒Главное меню). В опции Display in (отображать в) укажите Joomla! 1.5 Book.

Выберите Menu⇒Main Menu и щелкните на пиктограмме New (Создать). Затем щелкните на типе пункта меню Articles (Статьи), как показано на рис. 8.21.

Доступно множество вариантов выбора типа отображаемого формата. Сейчас нужно отобразить раздел News, и есть выбор между следующими двумя компоновками:

- Section Blog Layout (Компоновка блога раздела)
- Section Layout (Компоновка раздела)

Статьи можно связывать как списки или так называемые блог-отображения (как на текущей титульной странице).

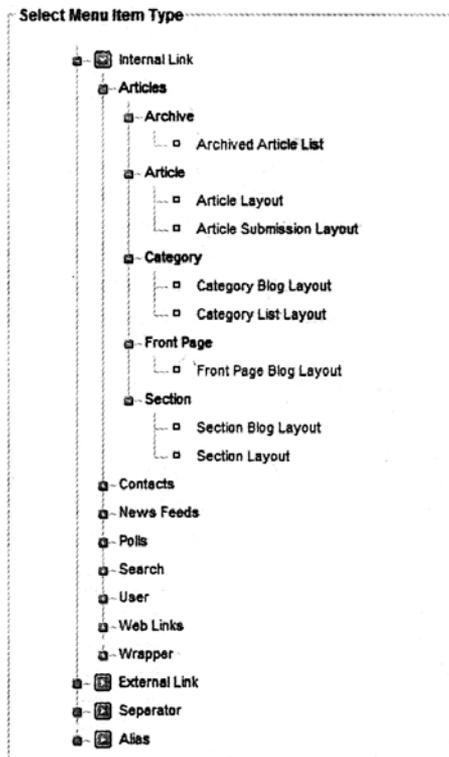


Рис. 8.21. Выбор типа пункта меню

Формат списка показывает все категории, назначенные в раздел. Заголовок берется из раздела, который мы только что изменили. В данном случае сюда входят лишь две категории – Latest News (Последние новости) и Newsflash (Экстренное сообщение). Это должно дать табличное представление контента при простом щелчке на ссылке категории (рис. 8.22).



Рис. 8.22. Внешний вид титульной страницы

Блог-отображение в Joomla! – это список элементов с их текстами (анонсами, рекламными фрагментами) и, в зависимости от статьи, со ссылкой Read more. В обоих случаях в браузере присутствует окрашенная в оранжевый цвет кнопка, за которой вы найдете новостную ленту для соответствующих элементов. Формат отображения Archive Blog (Блог архива) в точности повторяет представление блога, но контент на этот раз берется из архива (архивировать статьи можно в интерфейсе управления статьями).

Попробуйте оба варианта. После щелчка на выбранном типе отображения вы попадаете в экран редактирования для этой ссылки. Там вы будете иметь дело с десятками самоочевидных параметров, которые можно изменять и наблюдать за результатом.

Диспетчер категорий

Категории можно редактировать точно так же, как разделы. В обзорной таблице на экране Category Manager (Диспетчер категорий) представлена информационная структура, которая вам уже знакома по отображению разделов, но на этот раз она расширена присвоенным разделом, количеством активных статей, содержащихся в категории, и количеством статей, находящихся в корзине (рис. 8.23).

#	<input type="checkbox"/>	Title	Published	Order	Access Level	Section	# Active	# Trash	ID
1	<input type="checkbox"/>	The Project	<input checked="" type="checkbox"/>	1	Public	About Joomla!	3	0	25
2	<input type="checkbox"/>	The CMS	<input checked="" type="checkbox"/>	2	Public	About Joomla!	5	0	29
3	<input type="checkbox"/>	The Community	<input checked="" type="checkbox"/>	3	Public	About Joomla!	2	0	30
4	<input type="checkbox"/>	General	<input checked="" type="checkbox"/>	1	Public	FAQs	8	0	31
5	<input type="checkbox"/>	Current Users	<input checked="" type="checkbox"/>	2	Public	FAQs	6	0	28
6	<input type="checkbox"/>	New to Joomla!	<input checked="" type="checkbox"/>	3	Public	FAQs	3	0	27
7	<input type="checkbox"/>	Languages	<input checked="" type="checkbox"/>	4	Public	FAQs	5	0	32
8	<input type="checkbox"/>	Latest	<input checked="" type="checkbox"/>	1	Public	News	5	0	1
9	<input type="checkbox"/>	Newsflash	<input checked="" type="checkbox"/>	2	Public	News	5	0	3

Рис. 8.23. Страница Category Manager

Когда вы посредством пункта меню Contents Category Manager⇒Edit (Контент⇒Диспетчер категорий⇒Редактировать) назначаете вновь созданные статьи категории Latest News и разделу News (или Newsflash), они автоматически появляются во вновь созданных списках разделов (рис. 8.24).

Category: [Edit]

Save Apply Close Help

Details

Title: Latest

Alias: latest-news

Published: No Yes

Section: News

Category Order: 1 (Latest)

Access Level: Public

Image: taking_notes.jpg

Image Position: Left

Рис. 8.24. Экран редактирования категории

Диспетчер титульных страниц

Титульная страница (**Frontpage**) имеет специальное предназначение. Титульная граница – это заглавная страница вашего веб-сайта. На ней содержатся наиболее важные статьи.

Часто на простых веб-сайтах на стартовой странице мало что представляется. В некоторых ситуациях оказывается достаточно единственной статьи. Тем не менее, возникает вопрос: как будет выглядеть сайт позднее?

Если позже вы захотите иметь стартовую страницу¹ с несколькими статьями, то данный контент должен быть представлен в формате блог-отображения. При редактировании статей можно активизировать переключатель Front Page (Титульная статья). После этого статья будет отображаться на экране Front Page Manager (Диспетчер титульных страниц), показанном на рис. 8.25. Представленный здесь список подобен списку статей. Здесь можно сортировать статьи внутри титульной страницы. Для каждой статьи указывается заголовок, раздел, категория и автор. Некатегоризированные статьи здесь помечаются как Static Content (Статический контент). Список можно ограничить с помощью фильтрации. Если статей больше, чем помещается на титульной странице, на веб-сайте активизируются ссылки навигации, которые расположены внизу статей на титульной странице.

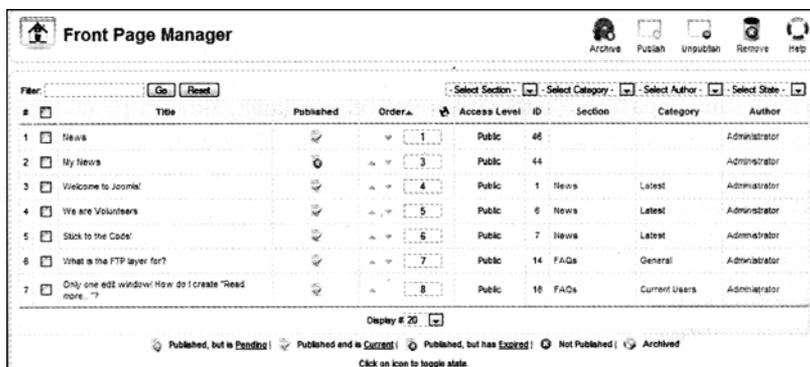


Рис. 8.25. Страница Front Page Manager

Изменить компоновку стартовой страницы можно в меню ссылок (Menu⇒Main Menu⇒Home⇒Edit (Меню⇒Главное меню⇒Домой⇒Редактировать)), как показано на рис. 8.26.

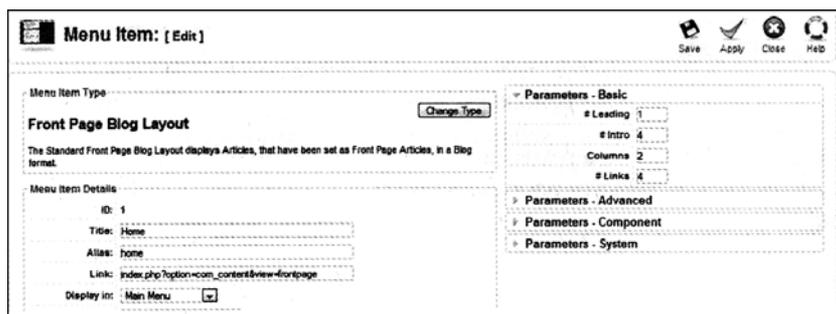


Рис. 8.26. Страница редактирования пункта меню

9

Меню Components

При разработке программного обеспечения компонентом обычно называют программу или фрагмент программного кода, содержащий бизнес-логику, который доступен через определенные интерфейсы, и который может иметь пользовательский интерфейс.

Воспринимайте компонент как черный ящик. Вы помещаете что-то на передний план, а что-то уходит на задний; вам не нужно знать, что происходит внутри; благодаря этому вы сможете применять этот черный ящик для множества совершенно разных целей.

Компоненты могут проектироваться некоторым обобщенным образом и собираться в удобные пакеты. Идея программного компонента подобна и в Joomla! Бизнес-логика, такая как администрирование баннеров или форума, пишется некоторым обобщенным образом и затем работает в Joomla! совместно с шаблонами и администрированием. Модуль часто предполагает отображение веб-сайта, а подключаемый модуль добавляет дополнительную функциональность к существующему тексту (подобно языку сценария). На сегодняшний день существуют более 2000 полезных компонентов для Joomla! 1.0, и все больше и больше разрабатывается для Joomla! 1.5. Но сначала давайте рассмотрим компоненты, поставляемые вместе с Joomla! 1.5.

Меню Banners

Меню Banners (Баннеры) содержит пункты Banners (Баннеры), Clients (Клиенты) и Categories (Категории).

Баннеры

Компонент-баннер позволяет отображать на сайте рекламные баннеры. Баннер может включать графику или текст. Подобно невероятно успешной модели Google, веб-сайты Joomla! также могут продавать рекламные ссылки. При каждом посещении вашего сайта отображается другой баннер из администрируемого набора. Каждый из них подсчитывает посещения. Текстовая ссылка баннера активна и ведет на сайт клиента. В дополнение к текстовым ссылкам могут также размещаться и графические баннеры.

Компонент-баннер поддерживает администрирование клиента, категории и собственно баннера. Графические баннеры часто переключаются в так называемые полные баннеры. Полный баннер имеет размер 468×60 пикселей и не должен по размеру превышать 20 Кбайт. Используемые форматы: .gif, .jpg или .png.

Давайте рассмотрим одну из двух версий баннерных переключателей. Создадим баннер размером 468×60 пикселей, показанный на рис. 9.1.

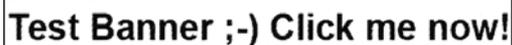


Рис. 9.1. Пример графического баннера

Клиенты

Перед тем, как включать баннер, нужно создать заказчика.

Выберите в меню Components⇒Banners⇒Clients⇒New (Компоненты⇒Баннеры⇒Клиенты⇒Создать), создайте учетную запись нового заказчика и сохраните ее щелчком на пиктограмме Save (Сохранить). Экран Banner Client: [Edit] (Клиент баннера: [редактирование]) показан на рис. 9.2.

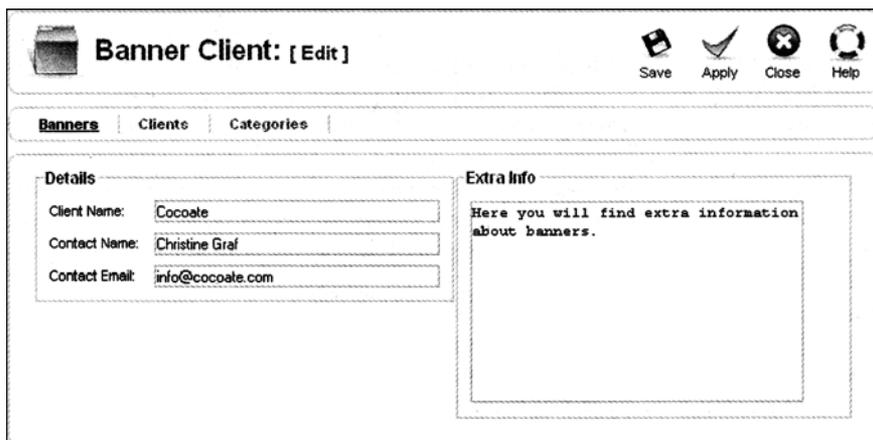


Рис. 9.2. Редактирование клиента баннера

На экране Banner Client Manager (Диспетчер клиентов баннеров), куда вы попадаете после сохранения, теперь отобразится ваш новый заказчик вместе с количеством его активных баннеров.

Управление баннерами

Теперь необходимо ввести баннер и/или текстовую ссылку для этого клиента. Щелкните на Components⇒Banners⇒Banners (Компоненты⇒Баннеры⇒Баннеры). На экране Banner Manager (Диспетчер баннеров) показаны существующие баннеры (рис. 9.3).

Здесь доступны перечисленные ниже колонки.

- Name (Имя). Имя баннера.
- Client (Клиент). Клиент баннера.
- Category (Категория). Категория, назначенная баннеру.
- Published (Опубликовано). Признак опубликования баннера.
- Order (Порядок). Определяет порядок включения баннеров.
- Sticky (Приклеенный). Имеют ли предпочтение "приклеенные" баннеры.

- Impressions (Показы). Количество успешных показов и количество оставшихся показов.
- Clicks (Щелчков). Щелчки на баннере в количественном и процентном выражении.
- Tags (Теги). Теги (категории), назначенные баннеру. Отображение баннеров может управляться этими тегами.
- ID (Идентификатор). Ключ набора данных.

#	Name	Client	Category	Published	Order	Sticky	Impressions	Clicks	Tag	ID
1	OSM 1	Open Source Matters	Joomla	✓	1	No	43 of Unlimited	0 - 0.00%		1
2	OSM 2	Open Source Matters	Joomla	✓	2	No	49 of Unlimited	0 - 0.00%		2
3	Joomla! Promo Shop	Open Source Matters	Joomla! Promo	✓	3	No	30 of Unlimited	1 - 3.33%		7
4	Joomla! Promo Books	Open Source Matters	Joomla! Promo	✓	4	No	28 of Unlimited	0 - 0.00%		8
5	Joomla!	Open Source Matters	Text Ads	✓	1	No	45 of Unlimited	0 - 0.00%		3
6	JoomlaCode	Open Source Matters	Text Ads	✓	2	No	45 of Unlimited	0 - 0.00%		4
7	Joomla! Extensions	Open Source Matters	Text Ads	✓	3	No	40 of Unlimited	0 - 0.00%		5
8	Joomla! Shop	Open Source Matters	Text Ads	✓	4	No	40 of Unlimited	0 - 0.00%		6

Рис. 9.3. Страница Banner Manager

Графические баннеры

Чтобы включить новый полный баннер, сначала загрузите файл с помощью диспетчера медиа (Site⇒Media Manager (Сайт⇒Диспетчер медиа)) в каталог /image/banner (рис. 9.4). Для удаления файла баннера переключитесь на подробный вид в разделе и щелкните на пиктограмме Delete (Удалить).

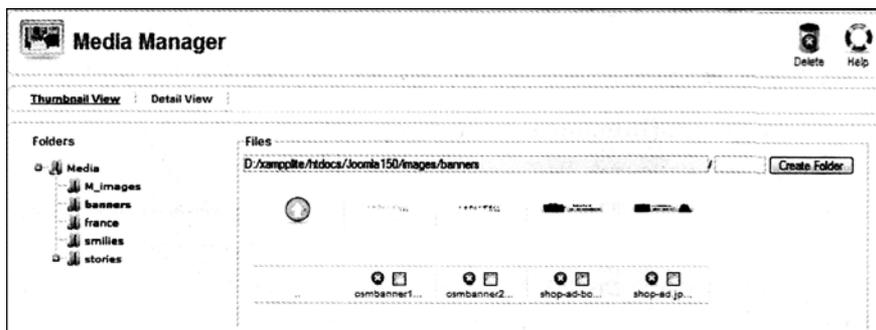


Рис. 9.4. Загрузка файла баннера

После загрузки баннера щелкните на пиктограмме New (Создать) на вкладке Banners (Баннеры) страницы Banner Manager и заполните форму Details (Подробности), показанную на рис. 9.5.

Test Banner ;-). Click me now!

Рис. 9.5. Страница Details

Потребуется указать следующие подробности, связанные с баннером.

- Banner Name (Имя баннера). Присвойте баннеру осмысленное имя, чтобы узнать его в списке диспетчера баннеров.
- Alias (Псевдоним). Это поле пока не имеет применения в Joomla!. Предполагается, что оно будет использоваться для URL баннеров в будущем.
- Show Banner (Показывать баннер). Вы можете прекратить ротацию баннера, выбрав No (Нет).
- Sticky (Приклеенный). Баннеры, помеченные как Sticky, отображаются первыми.
- Order (Порядок). Здесь определяется порядок, в соответствии с которым отображаются баннеры.
- Category (Категория). Можно назначать категории и затем отображать баннеры из определенных категорий на веб-сайте. Каждый баннер должен иметь присвоенную ему категорию, которые можно администрировать через меню Components⇒Banners⇒Categories (Компоненты⇒Баннеры⇒Категории).
- Client Name (Имя клиента). Выберите клиента из списка существующих клиентов.
- Impressions Purchased (Приобретенные показы). Введите количество приобретенных показов или отметьте флажок Unlimited (Неограниченно).
- Click URL (URL-адрес щелчка). Введите URL-адрес веб-сайта, на который должен указывать баннер.
- Custom Banner Code (Специальный код баннера). Здесь вводится специальный код баннера из программы Affiliate-Partner.

Поскольку этот шаблон также предназначен для редактирования баннеров, вы также найдете здесь информацию о количестве успешных щелчков и кнопку Reset Clicks (Сбросить щелчки), которая устанавливает счетчик щелчков в 0.

- Description/Notes (Описание/примечания). Внутренняя информация для этого баннера.
- Banner Image Selector (Селектор изображений баннера). Здесь выбирается изображение для баннера. После выбора оно будет отображаться под этим полем.
- Tags (Теги). Здесь задаются теги для этого баннера.

После щелчка на пиктограмме Save (Сохранить) баннер должен запускаться в ротацию и отображаться на вашем веб-сайте.

Текстовые ссылки

Для интегрирования текстовой ссылки введите ссылку в формате HTML в поле Custom Banner Code. Чтобы иметь возможность различать разные типы баннеров, потребуется заранее определить соответствующие категории (например, Text Ads).

Баннеры отображаются на веб-сайте посредством модулей. Подробнее о модулях ганнеров вы узнаете из раздела "Модули сайта" главы 10.

Контакты

Часто клиенту, посетившему ваш сайт, бывает трудно связаться с вами. Многие сотрудники обычно работают в разных отделах компании, и часто на домашней странице или форме приведен только один адрес электронной почты (например, info@company.com), потому клиент не имеет представления, кому он принадлежит.

Чтобы избежать этого, Joomla! позволяет заводить категории контактов. Эти категории можно, например, определить для разных отделов компании, и затем ввести в них контактные персоны. В зависимости от конфигурации, Joomla! затем отображает таблицу сотрудников и/или форму контакта для каждого члена персонала на сайте.

Диспетчер контактов

В диспетчере контактов осуществляется администрирование индивидуальных форм контактов. В примерах данных имеется сотрудник по имени Name, который включен в группу Contacts (Контакты). Давайте создадим нового члена персонала. Щелкните на Components⇒Contacts⇒Contacts (Компоненты⇒Контакты⇒Контакты). На появившейся странице Contact Manager (Диспетчер контактов) вы увидите раздел Contacts (Контакты) с упомянутым контактом из примеров данных (рис. 9.6).

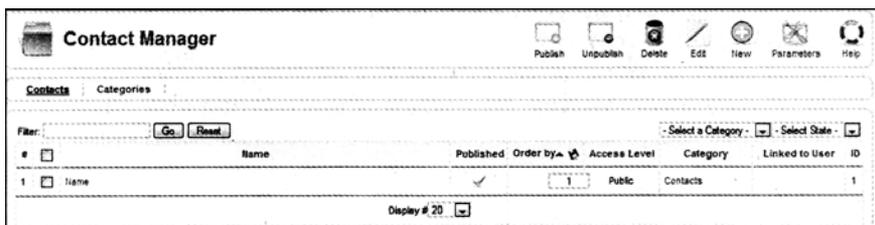


Рис. 9.6. Страница Contact Manager

Щелкните на пиктограмме New (Создать) и настройте новый контакт. Форма, которая появится, разделена на три области – Details (Подробности), Information (Информация) и Parameters (Параметры), как показано на рис. 9.7.

Рис. 9.7. Форма указания информации о контактной персоне

Область *Details*

В этой области содержатся перечисленные ниже опции.

- Name (Имя). Имя контакта
- Alias (Псевдоним). Краткий URL.
- Published (Опубликовано). Опубликован или нет.
- Category (Категория). Выбор категории контакта. На данный момент имеется только пример категории Contacts.
- Linked to User (Связан с пользователем). Эта опция позволяет связать контакт с определенной учетной записью пользователя.
- Access Level (Уровень доступа). Группа пользователей, имеющих доступ к этому контакту.
- ID (Идентификатор). Номер набора данных контакта.

Область *Information*

В этой области собраны информационные поля данного контакта. Поля, не имеющие никаких значений, автоматически скрываются. Поле Miscellaneous Information (Прочая информация) может использоваться для индивидуальных описаний. В поле Image (Изображение) можно указать изображение контактной персоны из подкаталога /images/stories/. Загружается изображение с помощью диспетчера медиа.

Область *Parameters*

В этой области указывается, какие поля контакта необходимо отображать, а какие – скрыть. vCard представляет собой формат файла, который может быть автоматически введен в многочисленные адресные каталоги.

Сохраните новый контакт щелчком на пиктограмме Save (Сохранить). Новый контакт сразу же отобразится в списке.

Создание ссылки меню для веб-сайта

Чтобы встроить ссылку на контакт в верхнее меню веб-сайта, щелкните на **Menus**⇒**Top Menu**⇒**New** (Меню⇒Верхнее меню⇒Создать), а затем – на **Contacts** (Контакты) в типе пункта меню. После этого можно выбирать между категорией контакта, отображающей список всех контактов в этой категории (**Contact Category Layout**), или отдельным контактом (**Standard Contact Layout**), как показано на рис. 9.8. Выберите **Standard Contact Layout**.

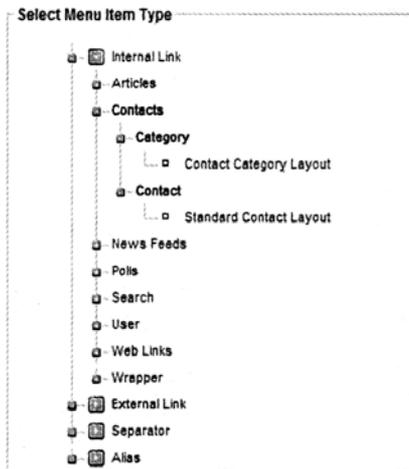


Рис. 9.8. Выбор типа пункта меню

В разделе **Categories** (Категории) можно настраивать новые категории и редактировать существующие. Этот раздел доступен через меню **Components**⇒**Contacts**⇒**Categories** (Компоненты⇒Контакты⇒Категории).

Необходимо ввести имя и псевдоним для ссылки меню, а также корректный контакт в параметрах на странице редактирования **Menu Item**: [Edit] (рис. 9.9). Здесь также можно настроить и параметры.

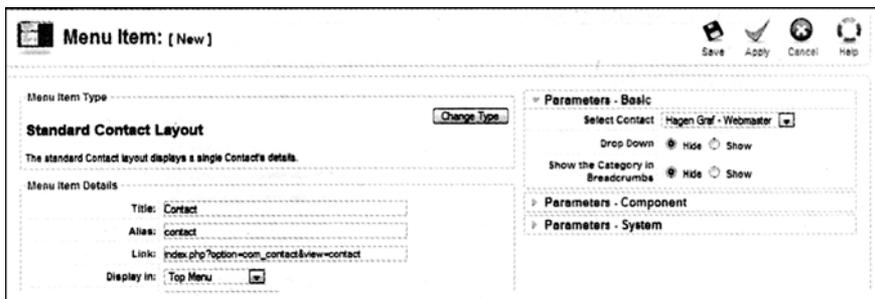


Рис. 9.9. Страница ввода нового пункта меню

Теперь взгляните на веб-сайт и щелкните на ссылке **Contact** (Контакт) в верхнем меню. Вы увидите данные контакта и автоматически сгенерированную форму для установки контакта (рис. 9.10). Адрес электронной почты останется при этом невидимым, что позволит избежать злоупотреблений со стороны спамеров.



Рис. 9.10. Контактная информация на сайте

Еще одной возможностью является список контактов. Этот список обрывается на телефонном номере, поскольку нет необходимости показывать номер мобильного телефона и номер факса.

Чтобы изменить перечень отображаемых в списке полей, вам придется отредактировать ссылку меню. Щелкните на Menus⇒Top Menu (Меню⇒Верхнее меню) и переведите элемент Contact (Контакт) в режим редактирования. В области справа на странице можно должным образом сконфигурировать параметры (рис. 9.11).

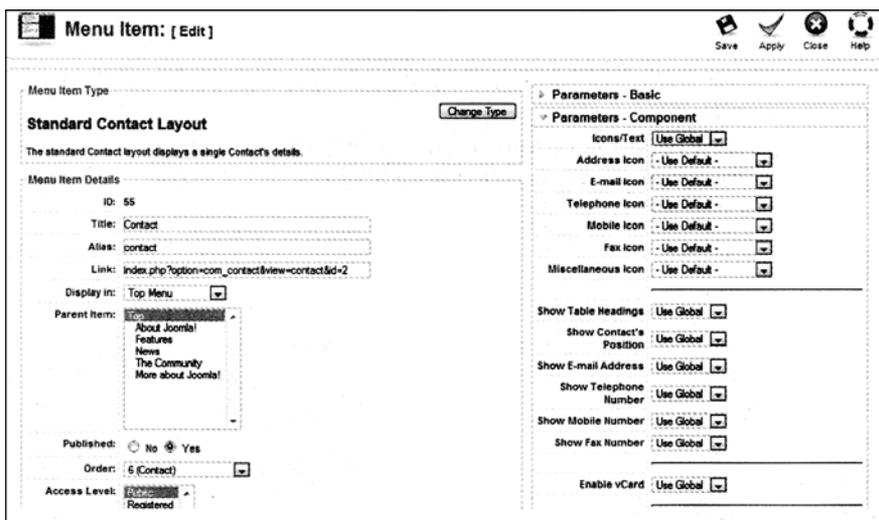


Рис. 9.11. Редактирование пункта меню Contact

В области Parameters – Component (Параметры – компонент) можно исключить определенные адреса электронной почты и слова. По завершении редактирования таблицы следует щелкнуть на имени контакта. В результате в форме контакта отобразится нужная информация.

Категории

На странице Category Manager (Диспетчер категорий) определяются новые и модифицируются существующие категории (рис. 9.12). Вы попадаете на нее через меню Components⇒Contacts⇒Categories (Компоненты⇒Контакты⇒Категории) и вкладку Categories (Категории) в диспетчере контактов.

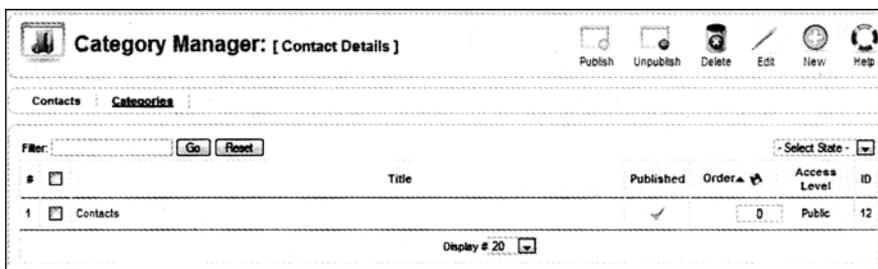


Рис. 9.12. Страница Category Manager

После щелчка на пиктограмме Edit (Редактировать) отображается форма редактирования, показанная на рис. 9.13. Здесь можно добавить изображение и редактировать описание в WYSIWYG-редакторе.

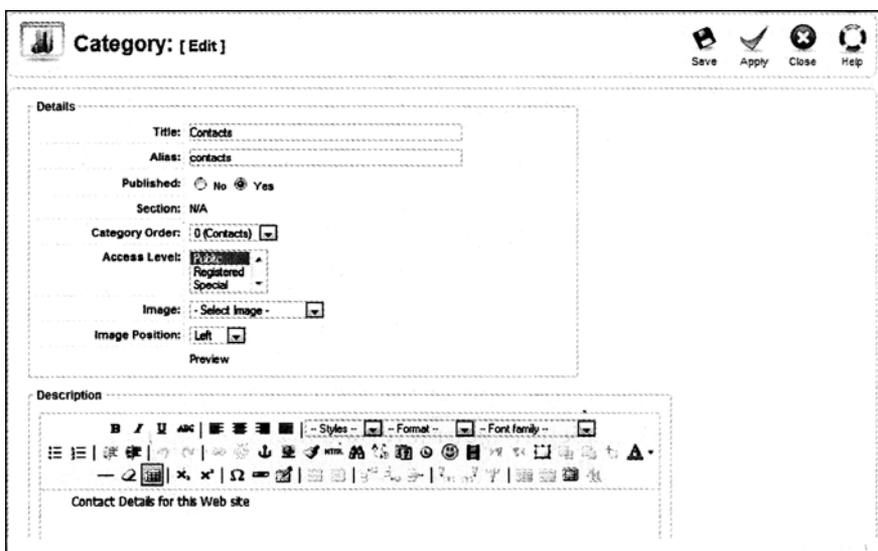


Рис. 9.13. Страница редактирования категории

Ленты новостей

Ленты новостей (news feeds) совершенно потрясающи. Постоянно растущее изобилие информации в Интернете вызвало потребность в эффективных методах ее организации. Регулярное посещение даже десятка веб-сайтов для проверки, что там появилось нового, занимает слишком много времени. Когда же сайтов пятьдесят или сто, то выполнять регулярный обзор совершенно не реально. Ленты новостей призваны решить эту проблему.

С компонентом — лентой новостей вы можете слить наполнения с других сайтов на ваш собственный сайт. Для этого в вашем распоряжении имеются такие инструменты, как Category Manager и Content Manager. Примеры данных содержат несколько категорий и множество лент новостей.

Чтобы интегрировать их в собственную ленту новостей, вы можете поискать на посещаемых вами сайтах маленькую оранжевую кнопку RSS.

Наполнения

Чтобы продемонстрировать пример, добавим наполнение (feed) из собственного блога автора, находящегося по адресу <http://www.bloghouse.org/blog/7/feed>. Щелкните на Components⇒News feeds⇒Feeds⇒New (Компоненты⇒Ленты новостей⇒Наполнения⇒Создать). Появится страница News Feed: [New] (Лента новостей: [новая]), показанная на рис. 9.14.

Рис. 9.14. Страница создания новой ленты

Некоторые опции на этой странице перечислены ниже.

- Name (Имя). Это имя ленты новостей, появляющееся на вашей странице.
- Alias (Псевдоним). Краткий URL.
- Published (Опубликовано). Для немедленной публикации выберите Yes (Да).
- Category (Категория). Выберите соответствующую категорию из числа существующих.
- Link (Ссылка). Это ссылка на ленту новостей, в данном случае — ссылка на блог.
- Number of Articles (Количество статей). Указывает количество статей, которые должны быть интегрированы.
- Cache Time (Время кэширования). Период времени в секундах между обновлениями ленты новостей.
- Order (Порядок). По умолчанию лента новостей находится в начале. Этот порядок может быть изменен после сохранения.

Предполагая наличие доступа в Интернет, ваша новая лента новостей с этого момента будет отображаться на веб-сайте, как показано на рис. 9.15.



Рис. 9.15. Веб-сайт с лентой новостей

Категории

В области Categories (Категории) администрируются категории лент новостей, - тот диспетчер работает аналогично Category Manager для контактов.

Опросы

Интегрированный модуль опросов позволяет публиковать на вашем сайте опросы, 'дин такой опрос входит в примеры данных. Страница редактирования этого опроса показана на рис. 9.16.

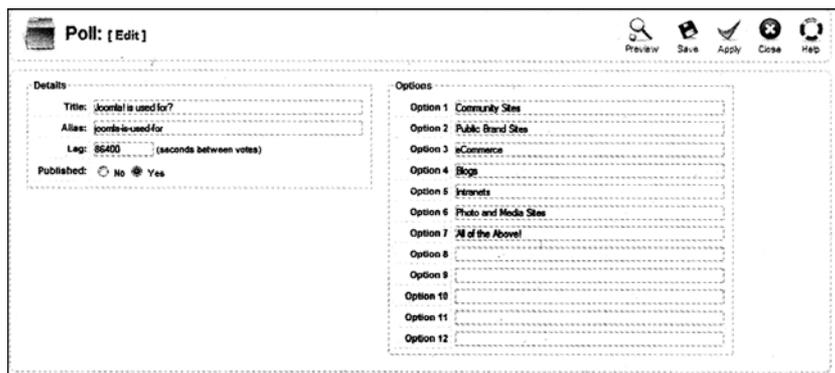


Рис. 9.16. Страница редактирования опроса

- Title (Заголовок). Это заголовок вашего опроса.
- Lag (Интервал). Задаёт время в секундах, которое должно пройти до следующего выбора. Это средство обеспечивает некоторую защиту от фальсификации данных опроса.
- Options (Варианты). Здесь можно ввести до 12 вариантов.

Щелкните на пиктограмме Preview (Предварительный просмотр), чтобы включить предварительный просмотр вашего опроса. Чтобы действительно отобразить его на сайте, удостоверьтесь, что модуль опроса находится в желаемой позиции. Модуль располагается по умолчанию справа (рис. 9.17).

Latest News

- Content Layouts
- The Joomla! Community
- Welcome to Joomla!
- Newsflash 4
- Newsflash 5

Popular

- Joomla! Overview
- Extensions
- Joomla! License Guidelines
- Welcome to Joomla!
- What's New in 1.5?

Welcome to the Frontpage

News

Written by Administrator
Tuesday, 18 March 2008 14:40

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 appears with a large editor text field.

Last Updated (Wednesday, 19 March 2008 03:59)

Read more... >>

Polls

Joomla! is used for?

Community Sites

Public Brand Sites

eCommerce

Blogs

Intranets

Photo and Media Sites

All of the Above!

Рис. 9.17. Внешний вид титульной страницы с опросом

После выбора ответа появится отчет по результатам опроса (рис. 9.18). Сам опрос после этого не отображается, поскольку по умолчанию он назначен только для титульной страницы.

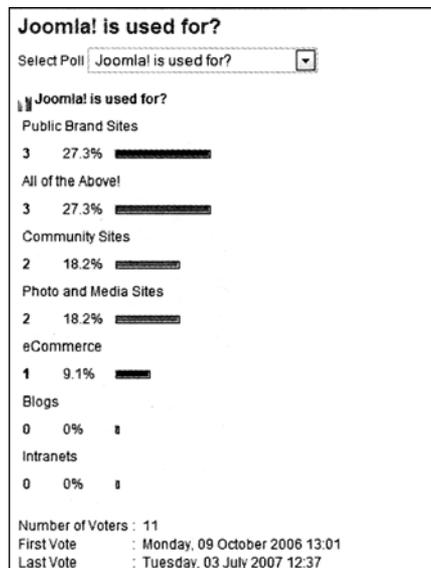


Рис. 9.18. Отчет по результатам опроса

Статистика поиска

Joomla! 1.0 предлагала множество опций для сбора статистики о доступе пользователей к вашему веб-сайту. Протоколирование доступа к данным иногда требовало существенных объемов хранилища на сервере, и потому основная команда разработчиков Joomla! решила избавиться от статистической части (в пользу понятного решения аналитики Google).

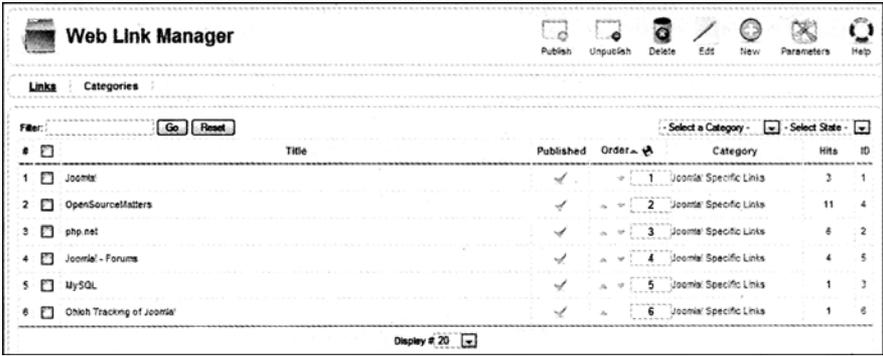
Возможно, наиболее важная часть информации для оператора веб-сайта связана с ответом на вопрос: "Что ищут пользователи на моем веб-сайте?". В Joomla! 1.5 вы можете просмотреть поисковые термины и их частоту в статистике поиска. Чтобы это работало, потребуется явно включить в предварительных установках захват поисковых слов.

Веб-ссылки

С помощью веб-ссылок (Web Links) можно сконструировать список ссылок или загружаемый раздел, который затем интегрировать в свой веб-сайт. Здесь Joomla! снова предоставляет категории и счетчики щелчков на ссылках. В пользовательском меню вы можете предложить зарегистрированным посетителям рекомендуемые ссылки для включения в список. Эти ссылки затем появятся в списке, который может быть запрошен через меню Components⇒Web Links⇒Links (Компоненты⇒Веб-ссылки⇒Ссылки). В зависимости от прав пользователя, эти ссылки могут требовать опубликования.

Ссылки

Просмотреть все ссылки, которые были введены до сих пор, вместе с назначенными им категориями, а также количество вызовов (hits) этих ссылок, можно в обзорном списке ссылок на странице Web Link Manager (Диспетчер веб-ссылок), которая показана на рис. 9.19.



#	Title	Published	Order	Category	Hits	ID
<input type="checkbox"/>	Joomla!	<input checked="" type="checkbox"/>	1	Joomla! Specific Links	3	1
<input type="checkbox"/>	OpenSourceMatters	<input checked="" type="checkbox"/>	2	Joomla! Specific Links	11	4
<input type="checkbox"/>	php.net	<input checked="" type="checkbox"/>	3	Joomla! Specific Links	6	2
<input type="checkbox"/>	Joomla! - Forums	<input checked="" type="checkbox"/>	4	Joomla! Specific Links	4	5
<input type="checkbox"/>	MySQL	<input checked="" type="checkbox"/>	5	Joomla! Specific Links	1	3
<input type="checkbox"/>	Online Tracking of Joomla!	<input checked="" type="checkbox"/>	6	Joomla! Specific Links	1	6

Рис. 9.19. Страница Web Link Manager

Параметры для всех ссылок могут редактироваться и устанавливаться предварительно. Вводить ссылки "с нуля", принимать решение об их публикации и редактировать их можно на форме редактирования веб-ссылок (рис. 9.20).

Ниже перечислены элементы этой формы.

- Name (Имя). Имя ссылки, которое появляется на вашем веб-сайте.
- Alias (Псевдоним). Краткий URL.
- Published (Опубликовано). Должна ли ссылка быть опубликована немедленно.
- Category (Категория). Выбор категории из списка существующих.
- URL (URL-адрес). URL ссылки.
- Order (Порядок). Сортировка ссылок.

- Target (Цель). Здесь можно выбирать, должна ссылка отображаться в новом окне (с навигацией или без) или в том же самом окне.
- Description (Описание). Здесь можно ввести исчерпывающее описание ссылки.

Рис. 9.20. Форма редактирования веб-ссылки

Категории

Категории ссылок, которые доступны в соответствующем разделе диспетчера веб-ссылок, администрируются через меню Components⇒Web Links⇒Categories (Компоненты⇒Веб-ссылки⇒Категории). Страница Category Manager [Web Links] (Диспетчер категорий [веб-ссылки]) показана на рис. 9.21.

#	Title	Published	Order	Access Level	ID
1	Joomla! Specific Links	✓	1	Public	2
2	Other Resources	✓	2	Public	19

Рис. 9.21. Страница Category Manager [Web Links]

Резюме

В этой главе было рассмотрено меню Components. Следующая глава посвящена меню Extensions (Расширения).

10

Меню Extensions

Помимо компонентов, в меню Extensions (Расширения) перечислены все доступные опции расширений. Это модули, подключаемые модули, шаблоны и языки. Имеется также централизованный инсталлятор, который позволяет устанавливать и деинсталлировать расширения Joomla! всего несколькими щелчками кнопкой мыши (рис. 10.1).

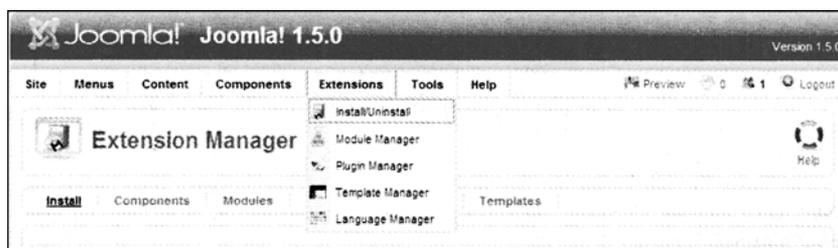


Рис. 10.1. Меню Extensions

Пункт Install/Uninstall

В принципе любой может написать расширение Joomla!, упаковать его в соответствии с определенным набором правил и интегрировать в веб-сайт. В этом разделе вы найдете обзор установленных расширений. После вызова пункта меню Extensions⇒Install/Uninstall (Расширения⇒Установить/деинсталлировать) появляется страница, на которой предлагаются три опции для установки расширений (рис. 10.2):

- Upload Package File (Загрузка файла пакета)
- Install from directory (Установка из каталога)
- Install from URL (Установка из URL)

Каждая из вкладок Install (Установка), Components (Компоненты), Modules (Модули), Plugins (Подключаемые модули), Templates (Шаблоны) и Languages (Языки) содержит списки установленных расширений. В главе 12 мы опишем установку дополнительных компонентов.

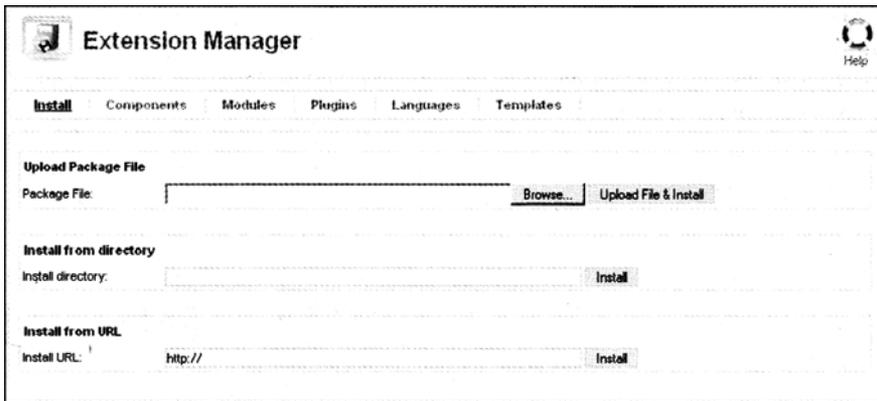


Рис. 10.2. Страница Extension Manager

Вкладки Install, Components, Modules, Plugins, Languages и Templates

Каждая из этих вкладок, например, Components, отображает список компонентов, которые были установлены по умолчанию. Вы увидите данные о версии компонента, дате его создания и информацию об авторе. Чтобы деинсталлировать компонент, выберите его и щелкните на пиктограмме Uninstall (Деинсталлировать) в панели инструментов.



Будьте осторожны! После деинсталляции компонент действительно исчезает, и иногда вместе со всеми хранимыми данными. Не существует такой вещи, как мусорная корзина для компонентов! Это сделано намеренно. Выбор стратегии оставлен за разработчиком компонента. Временами имеет смысл не удалять таблицы, например, когда вы осуществляете обновление. Однако иногда безвозвратное удаление всех таблиц, созданных ранее, смысл имеет. В главе 15 вы узнаете больше об этой технике. Обязательно ознакомьтесь с документацией по соответствующему компоненту, чтобы знать его характеристики.

Диспетчер модулей

Структура модуля намного проще, чем структура компонента. Это фрагмент кода, который интегрирован и интерпретирован другой частью программы. Благодаря возможностям языка сценариев PHP, модули могут собирать данные из источников любого рода. Источники могут находиться на вашем собственном веб-сайте (скажем, свежие пять статей), или быть веб-службами погоды, биржевых котировок, Amazon или eBay.

Модуль содержит бизнес-логику и пользовательский интерфейс. Он не имеет собственного раздела администрирования, как большинство компонентов. Модули могут, однако, управляться параметрами. Шаблоны вашего веб-сайта обращаются к различным модулям непосредственно и интегрируют их.

Поскольку модули — это самодостаточные программы, они могут выполнять отдельные задачи в домене шаблона, такие как отображение баннера. Шаблон не делает ничего кроме группирования нескольких разных модулей в визуально привлекательной манере. Структура модуля позволяет легко и просто расширять веб-сайт.

Поскольку шаблоны можно использовать и для веб-сайта, и для администрирования Joomla!, для этих шаблонов различные модули также доступны. Вы администрируете модули в диспетчере модулей, доступном через меню Extensions⇒Module Manager (Расширения⇒Диспетчер модулей) (рис. 10.3).

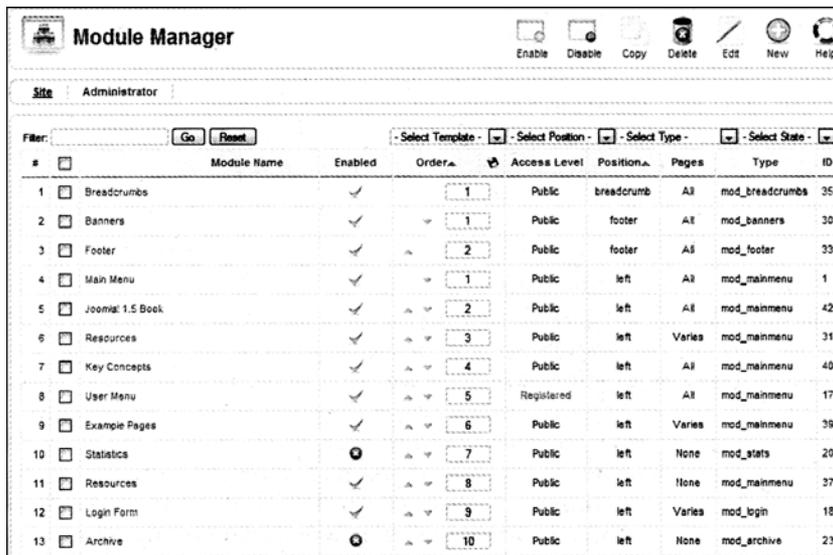


Рис. 10.3. Страница Module Manager

Колонки в списке диспетчера модулей описаны ниже.

- **Module Name** (Имя модуля). Имя модуля и заголовок веб-сайта.
- **Enabled** (Включен). Включен модуль или нет.
- **Order** (Порядок). Щелкая на стрелках зеленого цвета, можно изменять порядок модулей. Подобным образом можно, например, разместить меню Joomla! 1.5 Book выше или ниже главного меню. Можно также непосредственно указать позицию в поле ввода и щелкнуть на пиктограмме диска возле слова Order; это избавит от утомительных многократных щелчков на стрелках.
- **Access Level** (Уровень доступа). Здесь задаются права доступа для модуля (Public, Registered или Special).
- **Position** (Позиция). Сообщает шаблону, где необходимо отображать модуль. Доступны следующие стандартные позиции, определенные в шаблоне:
 banner (раздел рекламы)
 left (левая сторона)
 right (правая сторона)
 top (вверху)
 user1 (определяемое пользователем 1)
 user2 (определяемое пользователем 2)
 user3 (определяемое пользователем 3)
 user4 (определяемое пользователем 4)

В дополнение к этим позициям можно фильтровать отображение с помощью списка опций в верхней области, добиваясь лучшего общего вида. Далее в главе вы узнаете больше о позиционировании в шаблоне.

- **Pages (Страницы).** Модуль может отображаться на всех страницах или только на определенных.
- **Type (Тип).** Существуют разные типы модулей. Тип `mod_mainmenu`, например, соответствует главному меню. Индивидуальные меню различаются только их параметрами. С помощью списка опций вверху вы можете фильтровать отображение соответственно этим типам для лучшего обзора.
- **ID (Идентификатор).** Номер набора данных из базы.

Фильтры модулей

Шаблон обладает многочисленными опциями фильтрации, с помощью которых можно индивидуализировать отображение. Эти механизмы фильтрации невероятно полезны, и Joomla! по умолчанию включает более 20 таких механизмов.

Большинство этих модулей обрабатываются одинаковым образом. Помимо ввода имени, информации доступа и т.д. вы должны также решить, на каких страницах и в каком месте шаблона должен отображаться модуль.

Список параметров особенно важен для модулей. Мы опишем каждый из них по мере использования. Вы можете копировать модули по своему желанию и отображать их на веб-сайте с модифицированными параметрами и заголовками в разных позициях.

Модули сайта

Breadcrumbs

Breadcrumbs (меню указаний, или "хлебные крошки") — это иерархические индикаторы в статьях (рис. 10.4).

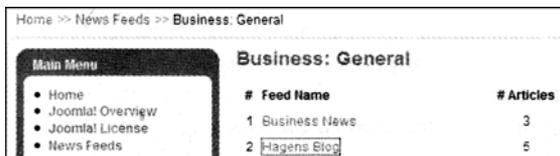
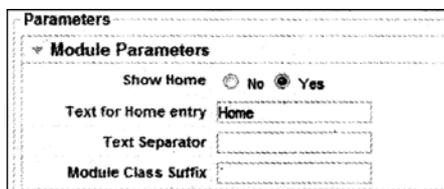


Рис. 10.4. Пример применения *breadcrumbs*

Индивидуальные соподчиненные элементы статьи, раздел **News** и категория **Latest** связаны между собой и потому предоставляют посетителю возможности для ориентации и навигации. Вы можете определить в качестве параметра, должна ли отображаться домашняя/титовая страница (рис. 10.5). Если деактивизировать этот модуль, меню указаний исчезнут.



Footer

Этот модуль отображает информацию о лицензии Joomla! в нижнем колонтитуле (footer). Вы можете деактивизировать его, и тем самым подавить эту информацию на титульной странице.

Banner

Этот модуль управляет отображением баннеров. Помимо знакомых установок деталей и размещения на странице, вы можете также конфигурировать перечисленные ниже параметры (рис. 10.6).

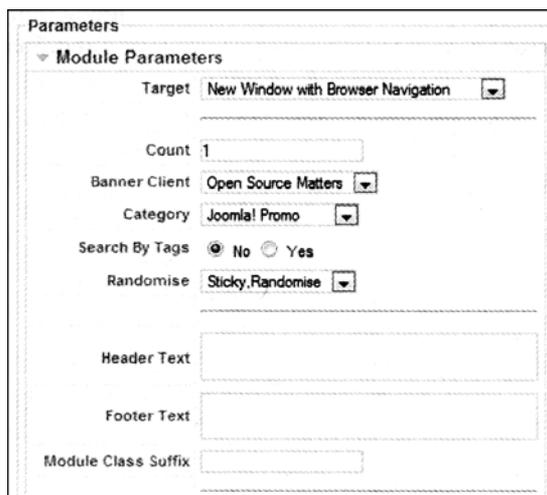


Рис. 10.6. Параметры модуля Banner

- Target (Цель). Где должен отображаться целевой URL— в том же окне или новом, с кнопками навигации или без.
- Count (Счетчик). Количество показов.
- Banner Client (Клиент баннера). Выберите здесь клиента, если данный модуль предназначен для показа баннеров только одного клиента.
- Category (Категория). Если должны отображаться только баннеры одной определенной категории (например, только текстовые ссылки), выберите соответствующую категорию.
- Search By Tags (Поиск по тегам). Индивидуальным баннерам могут быть назначены теги (категории). Здесь можно решить, хотите ли вы отображать баннеры, например, только с тегом Joomla!.
- Randomise (Произвольный порядок). С помощью этого параметра можно управлять ротацией баннеров. Sticky, Ordering означает, что баннеры, помеченные как sticky, отображаются первыми, а далее порядок сортировки соответствует определенному в баннере. Sticky, Randomise означает, что баннеры, помеченные как sticky, отображаются первыми, а затем — в произвольном порядке.
- Header Text (Текст заголовка). Текст перед баннером.

- Footer Text (Текст нижнего колонтитула). Текст после баннера.
- Module Class Suffix (Суффикс класса модуля). Вы можете ввести суффикс, который добавляется к имени класса CSS. Предположим, что вы сохранили в поле только `table`. Это превращается в класс `module_table`, который затем активизируется. Далее потребуется реализовать ваш класс в соответствующем CSS-файле шаблона.

Main Menus

Внутреннее назначение `mod_mainmenu` используется для всех модулей меню. Меню Joomla! 1.5 Book, которое мы создали, имеет тип `main_menu`. Бывают вертикальные меню (Main Menu) и горизонтальные меню (Top Menu). Для вертикальных меню также имеется опция `flat list` (плоский список). Плоский список — это просто листинг индивидуальных элементов. В параметрах модуля задаются наиболее важные установки (рис. 10.7).

Рис. 10. 7. Параметры модуля Main Menus

- Menu Name (Имя меню). Имя меню, назначенное модулю.
- Menu Style (Стиль меню). Вертикальное (Main Menu), горизонтальное (Top Menu) или плоский список (List).
- Start Level (Начальный уровень). Уровень вложенности, на котором это меню должно начинаться.
- End Level (Конечный уровень). Уровень вложенности, который должно включать меню для своего последнего пункта.
- Always show sub-menu items (Всегда показывать элементы подменю). Вы можете держать пункты меню всегда открытыми, даже если выполнен щелчок на другом элементе. Эта функция имеет смысл только для вложенных структур меню.
- Target Position (Позиция цели). Введенные здесь значения используются со всеми пунктами меню, которые имеют установки в новом окне браузера без меню для атрибута `On Click, Open in` (При щелчке открыть в). Например: `top=10, left=10, width=200, height=300`.

С помощью унаследованных параметров (Legacy Parameters) пиктограммы меню гут отображаться в стиле Joomla! 1.0.

- Show Menu Icons (Показывать пиктограммы меню). Должны ли отображаться пиктограммы меню. Отображение пиктограмм в каждом случае зависит от активного шаблона.
- Full Active Highlighting (Полная подсветка активных). Этот параметр ссылается на родительские элементы, указывая на то, следует ли их помечать как активные. Обычно это делается только для индивидуальной ссылки.



Пометка как активной более чем одной ссылки противоречит стандарту HTML.

- Menu Icon Alignment (Выравнивание пиктограмм меню). Где должны располагаться пиктограммы меню – слева или справа.
- Indent Image (Изображение иерархии). Можно специфицировать элемент для отображения с подструктурами пункта меню. Доступен выбор использовать значения Joomla! по умолчанию, применить отдельное изображение для каждого уровня иерархии, или же не отображать никаких пиктограмм вообще.
- Indent Image (1-6) (Изображение иерархии (1-6)). Можно определить шесть пиктограмм для шести шагов иерархии.
- Spacer (Разделитель). Здесь определяется символ-разделитель, который должен отображаться между пунктами горизонтального меню.
- End Spacer (Завершающий символ). Здесь определяется завершающий символ, отображаемый в конце элементов горизонтального меню.

Statistics

По умолчанию модуль Statistics отключен. Если вы активизируете его, то должны определить страницы, на которых он будет отображаться. Модуль отображает информацию о вашем сервере.

- Server Info (Информация о сервере). Установите это для отображения серверной информации.
- Site Info (Информация о сайте). Установите это для отображения информации о веб-сайте.
- Hit Counter (Счетчик попаданий). Здесь можно активизировать или деактивизировать счетчик посещений.
- Increase Counter (Увеличить счетчик). Здесь можно установить начальное значение счетчика посещений.

Login Form

Существуют два доступных представления модуля входа (login module). Если посетители еще не вошли на сайт, они получают форму входа, в которой могут вводить свое имя и пароль. В зависимости от установок Site⇒Global Configuration⇒Site (Сайт⇒Глобальная конфигурация⇒Сайт) можно также регистрировать новых пользователей. Форма Login Form (Форма входа) показана на рис. 10.8.

Рис. 10.8. Форма Login Form

Рис. 10.9. Форма Login Form после успешного входа

После успешного входа отображение изменяется, предоставляя возможность выхода (рис. 10.9).

Параметры включают перечисленные ниже опции.

- Caching (Кэширование). Содержимое меню можно кэшировать для экономии времени загрузки.
- Module Class Suffix (Суффикс класса модуля). Здесь может быть задан специальный класс CSS для визуальной конфигурации меню.
- Pre-text (Предварительный текст). Введенный здесь текст отображается перед тем, как форма оказывается в режиме входа.
- Post-text (Окончательный текст). Введенный здесь текст отображается после того, как форма покинет режим входа.
- Login Redirection URL (URL переадресации входа). Здесь указывается URL-адрес, по которому пользователь перенаправляется после успешного входа.
- Logout Redirection URL (URL переадресации выхода). Здесь указывается URL-адрес, по которому пользователь перенаправляется после выхода.
- Greeting (Приветствие). После успешного входа внешний вид модуля изменяется и отображает приветствие, а также кнопку Logout (Выход). Вы должны решить, отображать ли текст "Hi, имя_пользователя".

Archive

По умолчанию модуль Archive отключен. Если вы активизируете его, то должны определить страницы, на которых он отображается. Этот модуль показывает информацию о содержимом вашего архива. Информация сгруппирована по месяцам.

С помощью параметра Count (Количество) указывается число отображаемых элементов.

Sections

По умолчанию модуль Sections отключен. Этот модуль отображает различные разделы веб-сайта. Если вы активизируете его, то должны определить страницы, на которых он будет отображаться.

С помощью параметра Count (Количество) указывается число отображаемых разделов.

Related Items

Модуль Related Items (Схожие статьи) отображает заголовки других статей, которые связаны с данной.

Отношения между статьями основаны на ключевых словах, которые вводились в метаданных. Все ключевые слова текущей отображаемой статьи сравниваются со всеми ключевыми словами других опубликованных статей.

Если, например, вы ввели `test` в качестве ключевого слова в Joomla! License Guidelines (Принципы лицензирования Joomla!), и это же слово присутствует в новой метке новостей, такая заметка и условия лицензирования отображаются в Related Items, когда запрашивается один из двух материалов. Можно также установить параметр, чтобы отображалась дата создания связанной статьи.

Wrapper

Модуль Wrapper (Упаковка) связывает внешний, созданный не в Joomla! контент, внутри так называемого фрейма `iframe`. HTML-дескриптор `iframe` создает прокручиваемую область внутри веб-сайта. С помощью этого модуля в контент Joomla! можно интегрировать целые веб-сайты, расположенные на других серверах. На рис. 10.10 показан пример PDA-портала Google.

Ниже перечислены некоторые параметры.

- URL (URL-адрес). Укажите здесь URL-адрес желаемого веб-сайта.
- Scroll Bars (Линейки прокрутки). Если необходимо настроить линейки прокрутки для отображения в `iframe`, имеется выбор между Yes (Да), No (Нет) и Auto (Автоматически) — для автоматической активизации.
- Width/Height (Ширина/высота). Ширина и высота `iframe` в процентном отношении или в пикселях.
- Auto Height (Автоматическая высота). Установите в Yes (Да), если хотите, чтобы высота подгонялась автоматически.
- Auto Add (Автоматическое добавление). По умолчанию перед URL помещается `http://`, если не указано явно `http://` или `https://`. Этот переключатель можно включить или выключить.



Рис. 10.10. PDA-портал Google внутри модуля Wrapper

Feed Display

С помощью модуля Feed Display (Отображение лент) вы можете отображать ленты новостей на веб-сайте (рис. 10.11).

Ниже перечислены некоторые параметры.

- Feed URL (URL-адрес ленты). Вы можете вводить здесь URL-адрес нужной ленты новостей.
- RTL Feed (Лента справа налево). Здесь можно установить прокрутку справа налево.
- Feed Title (Заголовок ленты). Здесь задается отображаемый заголовок ленты новостей.
- Feed Description (Описание ленты). Описание ленты новостей.

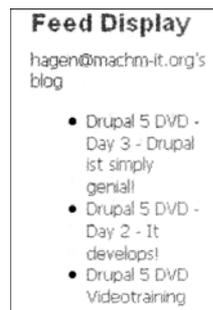


Рис. 10.11. Модуль Feed Display

- Feed Image (Изображение ленты). Здесь можно установить логотип ленты новостей.
- Items (Элементы). Количество отображаемых в ленте статей.
- Item Description (Описание элементов). Отображать текст статей.
- Word Count (Количество слов). Здесь определяется количество отображаемых слов в описании статьи.



Рис. 10.12. Модуль Who's Online

Who's Online

Модуль Who's Online (Кто в онлайн?) отображает список пользователей, находящихся в данный момент на сайте (рис. 10.12). Модуль различает гостей и зарегистрированных пользователей.

Параметр Display (Отображать) позволяет принять решение, касающееся отображения модуля. Можно выбрать один из следующих режимов:

- количество гостей и членов;
- имена зарегистрированных пользователей;
- комбинация предыдущих двух.

Polls

Здесь глобально включается или выключается функциональность для отображения опросов. Сами опросы конфигурируются в компоненте Polls (доступном через меню Components⇒Polls (Компоненты⇒Опросы)). Опрос определяется с помощью параметров модуля.

Advertisement

Этот модуль является копией модуля Banner. По умолчанию он имеет дело с баннерами, которые относятся к категории текстовых и содержат текстовые ссылки.

Random Image

Модуль Random Image (Случайное изображение) показывает случайно выбранные изображения из указанной папки. Этот модуль по умолчанию активизируется, но странице не назначается, поэтому размещать его на страницах необходимо явно.

Ниже перечислены некоторые параметры.

- Image Type (Тип изображения). Здесь задается тип изображения (jpg/png/gif). Допускается выбирать только один тип за раз.
- Image Folder (Папка изображений). Здесь необходимо указать каталог, где находятся изображения. Для данного примера выбран images/stories без ведущего слэша.
- Link (Ссылка). Если вы введете здесь URL-адрес, изображение будет реагировать на щелчки. Целью ссылки будет этот указанный URL.
- Width (px)/Height (px) (Ширина (в пикселях)/Высота (в пикселях)). Ширина и высота отображаемого изображения в пикселях. Если вы не определяете их, изображение отображается с максимально хорошими характеристиками.

Syndication

Лента новостей веб-сайта доставляется модулем Syndication (Синдикация). Поддерживаемые стандарты можно определить в его параметрах. Статьи на титульной странице представляют содержимое ленты новостей (рис. 10.13).



Рис. 10.13. Представление ленты новостей на титульной странице

Newsflash

Модуль Newsflash (Экстренное сообщение) отображает случайные фрагменты текста для представления ваших статей (рис. 10.14).

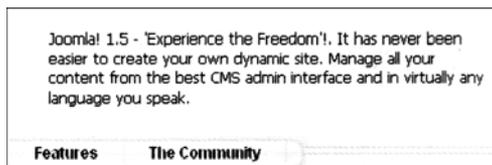


Рис. 10.14. Модуль Newsflash

Ниже перечислены параметры этого модуля.

- Category (Категория). Выберите из списка категорий одну определенную или же укажите все категории.
- Layout (Компоновка). Здесь можно сделать выбор между горизонтальным и вертикальным представлением.
- Show Images (Показывать изображения). Должны ли отображаться изображения, включенные в контент.
- Title Linkable (Заголовки в виде ссылок). Здесь можно указать, должны ли заголовки также быть ссылками на целые статьи.
- Read More...Link (Ссылка "Читать далее..."). Активизировать ссылку Read More или нет.
- Article Title (Заголовок статьи). Здесь можно указать, отображать заголовок статьи или нет.
- # of Articles (Количество статей). Здесь задается количество одновременно отображаемых статей.

Latest News

Модуль Latest News (Последние новости) отображает последние (новейшие) статьи (рис. 10.15). По умолчанию он устанавливается в позицию user1. Однако ему можно назначить и другую позицию, например, разместить справа.

Параметры этого модуля включают перечисленные ниже.

- Count (Количество). Количество отображаемых элементов.
- Order (Порядок). Должны отображаться новейшие или же недавно модифицированные статьи.



Рис. 10.12. Модуль Latest News

- **Authors (Авторы).** Позволяет ограничить статьи определенным автором.
- **Front Page Articles (Статьи на титульной странице).** Должны ли включаться статьи, присутствующие на титульной странице.
- **Section ID (Идентификатор раздела).** Если вы вводите номера наборов данных разделов для отображения в разделенном запятыми списке, то можете ограничить выбор контента только этими разделами.
- **Category ID (Идентификатор категории).** Если вы вводите номера наборов данных категорий для отображения в разделенном запятыми списке, то можете ограничить выбор контента только этими категориями.

Popular

Модуль Popular (Популярное) отображает наиболее популярные статьи (рис. 10.16).

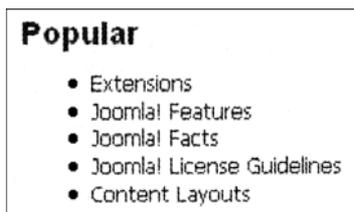


Рис. 10.16. Модуль Popular

Параметры – в точности те же, что и у модуля Latest News.

Search

В шаблоне по умолчанию модуль Search (Поиск) предназначен только для ввода (рис. 10.17). Этот шаблон не изменяет параметров.

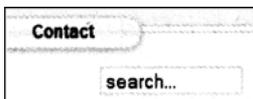


Рис. 10.17. Модуль Search

Ниже перечислены параметры этого модуля.

- **Box Width (Ширина поля).** Размер текстового поля в символах. В данном примере – 30 символов.
- **Text (Текст).** Текст, который будет отображаться в поисковом поле.
- **Search Button (Кнопка поиска).** Здесь можно указать, нужна кнопка поиска или нет.
- **Button Position (Позиция кнопки).** Если выбрано представление кнопки поиска, здесь задается ее положение (справа, слева, вверх, вниз).
- **Search Button as Image (Кнопка поиска в виде изображения).** В качестве кнопки поиска можно использовать графическое изображение.
- **Button Text (Текст на кнопке).** Здесь задается текст, отображаемый на кнопке поиска.

Копирование модуля

Предположим, что необходимо вывести два случайных изображения. Один модуль отображает изображения из каталога А, а другой должен получать изображения из каталога В. В подобных случаях следует выбрать модуль Random Image, отметив флажок перед ним, а затем щелкнуть на пиктограмме Copy (Копировать).

В списке появится новый модуль по имени Copy of Random Image (Копия Random Image). Измените информацию должным образом, и вы получите новый модуль (рис. 10.18).

Если вы деактивируете модули Latest News и Popular и переключите два модуля изображений в позиции user1 и user2, новые модули отобразятся в разделе контента шаблона над новостями и/или на титульной странице.



Рис. 10.18.
Два модуля
Random Image

Модули администратора

На вкладке Administrator (Администратор) диспетчера модулей представлен структурно идентичный раздел с модулями, однако, предназначенный для административной области (рис. 10.19).

#	Module Name	Enabled	Order	Position	Pages	Type	ID
1	Welcome to Joomla!	✓	1	cpanel	None	mod_custom	41
2	Logged in Users	✓	2	cpanel	None	mod_logged	10
3	Popular	✓	3	cpanel	None	mod_popular	3
4	Recent added Articles	✓	4	cpanel	None	mod_latest	4
5	Menu Stats	✓	5	cpanel	None	mod_stats	5
6	Footer	✓	0	footer	None	mod_footer	11
7	Unread Messages	✓	1	header	None	mod_unread	6
8	Online Users	✓	2	header	None	mod_online	7
9	Quick Icons	✓	1	icon	None	mod_quickicon	9
10	Login	✓	1	login	None	mod_login	2
11	Admin Menu	✓	1	menu	None	mod_menu	12
12	User Status	✓	1	status	None	mod_status	14

Рис. 10.19. Модули администратора в диспетчере модулей

Logged in Users

Модуль Logged in Users (Вошедшие пользователи) отображает список текущих активных пользователей на сайте, в позиции cpanel, а также на вкладке в панели управления (Control Panel). Модуль показан на рис. 10.20.

Welcome to Joomla!					
Logged in Users					
#	Name	Group	Client	Last Activity	Logout
1	admin	Super Administrator	administrator	0.0 hours ago	
Popular					
Recent added Articles					
Menu Stats					

Рис. 10.20. Модуль Logged in Users

Popular

Модуль Popular (Популярное) представляет список наиболее часто посещаемого контента в виде вкладки в панели управления.

Recent added Articles

Модуль Recent added Articles (Недавно добавленные статьи) отображает самые последние опубликованные статьи в виде вкладки в панели управления. Статьи, которые были опубликованы на титульной странице, в список не включаются.

Menu Stats

Модуль Menu Stats (Статистика по меню) отображает статистику о конфигурации индивидуальных элементов меню в виде вкладки в панели управления.

Footer

Модуль Footer (Нижний колонтитул) отображает информацию об авторских правах Joomla! в административной части.

Unread Messages

Модуль Unread Messages (Непрочитанные сообщения) информирует о количестве непрочитанных административных сообщений в позиции header (т.е. вверху справа).

Online Users

Модуль Online Users (Пользователи в онлайн) отображает счетчик текущих зарегистрированных пользователей на сайте в позиции header.

Quick Icons

Модуль Quick Icons (Пиктограммы быстрого доступа) предоставляет быстрый доступ к панели управления (Control Panel).

Login

Модуль Login (Вход) отображает форму входа в административную область. Не отключайте его!

Admin Menu

Модуль Admin Menu (Меню администрирования) отображает JavaScript-элементы навигации в административной части. Не отключайте его!

User Status

Модуль User Status (Состояние пользователей) полностью включает или отключает информационный раздел, находящийся вверху справа. Не отключайте его!

Admin Submenu

Модуль Admin Submenu (Подменю администрирования) отключает раздел, в котором отображаются вкладки. Не отключайте его!

Title

Модуль Title (Заголовок) определяет отображение и пиктограмму в левой части панели инструментов. Не отключайте его!

Toolbar

Модуль Toolbar (Панель инструментов) определяет отображение панели инструментов. Не отключайте его!

CSS Admin Menu

Модуль CSS Admin Menu (Меню администрирования CSS) отвечает за отображение административного меню Joomla!.

Диспетчер подключаемых модулей

Подключаемые модули можно сравнить с языком сценариев Joomla!. Вы уже имели дело с интегрированным подключаемым модулем, когда работали с редактором TinyMCE.

Подключаемые модули всегда относятся к определенному типу. На данный момент подключаемые модули принадлежат к разделам authentication, content, editors, editors-xtd, search, system и xmlrpc.

Активизировать и деактивизировать индивидуальные подключаемые модули можно в диспетчере подключаемых модулей (Plugin Manager), страница которого показана на рис. 10.21. За небольшими исключениями, подключаемые модули имеют мало или вообще не имеют изменяемых установок, поскольку в основном запрограммированы и оптимизированы для весьма специфических целей.

Подключаемые модули аутентификации

В Joomla! 1.5 пользователи могут аутентифицировать себя множеством разных способов. Это обладает тем преимуществом, что можно работать, минуя утомительный процесс регистрации на веб-сайте, и не нужно помнить другое имя пользователя и другой пароль. Для того чтобы можно было применять разнообразные методы аутентификации, необходимо опубликовать подключаемый модуль.

Модуль Authentication — Joomla!

"Нормальной" аутентификации предшествует предварительная регистрация на веб-сайте.

#	Plugin Name	Enabled	Order	Access Level	Type	File	ID
1	Authentication - Joomla	✓	1	Public	authentication	joomla	1
2	Authentication - LDAP	⊗	2	Public	authentication	ldap	2
3	Authentication - OpenID	⊗	3	Public	authentication	openid	4
4	Authentication - Gmail	⊗	4	Public	authentication	gmail	3
5	Content - Page Navigation	✓	2	Public	content	pagenavigation	17
6	Content - Rating	✓	4	Public	content	vote	13
7	Content - Email Cloaking	✓	5	Public	content	emailcloak	14
8	Content - Code Highlighter (GeSHi)	⊗	5	Public	content	geshi	15
9	Content - Load Module	✓	6	Public	content	loadmodule	16
10	Content - Pagebreak	✓	10000	Public	content	pagebreak	12
11	Editor - No Editor	✓	0	Public	editors	none	18
12	Editor - TinyMCE 2.0	✓	0	Public	editors	tinymce	19
13	Editor - XStandard Lite 2.0	⊗	0	Public	editors	xstandard	20
14	Editor Button - Image	✓	0	Public	editors-xtl	image	21

Рис. 10.21. Страница Plugin Manager

Модуль Authentication — LDAP

LDAP — это сетевой протокол, используемый в так называемых службах каталогов. Она обеспечивает коммуникации между клиентом LDAP (в нашем случае — веб-сайтом Joomla!) и каталогом, из которого читаются персональные данные. Каталоги LDAP популярны в крупных компаниях, и потому имеет смысл организовать доступ к персональным данным через корпоративную сеть.

С подключаемым модулем Authentication — LDAP связаны многочисленные параметры. Установка параметров достаточно очевидна.

Модуль Authentication — OpenID

Идея, положенная в основу OpenID, состоит в том, что пользователь, имеющий учетную запись на сервере OpenID, может входить на ваш веб-сайт, не заводя на нем учетной записи с паролем. В зависимости от ориентации вашего веб-сайта, это может оказаться значительным преимуществом.

Модуль Authentication — Gmail

Посредством службы Gmail компания Google предоставляет вариант аутентификации с использованием адреса электронной почты. Если у пользователей имеются учетные записи Gmail, они могут входить на ваш веб-сайт, указывая свои пользовательские идентификаторы (ID) Gmail. В зависимости от ориентации вашего веб-сайта, это может быть большим преимуществом.

Подключаемые модули контента

Модуль Content — Image

Подключаемый модуль Content — Image (Контент — изображение) сохранен для совместимости с версиями Joomla! 1.0.x. Он интерпретирует команду {mosimage} в тексте статьи и затем выводит в этом месте назначенное изображение. Кардинальное преимущество состоит в том, что эти изображения можно легко редактировать с помощью подключаемого модуля. Внешнее и внутреннее поле изображения задается параметрами.

Модуль Content — Page Navigation

Подключаемый модуль Content — Page Navigation (Контент — страничная навигация) отвечает за функции Next (Следующая) и Previous (Предыдущая) под статьями. Перед использованием этот модуль должен быть активизирован.

Модуль Content — SEF

SEF представляет собой сокращение от "Search Engine Friendly" (дружелюбность с поисковыми механизмами). Этот подключаемый модуль создает дружелюбные к механизмам поиска URL (SEF URL) для элементов контента. Перед использованием этот модуль должен быть активизирован.

Модуль Content — Rating

Подключаемый модуль Content — Rating (Контент — рейтинг) отображает панель рейтинга над контентом. Перед использованием этот модуль должен быть активизирован.

Модуль Content — Email Cloaking

Подключаемый модуль Content — Email Cloaking (Контент — маскировка адреса электронной почты) трансформирует адрес электронной почты, который вы вводите в элементе контента в виде name@example.com, в ссылку, и скрывает действительный адрес электронной почты посредством JavaScript. Это дает то преимущество, что программам сбора почтовых адресов прочесть такой адрес оказывается нелегко.

Модуль Content — Code Highlighter (GeSHi)

Подключаемый модуль Content — Code Highlighter (GeSHi) (Контент — подсветка синтаксиса (GeSHi)) форматирует код. Однако, кроме того, GeSHi может выполнять подсветку синтаксиса и создавать впечатляющий листинг на веб-сайте, если вы встроите код, подлежащий форматированию, в HTML-дескрипторы `<pre>` `</pre>`:

```
<pre>
if ($number > 0){
  echo $number;
} else{
  $number++;
}
</pre>
```

Модуль Content – Load Module

Подключаемый модуль Content – Load Module (Контент – загрузить модуль) позволяет загружать модули внутри статей. Он может быть вызван, например, конструкцией {loadposition user1}.

Модуль Content – Pagebreak

Подключаемый модуль Content – Pagebreak (Контент – разрыв страницы) занимается разбиением страниц Joomla! 1.0.x в статьях. Подобно подключаемому модулю Content – Image, его легко интегрировать в контент. Помимо простого разбиения страниц, он позволяет определить разнообразные заголовки страниц. В Joomla! 1.5 эту роль исполняет диалог pagebreak (см. раздел "Изображения" в главе 8).

Ниже показан синтаксис:

```
hr title="page title" alt=" page title" class="system-pagebreak">
```

Подключаемые модули редакторов

Модуль Editor – No Editor

Подключаемый модуль Editor – No Editor (Редактор – нет редактора) следует активизировать, если необходимо предоставлять поля textarea без редактора.

Модуль Editor – TinyMCE 2.0

Подключаемый модуль Editor – TinyMCE 2.0 (Редактор – TinyMCE 2.0) должен быть активизирован, если необходимо предоставлять поля textarea с редактором TinyMCE 2.0.

Подключаемые модули Editors-xtd

Тип Editors-xtd содержит в себе подключаемые модули Editor Button – Image (Редактор – кнопка вставки изображения), Editor Button – Pagebreak (Редактор – кнопка вставки разрыва страницы) и Editor Button – Read more (Редактор – кнопка вставки Read more). Эти три подключаемых модуля генерируют три кнопки под окном редактора. Щелчки на этих кнопках вызывают диалоги Insert Image (Вставить изображение), Pagebreak (Разрыв страницы) и Read more (Читать далее).

Подключаемые модули поиска

При желании можно активизировать подключаемые модули поиска: Search – Contents (Поиск – контент), Search – Weblinks (Поиск – веб-ссылки), Search – Contacts (Поиск – контакты), Search – Categories (Поиск – категории), Search – Sections (Поиск – разделы) и Search – Newsfeeds (Поиск – ленты новостей). Они управляют функцией поиска модуля Search. Эти подключаемые модули должны быть активизированы, если необходимо получать результаты поиска от соответствующих разделов. Если вы хотите искать в дополнительных компонентах, то для них также предусмотрены соответствующие подключаемые модули.

Системные подключаемые модули

Модуль System — Log

Подключаемый модуль System — Log (Система — журнал) делает доступным системный журнал. Вы можете указать местоположение файла журнала. Ниже приведен пример этого файла.

```
#Version: 1.0
#Date: 2007-10-29 23:39:56
#Fields: date time level c-ip status comment
#Software: Joomla! 1.5.0 Production/Stable [ Takriban ] 5-October-2007 21:00 GMT
2007-10-29 23:39:56 - 127.0.0.1 - stories/franee
2007-10-30 12:00:12 - 127.0.0.1 FAILURE: Invalid password
2007-10-31 13:44:11 - 127.0.0.1 FAILURE: Invalid password
```

Модуль System — Debug

Модуль System — Debug (Система — отладка) обеспечивает доступ к функциям отладки. Вы можете конфигурировать параметры для определения отображаемой информации.

Модуль System — Legacy

О подключаемом модуле System — Legacy (Система — унаследованная) велись и продолжают проводиться жаркие дискуссии. Это элемент, связывающий с миром Joomla! 1.0. Активизация этого подключаемого модуля существенно облегчает переход от Joomla! 1.0 на Joomla! 1.5.

Модуль System — Cache

Подключаемый модуль System — Cache (Система — кэш) устанавливает операции с кэшем. Вы можете определить кэш браузера на стороне клиента и установить базовое время кэширования в минутах.

Модуль System — Remember Me

Модуль System — Remember Me (Система — запомнить меня) представляет собой метод локального сохранения данных доступа в cookie-наборе. При повторном заходе на веб-сайт ваши данные уже будут находиться в форме. Такое сохранение возможно только при явной отметке соответствующего флажка.

Модуль System — Backlink

Подключаемый модуль System — Backlink (Система — обратная ссылка) выполняет задачу корректной трансформации "старых" ссылок Joomla! 1.0 в логику Joomla! 1.5.

Подключаемые модули пользователя

Модуль User — Joomla!

Подключаемый модуль User — Joomla! (Пользователь—Joomla!) создает пользователя в таблицах базы данных после первой успешной аутентификации.

Подключаемые модули XML-RPC

Интерфейс XML-RPC предоставляет возможность удаленного управления Joomla!.

Модуль XML-RPC – Joomla!

Этот подключаемый модуль включает контроль специфичных функций каркаса Joomla! из интерфейса XML-RPC.

Модуль XML-RPC – bloggerAPI

Подключаемый модуль XML-RPC – blogger API (XML-RPC – API блоггера) включает интеграцию контента из других платформ, например, фотографий Flickr. На данный момент поддерживается интерфейс блоггера. Множество платформ Web 2.0 интегрируют MetaWeblog API. Это расширение также будет доступно в Joomla! 1.5. Вы можете выбрать раздел и категорию, которые должны быть вставлены в эти статьи в качестве параметров.

Диспетчер шаблонов

Вы уже ознакомились с шаблонами в главе 4, где конфигурировались разные шаблоны для веб-сайта. Joomla! различает шаблоны сайта (Site) и администратора (Administrator). Страница Template Manager (Диспетчер шаблонов) показана на рис. 10.22.

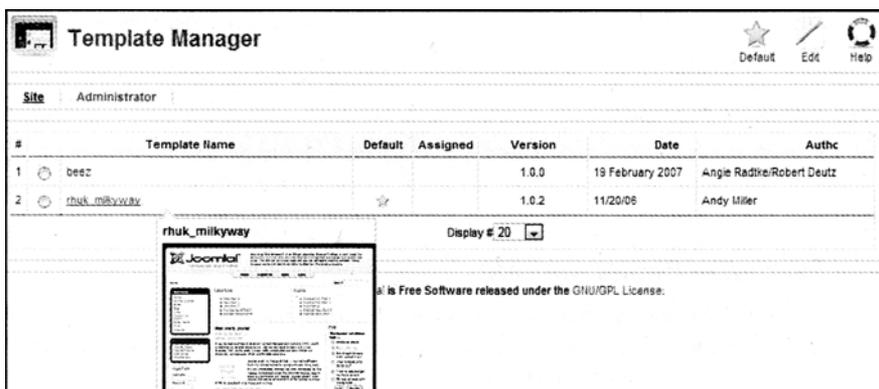


Рис. 10.22. Страница Template Manager

Шаблоны сайта

На вкладке Site (Сайт) перечислены установленные шаблоны. Вы можете задать один шаблон в качестве шаблона по умолчанию и присвоить другие шаблоны определенным ссылкам меню. Чтобы сделать шаблон шаблоном по умолчанию, отметьте флажок рядом с нужным шаблоном и щелкните на пиктограмме Default (По умолчанию). После этого веб-сайт соответствующим образом изменит внешний вид.

Редактирование шаблона

После щелчка на шаблоне можно модифицировать и конфигурировать все его свойства. Помимо прочего, список параметров покажет ссылки меню, которые размещены в этом шаблоне.

Отметьте ссылки меню, которым вы хотите назначить шаблон. Допускается делать множественный выбор, щелкая на нужных элементах один за другим и удерживая при этом клавишу <Ctrl>. Можно также внести изменения в цвета шаблона и ширину отображения (рис. 10.23).

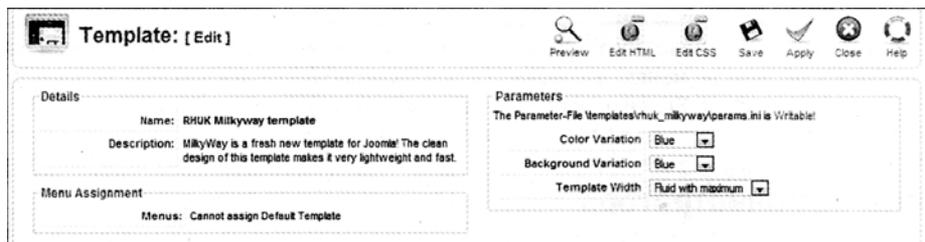


Рис. 10.23. Страница редактирования шаблона

Пиктограмма Preview

Щелчок на пиктограмме Preview (Предварительный просмотр) покажет предварительный вид шаблона по умолчанию для вашего веб-сайта с графическим указанием размещения модулей. В указанных местах можно размещать модули.

Пиктограмма Edit HTML

С помощью пиктограммы Edit HTML (Редактировать HTML) можно редактировать исходный код HTML выбранного шаблона. Шаблоны всегда состоят из файла HTML. В верхнем его разделе находится группа фрагментов шаблона. Эти фрагменты представляют собой заранее заданные команды для определенных действий над шаблонами. Редактор HTML-кода шаблона показан на рис. 10.24.

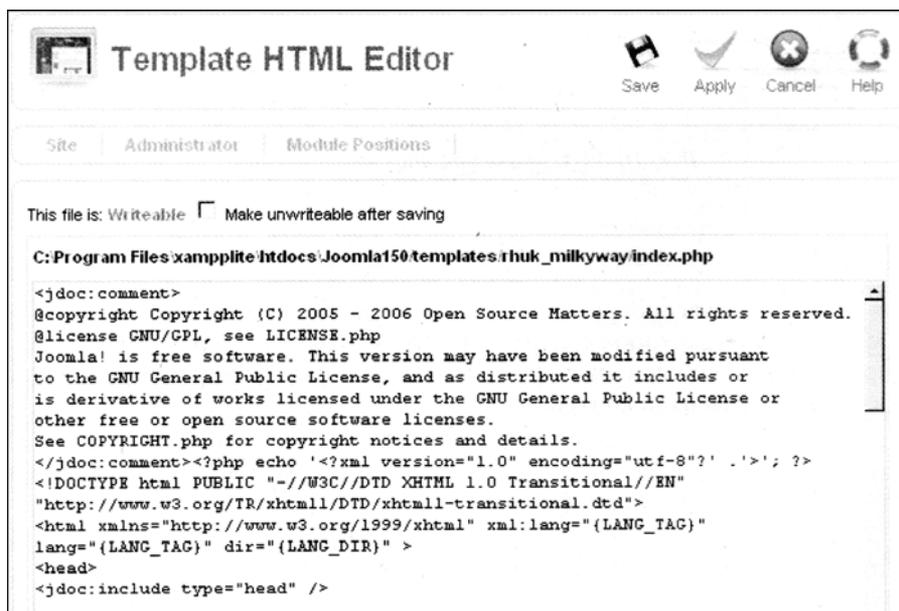


Рис. 10.24. Редактор HTML шаблона

Пиктограмма Edit CSS

С помощью пиктограммы Edit CSS (Редактировать CSS) можно редактировать исходный код CSS выбранного шаблона. Шаблоны могут иметь любое количество файлов CSS. Выберите нужный файл и щелкните на пиктограмме Edit CSS.



Вы должны четко представлять себе, что собираетесь делать. Здесь необходимы знания HTML и CSS. Тем не менее, это может быть интересно и для начинающих — посмотреть, как устроен шаблон. Более подробную информацию на эту тему ищите в главе 13.

Шаблоны администратора

Все, что справедливо для вашего веб-сайта, разумеется, также справедливо и для интерфейса администрирования. Административному интерфейсу можно назначить другие шаблоны — точно так же, как это делается с другими шаблонами сайта. В инсталляцию Joomla! входит один шаблон по умолчанию.

Диспетчер языков

Тема языков уже затрагивалась в главе 4. В диспетчере языков (рис. 10.25) также имеются вкладки Site (Сайт) и Administrator (Администратор), на которых можно выбирать языки для веб-сайта и административного интерфейса.

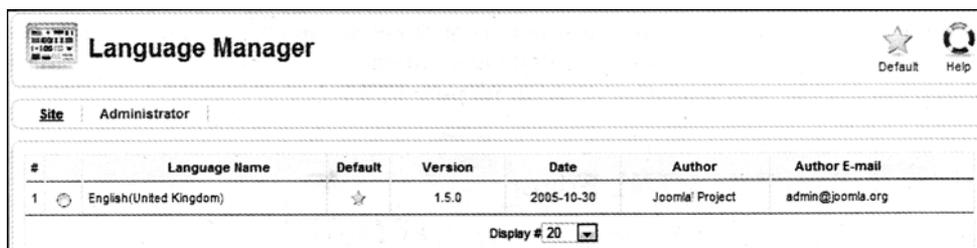


Рис. 10.25. Страница Language Manager

Резюме

В этой главе было рассмотрено меню Extensions. В следующей главе речь пойдет о меню Tools (Сервис), которое содержит инструменты для администратора: систему обмена приватными сообщениями, функцию массовой почтовой рассылки и глобальную регистрацию элементов контента.

11

Меню Tools

Меню Tools (Сервис) содержит административные инструменты: систему обмена приватными сообщениями, функцию массовой почтовой рассылки и глобальную регистрацию на сайте элементов контента.

Система обмена приватными сообщениями

Страница Private Messaging (Обмен приватными сообщениями), вопреки своему названию, содержит все сообщения (рис. 11.1). По сути, это почтовый ящик.

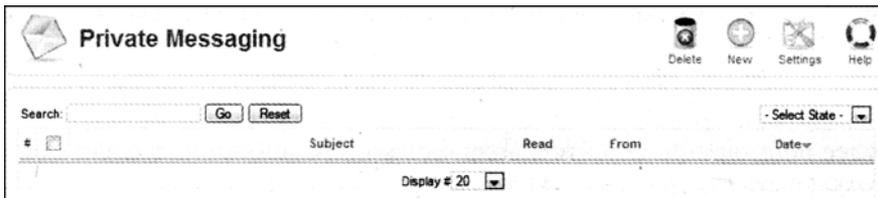


Рис. 11.1. Страница Private Messaging

Если вы щелкнете на теме или на отправителе сообщения, то сможете прочитать его, и состояние автоматически изменится на Read (Прочитано). Системные сообщения также доставляются сюда, например, когда кто-либо разместит новую статью. Щелкнув на пиктограмме New (Создать) или выбрав Tools⇒Write Message (Сервис⇒Написать сообщение), можно послать новое сообщение пользователям, имеющим доступ к администрированию Joomla!.

Щелчок на пиктограмме Settings (Настройки) позволит сконфигурировать систему обмена сообщениями.

Lock Inbox (Заблокировать входной ящик). Вы можете закрыть свой входной ящик и прекратить принимать почту. Если вы – единственный администратор, то это нормально, но в противном случае лучше оставьте его открытым.

Mail me on New Message (Уведомлять по почте о новом сообщении). Это средство очень удобно. Joomla! посылает сообщения по адресу электронной почты, который был определен при администрировании пользователя.

Auto Purge Messages (Автоматически удалять сообщения). Вы можете заставить систему автоматически удалять сообщения по истечении некоторого количества дней.

Массовая рассылка

Порадуйте пользователей массовыми рассылками! Страница Mass Mail (Массовая почта) показана на рис. 11.2.



Рис. 11.2. Страница Mass Mail

Как бы цинично это не звучало в наши дни, когда настолько распространен спам, все же это — лучший способ связи с зарегистрированными пользователями. Компонент Mass Mail предлагает необходимый для этого инструмент.

- Mail to Child Groups (Отправлять почту дочерним группам). Если отметить этот флажок, то подгруппы выбранной группы пользователей также будут включены в рассылку.
- Send in HTML mode (Посылать в режиме HTML). Отметьте этот флажок, если хотите выполнять рассылку в формате HTML. Почта HTML становится все более и более популярной. Имейте в виду, однако, что многие почтовые клиенты могут отключать отображение HTML, а некоторые адресаты по разным причинам не любят почт)" в виде HTML.
- Group (Группа). Здесь можно выбрать группу пользователей, которым адресуется рассылка.
- Subject (Тема). Тема сообщения.
- Message (Сообщение). Собственно текст для рассылки.

Для массовой рассылки потребуется корректно настроить параметры, связанные с почтой (Site⇒Global Configuration⇒Server⇒Mail Settings (Сайт⇒Глобальная конфигурация⇒Сервер⇒Настройки почты)). Для того чтобы иметь возможность выполнять массовую почтовую рассылку из локальной среды, не имея почтового сервера, можно ввести установки SMTP-сервера вашего почтового провайдера.

Щелкая на пиктограмме Parameters (Параметры), вы попадаете на экран, где можете определить значения Mailbody Suffix (Суффикс почтового сообщения) и Subject Prefix (Префикс темы). Префикс довольно удобен, так как позволяет клиентам электронной почты сортировать письма в определенном порядке.

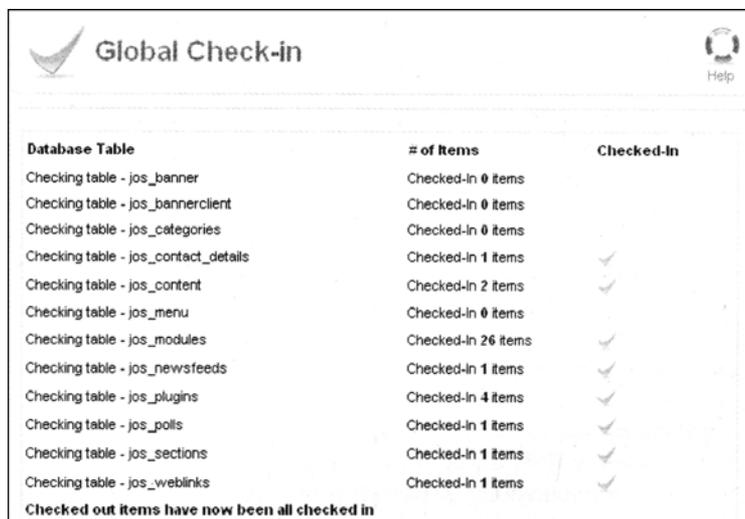
Глобальная регистрация контента

Если авторизованный пользователь запрашивает режим редактирования элемента контента, этот элемент *извлекается* (checked out) из сайта. Это означает лишь, что пользователю, запросившему элемент, разрешено его редактировать. В процессе редактирования другие пользователи будут видеть пиктограмму с символом замка перед именем элемента. Как только документ сохраняется после модификации, он автоматически вновь *возвращается* (checked in), и символ замка исчезает.

Но если пользователь, например, закрывает окно браузера или теряет соединение с Интернетом, элемент остается извлеченным и не может быть модифицирован.

И здесь вступает в действие страница Global Check-in (Глобальное возвращение), показанная на рис. 11.3. Щелчок на одноименном пункте меню приводит к тому, что все редактируемые элементы принудительно возвращаются, и вы получаете их список.

Недостаток такого глобального возвращения состоит в том, что на самом деле при этом возвращаются все элементы. Если кто-то только собрался провести модификацию, а элемент уже возвращен, то кто-то другой тут же сможет приступить к его редактированию. Поэтому будьте осторожны с этой функцией, и проверьте, кто в данный момент находится в онлайн.



The screenshot shows the 'Global Check-in' page with a table listing database tables and their checked-in status. The table has three columns: 'Database Table', '# of Items', and 'Checked-In'. Below the table, a message states: 'Checked out items have now been all checked in'.

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

Рис. 11.3. Страница Global Check-in

Резюме

В этой главе вы узнали о системе обмена приватными сообщениями, функциях массовой рассылки почты и глобальной регистрации элементов в меню Tools.

12

Дополнительные сведения о расширениях

Функциональность Joomla! легко поддается расширению. Благодаря популярности Joomla! 1.0.x, доступно множество готовых специализированных расширений. Команда разработчиков проекта Joomla! открыла собственный раздел расширений на веб-сайте, который доступен по адресу <http://extensions.joomla.org/>. Каждый посетитель может свободно загружать оттуда нужные ему расширения, а каждый зарегистрированный пользователь – посылать свои собственные расширения.

В главе 15 вы узнаете о том, как создавать собственные расширения. В конечном счете, если вы разработаете функциональное расширение, почему бы ни предложить его другим на упомянутом сайте? Другие пользователи загрузят его, оценят, и если разработанное вами действительно окажется полезным для других, вы вскоре получите некоторые отклики, которые, возможно, помогут в дальнейшей разработке.

Расширения могут состоять из компонентов, модулей, подключаемых модулей или внешних программ.

Благодаря модульной структуре Joomla! и удобному инсталлятору, устанавливать расширения очень легко. По состоянию на ноябрь 2007 г. было представлено всего 2267 расширений, 97 из которых – компоненты, работающие также с Joomla! 1.5.

К сожалению, некоторые из наиболее популярных компонентов, вроде Virtuemart (торговая система) или Pony Gallery (галерея изображений), на момент написания нашей книги еще не доступны для версии Joomla! 1.5. Однако все эти популярные расширения сейчас обновляются для Joomla! 1.5, и вероятно, скоро будут готовы. Работа с этими расширениями в Joomla! 1.5 будет похожа на работу в версиях Joomla! 1.0.x. Модификации в основном коснутся исходного кода и усовершенствованной интеграции.

В этой главе вы инсталлируем популярное программное обеспечение форумов Fireboard, которое создано в рамках проекта Joomla!Board, затем систему управления документами DocMan и, наконец, галерею изображений по имени Exposé.

Для того чтобы все эти компоненты работали гладко, необходимо активизировать подключаемый модуль Legacy в меню Plugin Manager. О его корректной работе говорит наличие элемента System – Legacy в области меню административного раздела страницы Plugin Manager (рис. 12.1).

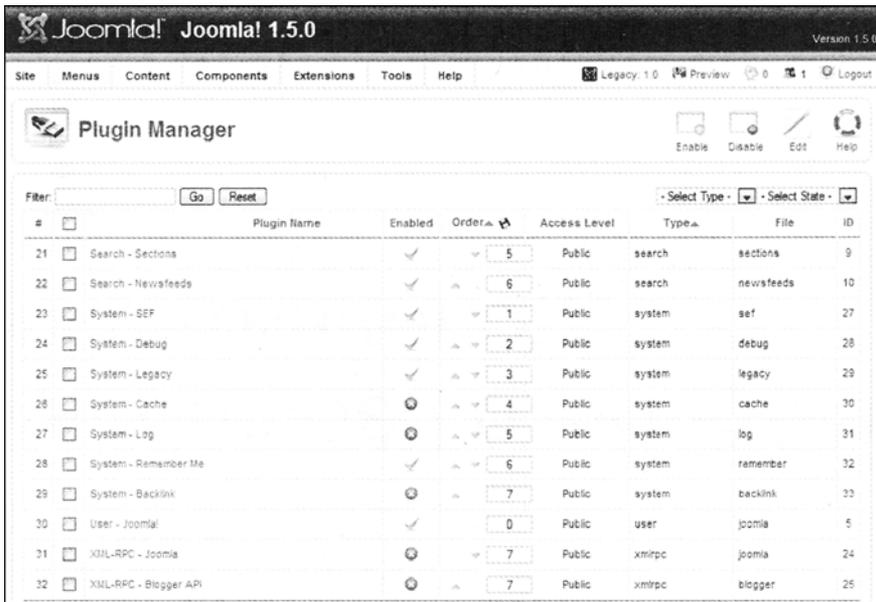


Рис. 12.1. Страница Plugin Manager с активизированным подключаемым модулем Legacy

Инсталляция

Компонент инсталлируется через меню Extensions⇒Install/Uninstall (Расширения⇒Инсталлировать/деинсталлировать) — точно так же, как языковые пакеты, шаблоны, модули и подключаемые модули.

Каждый пакет расширения состоит из архива с разными файлами и, по крайней мере, одного XML-файла инструкций для инсталлятора Joomla!. Файл XML сообщает тип расширения, с которым вы имеете дело. Благодаря этому методу инсталляции, все типы расширений могут быть установлены из одного шаблона инсталляции Joomla! 1.5.

В главах 13 и 15 вы создадите некоторые собственные пакеты, подобные этому.

Fireboard Forum

Пакет Fireboard (<http://www.bestofjoomla.com/>) — это потрясающий форум с множеством средств, часть из которых перечислена ниже.

- Определяемые пользователем вложенные категории форума.
- Столько форумов, сколько нужно, причем с поддержкой прав доступа.
- Профили и аватары пользователей.
- Загрузка файлов и изображений в сообщения форума.

Инсталляция

Инсталляция достаточно проста. Загрузите программный пакет Fireboard_Forum_1.0.3_UnzipFirst.zip и распакуйте его. Вы найдете в этом пакете файл по имени component_Fireboard_Forum_1.0.3.zip.

Щелкните в меню на пункте Extensions⇒Install/Uninstall, выберите файл component_Fireboard_Forum_1.0.3.zip и щелкните на кнопке Upload File & Install

(Загрузить файл и установить). Страница Extension Manager (Диспетчер расширений) показана на рис. 12.2.

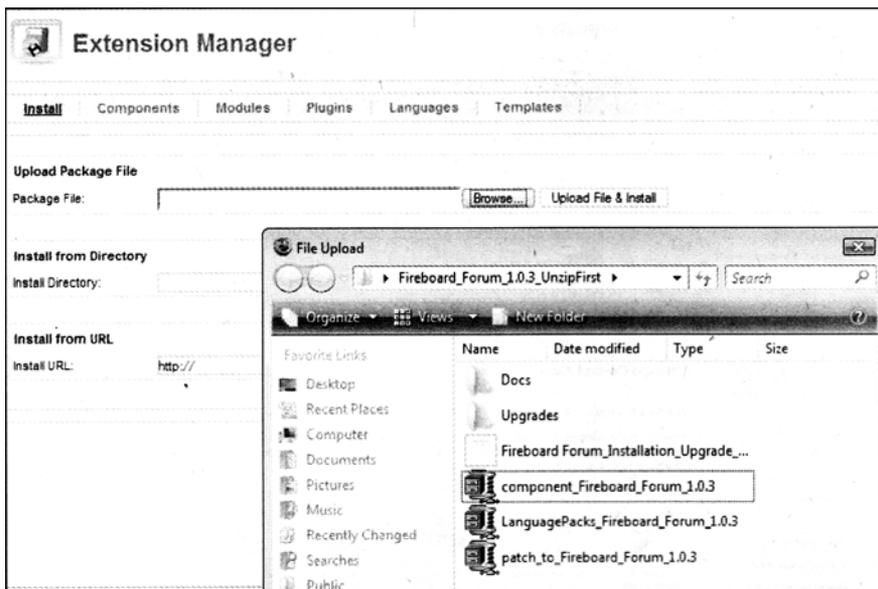


Рис. 12.2. Страница Extension Manager в процессе установки расширения

После загрузки вы получите экран, полный сообщений зеленого цвета. Если прокрутить его вниз, вы должны увидеть сообщение, показанное на рис. 12.3.

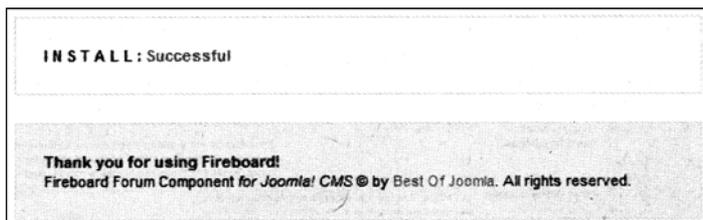


Рис. 12.3. Сообщение об успешной установке расширения

Если теперь вы проверите меню Components (Компоненты), то увидите там новый элемент – Fireboard Forum. Щелкните на нем, и появится панель управления Fireboard Control Panel. Но прежде чем приступить к работе, необходимо еще установить таблицы базы данных (рис. 12.4).

Ниже текста Clean Installation (Чистая установка) щелкните на кнопке Apply! (Применить!).

Расширение форума установлено.

Конфигурирование

Теперь, когда все загружено, можно определить некоторые из наиболее важных параметров. Fireboard имеет многочисленные уже установленные опции, описание которых может занять еще несколько глав, поэтому мы остановимся на некоторых

наиболее важных. Раздел конфигурации также хорошо документирован; у вас не должно быть проблем с нахождением того, что вам нужно.

Введите имя и адрес электронной почты форума в Fireboard Configurations⇒Basics (Конфигурация Fireboard⇒Базовые параметры), (рис. 12.5)

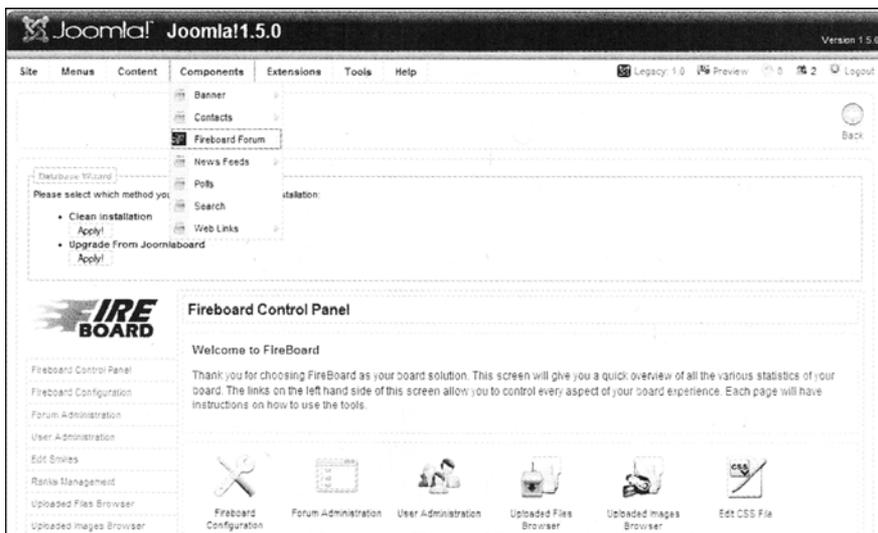


Рис. 12.4. Меню Components после инсталляции расширения

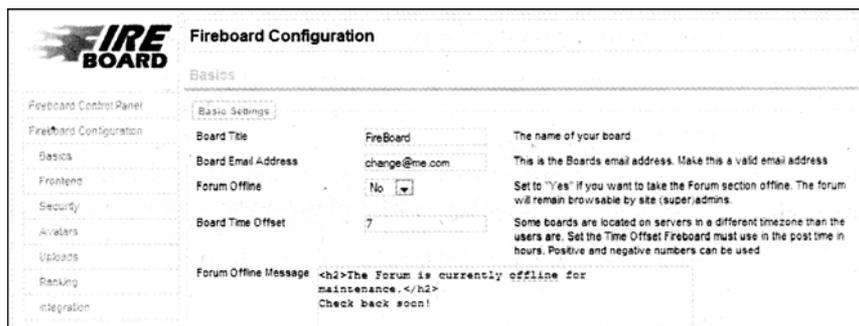


Рис. 12.5. Базовые параметры конфигурации Fireboard

Создание форума

Если вы хотите работать с форумом, то, разумеется, сначала должны его создать. При этом обязательно потребуется установить категорию, которая будет служить контейнером для содержащихся внутри форумов.

После создания категории можно настроить сам форум, который затем включить в эту категорию.

Теперь вы можете видеть категорию и форум. Форум немного отодвинут вправо, чтобы символизировать соединение. Нужно еще явно опубликовать категорию и новый форум.

Далее потребуется выполнить основные задачи по интеграции форума в ваш веб-сайт.

Интеграция в веб-сайт

Давайте создадим ссылку Forum (Форум) в меню Joomla! 1.5. Чтобы сделать это, щелкните на Menus⇒Joomla 1.5 Book⇒New (Меню1⇒Книга Joomla⇒Создать). Отобразится страница выбора пункта меню. Щелкните на Fireboard Forum (рис. 12.6).

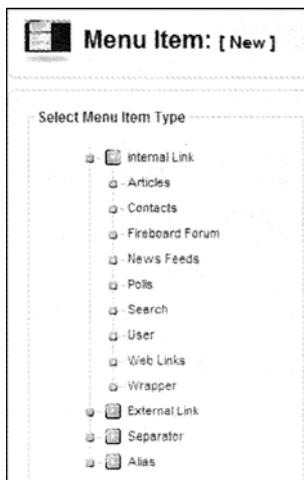


Рис. 12.6. Выбор пункта меню Fireboard Forum

Откроется экран редактирования ссылки меню (рис. 12.7). Введите заголовок и псевдоним, после чего щелкните на пиктограмме Save (Сохранить).

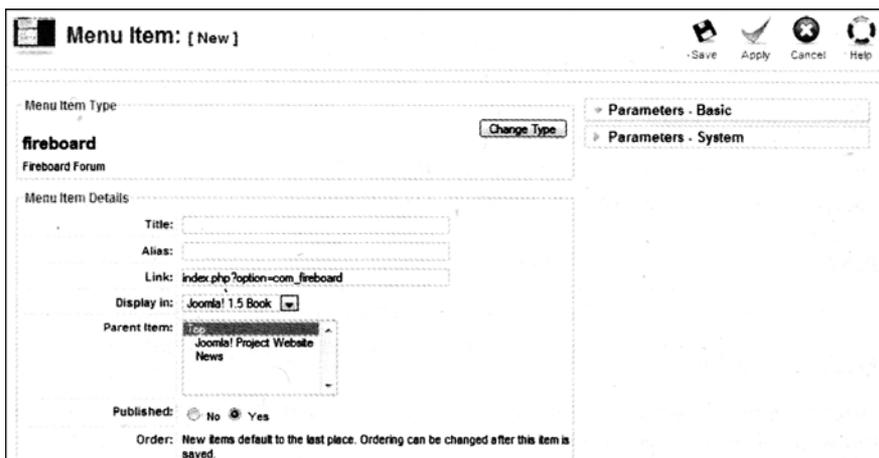


Рис. 12.7. Страница редактирования нового пункта меню

Если теперь перегрузить и обновить веб-сайт, появится новая ссылка меню. Щелкнув на ней, вы увидите новый форум (рис. 12.8).

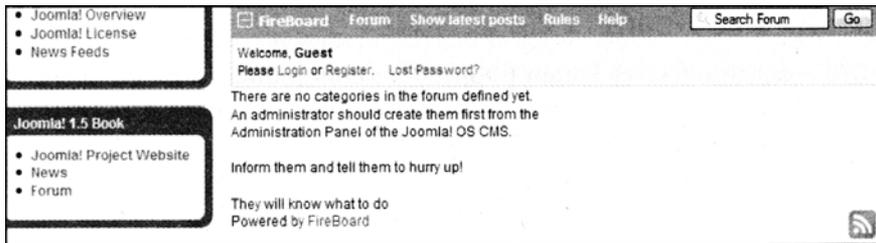


Рис. 12.8. Внешний вид форума после установки

DOCman (раздел Download, Document Management)

Концепция, положенная в основу DOCman (<http://www.nambodocman.com/>), заключается в том, чтобы связать файлы с описанием и лицензией, а затем предоставить его различным группам пользователей для загрузки.

Ниже перечислены возможности DOCman.

- Можно вкладывать свои избранные документы в любое количество категорий и подкатегорий.
- Можно сохранять эти файлы на собственном или же на удаленном сервере.
- Документы могут быть защищены правами доступа.
- Можно устанавливать любое необходимое количество групп, чтобы контролировать доступ к документам.
- Можно накапливать развернутую статистику о загрузках.

DOCman оснащен собственной системой поиска. Документы можно искать по имени файла или его описанию. Предусмотрен дополнительный подключаемый модуль, интегрирующий поиск в систему Joomla!.

Действительный путь к документам не имеет значения. На вашем веб-сайте доступно множество шаблонов для модификации. Кроме того, существует растущая онлайн-вая документация в форме wiki.

Инсталляция

Инсталляция идентична той, что была описана для форума. Загрузите файл `com_docman_1.4.Orcl.zip` и инсталлируйте пакеты через пункт меню Extensions⇒ Install/Uninstall (рис. 12.9).

Под логотипом DOCman находится кнопка для инсталляции данных примеров. Щелкните на ней, после этого DOCman инсталлирует:

- файл примера;
- описание этого файла;
- категорию, назначенную описанию;
- группу пользователей;
- пример лицензии.

После инсталляции данных примера DOCman готов к работе.

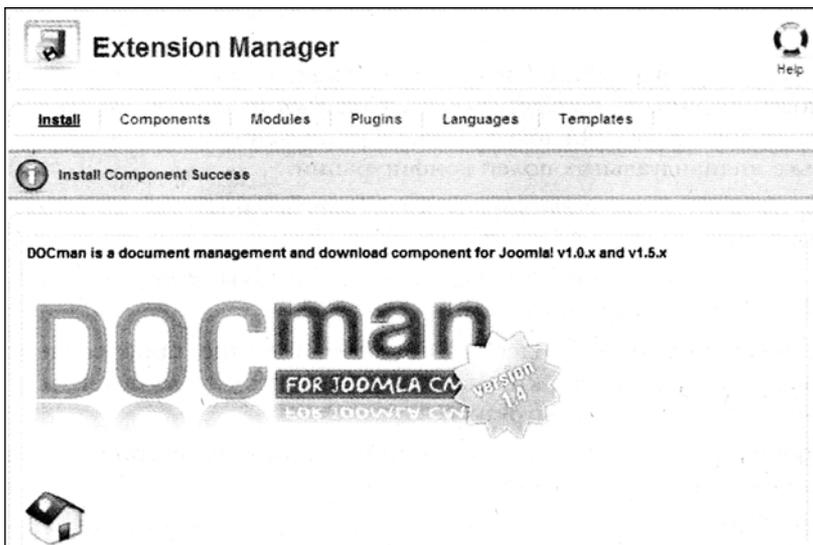


Рис. 12. 9. Инсталляция DOCman

Конфигурирование

DOCman — это один из компонентов, которые полностью поддерживают концепцию меню Joomla! 1.5 (рис. 12.10).

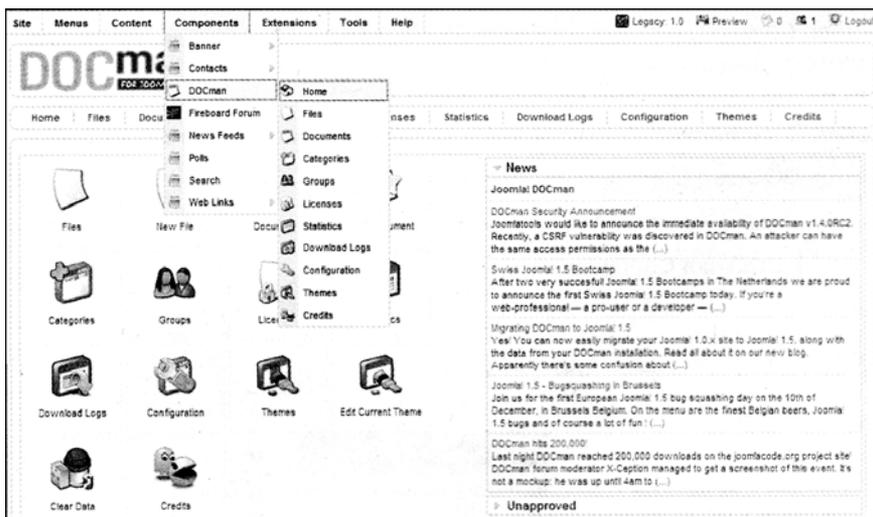


Рис. 12.10. Меню DOCman

Индивидуальные элементы, опять-таки, находятся в строке под логотипом. Существуют также пиктограммы быстрого доступа в центр управления DOCman Control Center и обзор информации в правой части.

Если вы щелкнете на ссылке Configuration (Конфигурация), то увидите расширенные опции DOCman.

Язык

Поначалу для всех параметров можно оставить значения по умолчанию. Ознакомившись с многочисленными настройками DOCman вы наверняка уже убедились в его мощи в отношении обработки документов. Обратите внимание на несколько перечисленных ниже индивидуальных полей конфигурации.

- Home (Домой). Это панель управления (Control Panel), другими словами, обзорная панель.
 - Files (Файлы). Это место, куда вы загружаете файлы, которые хотите предоставить для загрузки пользователям.
 - Documents (Документы). Здесь вы создаете описание, прикрепляемое к файлу. Здесь же вы определяете, кто может иметь доступ к этому файлу, и какая лицензия ассоциирована с его загрузкой.
 - Categories (Категории). Здесь вы администрируете категории, в которые включены предлагаемые документы.
 - Groups (Группы). Здесь вы создаете и администрируете группы пользователей.
 - Licenses (Лицензии). Здесь вы определяете и администрируете условия лицензирования. Администрировать можно произвольное количество лицензий.
 - Statistics (Статистика). Здесь можно просматривать статистику загрузок (насколько часто загружался тот или иной файл).
 - Download Logs (Журнал загрузок). Отсюда можно загружать файлы журналов.
 - Configuration (Конфигурация). Базовая конфигурация DOCman.
 - Themes (Темы). DOCman поддерживает и управляет собственными темами. Вы можете загружать дополнительные темы с веб-сайта проекта и затем использовать их для оформления интерфейсной части.
 - Credits (Участники). Здесь вы найдете протокол изменений, начиная с 2004 г.
- Страница DOCman Configuration (Конфигурация DOCman) показана на рис. 12.11.

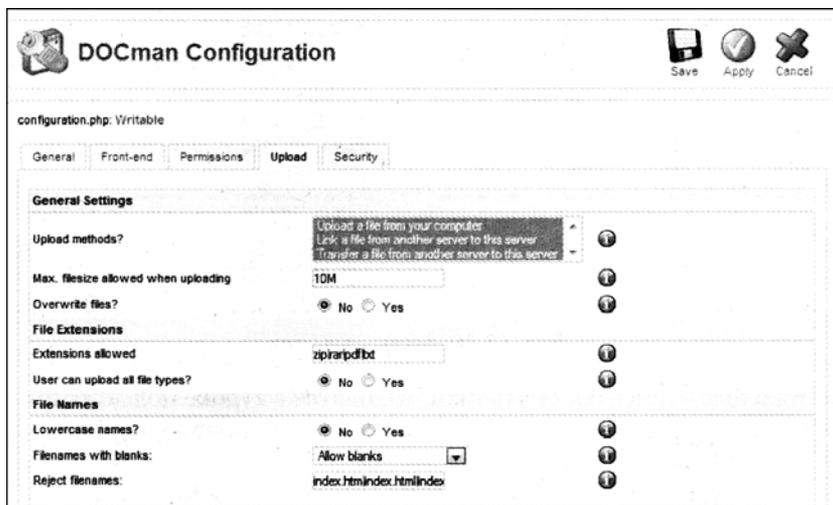


Рис. 12.11. Страница DOCman Configuration

Интеграция в веб-сайт

Эта процедура идентична тому, что мы делали с форумом Fireboard. Вы должны создать ссылку в Joomla! 1.5 Book на ваши документы. Для этого потребуется создать новый элемент в меню Joomla! 1.5 Book. В качестве типа пункта меню выберите DOCman (рис. 12.12).

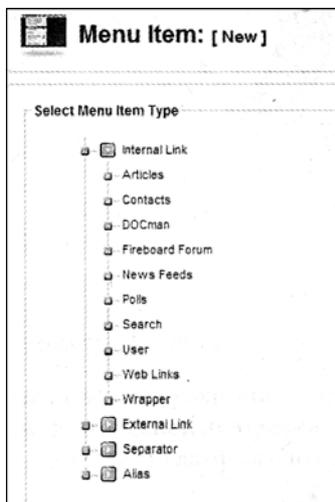


Рис. 12.12. Выбор типа пункта меню

Откроется экран редактирования, показанный на рис. 12.13.

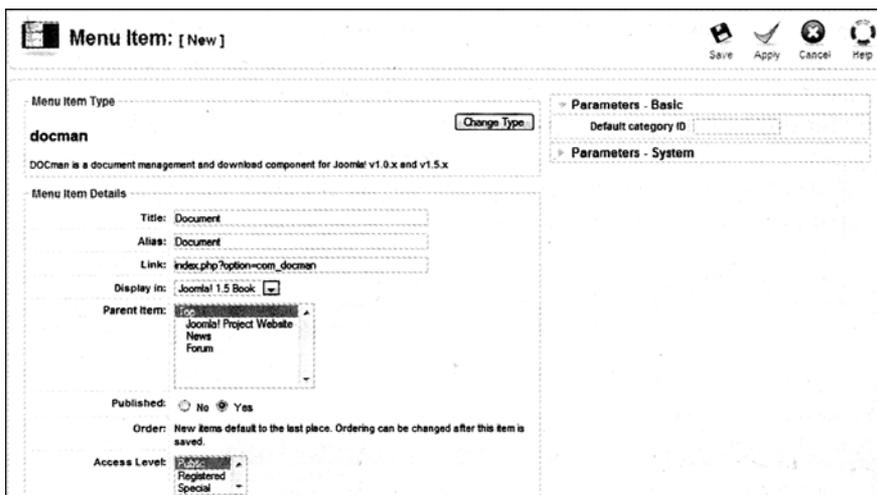


Рис. 12.13. Страница редактирования нового пункта меню типа DOCman

Установите заголовок и псевдоним для ссылки меню. Вы можете также активизировать определенную категорию документов непосредственно. Просто введите идентификатор категории в параметрах. Впоследствии вы найдете этот идентификатор в разделе категорий администрирования DOCman.

После щелчка на пиктограмме Save (Сохранить) и перезагрузки веб-сайта увидите профессионально выглядящий каталог загрузки (рис. 12.14).

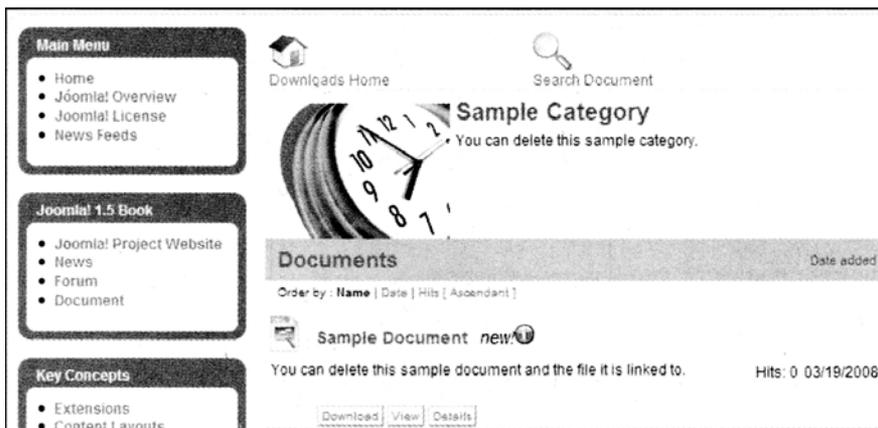


Рис. 12.14. Каталог загрузки с примером документа

Теперь если кто-то щелкнет на кнопке Download (Загрузить), будет отображена лицензия для этой загрузки. Пользователь должен подтвердить свое согласие с условиями лицензирования, и после этого загрузка стартует.

Подготовка контента

Вы можете определить новые файлы и описания для загрузки из области администрирования либо с интерфейсной части. После входа на свой веб-сайт в качестве пользователя admin вы увидите пиктограмму и ссылку Submit File (Отправить файл) в верхней области. Загрузка документа на сервер выполняется за три шага.

На первом шаге вы указываете, откуда поступает документ: с вашего компьютера (Upload a file from your computer (Загрузить файл с вашего компьютера)), с другого сервера (Transfer a file from another server to this server (Передать файл с другого сервера на этот сервер)) или же это должна быть просто ссылка на некоторый существующий документ (Link a file from another server to this server (Ссылка на файл с другого сервера на этом сервере)), как показано на рис. 12.15.

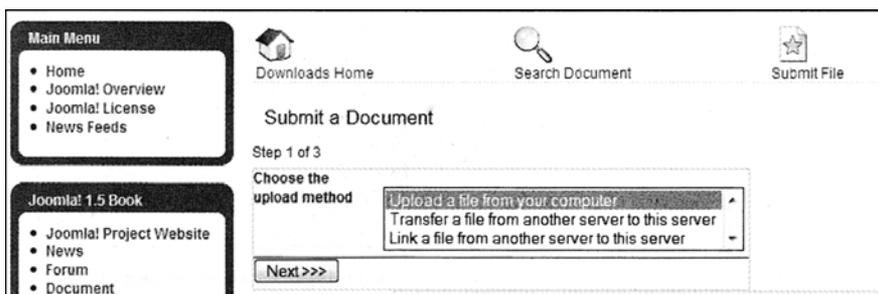


Рис. 12.15. Выбор метода загрузки документа

Необходимо выбрать вариант Upload a file from your computer и загрузить zip-файл компонента (рис. 12.16).

Submit a Document

Step 3 of 3

Document Permissions License Details

Title
router

Category
Select Category

Date
2008-03-19 16:09:41

Thumbnail
Select Image

File
router.php

Homepage

(Make sure to start the URL with http://)

Approved
 no yes

Description

Рис. 12.16. Загрузка документа на сервер

После успешной загрузки на шаге 3 потребуется указать значения для Description (Описание), Permissions (Права доступа), Category (Категория) и License (Лицензия). По щелчку на пиктограмме с галочкой зеленого цвета в верхней области новая загрузка сохраняется и немедленно публикуется, если вы вошли как admin. Пользователь admin видит намного больше кнопок, чем рядовой пользователь.

Обычно документ требует явной публикации в административной области (рис. 12.17).

Documents

Home Files Documents Categories Groups Licenses Statistics Download Logs Configuration Themes Credits

Filter: Select Category [1 Documents not approved] [1 Documents not published]

Name	File	Category	Date	Viewers	Published	Approved	Size	Hits	Checked out
framework	framework.php	Sample Category	Thursday, 20 March 2008	Everybody	<input checked="" type="radio"/>	<input checked="" type="radio"/>	1.87 kB	0	---
Sample Document	sample_file.png	Sample Category	Wednesday, 19 March 2008	Everybody	<input type="radio"/>	<input checked="" type="radio"/>	53.88 kB	0	---

Display # 20

Рис. 12.17. Список документов, доступных для загрузки

Подключаемый модуль поиска

Для того чтобы документы можно было включить во внутреннюю функцию поиска Joomla!, потребуется установить дополнительный подключаемый модуль. Вновь воспользуйтесь пунктом меню Extensions⇒Install/Uninstall. Загрузите файл bot_docman_search_1.4.0rc1.zip и устанавливайте его.

Теперь вы увидите подключаемый модуль поиска DOCman в диспетчере подключаемых модулей (Plugins Manager), и вам нужно будет его активизировать.

С этого момента при использовании обычного поля поиска веб-сайта вновь добавленный документ появится в списке найденных вместе со своей категорией.

Дополнительный модуль

В случае предоставления множества документов для загрузки может понадобиться отображать наиболее часто загружаемые документы. Для этого нужен модуль, который показывает список наиболее часто загружаемых документов. Воспользуйтесь пунктом меню Extensions⇒Install/Uninstall для инсталляции файла `mod_docman_mostdown_1.4.0rc1.zip` и активизируйте его в диспетчере расширений (Extension Manager).

По умолчанию этот модуль размещается слева. С помощью пиктограммы Edit (Редактировать) его можно позиционировать где угодно; разумеется, можно также изменить его заголовок. В параметрах можно задать количество элементов и то, должна ли отображаться пиктограмма и количество.

Галерея Exposé

Галереи изображений и видеороликов становятся все более и более популярными в современном мире Web 2.0. Такие приложения, как фото-платформа Flickr (<http://www.flickr.com/>) и видео-платформа YouTube (<http://www.youtube.com/>), обеспечили простую возможность публикации мультимедиа-контента для индивидуальных пользователей в современной сети Интернет с постоянно возрастающей пропускной способностью.

Требования к дизайну также все время ужесточаются. Простой список фотографий теперь выглядит несколько старомодно.

Exposé – одна из мощных, основанных на формате Adobe Flash галерей изображений, графики и видео.

Ниже перечислены основные возможности Exposé.

- Создание произвольного количества альбомов изображений.
- Сжатие изображений во время загрузки до разных размеров.
- Создание миниатюр для альбомов.
- Воспроизведение видеороликов посредством Flash Player в сжатом формате.
- Добавление аудио-комментариев к отдельным изображениям.

Версия Exposé, доступная для инсталляции, позволяет собирать изображения на вашем собственном компьютере. Благодаря компактному формату Flash, их можно просматривать, не располагая высокоскоростным соединением.

Интерфейс Album Manager (Диспетчер альбомов), который позволяет управлять данными в административной области, предъявляет несколько требований к интерпретатору PHP, а именно: должны быть установлены расширения GD, DOMXML и iconv. О выполнении этих условий на сервере позаботилась локальная среда XAMPP Lite.

Может случиться, что ваш провайдер хостинга не предоставляет таких средств PHP, но и на этот случай существует решение (см. раздел "Диспетчер альбомов в административном разделе" далее в главе). Когда дело касается интерфейсной части, посетители должны иметь, как минимум, Flash Player версии 8, доступный в браузере. В наше время Flash Player инсталлирован на 96% всех браузеров.

Инсталляция

Чтобы установить галерею Exposé, загрузите пакет `com_expose_4.6.2.zip` с http://joomla.org/gf/project/expose/frs/?action=FrsReleaseBrowse&frs_package_id=2985. Щелкните на кнопке **Browse (Обзор)** в инсталляционном шаблоне, выберите файл и щелкните на кнопке **Upload File & Install (Загрузить файл и установить)**. Инсталлятор Joomla! загрузит файлы, обработает пункты меню и отобразит сообщение об успехе, как показано на рис. 12.18. Прочтите внимательно это сообщение и запомните пароль.

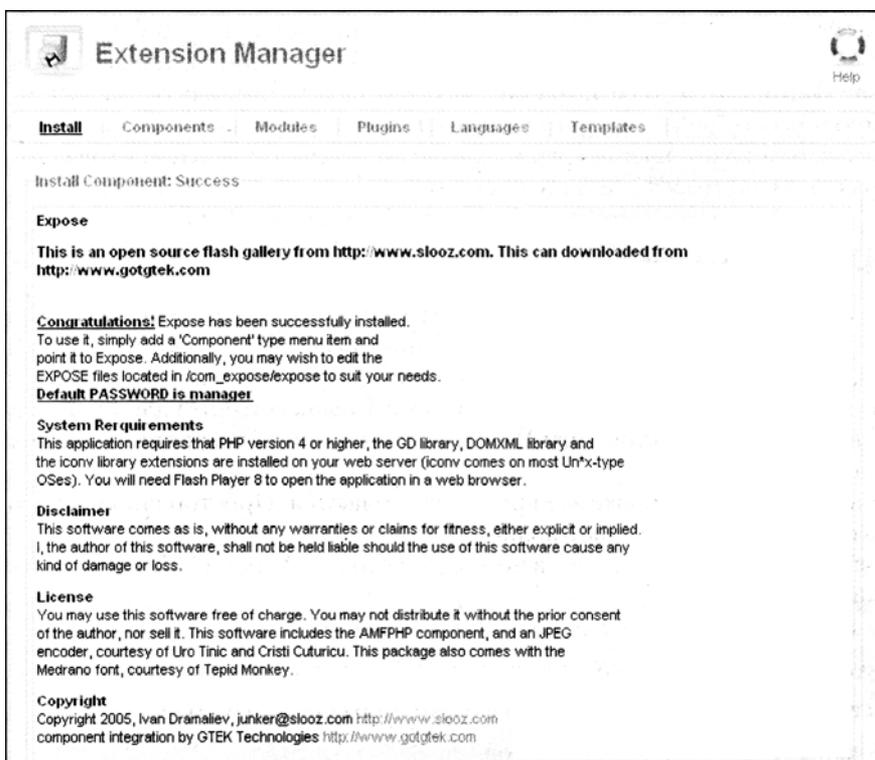


Рис. 12.18. Страница Extension Manager по завершении инсталляции Exposé



На момент написания этой книги галерея имела проблемы совместимости с подключаемым модулем Joomla! Legacy. Это довольно странно, поскольку Joomla! "просто" использует галерею в качестве контейнера, и она выполняется независимо. Существует обходной путь, представленный на форуме разработчиков галереи (<http://www.gotgtek.net/forum/index.php?topic=1345.0>). К моменту выхода книги в свет, вероятно, появится исправленная инсталляция пакета. Если нет вам нужно будет вручную ввести следующую команду:

```
INSERT INTO 'jos_components' ('id', 'name', 'link', 'menuid', 'parent', 'admin_menu_link', 'admin_menu_alt', 'option', 'ordering', 'admin_menu_img', 'iscore', 'params') VALUES (51, 'Expose', 'option=com_expose', 0, 0, '', 'Expose', 'com_expose', 0, './administrator/componentets/com_epose/expose_icon.png', 0, ''), (52, 'Manage Albums', '', 0, 51, 'option=com_expose&task=manage',
```

```
'Manage Albums', 'com_expose', 0,
'../administrator/components/com_expose/manage.png', 0, ''),
(53, 'Configuration', '', 0, 51, 'option=com_expose&task=config',
'Configuration', 'com_expose', 1,
'../administrator/components/com_expose/config.png', 0, ''),
(54, 'Manual', '', 0, 51, 'option=com_expose&task=manual', 'Manual',
'com_expose', 2, '../administrator/components/com_expose/docs.png',
0, '');
```

Затем потребуется распаковать файл пакета и скопировать содержащиеся в нем файлы в соответствующие каталоги, как показано на рис. 12.19 и 12.20. В файле `expose.xml` хранится информация о том, что куда пошло.

На рис. 12.19 показано содержимое каталога `/components/com_expose`.

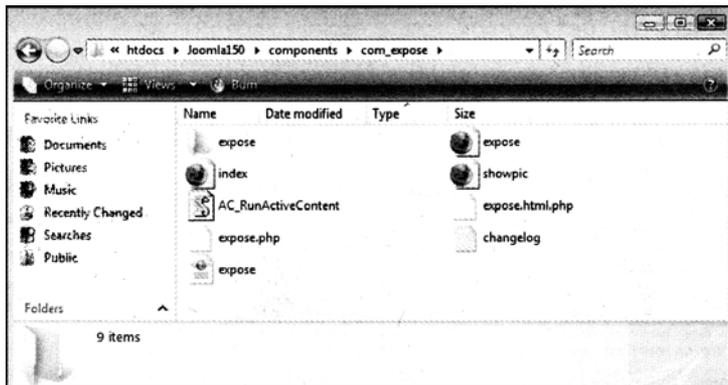


Рис. 12.19. Содержимое каталога `/components/com_expose`

А содержимое каталога `/administrator/com_expose` представлено на рис. 12.20.

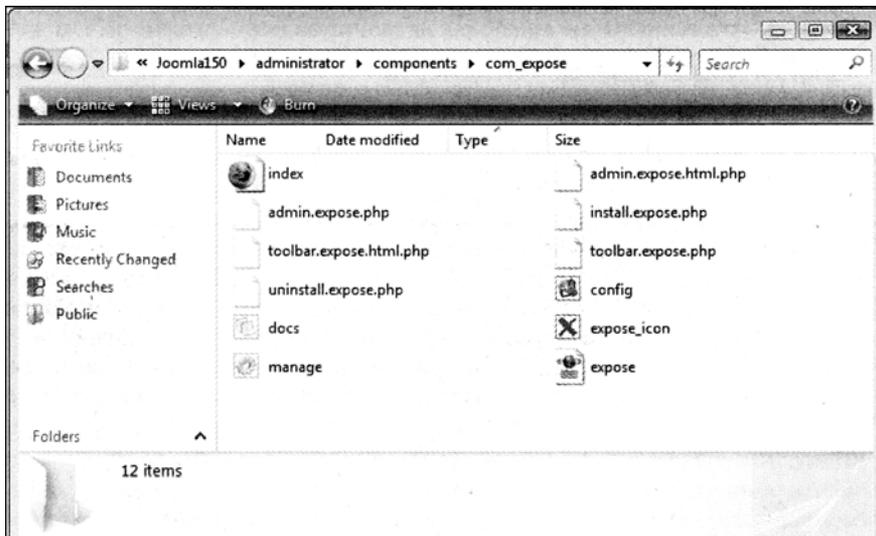


Рис. 12.20. Содержимое каталога `/administrator/com_expose`



Если ранее описанная процедура инсталляции галереи привела в замешательство, взглядывайте на упомянутый форум и ждите, когда разработчики предложат более простое решение.

Интеграция в веб-сайт

Компонент теперь установлен вместе с примерами данных и нуждается в соответствующем пункте меню в интерфейсной части сайта. Поместим его в меню Joomla! 1.5 Book.

Щелкните на Menu⇒Joomla! 1.5 Book (Меню⇒Книга Joomla! 1.5), а затем на пиктограмме New (Создать). Теперь создайте ссылку меню Gallery (Галерея), как описано в разделе "Создание нового меню" главы 7. Если вы теперь зайдете на веб-сайт, то увидите ссылку Gallery в меню Joomla! 1.5 Book. Щелчок на этой ссылке приводит к отображению двух папок примеров – Sample Collection (Пример коллекции) и Sample Album (Пример альбома) – и папки Videos (Видео).

Если рядом с миниатюрой находится стрелочка, значит, она содержит дополнительные альбомы. Если здесь же присутствует число, оно указывает количество медиа-файлов, содержащихся внутри (рис. 12.21). Щелчок на Sample Collection приводит к отображению содержащихся внутри альбомов.

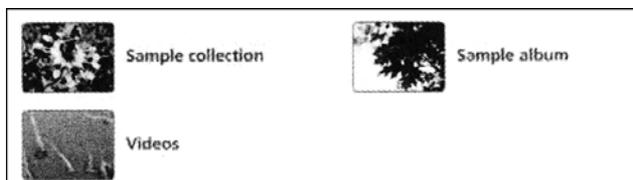


Рис. 12.21. Папки галереи

Щелчок на Sample Album переносит в так называемую полосу изображений (Image Strip), которая показана на рис. 12.22.

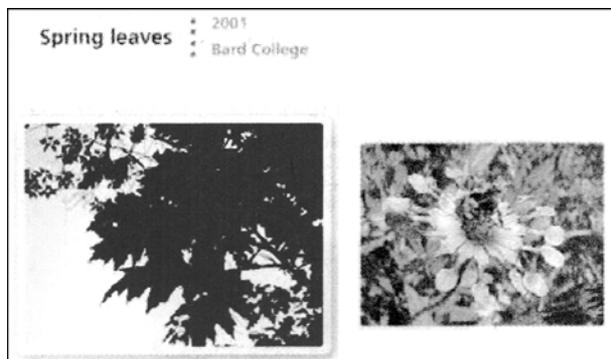


Рис. 12.22. Полоса изображений

Эта полоса может перетаскиваться вручную с помощью мыши. Изображение в центре увеличено. Справа и слева видны треугольники навигации. Вы можете перемещать ленту изображений, щелкая на них.

Если щелкнуть на изображении, появится индивидуальное его представление (рис. 12.23).



Рис. 12.23. Индивидуальное изображение

На странице индивидуального представления изображения можно:

- прокручивать вручную изображения вперед и назад;
- запустить автоматическое слайд-шоу;
- вернуться к полосе изображений;
- вернуться к альбому;
- щелкнуть на пиктограмме с увеличительным стеклом и увидеть изображение в его исходном размере;
- получить подсказку, какой альбом содержит изображения;
- вызвать экран справки, объясняющий все символы.

Видео

Вернитесь к альбому и щелкните на папке Videos. Здесь выводится лента (Image Strip) с изображением предварительного просмотра видео. Щелчок на этом изображении приводит к появлению индивидуального представления видеоролика (рис. 12.24).

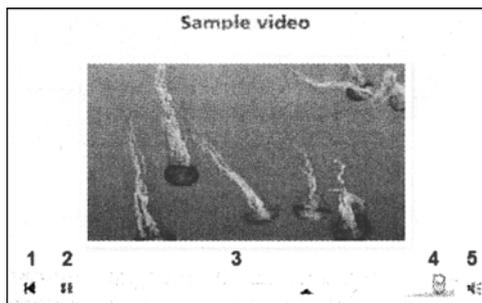


Рис. 12.24. Индивидуальный видеоролик

На странице индивидуального представления видеоролика можно:

- двигаться по видеоролику;
- включать и выключать звук, щелкая на символе громкоговорителя, и регулировать громкость звука с помощью ползунка.

Диспетчер альбомов в административном разделе

Чтобы загрузить собственные изображения и видеоролики в галерею, зайдите в раздел Administration и обратитесь к пункту меню Components⇒Extensions (Компоненты⇒Расширения). Появится экран входа, запрашивающий пароль (рис. 12.25). Пароль по умолчанию выглядит как manager. Вы можете (и должны) изменить пароль, щелкнув на кнопке Change Password (Изменить пароль). Если отметить флажок Save (Сохранить), пароль будет сохранен и не придется вводить его снова.

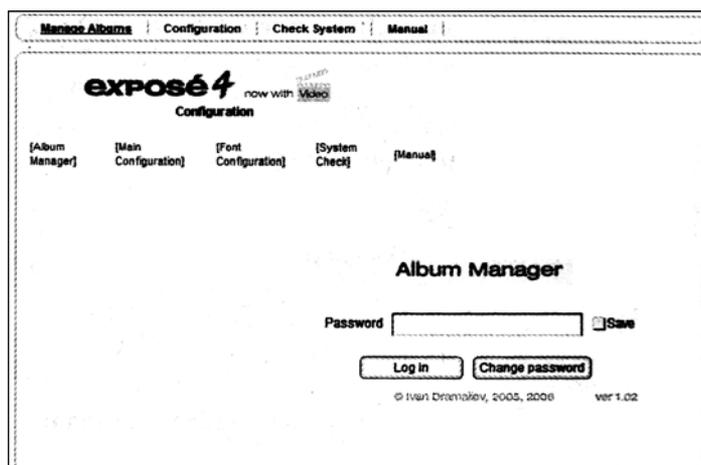


Рис. 12.25. Экран входа диспетчера альбомов

Область администрирования разделена перечисленные ниже части, как показано на рис. 12.26.

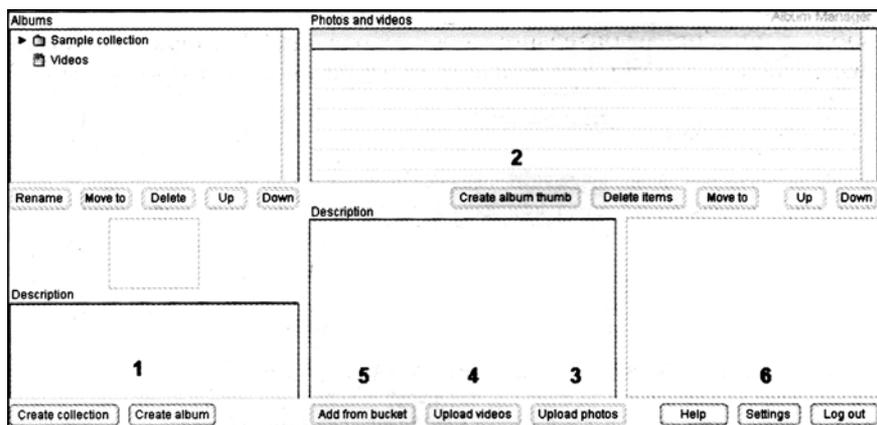


Рис. 12.26. Части области администрирования

- (1) Здесь можно создавать коллекцию и индивидуальные альбомы. В области, расположенной выше, можно перемещать (Move To) альбомы, переименовывать их (Rename) и удалять (Delete).
- (2) Чтобы создать изображение предварительного осмотра для альбома, выберите нужное изображение и щелкните на кнопке Create album thumb (Создать миниатюру альбома).
- (3) Щелчок на кнопке Upload photos (Загрузить фотографии) приводит к запуску диалога, в котором можно загрузить в галерею изображение с жесткого диска. Изображение автоматически сжимается, то есть вы можете загружать его в том размере, в каком оно хранится в камере.
- (4) С помощью кнопки Upload videos (Загрузить видео) можно загружать видеоролики в галерею в формате flv. Для конвертирования видеофайлов в этот формат можно воспользоваться пробной версией программы Dreamweaver CS3 или же бесплатным кодировщиком Riva FLV Encoder 2.
- (5) Щелчок на кнопке Add from bucket (Добавить из корзины) открывает доступ к весьма полезному средству. С помощью FTP можно переместить файлы в каталог [путь_к_Joomla!]components/com_expose/expose/manager/bucket, а позднее добавить их посредством Add from bucket.
- (6) Щелчок на кнопке Configuration (Конфигурация) открывает доступ к странице конфигурирования параметров, с которыми должны загружаться на сервер изображения и видеоролики (рис. 12.27). Здесь можно даже включить "водяные знаки" (watermark) в изображения.

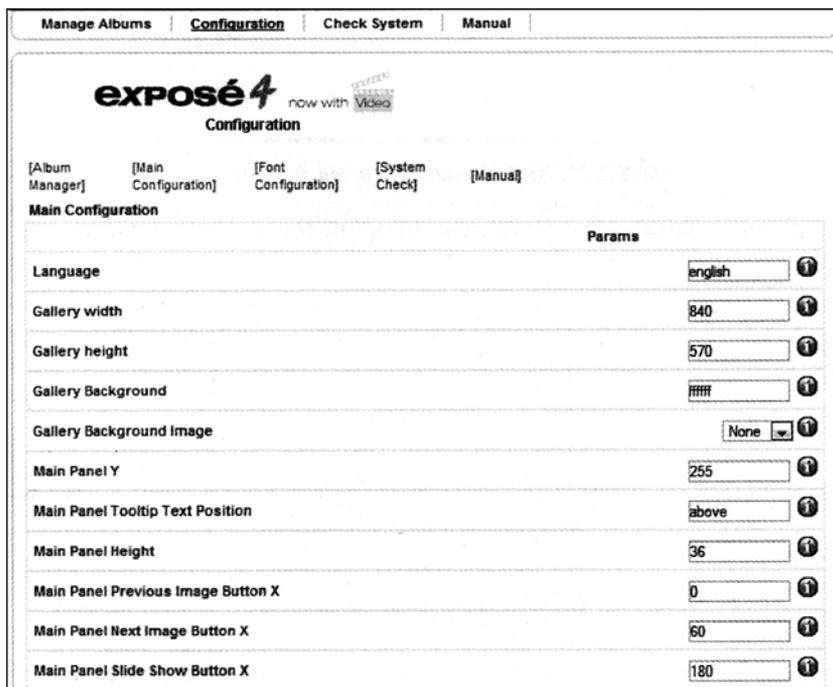


Рис. 12.27. Страница конфигурирования Exposé

Завершить управления изображениями вы можете щелчком на кнопке Log out (Выход), расположенной справа внизу.

Деинсталляция

Если необходимо удалить компонент Exposé из системы, обратитесь к меню Extensions⇒Install/Uninstall и перейдите на вкладку Components (Компоненты). Выберите элемент Exposé и щелкните на пиктограмме Uninstall (Деинсталлировать).

Диспетчер альбомов как автономная программа

Поскольку работа в онлайн-режиме иногда утомительна, особенно при большом размере файлов, доступна автономная версия (Stand Alone) программы Album Manager для операционных систем Windows и Mac OS X.

Программа устанавливается под Windows двойным щелчком на файле AlbumManager-2.7-Setup.msi.

Диспетчер альбомов работает с каталогом [Путь_к_Joomla !]components/com_expose/expose.

Если вы работаете в локальной серверной среде, то можете просто обратиться к этому каталогу при запуске Album Manager.

После запуска диспетчер альбомов позволит работать с существующей структурой индентично своей онлайн-версии (рис. 12.28).

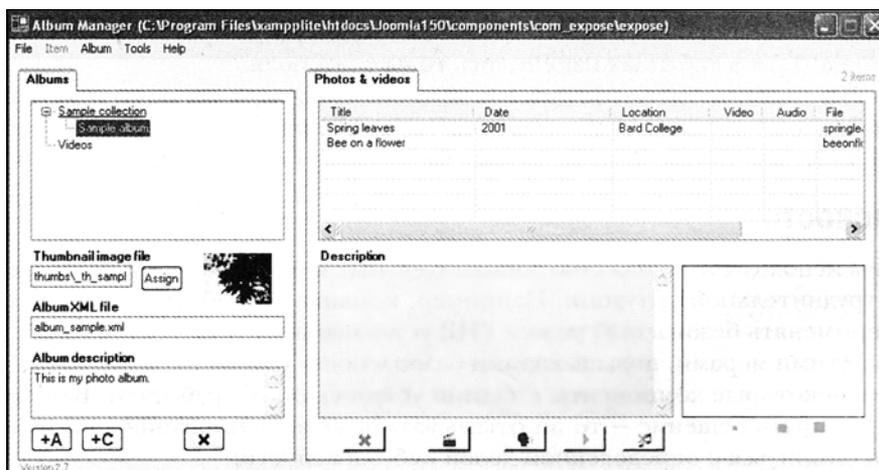


Рис. 12.28. Автономная версия диспетчера альбомов

Если же необходимо использовать Album Manager для какого-то веб-сайта в Интернете, просто загрузите каталог перед тем, как начать работу с ним, и выгрузите обратно на сервер по завершении работы.

Интеграция в каркас Joomla!

Поскольку Joomla! выросла из системы управления содержимым до каркаса, теперь стало возможным писать компоненты Joomla!, которые выглядят совершенно иначе. Компонент галереи, например, очень слабо связан с Joomla! 1.5. Галерея отображает-

ся в компоненте-оболочке, подобном `iframe`. Администрирование также имеет дело с `iframe`, поэтому требуется запрос пароля. Другие компоненты, например, инсталлированный форум или система управления документами, глубже интегрированы в систему, а, скажем, в случае Community Builder даже подменяют процедуру входа.

С появлением Joomla! 1.5 эти средства теперь могут быть интегрированы более элегантно, и все независимые команды разработчиков уже работают над интеграцией своих компонентов, начиная с 2007 г.

Проблемы с компонентами от независимых поставщиков

Проблемы обновления сторонних компонентов растут вместе с популярностью Joomla!.

Обновления

За последний год новые усовершенствованные версии Joomla! выходили по одной в месяц, и множество ранее обнаруженных проблем безопасности Joomla!, помимо прочего, были успешно преодолены.

Если вы работаете с веб-сайтом, который состоит только из центрального кода Joomla!, то вы в безопасности. Загрузите пакет обновлений, перепишите старые файлы, и вы получите "свежую" систему.

Обновления таблиц базы данных требуются редко. Даже при переходе от Joomla! 1.0.x к Joomla! 1.5.x в таблицах изменились только два поля.

Однако что касается дополнительных компонентов, то здесь часто обновления не проводятся или проводятся лишь поверхностно. Операторы веб-сайта часто боятся новых инсталляций и потому продолжают работать со старыми версиями.

Безопасность

Если вы используете множество дополнительных компонентов, то часто оказываетесь в затруднительной ситуации. Например, команда разработчиков Joomla! рекомендует применять безопасный режим PHP, устанавливать `register_globals` в OFF, наряду с другими мерами, повышающими безопасность Joomla!.

Однако некоторые компоненты с такими установками не работают. Вам придется теперь принимать решение — то ли отказываться от использования этих компонентов, то ли смириться с определенной долей небезопасности.

Что делать?

Принимая такие решения, всегда думайте о последствиях взлома веб-сайта (и сервера) для вашего бизнеса. Насколько важна информация, хранящаяся на сервере?

В некоторых случаях может оказаться приемлемым поручить управление сайтом или почтовой системой третьей стороне, или хотя бы хранить там информацию о зарегистрированных пользователях, но не более. Из-за сложности компонентов возникают проблемы с безопасностью, и иногда этим пренебрегают независимые разработчики и операторы веб-сайтов.

Команда программы "Google Summer of Code" в этом году занимается разработкой системы обновления Joomla!, которая станет загружать обновления при необходимо-

сти и, возможно, даже автоматически устанавливать их. Система обновления подобного рода также сама по себе представляет объект для расширений.



Я не собираюсь отговаривать вас от использования в онлайн-версиях Release Candidates (RC) или даже бета-версий. Нам всем приходится жить в реальном мире, где иногда клиенты требуют этого. В зависимости от проекта, некоторые бета-версии бывают весьма стабильными. База данных MySQL долгое время существовала в состоянии бета-версии, и то же самое можно сказать о веб-сервере Apache. Просто помните об основных рисках! Когда только возможно, старайтесь работать со стандартными версиями Joomla!. Поступая подобным образом, вы обезопасите себя от неприятностей.

Резюме

В этой главе вы узнали больше о расширениях, а также о том, как использовать специализированные настраиваемые компоненты, доступные для Joomla!. И, наконец, были обсуждены проблемы, связанные с применением компонентов от независимых разработчиков. В следующей главе речь пойдет о создании собственных шаблонов.

13

Написание собственных шаблонов Joomla!

Для того чтобы настроить внешний вид вашего веб-сайта в соответствии с имиджем компании, необходимо модифицировать существующий шаблон или создать новый. В этой главе вы узнаете основы построения собственных шаблонов Joomla!.

Корпоративная идентичность

Корпоративная идентичность (Corporate Identity – CI) – понятие, касающееся внешнего образа предприятия. Эта внешность, т.е. идентичность, происходит из традиций предприятия или разрабатывается искусственно и навязывается руководством. Такая идентичность важна для создания у клиента впечатления от предприятия и обеспечения его узнаваемости.

Корпоративная идентичность включает:

- корпоративный имидж (цена, продукт, рекламная стратегия);
- корпоративный дизайн (визуальное представление);
- корпоративная коммуникация;
- корпоративное поведение (поведение сотрудников по отношению друг к другу и к внешнему миру).

Все эти области учитываются при разработке веб-сайта. В этой главе рассматривается корпоративный дизайн. Как минимум, он включает логотип, шрифт символов и цвета, используемые предприятием.

Посетители вашего веб-сайта должны узнавать ваше предприятие с первого же посещения.

HTML/ XHTML, CSS и XML

Аббревиатуры HTML/XHTML, CSS и XML касаются Интернет-технологий, с которыми работает Joomla!. Консорциум World Wide Web Consortium стандартизует эти технологии.

HTML/ХТМЛ

"Всемирная паутина" (World Wide Web) основана на HTML., который представляет собой не язык программирования, а язык описания текста.

Каждая часть текста состоит из структур типа заголовков, списков, выделенных курсивом областей, таблиц и многого другого. Язык HTML имеет дело с так называемыми дескрипторами. У дескриптора есть открывающая и закрывающая части. Например, заголовок первого уровня выглядит следующим образом:

```
<h1>Текст заголовка</h1>
```

Дескрипторы интерпретируются браузером, и текст отображается в соответствии с его важностью.

Выучить HTML очень легко, и на эту тему можно найти очень много онлайн-руководств. Разработка HTML на сегодняшний день прекращена. Наследником HTML стал XHTML версии 1.0.

CSS

Каскадные таблицы стилей (Cascading Style Sheets – CSS) – это расширение HTML. CSS также не является языком программирования, а словарем для определения свойств форматирования индивидуальных элементов HTML.

С помощью команд CSS можно, например, специфицировать, что заголовки первого уровня должны иметь шрифт Arial с размером символов в 18 пунктов, не полужирный, с расстоянием в 1,9 см между абзацами. Это невозможно сделать на чистом HTML, да и не нужно было, когда HTML только разрабатывался. С прогрессирующей коммерциализацией Интернета дополнительные возможности форматирования стали важными.

Данные CSS могут быть интегрированы в HTML следующими тремя способами.

В центральном файле HTML

Команды CSS определены в разделе заголовка файла HTML, как показано ниже:

```
<head>
<title>Текст заголовка</title>
<style type="text/css">
<! --
/* ... здесь определены команды CSS... */
-->
</style>
</head>
```

В отдельном файле CSS

Если команды CSS применяются к нескольким файлам HTML, они могут размещаться в отдельном файле, и путь к этому файлу указывается в заголовке HTML. Вот версия, используемая Joomla!:

```
<head>
<title>Текст заголовка</title>
<link rel="stylesheet" type="text/css" href="formats.css">
</head>
```

Внутри дескриптора HTML

Команды CSS могут также помещаться внутри дескриптора HTML:

```
<body>  
<h1 style="... команды CSS ..." >...</h1>  
</body>
```

Комбинации

Все перечисленные три метода могут комбинироваться без каких-либо проблем в дном HTML-файле. Например, допускается последовательно переписать команды CSS, которые были определены в центральном файле, в дополнительном коде на странице HTML. Такая практика, однако, быстро ведет к запутанным структурам; лучше все-таки подкорректировать центральный файл.

XML

Расширяемый язык разметки (Extended Markup Language – XML) сам по себе универсален. Он представляет собой мета-язык, и ведет свою родословную от более сложного SGML (Standardized Generalized Markup Language – стандартизованный обобщенный язык разметки), который был разработан еще в шестидесятых годах. XML часто используется для конфигурационных файлов и в качестве формата обмена. Для наших целей XML понадобится как язык описания метаданных шаблонов, которые вы хотите создавать. Эти метаданные, прежде всего, важны для инсталлятора шаблонов (Template Installer) и отображения в диспетчере шаблонов (Template Manager).

В принципе, эти данные также состоят из открывающего и закрывающего дескриптора. Например:

```
<name>Joomla! Book</name>
```

Разница между HTML и XML состоит в том, что в XML никаких дескрипторов не предусмотрено. По этой причине вы полностью свободны в организации структуры и именовании ваших дескрипторов.

Создание ваших собственных шаблонов

Итак, требуется создать собственный шаблон. Существуют несколько вещей, которые нужно продумать, прежде чем создавать завершенные пакеты шаблонов. Давайте двигаться постепенно – шаг за шагом.

Концепция

Прежде чем приступить к работе, вы должны подготовить концепцию. Работа начинается с наброска диаграммы – особенно при построении шаблонов. Это ваше дело, как именно вы будете создавать этот набросок: в программе редактирования изображений вроде Adobe Photoshop, в Microsoft Paint, поставляемом вместе с Windows, в программе с открытым кодом GIMP или просто карандашом на куске бумаги.

Фиксированный размер, переменный (текущий) или то и другое

Вы можете создавать шаблоны двух типов – шаблоны, которые адаптируют свою структуру к размеру окна браузера, и шаблоны, имеющие фиксированный размер. Примером гибкой компоновки может быть ситуация, когда ваша страница растягивается при экране шириной 2048 пикселей и развернутом окне браузера. Это может

показаться странным, если вы используете в шаблоне графический, не масштабируемый элемент вроде логотипа или подписи. Вы не можете контролировать то, как он будет выглядеть.

Другой выбор основан на определенном разрешении и позиционировании всех элементов в точности по пикселям шаблона. Это обладает тем преимуществом, что веб-страница всегда выглядит так, как вы хотите. К сожалению, вы не можете заранее знать разрешения монитора, на котором будет отображаться страница. Если монитор имеет разрешение 800×600 пикселей, страница заполнит весь экран. На более крупных экранах с разрешением 1600×1200 пикселей она займет около четверти поверхности экрана, и будет выглядеть несколько потерянной на этом пространстве.

Вы должны взвесить все "за" и "против" и принять то или иное решение, или же рассмотреть решение, свободное от барьеров (см. главу 14), и предоставить обе версии. Вам наверняка доводилось видеть веб-сайты, где можно даже изменять размер шрифта. В дополнение к кнопкам размера шрифта на них часто также присутствует кнопка для выбора разных компоновок.

Если вы предпочитаете фиксированный размер, то должны выбрать размер, который выглядит презентабельно на большинстве экранов, другими словами – 800×600 пикселей. Поскольку браузер имеет справа линейку прокрутки, и окно браузера оснащено рамкой, доступное пространство окажется даже еще меньше, т.е. в вашем распоряжении будет максимум 780 пикселей в ширину.

Структура

Вы имеете дело со структурированными данными и, прежде всего, должны позаботиться об общем размещении контента. Joomla! обычно использует структуру, показанную на рис. 13.1.

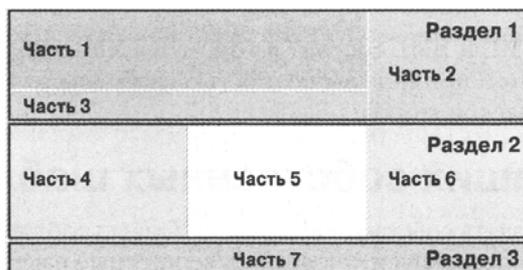


Рис. 13.1. Общая структура страницы Joomla!

Раздел 1:

- часть 1: здесь размещается логотип или картинка и имя веб-сайта;
- часть 2: здесь находится поле поиска;
- часть 3: здесь находятся связанные пути навигации (Breadcrumbs).

Раздел 2:

- часть 4: наиболее важные меню отображаются в левой колонке;
- часть 5: контент страницы попадает сюда;
- часть 6: правая колонка – место для дополнительных меню.

Раздел 3:

- часть 7: нижний колонтитул.

Преобразование в HTML

Теперь вы должны преобразовать эту концепцию в HTML и CSS. В зависимости от программы редактирования графики, которую вы используете для ее создания, существует возможность автоматического экспорта изображения в код HTML. Преобразование можно также выполнить вручную в текстовом редакторе, в редакторе HTML типа Dreamweaver (<http://www.adobe.com/products/dreamweaver>) или в одном из многочисленных бесплатных редакторов HTML (<http://www.thefreecountry.com/webmaster/htmleditors.shtml>).



<table> или <div>?

Дескриптор <div> — замена и дополнение дескриптора <table> в HTML.

С помощью этого дескриптора можно заключить несколько элементов HTML, таких как текст и графика, в одну общую область. Эта общая область не делает ничего, кроме того, что начинает новую строку непрерывного текста. Дескриптор <div> не имеет никаких других свойств. Однако использование комбинации дескрипторов <div> с командами CSS дает значительные преимущества. <div> был специально разработан для того, чтобы его можно было форматировать командами CSS.

До 2004 г. было принято определять структуру веб-сайта обильным применением таблиц HTML. По мере роста популярности CSS и HTML-элемента <div> и появления способных интерпретировать их браузеров, все больше и больше шаблонов стали структурироваться без использования таблиц HTML. Тем не менее, редко можно встретить веб-сайты, которые содержат только семантически корректный HTML и имеют компоновку, на 100% свободную от таблиц. Первый шаг в структурировании вашего веб-сайта состоит в использовании дескрипторов <div>.

Joomla! 1.5 также постепенно отходит от "табличного" пути и начинает доставлять семантически корректный HTML. Однако она оставляет возможности структурировать компоновку сайта с использованием таблиц HTML. Во включенном шаблоне `rhuk_milkyway` нет табличных дескрипторов; вся табличная структура создана посредством дескрипторов <div>. Взгляните на исходный код этого шаблона, чтобы познакомиться с этой технологией. Dreamweaver также поддерживает эту технологию.

Исходный код преобразования HTML выглядит немного похожим на приведенный ниже листинг. Код намеренно оставлен простым и он *не* соответствует стандарту XHTML в заголовке. Именем этого файла компоновки должно быть `index.php`, поскольку Joomla! ищет встроенные команды PHP.

Вот как выглядит содержимое HTML-файла `/index.php`.

```
<html>
<head>
  <link href="/joomla150/templates/joomla150buch/css/template.css"
    rel="stylesheet" type="text/css"/>
</head>
<body>
  <div id="part1">header / header<br /><br />
  <div id="section1">section1</div>
  <div id="section2">section2</div>
</div>
  <div id="part2">main display area / main<br /><br />
  <div id="section3">breadcrumbs</div>
  <div id="section6">right side</div>
  <div id="section4">left side</div>
```

```

    <div id="section5">content</div>
  </div>
  <div id="part3">footer / footer<br /><br />
  <div id="section7">section7</div>
</div>
</body>
</html>

```

Приведенный далее файл CSS из индивидуального шаблона интегрирован в область заголовка приведенного кода. Пока этот файл CSS содержит только одну команду, которая определяет гарнитуру шрифта.

Ниже показано содержимое CSS-файла /css/template.css.

```

body {
font-size: 12px;
font-family: Helvetica,Arial,sans-serif; }
#Part1{ /*header*/
float: left;
border: 2px dotted green; }
#Part2{ /*main*/
float: left;
border: 2px dotted yellow; }
#Part3{ /*footer*/
clear:all;
border: 2px dotted red; }
#Section1{ /*top right*/
float: left; width: 18em;
margin: 0 0 1.2em;
border: 1px dashed silver; background-color: #eee; }
#Section2{ /*top left*/
float: right; width: 12em;
margin: 0 0 1.1em;
background-color: #eee; ccrder: 1px dashed silver; }
#Section3{ /*breadcruobs*/
border: 1px dashed silver;
background-color: #eee; }
#Section4{ /*left side*/
float: left; width: 15em;
margin: 0 0 1.2 em;
border: 1px dashed silver; }
#Section5{ /*content*/
margin: 0 12em 1em 16em;
padding: 0 1em;
border: 1px dashed silver; }
#Section6{ /*right side*/
float: right; width: 12em;
margin: 0 0 1.1em;
background-color: #eee; border: 1px dashed silver; }
#Section7{ /*footer*/
margin: 0 0 1.1em;
background-color: #eee; border: 1px dashed silver; }

```

Вы создадите этот первый шаблон вручную в каталоге Joomla!. Когда шаблон будет готов, вы сможете превратить его в сжатый инсталляционный пакет, который затем будет инсталлирован кем-то другим (либо вами) с помощью инсталлятора Joomla!.

Сохраните HTML-файл компоновки под именем `index.php` во вновь созданном каталоге `[Путь_к_Joommla]/templates/joomla150book/`. Файл `template.css` сохраните в каталоге `[Путь_к_Joommla]/templates/joomla150book/css/`.

Базовая структура шаблона готова. Теперь потребуется определить шаблон более точно для Joomla! с помощью XML-файла, чтобы он отображался в разделе администрирования шаблона.

Структура каталога шаблона

Теперь стоит позаботиться о соблюдении некоторых соглашений. Как уже говорилось ранее, шаблон должен быть сохранен в определенной структуре каталогов:

```
[Путь_к_Joommla]/templates/[имя_шаблона]/  
[Путь_к_Joommla]/templates/[имя_шаблона]/CSS/  
[Путь_к_Joommla]/templates/[имя_шаблона]/images/
```

Имя шаблона не может содержать пробелов и других специальных символов. Когда шаблон позднее будет инсталлирован как пакет, `Template Installer` должен будет создать каталог с этим именем. В зависимости от операционной системы, экзотические комбинации символов могут стать причиной проблем. Вдобавок имя должно быть осмысленным. Здесь в качестве имени шаблона выбрано `joomla150book`.

В каталогах шаблона должны быть представлены различные файлы с предопределенными именами.

- **Файл компоновки (Layout File)**. Это HTML-файл, который был создан ранее — `/templates/joomla150book/index.php`. Он должен заканчиваться на `.php`, поскольку элементы динамического модуля Joomla!, которые мы вставим позднее, должны интерпретироваться PHP.
- **Изображение предварительного просмотра (Preview Picture)**. Файл `/templates/joomla150book/template_thumbnail.png` содержит изображение предварительного просмотра вашего шаблона, которое предназначено для раздела предварительного просмотра при администрировании Joomla! в меню `Extensions⇒Template Manager (Расширения⇒Диспетчер шаблонов)`. Изображения предварительного просмотра имеют формат примерно `200×150` пикселей. Вы можете создать этот файл позднее, когда увидите готовый шаблон.
- **Метаданные шаблона (Metadata of the Template)**. Файл `/templates/joomla150_book/templateDetails.xml` представляет техническое руководство для инсталлятора шаблонов и содержит инсталляции для выбора шаблона в диспетчере шаблонов. Здесь вы специфицируете место, куда должны копироваться файлы, кто их автор, а также дополнительные метаданные о шаблоне. Во время последующей инсталляции этого файла инсталлятором Joomla!, PHP читает его и копирует файлы в места, специфицированные XML-файлом. Для примера шаблона можно воспользоваться файлом из приведенного ниже листинга (`templateDetails.xml`), наполнив его собственными данными. Для каждого файла, который применяется в шаблоне, соответствующий контейнер XML должен быть наполнен именем файла и корректным путем.

```
<files>  
  <filename> ... имя файла из каталога TemplateRoot ...  
  </filename>  
  <filename> ... для каждого файла предусмотрен контейнер filename  
  </filename>  
</files>
```

Другие контейнеры XML-файла предназначены для описания шаблона. Ниже показано полное содержимое XML-файла.

```
templateDetails.xml:
<install version="1.5" type="template">
<name>joomla150book</name>
<version>1.0</version>
<creationDate>11.11.2007</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>
<filenarae>css/template.css</filenarae>
</files>
</install>
```

Создайте также файл `templateDetails.xml` в каталоге `[Путь_к_Joomla]/templates/joomla150book/`.

- **CSS-файл (CSS File).** Для разрабатываемого шаблона можно использовать несколько CSS-файлов. Какое имя вы дадите файлу CSS, и как создадите его — дело ваше. Однако для разных элементов CSS предусмотрены разные каталоги. Для первой попытки вам понадобится CSS-файл по имени `/templates/joomla150_book/css/template.css`.
- **Графика, изображения (Graphics, Images).** Здесь можно ввести определенные пользователем файлы изображений, которые понадобятся в шаблоне. Инсталлятор затем скопирует эти файлы в папку `images`. Имя файла выглядит как `/templates/joomla150_book/images/[определенный_пользователем_файл_изображения]`.

Первый пробный запуск

Воспроизведя все структуры в подкаталоге `[Путь_к_Joomla]/templates/`, вы сможете увидеть новый шаблон в меню `Extensions⇒Template Manager` области администрирования Joomla!, и сделать его шаблоном по умолчанию (рис. 13.2).

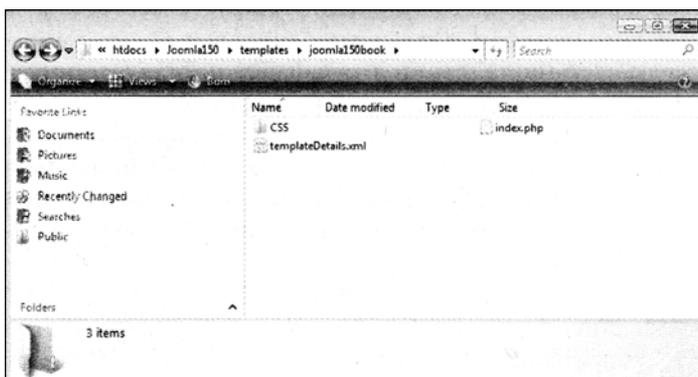


Рис. 13.2. Новый шаблон

Когда обратитесь к своему веб-сайту, то увидите новый шаблон. К сожалению, контент еще не виден. Поскольку контент генерируется динамически, вы должны интегрировать его по частям в новый шаблон.

Интеграция модуля Joomla!

Интеграция модуля Joomla! происходит посредством команд, встроенных в код HTML. Joomla! использует пространство имен `jdocus` для интеграции различных элементов в шаблон. Если вы вставите следующую выделенную строку в заголовок файла компоновки:

```
<head>
<jdocus:include type="head" />
</head>
```

то заголовок сайта и символ ленты новостей уже будут отображаться корректно (рис. 13.3).

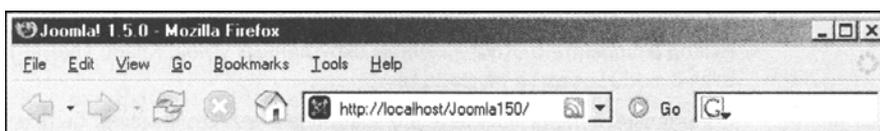


Рис. 13.3. Заголовок и символ ленты новостей на сайте

Если вы откроете исходный код сайта, то заметите, что Joomla! скопировала все метаданные, введенные вами в административной области, в код HTML. Вдобавок с помощью дескрипторов ссылки были интегрированы наполнения RSS, и они будут отображаться соответствующими символами в браузерах вроде Firefox, которые поддерживают эту технологию.

Вот как выглядят метаданные Joomla!:

```
...
<head>
<base href="http://localhost/joomla150/" />
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, joomla!, Joomla,
Joomla!, J!" />
<meta name="description" content="Joomla! - dynamic portal-engine
and Content-Management-System" />
<meta name="generator" content="Joomla! 1.5 - Open Source Content
Management" />
<title>Welcome to the FrontPage</title>
<link href="/joomla150/index.php?format=feed&type=rss"
rel="alternate"
type="application/rss+xml" title = "RSS 2.0" />
<link href="/joomla150/index.php?format=feed&type=atom"
rel="alternate"
type="application/atom+xml" title="Atom 1.0" />
<script type="text/javascript"
src="/joomla150/media/system/js/mootools.js"></script>
<script type="text/javascript"
src="/joomla150/media/system/js/caption.js"></script>
...
```

Поскольку это работает хорошо, мы не станем тратить время на рассмотрение других вставок дескриптора `jdoc`. Например, команда `<jdoc:include type="modules" name="top" style="none" />` ожидает тип вставки как параметр `type`, в нашем случае – `modules`. Параметр `name` задает позицию модуля (`top`, `right`, `left`, `user1...`) на веб-сайте. Эту позицию можно назначить модулю через меню `Extensions⇒Template Manager` (Расширения⇒Диспетчер модулей). Наконец, параметр `style` содержит значение, которое принадлежит типу HTML-кода, доставляемого модулем. Например, `none` обеспечивает чистый HTML-вывод, не окружая его дескрипторами `<div>`. В табл. 13.1 перечислены параметры модуля.

Таблица 13.1. Параметры модуля

Параметр	Вывод
<code>table</code>	Модуль отображается в таблице
<code>horz</code>	Модуль отображается горизонтально в ячейке окружающей таблицы
<code>xhtml</code>	Модуль выводится в XHTML-совместимый элемент <code><div></code>
<code>rounded</code>	Вывод в формате с отображением скругленных углов. Класс элемента меняется с <code>moduletable</code> на <code>module</code>
<code>beezDivision</code>	Специальный переключатель для шаблона <i>Beez</i> . Этим может быть изменен размер заголовка. Вне шаблона <i>Beez</i> не доступен
<code>none</code>	Модуль выводится без какого-либо форматирования

Описание вставок `jdoc` в файле `index.php` дано в следующем листинге. Содержимое файла `index.php` со вставками `jdoc`:

```
<html><head>
<jdoc:include type="head" />
<link href="/joomla150/templates/joomla150book/css/template.css"
      rel="stylesheet"
type="text/css" />
</head>
<body>
<div id="Part1">header / header
<div id="Section1">Section1 <jdoc:include type="modules"
      name="user3" style="xhtml" />
</div>
<div id="Section2">Section2 <jdoc:include type="modules"
      name="top" style="xhtml" />
</div>
</div>
<div id="Part2">main display area / main
<div id="Section3">breadcrumbs <jdoc:include type="modules"
      name="breadcrumb"
style="xhtml" /></div>
<div id="Section6">right side <jdoc:include type="modules"
      name="right"
style="xhtml" /></div>
<div id="Section4">left side <jdoc:include type="modules" name="left"
style="xhtml" /></div>
<div id="section5">content <jdoc:include type="component" style="xhtml"/></div>
</div>
```

```
<div id="Part3">footer /footer
<div id="Section7">Section7<jdoc:include type="modules" name="footer"
style="xhtml" />
</div>
</div>
</body></html>
```

Если вы откроете веб-сайт с этим модифицированным кодом HTML на локальном сервере, то уже сможете видеть динамический контент. Новый шаблон будет заполнен всеми данными. Визуальный аспект результата может еще потребовать некоторых усовершенствований, но в целом концепция уже работает.

Теперь потребуются упаковать шаблон в инсталляционный пакет.

Создание пакета шаблона

Чтобы передать созданный шаблон другим разработчикам, его необходимо упаковать в ZIP-архив. Прежде чем сделать это, создайте изображение для предварительного просмотра шаблона (`template_thumbnail.png`) размером 227×162 пикселя. Это изображение должно быть показано при наведении курсора мыши на имя-ссылку в оазделе выбора шаблонов.

Упакуйте все файлы из каталога `joomla150book` и всех его подкаталогов в ZIP-архив. Вдобавок выберите все файлы и папки в папке `[путь_к_Joomla]/templates/joomla150book/` и упакуйте их в файл `joomla150book.zip`. Создайте резервную копию этого файла и первого ZIP-файла.

Деинсталляция шаблона

Теперь можно передать этот пакет другим либо инсталлировать его самостоятельно. Если вы хотите попробовать инсталляцию, нужно будет предварительно удалить вновь созданный шаблон из системы Joomla!.

Чтобы удалить шаблон, сначала необходимо назначить в качестве шаблона по умолчанию другой шаблон. Щелкните на `Extensions⇒Template Manager` выберите желаемый шаблон и щелкните на пиктограмме `Default` (По умолчанию).

Сейчас можно деинсталлировать шаблон. Щелкните на `Extensions|Install/Uninstall` и перейдите на вкладку `Templates` (Шаблоны). Выберите вновь созданный шаблон `joomla150book` и щелкните на пиктограмме `Uninstall` (Деинсталлировать).

Здесь можно инсталлировать ZIP-пакет. Выберите файл `joomla150book.zip` и инсталлируйте его щелчком на кнопке `Upload File & Install` Загрузить файл и инсталлировать). Инсталлятор сообщит, что инсталляция выполнена успешно, и в этом сообщении также будет отображено описание из XML-файла.

Какой исходный код поступает от Joomla?

После того, как мы преодолели самый сложный участок, создав шаблон, давайте взглянем на HTML-код, который предоставляет Joomla!.

Joomla!, естественно, создает код, который вставляется в созданный нами контейнер. Этот код состоит из HTML, "приправленного" классами и идентификаторами (ID) CSS. Помимо прочего, Joomla! включает два файла CSS, описывающих определенные классы и идентификаторы, которые вставляются в каждый шаблон.

Этими файлами являются `/templates/system/css/system.css` и `/templates/system/css/general.css`.

Они вставляются в область <head>:

```
<link rel="stylesheet" href="<?php echo $this->baseurl ?>
/templates/system/css/system.ess" type="text/css" />
<link rel="stylesheet" href="<?php echo $this->baseurl ?>
/templates/system/css/general.ess" type="text/css" />
```

Оба эти файла содержат классы и идентификаторы, к которым мы будем обращаться далее на протяжении этой главы.

Позиция левого модуля в деталях

Давайте внимательной присмотримся к панели левого модуля, показанной на рис. 13.4.

Ниже представлен соответствующий HTML-код в файле компоновки index.php:

```
<div id="Section4">left side <jdoc:include type="modules"
name="left" style="xhtml" />
```

Дескриптор <div> для этого раздела формируется CSS ID по имени Section4.

Если вы взглянете на доставленный HTML-код в браузере (выполните щелчок правой кнопкой мыши на контенте сайта и выберите в контекстном меню пункт Display Source Code (Исходный код страницы)), то увидите, конечно, намного больше кода, чем было создано запросом jdocs.

Вот как выглядит фрагмент этого доставленного исходного кода HTML:

```
... дополнительные команды HTML
<div id="Section4">left side
<div class="moduletable_menu">
<h3>main menu</h3>
<ul class="menu">
<li id="current" class="active item1">
<a href="http://localhost/joomla150/">front page</a>
</li>
<li class="item2">
<a href="/joomla150/joomla-license">Joomla! License</a>
</li>
</ul>
</div>
```

... дополнительные команды HTML

Рис. 13.4. Внешний вид модуля

В этом коде достойны упоминания CSS-классы active item1 и active item2. Дополнительные суффиксы CSS вы найдете в одном из CSS-файлов шаблона по умолчанию rhuk_milkyway:

```
div.module_menu h3 {
font-family: Helvetica, Arial, sans-serif;
font-size: 12px;
font-weight: bold;
color: #eee;
margin: -23px -4px 5px -5px;
padding-left: 10px;
padding-bottom: 2px;
}
```

```

div.module_menu {
margin: 0;
padding: 0;
margin-bottom: 15px;
}
div.module_menu div div div {
padding: 10px;
padding-top: 30px;
padding-bottom: 15px;
width: auto;
}
div.module_menu div div div div {
background: none;
padding: 0;
}
div.module_menu ul {
margin: 10px 0;
padding-left: 20px;
}
div.module_menu ul li a:link, div.module_menu ul li a:visited {
font-weight: bold;
}

```

В данном случае блоки CSS заботятся о скругленных углах в оформлении меню. В файле CSS присутствуют дополнительные элементы, которые форматируют заголовки третьего уровня (<h3></h3>) и несортированный список (ссылки меню) для данного случая.

Пока все хорошо. Такого типа форматирование вполне удовлетворительно, и вывод необходимых дескрипторов <div> также может быть изменен атрибутами style. Мы применяли параметр style xhtml в рассмотренном примере шаблона: в этом шаблоне используется параметр style rounded. Он доставляет соответствующий дескриптор <div> из четырех частей.

Создание кода такого типа хорошо подходит для форматирования CSS. Если вы продолжите читать исходный код, то вновь встретите там таблицы. Компонент контента по-прежнему работает с таблицами! И здесь начнется проблемы, если вы захотите форматировать контент отдельно от таблиц. Однако Joomla! также предлагает готовое решение и для этого.

Переопределение шаблона (настройка вывода HTML без изменения файлов ядра)

Чтобы решить проблем с отображением таблиц, обычно нужно было бы изменить файлы ядра Joomla!. Контент тогда можно было бы избавить от таблиц, но этот файл мог бы быть переписан следующим обновлением.

Начиная с Joomla! 1.5.0, появились так называемые представления (views), которые предлагают решение этой проблемы: вы найдете множество различных представлений для отображения компонентов логики. В компоненте com_content, например, это archive, article, category, frontpage и section. Эти пять папок содержат файлы, которые делают доступными различные представления контента. Я надеюсь, что вы помните о представлениях таблицы и блога. Каждая из этих пяти папок содержит папку tmpl, которая, в свою очередь, содержит различные файлы, отвечающие за отображение (рис. 13.5).

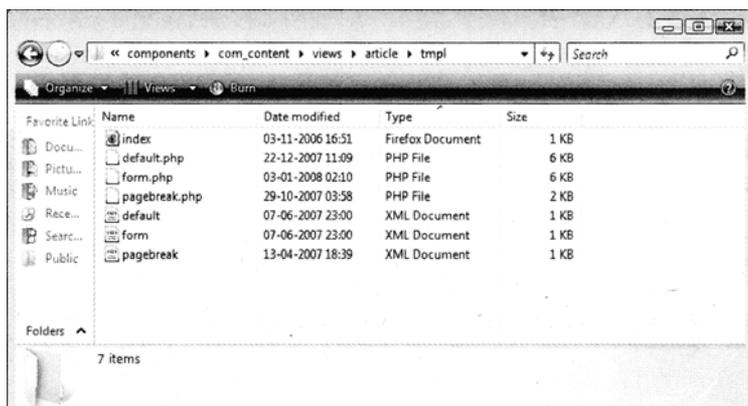


Рис. 13.5. Содержимое папки *tmpl*

Чтобы двигаться далее по маршруту, понадобится дополнительный каталог в каталоге вашего шаблона. В данном примере этот каталог должен называться `com_content/article`, поскольку вы хотите изменить компоновку компонента `content` в типе отображения `article`.

Другими словами, потребуется скопировать файлы из

`[Путь_к_Joomla]/components/com_content/view/article/tmpl/*.*`

в ваш новый каталог шаблона и затем осуществить необходимые изменения:

`[Путь_к_Joomla]/templates/[ВашШаблон]/html/com_content/article`

Это легко дублировать в шаблоне `Beez`. Упомянутые файлы здесь находятся в

`[Путь_к_Joomla]/templates/beez/html/com_content/article`

и, таким образом, переписывают (переопределяют) файлы ядра, не изменяя их.

Joomla! ищет эти файлы в каталоге шаблона. Если они там есть, они используются. Если же их там нет, используются файлы ядра компонентов.

Резюме

Эта глава была посвящена вопросам создания собственных шаблонов. В следующей главе рассматривается концепция свободы от барьеров в Joomla!.

14

Свобода от барьеров и Joomla!

Энджи Радтке (Angie Radtke)

Когда Хаген попросил меня написать эту главу для его книги, мне стало не по себе; ведь до срока сдачи оставалось очень мало времени.

Но потом, после зрелых размышлений я успокоилась. Большая часть материала этой главы уже изложена в детальной форме в книге *Barrier-free Web Design*, которую я написала вместе с Майклом Чарлиером (Michael Charlier).

Мне поручена только одна глава в книге Хагена, поэтому мне придется ограничиться только самыми важными вещами. Я изложу основы свободного от барьеров веб-дизайна и опишу концепцию, положенную в основу стандартного шаблона Beez, с помощью которого можно создавать свободные от барьеров веб-сайты в Joomla!.

Вопрос о том, как свобода от барьеров может сочетаться с Joomla!, вышел на первый план, начиная с версии 1.0.3. Это было возможно уже тогда, но объем работы, которую для этого приходилось выполнять, был огромным. Я хочу заверить вас, что теперь ничто не препятствует в создании свободного от барьеров контента веб-сайтов.

Что такое свобода от барьеров?

Сеть Интернет в наши дни стала для большинства людей чем-то само собой разумеющимся. Информация со всего мира постоянно доступна. И что, возможно, еще важнее — так это то, что предложения соседнего магазина "все для дома", режимы работы государственных учреждений, да и целые телефонные книги городов могут быть легко вызваны на экран вашего домашнего монитора. Теперь нет необходимости тратить часы на посещение библиотеки или телефонные звонки для поиска нужной информации. Просто загляните в Web.

Однако, как выяснилось, не все способны воспользоваться преимуществами этой технологии. В частности, люди с физическими и умственными недостатками, которые испытывают трудности в адаптации к общественной жизни, могли бы получить колоссальный выигрыш от применения этой технологии коммуникаций. Тем не менее, их часто останавливают барьеры, выражающиеся в затруднении или даже невозможности получить необходимую информацию. Большинство этих барьеров можно преодолеть при правильном проектировании технологий.

Операторы онлайн-магазинов или банков, которые предоставляют услуги Интернет-банкинга, должны быть знакомы с этой не такой уж и малочисленной целевой группой.

Целью свободного от барьеров веб-дизайна является сделать контент и взаимодействие во "Всемирной паутине" доступными для как можно большего числа групп пользователей и терминалов.

Есть люди с физическими недостатками, которые затрудняют для них доступ в Интернет. Веб-дизайн, свободный от барьеров, часто называют "Интернетом для слепых". Однако я хочу подчеркнуть, что это — не все; на самом деле это лишь малая часть концепции свободы от барьеров. Я не раз спрашивала себя, почему такой взгляд столь широко распространился. Скорее всего, дело в том, что монитор — неотъемлемый признак компьютера, и если вы не можете видеть, то не можете использовать монитор. Однако в процессе ежедневной деятельности я заметила, что плохо видящие люди легче справляются с труднодоступными сайтами, чем люди с другими физическими недостатками.

Слабо видящие люди — это те, у кого остаточный процент зрения составляет лишь небольшую часть от зрения среднестатистического человека. Некоторые из них в состоянии воспринимать текст и распознавать цветовые установки, увеличивая размер шрифта, в то время как другие вынуждены пользоваться средствами акустического вывода информации либо воспринимать текст наощупь через устройства вывода текста Брайля.

Процент людей с менее значительными дефектами зрения намного выше.

Примерно одна четверть всех людей трудоспособного возраста жалуются на различного рода аметропию (дефекты зрения вроде близорукости, дальнозоркости, астигматизма и т.п.). Этот процент значительно возрастает по мере старения. Некоторые из этих дефектов могут быть эффективно компенсированы использованием очков, другим же очки помогают лишь отчасти. Такие дефекты зрения, как катаракта и глаукома, могут быть устранены лишь хирургическим путем. Другие — типа пигментного ретинита или диабетического ретинита — приводят к постепенной потере зрения, вплоть до полной слепоты. Эффект "туннельного зрения" выражается в значительном сужении поля зрения, иногда до размера небольшой монеты, помещенной на расстоянии вытянутой руки.

Около 10% мужчин страдают от цветовой слепоты. Это обычно выражается в том, что они не могут различить определенные оттенки красного и зеленого. Невосприимчивость к другим цветам, полная цветовая слепота или слепота к красному и зеленому среди женщин встречается очень редко.

Еще одна потенциальная группа пользователей имеет проблемы с распространенными технологиями ввода. Не все могут оперировать мышью или пользоваться клавиатурой.

На то имеется масса причин: неподвижность конечностей или пальцев, либо их спонтанная подвижность, которую трудно контролировать. Кто-то просто не имеет рук или парализован от шеи вниз — целиком или наполовину. Но до тех пор, пока человек в состоянии вводить хотя бы двоичные сигналы — общеизвестные 0 и 1 — он может научиться работать с компьютером, обладая энергией и настойчивостью, а также соответствующим программным обеспечением.

Многие люди в мире страдают от недостатков слуха. Тысячи из них испытывали трудности в изучении языка — настолько, что остаются здесь на уровне знаний пятого или шестого класса школы. Это вызывает необходимость в предоставлении более понятного текста.

Когда эти люди общаются друг с другом, они применяют язык жестов. Они используют тот этот язык для понимания иностранного или более трудного контента. Язык жестов — это система жестов, которая в значительной мере независима от устного языка.



Не только плохо слышащие люди путешествуют по Web с выключенными динамиками компьютеров! И поэтому, например, недостаточно выводить только звуковые предупреждающие сигналы; их всегда следует подкреплять ясно воспринимаемыми визуальными сигналами.

Чем больше Интернет проникает во все области жизни, тем более существенными становятся ситуационные затруднения: медленное соединение в отеле, неудобные условия освещения в движущемся поезде, необходимость работы с выключенным звуком.

От доступности веб-сайта выигрывают все. И не обязательно выполнять все правила, которых придерживаются государственные веб-сайты. Даже небольшие шаги в сторону повышения доступности сайта могут существенно повысить его дружелюбность к пользователям.

Благодаря своей популярности, Joomla! может оказаться в авангарде этого движения. С помощью стандартного шаблона Veez теперь относительно легко создавать доступные веб-сайты, которые могут быть использованы многими.

Законодательная база

Усилия по повышению доступности компьютеров для людей с недостатками предшествовали появлению Интернет. В декабре 1982 г. ООН издала документ *World Programme of Action (WPA)*, который подчеркнул важность доступа к современным технологиям для людей с физическими недостатками. Крупные компании, такие как IBM, Microsoft и Sun, внесли значительный вклад в достижение этой цели за последующие годы. В 1993 г., когда протоколу HTTP исполнилось два года, Генеральная Ассамблея ООН приняла резолюцию, требующую равных прав на доступ к информации и коммуникациям для людей с физическими недостатками. Вскоре ведущие страны приняли законодательные акты, направленные на реализацию этих требований.

С основанием W3C в 1994 г. появилась группа экспертов, приступившая к разработке руководств для безбарьерного доступа к веб-сайтам в дополнение к другим мерам по стандартизации. Эта работа была в основном завершена, когда в 1998 г. в США был принят законодательный акт *Section 508 of the Rehabilitation Act Amendment*, основанный на этих руководствах. С этого момента выполнение определенных требований доступа стало требованием закона для государственных учреждений США и их поставщиков информации. Инициатива Web Accessibility Initiative WAI от W3C, наконец, привела к выпуску в 1999 г. набора руководств под названием *Web Content Accessibility Guidelines 1.0 (WCAG1.0)*. Эти руководства, почти в неизменной форме, стали основой для положения BITV (*Regulation for Barrier Free Information Technology*) закона Германии о равных возможностях для людей с физическими недостатками от 2002 г., а также множества других законодательных актов во многих странах по всему миру. За годы, прошедшие после смены тысячелетия, WAI разработала дополнительные руководства, которые касаются, например, разработки браузеров, поддерживающих специальные возможности, а также других пользовательских агентов (LAAG), в отношении авторских инструментов (ATAG).

Некоторые положения WCAG1 от 1998 г. несколько устарели в наши дни, и уже не подходят в качестве основы для дальнейшей работы.

Работа над WCAG2, которая должна была завершиться в 2001 г., все еще не окончена. Первый набросок WCAG2, который был опубликован в середине января 2007 г., в настоящее время еще активно обсуждается.

Канадский эксперт по доступу Джо Кларк (Joe Clark) опубликовал свою работу *WCAG Samurai Errata* в июне 2007 г., в которой изложил поправки к действующему WCAG1.0.

Таким образом, WCAG1 /BITV, несмотря на все недостатки, остается на сегодняшний день единственным руководством по созданию доступных веб-сайтов. Поэтому мы рекомендуем всем, кто работает в этой области, не только полагаться на рекомендации (за несколькими исключениями) этого документа, но также принять участие в открытом обсуждении версии WCAG2.

WCAG1 содержит четырнадцать руководств, каждое из которых состоит из нескольких пунктов. Эти пункты делятся на три уровня приоритетов — согласно категориям "обязано соответствовать", "должно соответствовать" и "может соответствовать". В зависимости от уровня соответствия — первого, второго и третьего, или всех трех категорий — веб-сайт может помечать себя как А, АА или ААА.

Общие положения о критериях доступных сайтов

Давайте рассмотрим эти критерии.

Отделение контента от компоновки

Первое и наиболее важное правило для разработчика — отделять контент от компоновки. Это обеспечивает:

- чистый и ясный код HTML для контента;
- отсутствие излишних таблиц данных;
- форматирование исключительно посредством CSS;
- логичная семантическая структура;
- метки для прыжков.

Одним из наиболее важных требований для практической доступности является контент, который может быть выведен в линейном формате и сформатированный исключительно с помощью CSS.

Только при таком условии Assistive Technology (Услужливая технология) может подготовить контент надлежащим образом, игнорируя его визуальное представление. Перемещение презентации в таблицы стилей позволяет, например, пользователям со слабым зрением определять собственные таблицы стилей в своих браузерах, подгоняя страницы к своим индивидуальным потребностям. Для пользователя, применяющего программы чтения экрана, линейное отображение контента и соответствующая семантическая структура особенно важна.

Программы — читатели экрана схватывают контент веб-сайта сверху вниз, другими словами, линейно. Если на сайте присутствуют сложные таблицы данных, эта линейность нарушается.

Термин *семантический Web* поначалу трудно понять, но он важен при написании веб-контента. Например, читатели экрана позволяют пользователю перепрыгивать от заголовка на заголовок или от списка к списку, обеспечивая, насколько возмож-

но, быстрый обзор всего документа. Если веб-сайт не имеет никаких заголовков, эта пункция не может быть использована. Формальная структура веб-документа должна преимущественно следовать структуре контента. В зависимости от веб-проекта, выбор соответствующих заголовков может оказаться непростой задачей.

Метки для прыжков

Однако с линейным отображением контента связан большой недостаток. В некоторых случаях нужно покрыть большую территорию, чтобы добраться к "необитаемым" областям контента. При компоновке в три столбца, по крайней мере, больше разделов могут начинаться от верха, и глаз, ориентированный на визуальные подсказки, может быстро перепрыгнуть к области, обещающей наиболее ценную информацию. На помощь здесь может прийти концепция меток для прыжков. Она предоставляет виртуальное, не визуальное дополнение графической компоновки и позволяет пользователю, работающему с устройствами линейного вывода, идентифицировать области важного контента в начале страницы и затем немедленно перепрыгивать к интересующей его части.

Вставка виртуальных меток для прыжков в основном включает позиционирование дополнительных меню и начало каждой страницы с элементов внутренней навигации. В большинстве случаев имеет смысл скрыть эти меню в графической компоновке; для пользователей с нормальным зрением они будут раздражающим фактором — щелкать на ссылке, которая, как кажется, не ведет никуда, поскольку цель ссылки присутствует в видимой области браузера. Это меню прыжков не должно быть слишком длинным, и его структура должна выбираться очень тщательно: в конце концов, оно само по себе удлинняет и усложняет восприятие с учетом требований линеаризации. Вообще рекомендуется сделать основной контент целью для первого прыжка: постоянные посетители знакомы с сайтом и могут манипулировать навигацией, выбирая краткий путь туда, куда им надо.

Теперь очевидно, что сайты с более сложным контентом не только требуют графической компоновки, но также соответствующего дизайн.: контент.", нацеленного на его структуризацию — таким образом, чтобы не возникало излишних барьеров для пользователей с линейными устройствами вывода.

Требования к дизайну контента

Презентация веб-сайта — это нечто большее, чем просто симпатичный дизайн; она должна помогать посетителям в получении нужной им информации. Она отображает интерактивные опции и в то же время выражает корпоративную идентичность оператора сайта. Она ведет наблюдателя через наиболее важный контент в логическом порядке, помогая ему ухватить общую концепцию.

В терминах свободной от барьеров презентации особого внимания требуют следующие моменты:

- логическая конфигурация одержимого;
- продуманный выбор цветов;
- достаточные контрасты;
- изменяемый размер шрифта;
- масштабируемая компоновка;
- никаких графических шрифтов;
- никаких прозрачных фонов для графики;

- осмысленный альтернативный текст для графики;
- достаточно крупные элементы навигации;
- осторожная обработка событий мыши.

Визуальная и соответствующая тексту организация контента

Визуальная организация индивидуальных разделов страниц — центральный элемент презентации контента и проведения пользователя по вашему веб-сайту структурированным путем. Чем лучше структурирована и более логична презентация, тем легче пользователям найти свой путь по веб-сайту.

Структура контента столь же важна, как и структура творчества. Концепция контента влечет за собой определенную ответственность. Важнейший принцип в структурировании контента гласит следующее.

Всегда структурируйте контент так, как пользователь ожидает его видеть.

Это не всегда легко, потому что вы можете изменять свою точку зрения. Оператор веб-сайта знает свое предприятие и его "внутреннюю кухню" настолько подробно, что он склонен структурировать вещи со своей точки зрения — такой, которая, возможно, не имеет смысла для пользователя.

С годами в Web эволюционировали несколько узнаваемых структурных соглашений. Заголовок обычно содержит информацию о поставщике, общем назначении сайта, а также центральные элементы навигации, такие как контактная информация, передовая статья и дополнительная помощь в навигации вроде ссылки на карту сайта и функции поиска. Все это обычно помещается перед глазами пользователя — в центре титульной страницы и области зрения пользователя, и может быть вызвано в любой момент без каких-либо проблем.

Глаза читателей, привыкших двигаться слева направо, естественно, сканируют страницы слева направо и сверху вниз. По этой причине логотип обычно размещается слева вверх, откуда и начинается обзор. Пользователь ожидает найти его именно там.

Большинство пользователей ожидают видеть навигационные элементы слева. Эта концепция часто обсуждается, и разработчики говорят, что она уже всем надоела и нуждается в обновлении. Однако люди путешествуют в Web в соответствии с привычными шаблонами. Они обладают собственным опытом и реагируют соответственно. Знакомое расположение элементов сокращает время на обзор всего контента в поисках того, что интересует пользователя.

Выбор цветов

Выбор цветов особенно важен в контексте свободы от барьеров. Даже люди с ограниченной восприимчивостью к цветам должны иметь возможность выполнять навигацию по вашему веб-сайту.

Если вы переключите свою компоновку в черно-белый режим с градациями серого, то получите представление о том, как видят дальтоники, не различающие цветов (рис. 14.1). Напомню, что восприятие очень индивидуально и, в зависимости от серьезности проблем со зрением, может изменяться значительно. Множество дальтоников научились угадывать, какие на самом деле цвета имеют изображения, которые они видят. Например, они знают, что трава зеленая, и, сравнивая, они могут даже идентифицировать различные оттенки зеленого.

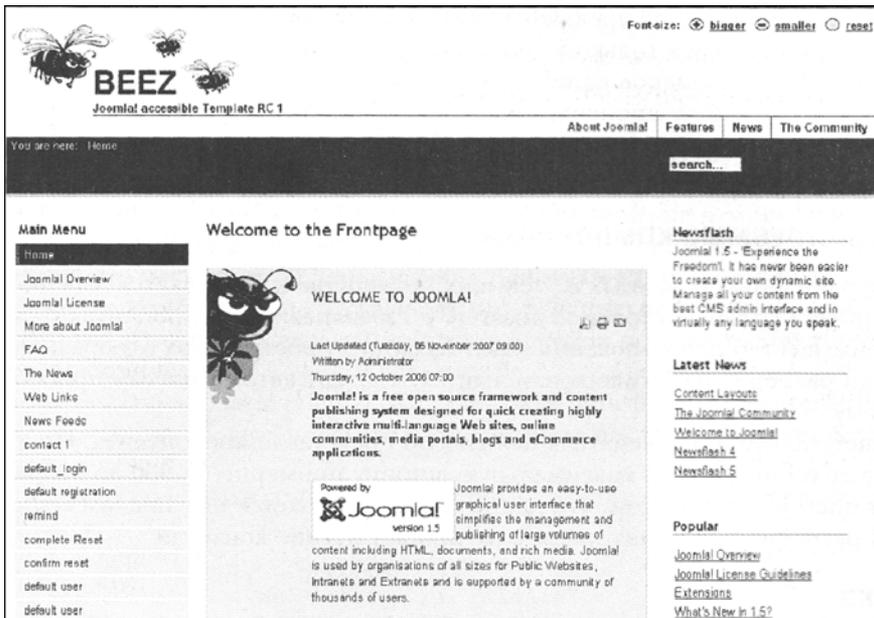


Рис. 14.1. Пример восприятия сайта дальтоником

Намного чаще, чем полная цветовая слепота, встречается так называемая слепота к красному-зеленому. Те, кто страдает ею, не могут отличить красный от зеленого из-за генетического отклонения. Комбинации цвета, которые содержат эти цвета, для них выглядят смутно.



Говоря простым языком, это значит: избегайте комбинаций цветов красного с зеленым и наоборот.

Контрасты

Цвета также важны для всех прочих дефектов зрения, поскольку не только сами цвета, а их сочетания — контрасты — могут помочь в использовании веб-сайта.

Цвета переднего плана и фона должны формировать четкий контраст между элементами текста, даже когда невозможно выбрать установки цвета и контрастности, подходящие для всех. Самый лучший цветовой контраст достигается черным текстом на белом фоне. Чтобы избежать эффекта ослепления, может быть, имеет смысл использовать светлую окраску фона. Множество людей с дефектами зрения нуждаются в очень четком контрасте, чтобы различать элементы контента на странице. Для них цветовые комбинации вроде белого текста на светло-оранжевом фоне не обеспечивают необходимого контраста. Для прочих, с другой стороны, сильные контрасты вызывают усталость, затрудняя чтение контента.

Изменяемые размеры шрифтов

Другое, также невероятно важное, правило заключается в том, что размеры шрифта должны быть изменяемыми.

Современные браузеры предлагают возможность увеличивать и уменьшать шрифты. Однако это работает только в том случае, если вы предоставим соответствующие спецификации для размеров шрифтов и не установили жесткого размера в пикселях. У вас есть выбор между процентной спецификацией и `em`-спецификацией. Обе эти спецификации также применимы в элементарно-специфичным размерам шрифтов, указываемом в свойстве `font-size`.

Масштабируемые компоновки

Веб-дизайн фиксированных и "текущих" компоновок различается. Спецификации ширины индивидуальных столбцов дизайна устанавливаются в процентах или `em`. Это дает преимущество использования максимального пространства отображения монитора. Если размер окна браузера изменяется, контент автоматически приспосабливается к нему.

В общем случае это замечательный способ представления контента. Но я считаю, что следует устанавливать максимальную ширину примерно от 950 до 980 пикселей. Это дает преимущество в том, что длина строки получается не слишком большой при высоких расширениях монитора, что затрудняет чтение контента.

Графика

Графика может включаться в веб-сайт различными способами. Вы можете вставлять ее в шаблон или непосредственно в действительную область контента.

Особенно важно для презентации, когда, например, вы вставляете логотип, избегать прозрачного фона. Представьте, что вы интегрируете логотип с черным текстом на прозрачном фоне.

Люди с определенными дефектами зрения иногда работают в инверсном режиме Windows, который, как правило, использует светлый шрифт на черном фоне (рис. 14.2).



Рис. 14.2. Инверсный режим Windows

Логотип *Beez* имеет белый фон и потому будет виден на черном фоне.

Ваш логотип (черный текст на прозрачном фоне) не будет видим, поскольку текст потеряется на черном фоне.

Вы должны избегать графических шрифтов вообще; в большинстве случаев следует быть готовыми предоставить текстовую альтернативу. Как правило, графические надписи не масштабируемы, и потому пользователь не может их подстроить под себя.

Иногда клиент может требовать использования определенного конкретного шрифта. В этом случае сделайте графику крупнее и затем примените спецификацию размера `em`, чтобы уменьшить его до желаемого размера. Таким образом, он, по крайней мере, может быть отчасти увеличен.

Также не забудьте предоставить осмысленную текстовую альтернативу графике.

Вы должны постоянно иметь все это в виду. Люди, которые не могут видеть вообще, и люди с ограниченным зрением не смогут иметь дело с графикой.

С помощью атрибутов `alt` и элементов `em` вы сможете предоставить альтернативные тексты.

Не всегда легко сочинить соответствующий текст. Он должен быть кратким, осмысленным и сжатым.

Картинки, которые не содержат никакой информации, не нуждаются ни в каком альтернативном (`alt`) тексте. Помните, что существует широкий диапазон изображений — от тех, что несут четкую, узнаваемую и однозначную информацию, до тех, что всего лишь создают настроение или выполняют функцию организации пространства. При некоторых условиях бывает действительно трудно подобрать текстовую альтернативу изображению.

Вы можете использовать атрибут `longdesc` для описания изображений с массой информации, например, графического представления последних первичных выборов президента.

Этот атрибут содержит ссылку на внешний источник, который содержит описательный текст:

```
img src="electionresults.jpg" width="271" height="265" alt="Summary  
of the results of the Presidential primary"  
longdesc="electionresults.html">
```

К сожалению, `longdesc` имеет некоторые минусы в реализации. Не все программы-читатели экрана могут интерпретировать его.

Достаточно большие навигационные элементы

Люди, которые по разным причинам не могут управлять мышью, вынуждены использовать альтернативные технологии. Многие из них ориентированы на применение либо клавиатуры, либо других технических устройств. Диагноз Стивена Хоукинга (Steven Hawking) — классический пример расстройства моторных функций. Этот ученый-физик страдает от бокового амиотрофического склероза (болезнь Шарко; amyotrophic lateral sclerosis) и использует рот для управления инвалидной коляской. Вещи, доступные во многих областях, представляют собой технические чудеса, которые делают возможным то, что считалось ранее невозможным.

Подход к обработке событий мыши существует уже много лет. Но мышь не может быть использована людьми с дефектами моторных функций, равно как и пользователями, применяющими программы-читатели экрана, а между тем технология достигла такого состояния, что позволяет преодолеть и эту проблему.

Тем не менее, люди, использующие только клавиатуру, не могут работать с событиями мыши.

В любом случае убедитесь, что вы предоставили на сайте крупные кнопки. На ссылках с небольшой чувствительной областью трудно щелкать, особенно людям с физическими недостатками.

Формы

Интерактивность Web становится все более и более важной для упрощения коммуникаций между пользователем и оператором сайта. Пользователь вводит персональные данные, а программные компоненты в фоновом режиме манипулируют ими.

При текущем состоянии дел формы HTML все еще остаются предпочтительным посредником для реализации такого взаимодействия.

Это хорошо для доступности, поскольку HTML предлагает по существу платформенно-независимую и нейтральную по отношению к устройствам возможность взаимодействия. И тут не может быть возражений — до тех пор, пока эти функции также применяются пользователями альтернативных технологий.

Дизайн повышенной доступности форм HTML — первая и главная проблема линеаризации и группирования контента.

Элементы *fieldset* и *label*

Веб-разработчики склонны проектировать формы в виде таблиц данных. Дизайн этих форм на самом деле существенно проще. К сожалению, это ведет к появлению структур, в которых теряется связь контента между описанием и элементами форм.

В (X)HTML предусмотрен элемент `label` для логического соединения между элементом формы и описанием:

```
<label for="first name" title="first name">firstname:</label>
<input id="first name" type="text" size="20" name="first name"
  value="" />
```

Поле ввода получает уникальное имя посредством универсального атрибута `ID`, на который ссылается атрибут `for` элемента `label`.

Если в форме присутствуют похожие поля ввода, такие как различные поля для мужа и жены, то на этот случай имеется удобный инструмент для группировки по имени `fieldset`, который разделяет эти поля явно:

```
http://www.lenta.ru/photo/2008/09/13/kenny/
<fieldset>
<legend> Wife's information</legend>
<label for="first name of wife">first name</label>
<input id="first name of wife" type="text" size="20" name="first
  name" value="" />
...
</fieldset>
<fieldset>
<legend> Husband's information</legend>
<label for="first name of husband">first name</label>
<input id="first name of husband" type="text" size="20" name="first
  name" value="" />
```



Большинство программ-читателей экрана также будут предварительно читать контент `legend` перед каждой меткой, и потому его следует делать кратким и сжатым.

Использование `legend` предоставляет дополнительную помощь в навигации пользователям читателя экрана Jaws, поскольку он может перепрыгивать от `fieldset` к `fieldset` и тем самым обеспечивать быстрый обзор элементов форм. А Webformator, например, такого делать не может.

Свобода от барьеров в Joomla! 1.5 возможна с помощью Beez

Joomla! — одна из наиболее популярных в мире систем управления содержимым (CMS). Весь Web теперь может испытать лавинообразный рост доступности, поскольку Joomla! поставляется с шаблоном Beez, обеспечивающим свободу от барьеров.

Шаблон Beez является результатом огромной работы программистов и в еще большей степени — результатом убеждений (рис. 14.3). Разработчики Beez ставили перед собой две цели. Одна из них состоит в том, чтобы упростить работу профессионалов при создании свободных от барьеров проектов. Команда разработчиков может быть существенно сокращена. Это позволит сэкономить деньги, и свобода от барьеров будет более экономичной. Это идеальное решение для местных органов власти и других институтов, для которых важен фактор свободы от барьеров.



Рис. 14.3. Логотип Beez

В то же время шаблон Beez структурирован таким образом, что даже люди с ограниченными знаниями могут разрабатывать сайты умеренной сложности с высокой степенью доступности.

Поэтому Joomla! с Beez — идеальный инструмент для создания веб-представлений для учреждений, имеющих дело с инвалидами.

Свободный от барьеров шаблон Beez — это только одно новое направление, которое открывает Joomla!. Это фундамент, который может быть модифицирован и расширен в будущем. На основе Beez дизайн веб-сайта может быть модифицирован почти неограниченно посредством изменения файлов CSS.

Разработчики могут писать собственные шаблоны на основе Beez, которые буквально удовлетворяют всем требованиям высокой вмести­мости веб-сайтов при соблюдении свободы от барьеров.

На данный момент шаблон Joomla! понимается просто как средство визуального представления. На первый взгляд, Beez выглядит не слишком привлекательно. Но имейте в виду, что визуальный дизайн и ассоциированный код CSS — лишь поверхность, которая может быть легко модифицирована при наличии соответствующих знаний CSS.

Структура файла `index.php` очень мало отличается от других шаблонов, и потому я не стану описывать ее слишком подробно.

HTML

Предыдущие версии Joomla! выводили контент по умолчанию в табличной компоновке. Новая система Joomla! использует так называемые переопределения шаблонов (template overrides). Это обеспечивает гибкость, которая необходима для создания доступных веб-сайтов, отвечающих стандартам. Не так просто внести изменения в

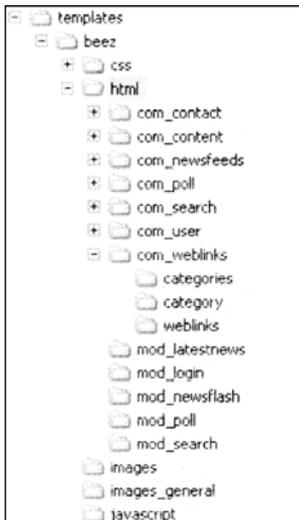


Рис. 14.4. Структура папок шаблона Beez

код HTML, даже если структура файлов логична и сама по себе согласована. Однако базовые знания PHP – это все, что вам нужно для выполнения собственной настройки.

Если Joomla! находит папку HTML с соответствующим контентом в каталоге нашего шаблона, она берет его; в противном случае Joomla! для обратной совместимости использует стандартный код, который работает с таблицами. Beez демонстрирует, как именно это делается.

Если вы взглянете на структуру Beez, то заметите в ней дополнительную папку HTML по сравнению со стандартным шаблоном (рис. 14.4).

Эта папка, в свою очередь, содержит все стандартные модули и компоненты, интегрированные в Joomla!. HTML-вывод всех этих файлов настроен и переработан. Я разработала разметку на основе принципа разделения контента и презентации, чья структура формата в основном согласована со структурой контента, и которую вы можете использовать для большинства приложений без дополнительной подстройки.

Это значит следующее.

- Все элементы документа организованы в корректном и логичном порядке, независимо от того, будут они отображаться на мониторе рядом друг с другом либо один над другим.
- Все элементы отображаются таким образом, что это согласуется с их позицией и значением в документе (семантически): заголовки – как заголовки, абзацы – как абзацы, кавычки – как кавычки, табличные данные – как табличные данные и т.д. Это делает возможным автоматическую манипуляцию документом различными способами, а не только максимально удобным для экранного читателя.

На рис. 14.5 представлен обзор видимой структуры заголовков Beez.



Рис. 14.5. Структура заголовков Beez

Метки для прыжков

Линейная презентация контента имеет существенный недостаток: для того, чтобы вернуться на некоторый предыдущий раздел контента, нужно выполнить трассировку е ашего пути назад на определенную дистанцию.

Beez помещает метки прыжков в двух местах. Первое — `index.php`, где позиция пределяется, а второе — в компонентах и модулях, содержащих элементы формы.

`index.php`

Здесь находится следующий код:

```
<ul>
<li><a href="#content" class="u2">
<?php echo JText::_('Skip to Content1'); ?></a></li>
<li><a href="#mainmenu" class="u2"><?php echo JText::_('Jump zo Main
Navigation and Login'); ?></a></li>
<li><a href="#additional" class="u2"><?php echo JText::_('
Jump to additional Information'); ?></a></li>
</ul>
```

Эти метки прыжков привязаны к соответствующим якорям внутри документов, и упорядочены семантически корректно посредством списка. Команды `JText` предназначены для трансляции и автоматически переводятся на язык по умолчанию, который выбран на основе используемого файла языка. CSS помешает заголовок из области просмотра перед каждым якорем. Этот заголовок обычно скрыт, но выводится, например, программой чтения экрана, значительно облегчая понимание структуры контента:

```
<h2 class="unseen">
<?php echo JText::_(' View , Navigation and Search'); ?>
</h2>
...
<a name="mainmenu" x/a>
```

Вы можете также поместить ссылки пропуска (skip links) непосредственно на ID окружающих элементов `div`, которые служат концепции структурированного документа даже еще лучше, но, к сожалению, существуют старые программы-читатели экрана, которые не могут интерпретировать такой вывод.

Ссылки пропуска в формах

Когда пользователи экранного читателя отправляют форму, не оснащенную метками прыжков, они оказываются в начале сайта и вынуждены мучительно пробираться обратно к нужному контенту. Чтобы предотвратить г*го. формы в Beez имеют свои собственные метки прыжков на действительный контент сайта.

Примером может служить следующая форма поиска:

```
<form action="<?php echo
JRoute::_('index.php?option=com_search#content' ) ?>"
method="post" class="search_result"><?php echo $this->params->get (
'pageclass_sfx') ?>>
```

Без и модули

Joomla! поставляется с различными модулями разнообразной функциональности. Свободно определяемый код HTML позволяет довольно немного в отношении вывода списка наиболее читаемых статей. Вы можете выделить различные позиции модулям в административной части Joomla!, которые затем помещаются в позицию в шаблоне, где предполагается их отображение. Имена позиций модулей определяются в XML-файле внутри шаблона. Это открывает массу возможностей для настройки структуры сайта под нужды конкретного проекта.

Следующий код связывает модуль с позицией left:

```
<jdoc:include type="module" name=" left" />
```

Без имеет свой собственный метод связывания модулей, который позволяет определить header-level, задающий семантическую спецификацию заголовка самого модуля.

При желании можно пометить каждый модуль заголовком из иерархического уровня. Выбор типа заголовка чрезвычайно важен, когда речь идет о глобальной целостности веб-сайта. Он должен быть интегрирован логически в общую структуру сайта, поскольку различный контент при некоторых обстоятельствах иметь разную "весовую категорию":

```
<jdoc:include type="modules" name="left" style="beezDivision"
headerLevel="3" />
```



Модуль Most Read (Наиболее читаемое) заслуживает особого внимания с точки зрения проектирования свободных от барьеров сайтов.

Он отображает статьи, которые открывались чаще всего, и тем самым помогает посетителям получать наиболее ценную информацию.

Компонент com_content

Компонент com_content контролирует вывод всего контента, а потому образует ядро нашего кода. Соответствующие файлы можно найти в templates/beez/html/com_content.

В стандартном шаблоне специфичный для Joomla! многостолбцовый вывод реализуется посредством таблицы. Это простейшее решение, которое ничего не сдвигает и все заполняет — даже без CSS.

Если же вы используете Без, вам нужно полностью пересмотреть это. Здесь применяются вложенные элементы div, которым назначены различные классы CSS, чтобы обеспечить максимальную гибкость дизайна. Предусмотрены классы для одиночных строк и колонок, что позволяет изменить почти все, добиваясь дизайна, который вам нравится. Если этот массив классов слишком велик, можете легко удалить их из кода функцией поиска с заменой.

Ниже показан соответствующий фрагмент из beez/html/comcontent/frontpage/default.php:

```
<div class = "article_row"<?php echo $this->params->get('pageclass_sfx'); ?>">
<?php for ($z = 0; $z < $colcount && $ii < $introcount && $i < $this->total;
$z++, $i++, $ii++) : ?>
```

```
<div class="article_column column<?php echo $z + 1; ?> cols
<?php echo $colcount; ?>" >
<?php $this->item =& $this->getItem($i, $this->params);
echo $this->loadTemplate('item'); ?>
</div>
<span class="article_separator">&nbsp;</span>
<?php endfor; ?>
<span class = "row_separator<?php echo $this->params->get('pageclass_sfx' );
?>"&nbsp;</span>
</div>
```

Формы

В Beez нет таблиц компоновки, даже для форматирования форм. Чтобы облегчить логическую связь между элементом формы и описанием, я использую элемент `label`, как того требует стандарт. Поле ввода получает уникальное имя через универсальный атрибут `ID`, и в атрибуте `for` элемента `label` указывается ссылка на него. Все существующие элементы формы в Beez были соответствующим образом перепроектированы. Я сгруппировала контент посредством `fieldset`, где это имело смысл, и снабдила его `legend` (например, в формах редактирования контента).

Таблицы данных

Таблицы не всегда являются проблемой. Если вы действительно хотите отображать структуры данных, то они представляют собой блестящий выбор.

Joomla! использует несложные таблицы данных в нескольких местах, например, при отображении веб-ссылок или в обзоре контакта.

В принципе, таблицы данных доступны, если они помечены как таковые и запрограммированы соответствующим образом.

Атрибут `headers` обеспечивает связь для каждой ячейки данных (`td`) между этой ячейкой и заголовком (`th`), указывая `ID` для соответствующего ассоциированного заголовка или, поскольку заголовков может быть несколько, указывая список `ID`. Такой дизайн также используется для внутренних таблиц данных Joomla!.

Вы можете скрыть представленные таблицы в конфигурации меню веб-ссылок или в обзоре контакта.



Не поступайте так, поскольку это повлияет на доступность ваших таблиц данных.

Дизайн и CSS

При проектировании Beez я умышленно пыталась ответить на постоянно возникающие вопросы CSS, используя "плавающий" контент почти так же интенсивно, как и абсолютное позиционирование внутри связанных элементов.



Веб-профессионалы не должны иметь никаких проблем с подгонкой ранее приведенного кода под свои нужды. Хотя для новичков это определенно представляет некоторую.

Внутренние CSS-файлы Beez

В папке CSS находится множество CSS-файлов различного назначения.

Позиционирование и презентация намеренно были помещены в разные файлы. Это дает то преимущество, что вам нужно модифицировать только файл `layout.css`, если требуется изменить цвета; при этом позиционирование останется нетронутым. Другими словами, будет меньше ошибок.

Список CSS-файлов:

- `position.css`
- `layout.css`
- `print.css`
- `template.css`
- `ie7only.css`
- `ieonly.css`
- `genereli.css`
- `template_rtl.css`

Позиционирование

Позиционирование управляется файлом `position.css`.

Здесь вы найдете все команды CSS для так называемого *документа с фреймами*, как определено в файле `index.php`, а также одно- или двухстолбцовую структуру в разделе контента и в ведущей истории (*leading story*). Вносить изменения в `position.css` необходимо только в том случае, если точно известно, что делается и чего планируется достичь.

Следующий код помещает фоновое изображение маленькой пчелки под ведущую историю:

```
.leading
{
background: #EFDEEA url(..images/bee.gif) no-repeat top left;
border: solid 1px #CCC;
color: #000;
margin: 30px 0px 10px 0px;
padding: 20px 20px 40px 120px;
position: relative;
}
```

Компоновка

Вы можете пересмотреть основной контент в `layout.css`, потому что все схемы форматирования и схемы цветов хранятся именно здесь. Здесь же вы найдете позиционирование определенных элементов контента.

Следующий код CSS форматирует заголовок ведущей истории и позиционирует ссылку *Read more* (Читать далее) в правом нижнем углу. Это происходит посредством абсолютного позиционирования в относительно позиционированной ведущей истории.

```
main .leading h2,#main2 .leading h2 {
background:#EFDEEA;
border-bottom:solid 0 #333;
color:#93246F;
```

```

font-family:trebuchet MS, sans-serif;
font-size:1.4em;
font-weight:normal;
margin:0 0 10px;
text-transform:uppercase;
}
#main .leading .readon,#main2 .leading .readon {
background:url(../images/arrow.gif) #93246F no-repeat;
border:solid 0;
bottom:0;
color:#FFF !important;
display:block !important;
margin-top:20px !important;
position:absolute;
right:0;
text-decoration:none;
padding:2px 2px 0 30px;
}

```

Прочее

Файл `template.css` всегда интегрирован в Joomla! — на случай появления всплывающего окна без браузерной навигации, например, в окне отправки почты другу, или когда вы настроите в этом стиле пункт меню для навигации. Форматирование этого представления задается исключительно в этом файле.

Файлы `ie7only.css` и `ieonly.css` заботятся об ошибках разных версий браузера Internet Explorer, и интегрированы посредством условных комментариев (Conditional Comments) в заголовок `index.php`.

Условные комментарии — это специальный вид комментариев, которые могут быть интерпретированы, начиная с версии Internet Explorer 5, и посылают инструкции только для таких браузеров.

```

<!--[if lte IE 6]>
<link href="<?php echo $this->baseurl ?>/templates/beeze/css/ieonly.css"
rel="stylesheet" type="text/css" />
<![endif]-->
<!--[if IE 7]>
<link href="<?php echo $this->baseurl ?>/templates/beeze/css/ie7only.css"
rel="stylesheet" type="text/css" />
<![endif]-->

```

Содержимое `print.css` достаточно очевидно. Оно формирует страницу для печати и используется только для этого. Может вызвать удивление, почему слово "content" находится перед действительным контентом с крупным размером шрифта, даже несмотря на то, что оно не появляется в этой форме в исходном коде.

CSS дает возможность писать контент в документ. Следующий код заботится об этом. Не все браузеры знают, что с этим делать, но те, что отвечают стандарту — знают.

```

#main2:before
{
content: " content ";
...
}

```

Файл `generell.css` уже находится в системной папке — одним уровнем выше в иерархии. Но если вы хотите подстроить внутренние сообщения об ошибках Joomla! в соответствии с дизайном — это подходящее место.

И, наконец, последний — по порядку, но не по важности — файл `template_rtl.css`. RTL — означает "справа налево" (right to left), что задает направление чтения на установленном языке. Направление чтения специфицировано в файле XML выбранного языка и может быть использовано в качестве переменной в файле `index.php` или шаблоне:

```
<?php if($this->direction == 'rtl') : ?>
<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/beez/css/template_rtl.css" type="text/css" />
<?php endif; ?>
```

Этот файл CSS изменяет отображение Beez в соответствии с направлением чтения, если вы устанавливаете такой язык, как иврит или арабский.

Если вы хотите позабавиться, посмотрите, что произойдет, если изменить направление чтения на английском языке, переключив соответствующую переменную с 0 на 1 (рис. 14.6):

```
Path: language/en-GB/en-GB.xml
...
<metadata>
...
<rtl>1</rtl>
...
</metadata>
```



Рис. 14.6. Внешний вид сайта после переключения направления чтения

Внутренние средства доступа Joomla!

Joomla! предоставляет возможность установки представлений статей, имеющих индивидуальные анонсы (teasers). Более старые версии Joomla! должны были иметь

кнопку Read more после каждого из анонсов. Вплоть до версии 1.5 эта ссылка всегда появлялась с одним и тем же текстом: Read more.

Поскольку такие вещи, как читатели экрана, отображают ссылки на страницы при необходимости, текст ссылки должен быть форматирован в соответствии с контентом, чтобы указывать на узнаваемую цель.

Кроме того, текстовый контент ссылки Read more семантически относится к контенту документа и потому должен быть свободно форматируемым. Если это требование не удовлетворено, ваш веб-сайт не пройдет простейший механический тест на доступность.

По умолчанию заголовок статьи позиционируется перед ссылкой Read more в Joomla! 1.5, поскольку это избыточная информация, которая не может быть отредактирована.

Вы уже знаете о параметрах статей, которые можно использовать для индивидуализации дизайна и/или компоновки информации каждой статьи на каждой странице. В Joomla! 1.5 есть параметр `readmore:text` (поле *Alternative Read more: text* (Альтернативный текст для *Read more:*)), который вы можете найти в области Parameters — Advanced (Параметры — дополнительные) в обзоре статей (рис. 14.7).



Рис. 14.7. Область Parameters — Advanced

Это может оказаться непростой задачей для редактора. Данный параметр призван не только описывать уникальный текст ссылки, но также заинтересовать посетителя и вынудить его прочесть больше.

Дополнительная информация

Вспомогательные технологии

- Webformator: www.webformator.com/englisch/inclex.php
- Jaws: http://www.freedomscientific.com/fs_downloads/jaws.asp
- Windows eyes: <http://www.gwmicro.com>

Инструменты проверки CheckerTools

- Colour Contrast Analyser:
<http://www.juicystudio.com/services/colourcontrast.php>
- <http://www.visionaustralia.org.au/info.aspx?page=628>
- Vischeck: <http://vischeck.com/>
- Cynthia Says: <http://www.contentr.quality.com>
- Bobby: <http://vrebxact.watchfire.com>
- Validator: <http://validator.w3.org>
- Validator: <http://www.htmlhelp.com>
- Wave: <http://www.wave.webaim.org/wave/index.jsp>
- Accessibility Toolbar Mozill/Firefox:
<http://cita.disability.uiuc.edu/software/mozilla/>

Рекомендуемые книги

- *Barrier-free Webdesign* Энджи Радтке (Angie Radtke) и Майкла Чарлиера (Michael Charlier).

Резюме

В этой главе были рассмотрены элементы, необходимые для построения презентаций, свободных от барьеров.

15

Собственные компоненты, модули и подключаемые модули

Предположим, что вы хотите решить некоторую задачу с помощью Joomla!, но не находите готового компонента, подходящего для этого. Например, вы – автомобильный дилер, и нуждаетесь в списке используемых грузовиков на веб-сайте, которым можете управлять административными средствами Joomla!, или же вам нужен список филиалов. Для этого можно просто расширить функциональность Joomla! новыми компонентами, модулями или подключаемыми модулями. То, что на первый взгляд кажется довольно трудным, на самом деле доступно даже тем, кто обладает лишь начальными знаниями PHP.

Сейчас, когда вы обладаете опытом построения шаблонов, вы уже можете придумать, что следует делать. Однако на этот раз упор будет сделан не на компоновке, а на программировании.



С 1 января 2008 г. разработка PHP 4 официально остановлена. Поэтому вы должны обновить все новые расширения в соответствии с рекомендациями версии PHP 5. Joomla! 1.5 совместима с обеими этими версиями. Дополнительную информацию можно найти на сайте <http://gophp.org/>.

Концепция “модель-представление-контроллер”

В Joomla! 1.5 представлена концепция (шаблон проектирования) для программирования расширений, которая называется “модель-представление-контроллер” (Model-View-Controller – MVC). Данный подход применяется в современной разработке программного обеспечения, чтобы внести некоторый порядок в эти вещи.

При решении программных задач почти всегда необходимы следующие три компонента:

- модель;
- представление;
- контроллер.

Отсюда возникает необходимость в стандартизации этих трех компонентов. Поскольку программисты испытывают аллергию к подобным стандартам, концепция должна нести в себе реальную выгоду. Давайте внимательно рассмотрим каждый из этих компонентов.

Модель

Модель содержит представление информации. Неважно, откуда поступают данные. Модель не знает ничего о выводе данных и не имеет представления о том, как эти данные изменяются.

Представление

Представление отображает данные из модели. Между моделью и представлением должна быть связь, чтобы было чего отображать.

Контроллер

Контроллер управляет всем в целом, т.е. реагирует на пользовательский ввод и другие события, администрирует модели и представления, а также обрабатывает вашу информацию для представлений.

Преимущество концепции MVC

Как уже указывалось выше, эти вещи принуждают к определенному порядку действий — в частности, в Joomla!. Если вы привыкли кодировать в точности, как делали это в Joomla! 1.0, знайте — в Joomla! 1.5 это уже нежелательно. Это продиктовано требованиями пользователей, то есть — вашими.

Что же необходимо? Свободные от барьеров веб-сайты, SEF URL, распределенная аутентификация, интернационализация, безопасность и удобство в обслуживании, а также многое другое... Все это требует некоторого порядка в системе. Если у вас компания с единственным сотрудником, вы можете осуществлять все администрирование в голове или с помощью карандаша и куска бумаги. Если у вас десять сотрудников, это становится несколько затруднительно. Если ваша компания начинает бурно развиваться и расти, вам нужна какая-то система для управления ею.

MVC — это разновидность системы для программного обеспечения. MVC позволяет создавать программный дизайн, который упрощает проведение последующих изменений и позволяет многократно использовать код. Конечно, здесь не обходится без трудностей, но это определенно неплохо. Поскольку у этой концепции столько приверженцев, это должно быть хорошо.

Разумеется, возникают вопросы, часть из которых перечислена ниже.

- Куда поместить бизнес-логику? В контроллер или в модель?
- Как насчет многократно используемого диалога? Поместить его в представление?
- Как управлять несколькими представлениями (таблицами, индивидуальным представлением, интерфейсной частью, административной частью)?

Поскольку сравнимой структуры в Joomla! 1.0 нет, эта "мантра" и ее реализация, на мой взгляд, является крупнейшим достижением разработчиков Joomla! 1.5. Будущее покажет, одинаково ли воспринимают это новшество пользователи и разработчики.

Так в чем же всеобщий выигрыш?

Это очень просто, и вы в этом убедитесь. Если ваша кухня, подвал, машина, чердак постоянно поддерживаются в порядке, то управлять ими легко. То же касается и программного обеспечения — чтобы его было легко развивать и обслуживать, необходимо выполнение определенных предварительных условий. И в этом отношении Joomla! находится в начале пути. Это значит, что если кто-то (команда проекта Joomla!) найдет сегодня время для приведения в порядок Joomla! 1.0, то вам будет легче адаптировать ее к существующим нуждам в будущем.

Пример компонента helloworld

Все компоненты делятся на компоненты интерфейсной части и компоненты административной части. Компоненты интерфейсной части отображаются на веб-сайте, а компоненты административной части разрабатываются для административного раздела — в основном для управления компонентами интерфейсной части. С точки зрения посетителя сайта, вы можете узнать компонент по способу запроса страницы Joomla!.

Так, например, URL-адрес `http://localhost/Joomla150/index.php?option=com_contact` вызывает компонент `com_contact` интерфейсной части.

Если вы заглянете в базу данных, то обнаружите множество компонентов в подкаталоге [Путь_к_Joomla!]/components, и среди них — компонент `com_contact` (рис. 15.1).

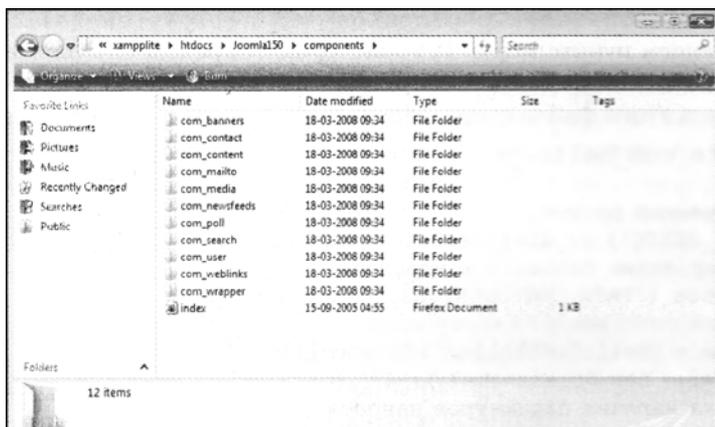


Рис. 15.1. Список компонентов

В зависимости от сложности компонентов, в этом каталоге появляются дополнительные подкаталоги для модели, представления и контроллера.

Домашний каталог helloworld

Каждый компонент располагается в собственном каталоге. Joomla! предлагает следующую последовательность обнаружения компонента.

1. Joomla! интерпретирует сгенерированные значения в URL: `/index.php?option=com_helloworld`.
2. Выполняет поиск таблицы для компоненты `com_helloworld`.
3. Ищет каталог `com_helloworld`.

4. В этом каталоге ищет файл `helloworld.php`.
5. Интерпретирует этот файл.

Для правильной интерпретации необходимо несколько файлов.

- `helloworld.php` — точка входа в ваш компонент.
- `controller.php` — контроллер.
- `views/helloworld/view.html.php` — представление, передающее данные в шаблон.
- `views/helloworld/tmpl/default.php` — файл стандартного шаблона, который будет переписан корректным шаблоном или использован как есть.
- `helloworld.xml` — уже знакомый XML-файл, который объясняет содержимое пакета контроллеру, сообщает ему, куда установить его, и кто за него отвечает.

Если вы растеряны и испытываете ностальгию по Joomla! 1.0 (там было привычно делать это одним файлом), доверьтесь нововведению.

Лучше позволить инсталлятору Joomla! установить пакет компонента `com_hello.zip` и установить пункт меню. Таким образом, вы немедленно размещаете все файлы в нужные каталоги и можете легко исследовать исходный код.

Точка входа(`components/com_hello/hello.php`)

Это первый файл, который будет вызван, когда вы указываете URL `http://localhost/joomla150/index.php?option=com_hello&view=hello` или щелкаете на соответствующем пункте меню. Результат будет определенно ожидаемым. Вы увидите текст "Hello World" в окне контента.

Исходный код этого файла представлен ниже.

```
/components/com_hello/hello.php:  
<?php  
// ограниченный доступ  
defined('_JEXEC') or die('Restricted access');  
// импортирование базового контроллера  
require_once (JPATH_COMPONENT.DS.'controller.php');  
// создание собственного контроллера  
$classname = 'HelloController'.$controller;  
$controller = new $classname( );  
// проверка наличия параметров запроса  
$controller->execute( JRequest::getVar('task'));  
// перенаправление внутри контроллера  
$controller->redirect();  
?>
```

Первая строка осуществляет контроль безопасности, проверяя, был ли файл вызван Joomla! или же непосредственно. Непосредственно вызванный сценарий немедленно останавливается функцией `die()`:

```
require_once {JPATH_COMPONENT.DS.'controller.php'};
```

После этого импортируется первый контроллер. Абсолютный путь к текущему компоненту (`components/com_hello`) выглядит как `JPATH_COMPONENT`, а `DS` представляет собой разделитель каталогов операционной системы. У Windows он отличается от Unix-подобных систем, т.е. это будет `/` или `\`. Joomla! устанавливает это значение автоматически.

```
$classname = 'HelloController'.$controller;  
$controller = new $classname();
```

Теперь можно создать экземпляр читающего контроллера, и тем самым получить контроллер, который затем использовать. Если нужен всего один контроллер, как это часто бывает в интерфейсной части, можно воспользоваться следующим оператором:

```
$controller=newHelloController();  
$controller->execute( JRequest::getVar('task'));
```

Операторы для компонентов устанавливаются в URL в следующей форме:

```
index.php?option=com_hello&task=task
```

(где `task` — одно из значений `save`, `edit`, `new...`).

Следующая строка используется для проверки, есть ли что-то для чтения:

```
$controller->redirect();
```

При этих условиях контроллер перенаправляет запрос на другой сайт, например, если что-то должно сохраниться.

Контроллер (/components/com_hello/controller.php)

Поскольку компонент действительно прост, перед контроллером теперь стоит задача отобразить что-нибудь. Нам не нужна модель данных, а достаточно одного метода `display`.

```
/components/com_hello/controller.php:  
<?php  
jimport('joomla.application.component.controller');  
class HelloController extends JController  
{  
function display()  
{  
parent::display();  
}  
}  
?>
```

Вызов метода `display()` задает имя и компоновку представления. Вы увидите результат этой версии отображения, если, например, установите новый пункт меню. Наш компонент распознает только стандартную компоновку.

Представление (/components/com_hello/views/hello/views.html.php)

Здесь представление уже имеется — представление по умолчанию.

```
/components/com_hello/views/hello/views.html.php:  
<?php  
jimport('joomla.application.component.view');  
class HelloViewHello extends JView  
{  
function display ($tpl = null)  
{  
$greeting = "Hello World!";  
$this->assignRef( 'greeting1', $greeting);  
parent::display ($tpl);  
}  
}  
?>
```

Представление обычно содержит данные (из модели), подготавливает их и посылает шаблону:

```
$greeting = "Hello World!";  
$this->assignRef( 'greeting', $greeting );  
parent::display($tpl);
```

Из-за присваиваний переменных модель здесь избыточна. Переменная `$greeting`, однако, могла бы содержать результат запроса к базе данных. Переменная передается шаблону методом `assignRef`:

```
parent::display($tpl);
```

Это вызывает появление шаблона.

Шаблон (/components/com_hello/views/hello/tmpl/default.php)

Здесь применяется шаблон по умолчанию. Он всегда называется `default.php` и выглядит в своем простейшем виде, как показано в следующем листинге.

```
/components/com_hello/views/hello/tmpl/default.php:  
defined('_JEXEC') or die('Restricted access'); ?>  
<h1>?>php echo $this->greeting; ?></h1>
```

Результат

В конечном итоге, получаем на веб-сайте ожидаемый результат.

Инсталляция

Все файлы теперь упакованы в ZIP-пакет и могут быть установлены инсталлятором Joomla!. Файлы интерфейсной части из `/components/com_hello` сохраняются в каталоге `site`, а файлы административной области из `/com_hello/administrator/components` – в каталоге `admin`. Все файлы описываются дополнительной информацией в `hello.xml`.

hello.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<!DOCTYPE install SYSTEM "http://dev.joomla.org/xml/1.5/component-install.dtd">  
<install type="component" version="1.5.0">  
<name>Hello</name>  
<creationDate>November</creationDate>  
<author>Nobody</author>  
<authorEmail>nobody@example.org</authorEmail>  
<authorUrl>http://www.example.org</authorUrl>  
<copyright>Copyright Info</copyright>  
<license>License Info</license>  
<version>Component Version String</version>  
<description>description of the component . . . </description>  
<!-- Site Main File Copy Section -->  
<files folder="site">  
<filename>index.html</filename>  
<filename>hello.php</filename>  
<filename>controller.php</filename>  
<filename>views/index.html</filename>  
<filename>views/hello/index.html</filename>  
<filename>views/hello/view.html.php</filename>
```

```

<filename>views/hello/tmpl/index.html</filename>
<filename>views/hello/tmpl/default.php</filename>
</files>
<administration>
<!-- Раздел меню администрирования -->
<menu>Hello World!</menu>
<!-- Раздел копии главных файлов администрирования -->
<files folder="admin">
<!-- Раздел копии главных файлов сайта -->
<filename>index.html</filename>
<filename>admin.hello.php</filename>
</files>
</administration>
</install>

```

При упаковке в ZIP-пакет необходимо сохранить структуру. Соответствующая структура для компонента `com_hello` показана на рис. 15.2.

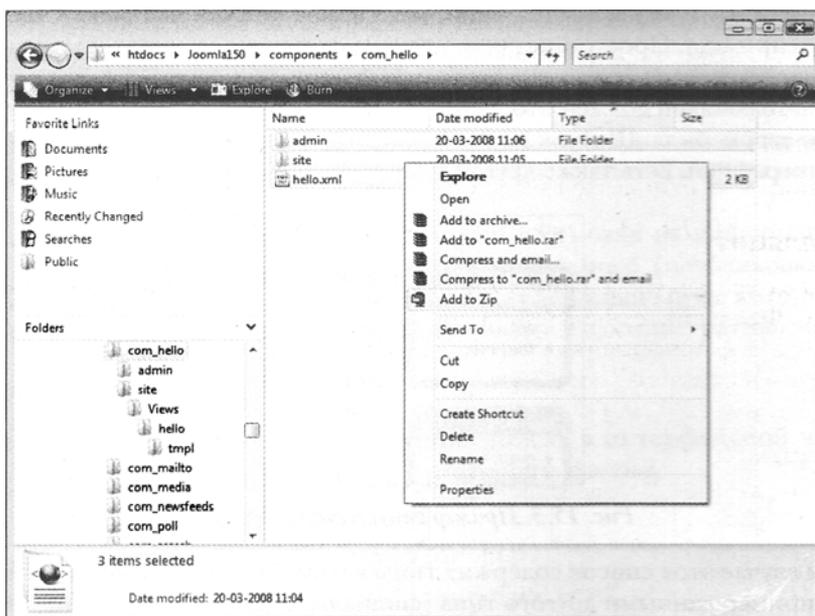


Рис. 15.2. Создание ZIP-пакта компонента

Теперь можно позволить инсталлятору Joomla! загрузить и установить пакет обычным образом.

Вывод

То, что вы только что создали — действительно легко в обслуживании, поддается расширению, и говоря откровенно, предельно ясно. На первый взгляд все эти файлы казались несколько путающими, но взгляните на результат — и вы избавитесь от страхов. Факт заключается в том, что в каждом файле присутствует очень немного исходного кода, и потому вы получаете гораздо лучший обзор. Именно на это и нацелена концепция MVC. И если вы хотите иметь специальный сложный шаблон, просто перепишите представление в вашем самодельном шаблоне.

Пример компонента

Теперь осуществим большой скачок в сторону сложности. Мы хотим настроить очень простой список используемых автомобилей, который можно было бы поддерживать из административной области.

Для этого вам понадобится:

- новый компонент интерфейсной части;
- компонент административной части;
- специальная таблица в базе данных;
- (необязательна) дополнительный модуль для отображения элементов на веб-сайте;
- (необязательно) подключаемый модуль поиска, чтобы можно было находить новый компонент.

Если вы поняли идею и представили, как список должен выглядеть на веб-сайте, приступим к примеру. Просто установите компонент `com_auto.zip` и исследуйте его самостоятельно.

Список автомобилей должен отображаться на веб-сайте с небольшим изображением для каждого автомобиля. Щелчок на изображении приводит к открытию окна с увеличенной фотографией. Есть также другой модуль со списком автомобилей (рис. 15.3).



Рис. 15.3. Пример списка автомобилей

В нашем случае этот список содержит типы автомобилей. При желании вы можете заполнить пример данными другого типа (филиалы, предложения и т.н.). Обычно посетителю достаточно видеть список. Редактировать его посетителю незачем.

Администратор, с другой стороны, должен управлять списком. Управление в данном случае означает следующее:

- ввод новых элементов;
- модификация существующих элементов;
- удаление существующих элементов.

Чтобы чересчур не усложнять этот пример, давайте просто отобразим и отредактируем четыре поля: два текстовых (тип и производитель) и два — с фотографиями (URL-адреса, указывающие на маленькие и большие фотографии). Принцип программирования компоненты в Joomla! станет яснее, и при необходимости вы сможете легко расширить этот пример дополнительными полями. Компонент должен быть интегрирован в существующую административную структуру.

Должна быть возможность выполнения функций, упомянутых ранее. В дополнение вам, конечно же, понадобится панель инструментов для отображения списка, и еще одна — для режима редактирования. Еще потребуются опции для публикации элементов и сокрытия их, а также функции Publish (Опубликовать) и Delete (Удалить), чтобы охватить несколько элементов сразу, отметив их флажками, находящимися рядом с каждым. Таким образом, например, вы можете опубликовать несколько элементов одновременно. Помимо списка также понадобится форма для модификации и добавления элемента.

Начнем с компонента; здесь он называется `com_auto`. Если вы хотите дублировать эту демонстрацию, можете загрузить готовый пример и установить его подобно любому другому компоненту. Затем можете модифицировать этот компонент вручную. Модификации и преобразования обладают тем преимуществом, что вы хорошо ознакомитесь со структурами и также сможете усовершенствовать компонент по своему желанию.

Таблица MySQL

Этот компонент использует таблицу MySQL, в которой хранятся записи об автомобилях. Таблица создается автоматически при установке. Вы можете видеть команды SQL после установки в файле `/administrator/com_auto/install.sql`. Есть также файл `uninstall.sql`, который удаляет таблицу при деинсталляции компонента.

Обратившись по адресу `http://localhost/phpmyadmin`, можете взглянуть на таблицу `jos_auto`.

Префикс `jos_` добавляется установщиком и основан на информации из вашей конфигурации Joomla!. Поле `Published` имеет значение 1 (опубликовано) или 0 (не опубликовано). Есть два поля для фотографий, содержащие путь, который вы, скорее всего, должны будете изменить. Это можно сделать в административном разделе.

Интерфейсная часть

Теперь приступим к программированию списка в интерфейсной части. Как и с компонентом `helloworld`, понадобятся следующие файлы:

- `/components/com_auto/auto.php`
- `/components/com_auto/controller.php`
- `/components/com_auto/views/auto/view.html.php`
- `/components/com_auto/views/auto/tmpl/default.php`
- `/components/com_auto/models/auto.php`

Как видите, на этот раз имеется модель, и потому понадобятся данные из базы.

Точка входа (`/components/com_auto/auto.php`)

Здесь также есть точка входа. Файл `/components/com_auto/auto.php` собирает все вместе и делегирует задачи.

```
/components/com_auto/auto.php:  
<?php  
// ограниченный доступ  
defined('_JEXEC') or die('Restricted access');
```

```
// импортирование базового контроллера
require_once (JPATH_COMPONENT.DS.'controller.php');
// создание собственного контроллера
$controller = new AutoController();
// проверка наличия параметров запроса
$controller->execute(JRequest::getCmd('task'));
// перенаправление внутри контроллера $controller->redirect();
?>
```

Этот код почти идентичен коду для компонента com_hello.

Контроллер (/components/com_auto/controller.php)

Здесь также вы найдете много фрагментов, знакомых по компоненту com_hello.

```
/components/com_auto/controller.php:
<?php
defined('_JEXEC) or die);
jimport('joomla.application.component.controller');
class AutoController extends JController
{
function display()
{
parent::display();
}
}
```

Представление (/components/com_auto/views/auto/view.html.php)

Здесь вы обнаружите первое отличие. Разумеется, для списка понадобятся данные.

```
/components/com_auto/views/auto/view.html.php:
<?php
jimport('joomla.application.component.view');
class AutoViewAuto extends JView
{
function display($tpl = null)
{
$model = &$this->getModel();
$rows = $model->getAutoList();
$this->assignRef('rows' , $rows);
parent::display($tpl);
}
}
?>
$model = &$this->getModel();
```

Экземпляр модели создается и доступен в виде объекта в переменной \$model:

```
$rows = $model->getAutoList();
```

Метод getAutoList() вызывается на объекте и возвращает массив \$rows:

```
$this->assignRef('rows', $rows);
parent::display($tpl);
```

Результирующие строки выделяются и передаются шаблону.

Шаблон (/components/com_a.nto/views/auto/tmpl/default.php)

Строки в переменной \$rows посылаются в цикле for, трансформируясь в единственную \$row, и затем выводятся по отдельности.

```
/components/com_auto/views/auto/tmpl/default.php:
<?php
defined('_JEXEC') or die('Restricted access');
?>
<script type="text/javascript">
function OpenWindow (Adresse) {
MyWindow = window.open(Adress, "Auto",
"width=400,height=300,left=100,top=200");
MyWindow.focus();
}
</script>
<h1><?php echo "&Uuml;berschrift"; ?></h1>
<ul>
<?
// Чтение наборов данных в массиве
foreach ($this->rows as $row) { ?>
<li><?php echo $row->text; ?> <small><em>(<?php echo
$row->manufacturer; ?>)
</em></small><br>
<a href="<?php echo $row->photo_large; ?>"
onclick="OpenWindow(this.href);
return false">_getAutoQuery( $options );
$result = $this->_getList( $query );
return @$result;
}
}
?>

```

В этой модели присутствуют два важных метода. Один из них— `_getAutoQuery`, который актуализирует доступ к данным и содержит команду SQL, а другой — метод `getAutoList`, контролирующий этот доступ и возвращающий результат. Вспомните файл `views.html.php`. Следующая строка — из этого файла:

```
$rows = $model->getAutoList ();
```

Она обращается к объекту модели и сохраняет результат в `$rows`. Этот результат актуализирован посредством метода `getAutoList ()` класса `AutoModelAuto`.

Вывод

Это хороший момент, чтобы познакомиться с PHP и объектно-ориентированным программированием. Вы не обязательно потеряны для веб-дизайна без знаний о классах, методах, наследовании и подобных вещах, но наверняка будете чувствовать себя неуверенно.

Тем не менее, однажды вступив на путь Joomla! MVC, вы вскоре увидите закономерности. Если, например, сравнить компонент `com_helloworld` с этой частью компонента `com_auto`, то 80% кода, который ориентирован на одинаковые цели, окажется идентичным (все, кроме модели).

Администрирование com_auto

Чистое отображение данных на веб-сайте относительно просто; администрирование же данных по своей природе несколько сложнее. Как администратор, вы должны иметь возможность отображать, модифицировать, вставлять, удалять и публиковать данные. Это требует намного большей степени интерактивности, чем при простом выводе списка на веб-сайте.

Таблица компонента

Joomla!, кстати, администрирует все пункты меню компонента в таблице [префикс] `components`. Элементы меню всех компонентов в административной области также должны быть записаны. Компонент `com_auto` также введен сюда.

Графика, отображаемая рядом с меню, тоже записана там ('js/ThemeOffice/component.png'). Вы найдете графику в папке [Путь_к_Joomla!]/includes/js/ThemeOffice.

Вам понадобится несколько файлов, чтобы иметь возможность создать административный компонент. В папке [Путь_к_Joomla!]/administration/components/com_auto/ находятся следующие файлы:

- /administration/components/admin.auto.php
- /administration/components/controller.php
- /administration/components/controllers/auto.php
- /administration/components/views/autos/view.html.php
- /administration/components/views/autos/tmpl/default.php
- /administration/components/views/auto/view.html.php
- /administration/components/views/auto/tmpl/form.php
- /administration/components/tables/autc.php
- /administration/components/install.sql
- /administration/components/uninstall.sql

Точка входа(/administration/components/admin.auto.php)

Разумеется, в административную область также имеется своя точка входа.

```
/administration/components/admin.auto.php:
<?php
defined('_JEXEC') or die('Restricted access');
$controller = JRequest::getVar('controller', 'auto');
require_once (JPATH_ADMINISTRATOR.DS.'controllers'.DS.$controller.'.php');
$classname = 'AutosController'.$controller;
$controller = new $classname();
$controller->execute( JRequest::getVar('task'));
$controller->redirect();
?>
```

Эксперты по точкам входа заметят, что это выглядит очень знакомо. Здесь знакомо все, кроме запроса if, который ищет дополнительные контроллеры.

Контроллер (/administration/components/controller.php)

Базовый контроллер также выглядит вполне знакомо.

```
/administration/components/conetroller.php:
<?php
jimport('joomla.application.component.controller');
class AutosController extends JController
{
function display()
{
parent::display();
}
}??>
```

Другой контроллер

(*/administration/components/controllers/auto.php*)

Теперь видим некоторые отличия. У нас есть дополнительный контроллер, в котором присутствует фрагмент кода.

```
/administration/components/controllers/auto.php:
```

```
<?php
defined('_JEXEC') or die();
class AutosControllerAuto extends AutosController
{
    function _construct() {
        parent::_construct();
        $this->registerTask( 'add', 'edit' );
        $this->registerTask( 'unpublish', 'publish' );
    }
    function edit() {
        JRequest::setVar( 'view', 'auto' );
        JRequest::setVar( 'layout', 'form' );
        JRequest::setVar( 'hidemainmenu', 1 );
        parent::display();
    }
    function save() {
        $model = $this->getModel( 'auto' );
        if ( $model->store( $post ) ) {
            $msg = JText::_ ( 'Auto Saved!' );
        } else {
            $msg = JText::_ ( 'Error Saving Auto' );
        }
        $link = 'index.php?option=com_auto';
        $this->setRedirect( $link, $msg );
    }
    function remove() {
        $model = $this->getModel( 'auto' );
        if ( !$model->delete() ) {
            $msg = JText::_ ( 'Error: One or more Autos could not be Deleted' );
        } else {
            $msg = JText::_ ( 'Auto(s) Deleted' );
        }
        $this->setRedirect( 'index.php?option=com_auto', $msg );
    }
    function publish() {
        $this->setRedirect( 'index.php?option=com_auto' );
        $db =& JFactory::getDBO();
        $user =& JFactory::getUser();
        $cid = JRequest::getVar( 'cid', array(), 'post', 'array' );
        $task = JRequest::getCmd( 'task' );
        $publish = ( $task == 'publish' );
        $n = count( $cid );
        if ( empty( $cid ) ) {
            return JError::raiseWarning( 500, JText::_ ( 'No items selected' ) );
        }
        JArrayHelper::toInteger ( $cid );
        $cids = implode ( ',', $cid );
        $query = 'UPDATE #__auto'
```

```

. ' SET published = ' . (int) $publish
. ' WHERE id IN ( '. $cids .' )'
;
$db->setQuery( $query );
if (!$db->query()) {
return JError::raiseWarning( 500, $row->getError() );
}

$this->setMessage( JText::sprintf( $publish ? 'Items published' :
'Items unpublished', $n ) );
}
function cancel(){
$msg = JText::_( 'Operation Cancelled' );
$this->setRedirect( 'index.php?option=com_auto', $msg );
}
}
?>

```

Этот контроллер реализует методы `edit`, `save`, `remove`, `publish` и `cancel`. Модель создается внутри этих методов, и при необходимости метод `store`, например, вызывается на модели. Сообщения об успехе или неудаче выводятся посредством статических классов `JText` и `JError` (рис. 15.4).

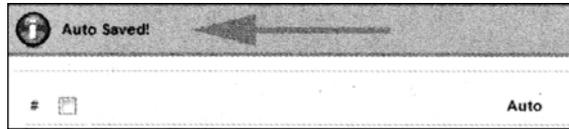


Рис. 15.4. Сообщение об успехе операции

Представление для списка

(/administration/components/views/autos/view.html.php)

На этот раз представление немного больше, поскольку добавлена панель инструментов.

```

/administration/components/views/autos/view.html.php:
<?php
defined( '_JEXEC' ) or die();
jimport( 'joomla.application.component.view' );
class AutosViewAutos extends JView
{
function display( $tpl = null )
{
JToolBarHelper::title( JText::_( 'Auto Manager' ), 'generic.png' );
JToolBarHelper::publishList();
JToolBarHelper::unpublishList();
JToolBarHelper::deleteList();
JToolBarHelper::editListX();
JToolBarHelper::addNewX();
$items = & $this->get( 'Data' );
$this->assignRef( 'items', $items );
parent::display( $tpl );
}
}

```

Об отображении заботится класс `JToolBarHelper`.

Шаблон списка

(/administration/components/views/autos/tmpl/default.php)

Разумеется, список также должен быть сформатирован, и потому читается соответствующий шаблон по умолчанию.

```
/administration/components/views/autos/tmpl/default.php:
<?php defined(1_JEXEC') or die('Restricted access'); ?>
<form action="index.php" method="post" name="adminForm">
<div id="editcell">
<table class="adminlist"><thead><tr>
<th width="5"><?php echo JText::_('NUM' ); ?></th>
<th width="20"> <input type="checkbox" name="toggle" value=""
onclick="checkAll(<?php echo count( $this->iteros ); ?>);" /></th>
<th class="title"><?php echo JHTML::_('grid.sort', 'Auto', 'a.text',
@$lists['order_Dir'], @$this->lists['order'] ); ?></th>
<th width="5%" align="center"><?php echo JHTML::_('grid.sort',
'Published',
'a.published', @$this->lists['order_Dir'], @$this->lists['order'] ); ?></th>
<th width="1%" nowrap="nowrap"><?php echo JHTML::_('grid.sort', 'ID',
'a.id',
@$this->lists['order_Dir'], @$this->lists['order'] ); ?></th>
</tr></thead>
<?php
$х = 0;
for ($i=0, $n=count( $this->items ); $i < $n; $i++)
{
$хrow = &$this->items[$i];
$хpublished = JHTML::_('grid.published', $хrow, $i);
$хlink = JRoute::_(
'index.php?option=com_auto&controller=auto&task=edit&cid[]='.
$хrow->id );
?>
<tr class="<?php echo "row$х"; ?>">
<td></td>
<td></td>
<td><a href="<?php echo $хlink; ?>"><?php echo $хrow->text; ?></a></td>
<td align="center"><?php echo $хpublished; ?></td>
<td align="center"><?php echo $хrow->id; ?></td>
</tr>
<?php
$х = 1 - $х;
}
?>
</table></div>
<input type="hidden" name="option" value="com_auto" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="boxchecked" value="0" />
<input type="hidden" name="controller" value="auto" />
</form>
```

Этот шаблон содержит весьма простой HTML, упакованный в форму. Он выполняет отображение таблицы, как показано на рис. 15.5

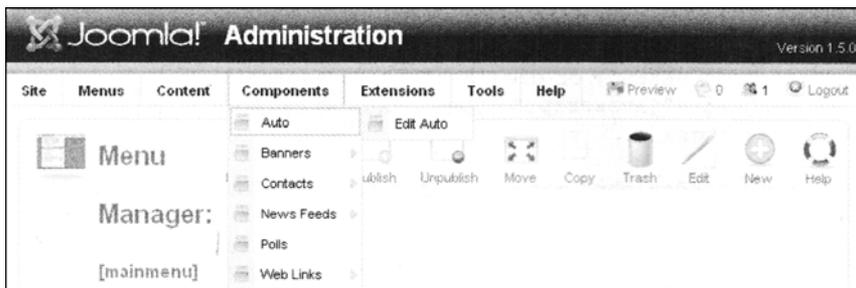


Рис. 15.5. Отображение таблицы

Представление формы

(/administration/components/views/auto/view.html.php)

Индивидуальный вид отдельных автомобилей также контролируется. Обратите внимание на имя подкаталога. Теперь мы – в папке auto; список находится в этой папке.

/administration/components/views/auto/view.html.php:

```
<?php
defined('_JEXEC') or die();
jimport( 'joomla.application.component.view' );
class AutosViewAuto extends JView
{
function display($tpl = null)
{
    $auto =& $this->get('Data');
    $isNew = ($auto->id < 1);
    $text = $isNew ? JText::_('New') : JText::_('Edit');
    JToolBarHelper::title( JText::_('Auto') . '<small>[ '.$text.' ]</small>' );
    JToolBarHelper::save();
    if ($isNew) {
        JToolBarHelper::cancel();
    } else {
        JToolBarHelper::cancel( 'cancel', 'Close' );
    }
    $this->assignRef('auto', $auto);
    parent::display($tpl);
}
}
```

В этом коде конструируется панель инструментов для индивидуального вида. Это представление может быть использовано для добавления и изменения наборов данных. Переменная \$isnew различает эти два случая.

Шаблон формы

(/administration/components/views/auto/tmpl/form.php)

Форма для индивидуального вида конструируется в стандартном шаблоне.

```

/administration/components/views/auto/tmpl/form.php:
<?php defined('_JEXEC') or die('Restricted access'); ?>
<script language="javascript" type="text/javascript">
... проверка ввода ...
</script>
<form action="index.php" method="post" name="adminForm" id="adminForm">
<div>
<fieldset class="adminform">
<legend><?php echo JText::_('Details' ); ?></legend>
<table class="admintable">
<tr>
<td width="110" class="key">
<label for="title">
<?php echo JText::_('Text' ); ?>:
</label>
</td>
<td>
<input class="inputbox" type="text" name="text" id="text" size="60"
value="<?php echo $this->auto->text; ?>" />
</td>
</tr>
<tr>
... дополнительные поля ...
</tr>
<tr>
<td width="120" class="key">
<?php echo JText::_('Published' ); ?>:
</td>
</tr>
</table>
</fieldset>
</div>
<div class = "clr"></div>
<input type="hidden" name="option" value="com_auto" />
<input type="hidden" name="id" value="<?php echo $this->auto->id; ?>" />
<input type="hidden" name="task" value="" />
<input type="hidden" name="controller" value="auto" />
</form>

```

Форма также состоит из чистого HTML с переменными PHP (`$this->auto->id`) и вызовами статического класса (`JText`).

Таблица автомобилей

(`/administration/components/tables/auto.php`)

И последнее — по порядку, но не по значению — это класс таблицы. Модель каким-то образом должна знать, с какими данными ей работать. Класс `JTable` чрезвычайно облегчает доступ к данным и их редактирование. Это абстрактный класс (интерфейс), который позволяет производным классам использовать структуры с их методами. Имя таблицы и первичный ключ указываются в конструкторе.

/administration/components/tables/auto.php:

```
<?php
defined('_JEXEC') or die('Restricted access');
class TableAuto extends JTable
{
var $id = 0;
var $text = '';
var $manufacturer = '';
var $photo_small = '';
var $photo_large = '';
var $published = 0;
function TableAuto(& $db) {
parent::_construct('#_auto', 'id', $db);
}
}
?>
```

Иnstалляция (/administration/components/install.sql) и деинсталляция (/administration/components/uninstall.sql)

Во время инсталляции/деинсталляции инсталлятор Joomla! должен установить или удалить необходимые таблицы. Для этого предусмотрены следующие два файла.

/administration/components/install.sql:

```
DROP TABLE IF EXISTS '#_auto';
CREATE TABLE '#_auto' (
'id' int(11) NOT NULL auto_increment,
'text' text character set utf8 NOT NULL,
'hersteller' varchar(100) character set utf8 NOT NULL,
'photo_gross' varchar(200) character set utf8 NOT NULL,
'photo_klein' varchar(200) character set utf8 NOT NULL,
'published' tinyint(1) NOT NULL,
PRIMARY KEY ('id')
) ENGINE=MyISAM AUTO_INCREMENT=5 ;
INSERT INTO '#_auto' ('id', 'text', 'manufacturer', 'photo_large',
'photo_small', 'published') VALUES
(2, 'Smart fortwo', 'Smart',
'http://localhost/joomlal50/images/stories/com_auto/smart_large.jpg',
'http://localhost/joomlal50/images/stories/com_auto/
smart_small.jpg', 1),
(4, 'Roadster', 'Smart',
'http://localhost/joomlal50/images/stories/
com_auto/roadster_large.jpg',
'http://localhost/joomlal50/images/stories/
com_auto/roadster_small.jpg', 1);
```

administration/components/uninstall.sql:

```
DROP TABLE IF EXISTS '#_auto';
```

Тестирование

После проверки всех файлов вы можете протестировать компонент и полностью администрировать наборы данных из Joomla!. Вы можете вводить новый текст, редактировать существующий и публиковать его. Попробуйте отредактировать и расширить несколько элементов. Это не слишком сложно.

Создание инсталляционного пакета

Чтобы поместить созданный компонент в инсталляционный пакет, помимо упомянутых файлов вам также понадобится обязательный XML-файл с метаданными.

Файл *auto.xml*

В этом файле созданный компонент описывается для инсталлятора Joomla!. Вы должны заключить всю информацию, такую как метаданные и все имена файлов, в дескрипторы XML. Инсталлятор Joomla! читает этот файл, создает новые подкаталоги, копирует файлы в нужные места и настраивает необходимые таблицы.

auto.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install SYSTEM "http://dev.joomla.org/xml/1.5/
component-install.dtd">
<install type="component" version="1.5.0">
<name>Auto</name>
<creationDate>November 2007</creationDate>
<author>Hagen Graf</author>
<authorEmail>hagen.graf@gmail.com</authorEmail>
<authorUrl>http://www.cocoate.com</authorUrl>
<copyright>All rights reserved</copyright>
<license>GNU/GPL</license>
<version>Component Version String</version>
<description>description of the component ... </description>
<files folder="site">
<filename>index.html</filename>
<filename>auto.php</filename>
<filename>controller.php</filename>
<filename>views/index.html</filename>
<filename>views/auto/index.html</filename>
<filename>views/auto/view.html.php</filename>
<filename>views/auto/tmpl/index.htrnK</filename>
<filename>views/auto/tmpl/default.php</filename>
<filename>models/auto.php</filename>
</files>
<install>
<sql>
<file charset="utf8" driver="mysql">install.sql</file>
</sql>
</install>
<uninstall>
<sql>
<file charset="utf8" driver="mysql">uninstall.sql</file> </sql>
</uninstall>
```

```

<administration>
<menu>Joomla! 1.5 Book Auto</menu>
<files folder="admin">
<filename>index.html</filename>
<filename>admin.auto.php</filename>
<filename>controller.php</filename>
<filename>controllers/auto.php</filename>
<filename>controllers/index.html</filename>
<filename>models/auto.php</filename>
<filename>models/autos.php</filename>
<filename>models/index.html</filename>
<filename>views/autos/view.html.php</filename>
<filename>views/autos/index.html</filename>
<filename>views/autos/tmpl/default.php</filename>
<filename>views/autos/tmpl/index.html</filename>
<filename>views/auto/view.html.php</filename>
<filename>views/auto/tmpl/form.php</filename>
<filename>views/auto/index.html</filename>
<filename>views/auto/tmpl/index.html</filename>
<filename>tables/auto.php</filename>
<filename>tables/index.html</filename>
<filename>install.sql</filename>
<filename>uninstall.sql</filename>
</files>
</administration>
</install>

```

Чтобы создать инсталляционный пакет, скопируйте все созданные файлы в каталог и упакуйте каталог в ZIP-файл с именем компонента, в данном случае — `com_auto.zip`. Файлы для интерфейсной части помещаются в папку `site`, в то время как файлы, предназначенные для административной области, — в папку `admin`.

Теперь можно установить ZIP-файл инсталлятором Joomla!, как обычно, и при желании предоставить его другим разработчикам для загрузки. Прежде чем делать это в вашей собственной инсталляции, воспользуйтесь инсталлятором Joomla! для деинсталляции версии, которая была установлена вручную. Для этого щелкните на `Extensions⇒Install/Uninstall` (Расширения⇒Установить/деинсталлировать), отметьте ваш компонент и щелкните на пиктограмме `Uninstall` (Деинсталлировать).

Модули

Модули намного проще. Обычно они не имеют административного интерфейса, но время от времени принимают параметры. Модули полностью связаны с презентацией на вашем веб-сайте и интегрированы в ваш шаблон. Обычно модули прикрепляются к существующим компонентам. (Потом) предполагается, что необходимые таблицы и контент уже существуют и могут использоваться.

Для того чтобы запрограммировать собственный модуль, понадобится два файла. Один — для логики и презентации, а другой — XML-файл для инсталлятора Joomla!. Имена обоих файлов начинаются с метки `mod_`.

Исходный код

Взглянем на исходный код этих файлов. Шаблоны здесь также применяются.

Вот их файлы:

- mod_auto.php
- helper.php
- tpl/default.php
- mod_auto.xml

Рассмотрим эти файлы вместе с остальными.

Точка входа (mod_auto.php)

Файл mod_auto.php — это управляющий файл для модуля.

mod_auto.php:

```
<?php
defined! '_JEXEC' or die('Restricted access');
require_once (dirname(__FILE__).DS.'helper.php');
$auto = modAutoHelper::getAuto($params);
require(JModuleHelper::getLayoutPath('mod_auto'));
?>
```

В этой точке интегрируется класс helper, а не базовый контроллер.

Класс helper (helper.php)

Класс helper комбинирует в себе контроллер и модель.

helper.php:

```
<?php
defined! '_JEXEC' ) or die( 'Restricted access' );
class modAutoHelper
{
function getAuto(&$params)
{
global $mainframe;
$db =& JFactory::getDBO();
$query = "SELECT *"
. "\n FROM #__auto"
. "\n WHERE published =1"
. "\n LIMIT 0,5"
;
$db->setQuery ( $query );
$rows = $db->loadObjectList();;
$auto = "<ul>\n";
if ($rows) {
foreach ($rows as $row) {
$auto .= " <li>". $row->text . "</li>\n";
}
}
$auto .= "</ul>\n";
return $auto;
}
}
?>
```

В классе helper запрос строк данных происходит после всего остального в цикле for.

Шаблон (*tmpl/default.php*)

В этой достаточно простой конструкции переменная `$auto` хранит вывод класса `helper`. Можно было бы также просто выполнить цикл `for` из класса `helper` и оказать большее влияние на код HTML, подлежащий выводу, и таким образом позволить дизайнеру шаблона переписать исходный код.

`tmpl/default.php`:

```
defined( '_JEXEC' ) or die( 'Restricted access' ); ?>
<?php echo $auto; ?>
```

Файл *mod_auto.xml*

Для инсталляции модуля понадобятся все необходимые данные для инсталлятора Joomla! в виде XML-файла (как и в случае компонента).

`mod_auto.xml`:

```
<?xml version="1.0" encoding="utf-8"?>
<install type="module" version="1.5.0">
  <name>Auto</name>
  <author>Hagen Graf</author>
  <creationDate>November 2007</creationDate>
  <copyright>(C) 2007 cocoate.com All rights reserved.</copyright>
  <license>GNU/GPL</license>
  <authorEmail>hagen.graf@gmail.com</authorEmail>
  <authorUrl>www.cocoate.com</authorUrl>
  <version>0.1</version>
  <description>Auto Module</description>
  <files>
    <filename module="mod_auto">mod_auto.php</filename>
    <filename>index.html</filename>
    <filename>helper.php</filename>
    <filename>tmpl/default.php</filename>
    <filename>tmpl/index.html</filename>
  </files>
</install>
```

Инсталляция

Скопируйте все файлы в подкаталог и упакуйте этот каталог в ZIP-пакет по имени `mod_auto.zip`. Как обычно, вы сможете затем инсталлировать этот пакет инсталлятором Joomla! и при желании позволить другим загружать его себе. Прежде чем приступить к инсталляции у себя, воспользуйтесь инсталлятором Joomla! для деинсталляции версии, установленной вами вручную. Для этого щелкните на `Extensions⇒Install/Uninstall`, отметьте компонент и щелкните на пиктограмме `Uninstall`.

После инсталляции вы должны будете активизировать модуль через меню `Extensions⇒Module Manager` (Расширения⇒Диспетчер модулей).

Представление на веб-сайте

Теперь вы можете видеть элементы из таблицы `joe_auto` в выбранной позиции.

Подключаемые модули

И последнее — по порядку, но не по значимости — мы хотим интегрировать компонент в общую функцию поиска Joomla!. Для этого понадобится подключаемый модуль типа search, чтобы включить вашу таблицу в общий путь поиска. В этом случае не каждый подключаемый модуль имеет подкаталог, но каждый тип подключаемого модуля имеет. По этой причине мы будем работать в каталоге [Путь_к_Joomla!]/plugins/search.

Исходный код

Вам понадобится, как минимум, один файл PHP с логикой (см. следующий листинг) и XML-файл с описанием подключаемого модуля. Имена этих файлов должны происходить от имени компонента, поэтому назовем их auto.php и auto.xml. Если, например, вы пишете пользовательский подключаемый модуль, то учитываете его функцию, выбирая имя. В данном же случае присутствует конкретная связь с компонентом com_auto. Подключаемые модули также должны быть анонсированы в таблице, в данном случае — в josplugins. Разумеется, инсталлятор сделает это за вас. Поиск — очень дорогостоящая операция, и может применяться с множеством параметров. Исходный код даст вам представление о возможных опциях. Поскольку наш компонент не отслеживает, когда элемент добавлен, или насколько часто к нему осуществляется доступ (у нас пока нет детальной страницы), пока многие из этих опций останутся неиспользованными.

auto.php:

```
<?php
defined ( '_JEXEC ) or die( 'Restricted access' );
$mainframe->registerEvent ( 'onSearch', 'plgSearchAuto' );
$mainframe->registerEvent ( 'onSearchAreas1', 'plgSearchAutoAreas' );
function splgSearchAutoAreas () {
    static $areas = array('auto' => 'Auto');
    return $areas;
}
function plgSearchAuto( $text, $phrase='', $ordering=' ', $areas=null ){
    $db =& JFactory::getDBO();
    $user =& JFactory::getUser();
    if (is_array( $areas )) {
        if (!array_intersect( $areas, array_keys( plgSearchAutoAreas() ) )) {
            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::_ ( 'Auto' );
    $wheres = array();
    switch ( $phrase ){
        case 'exact':
            $text = $db->getEscaped( $text );
            $wheres2 = array();
```

```

$wheres2[] = "LOWER(a.text) LIKE '%$text%'";
$wheres2[] = "LOWER(a.manufacturer) LIKE '%$text%'";
$where = '(' . implode( ' ) OR ( ', $wheres2 ) . ')';
break;
case 'all':
case 'any':
default:
$words = explode ( ' ' , $text );
$wheres = array ();
foreach (Swords as $word) {
$word = $db->getEscaped($word);
$wheres2 = array();
$wheres2[] = "LOWER(a.text) LIKE '%$word%'";
$wheres2[] = "LOWER(a.hersteller) LIKE '%$word%'";
$wheres[] = implode(' OR ', $wheres2 );
}
$where= '(' .implode( ($phrase == 'all' ? ' ) AND ( : ' ) OR ( ' , $wheres ) . ')';
break;
}
switch ( $ordering ) {
default:
$order = 'a.text ASC';
break;
}
$query = "SELECT * FROM #_auto AS a"
. "\n WHERE ( $where )"
. "\n AND published = '1'"
. "\n ORDER BY $order";
$db->setQuery( $query, 0, $limit );
$rows = $db->loadObjectList();
foreach ($rows as $key => $row) {
$rows[$key]->href = 'index.php?option=com_auto&view=auto';
}
return $rows;
}
?>

```

XML-файл содержит описательные данные о подключаемом модуле для инсталлятора. Его содержимое показано ниже.

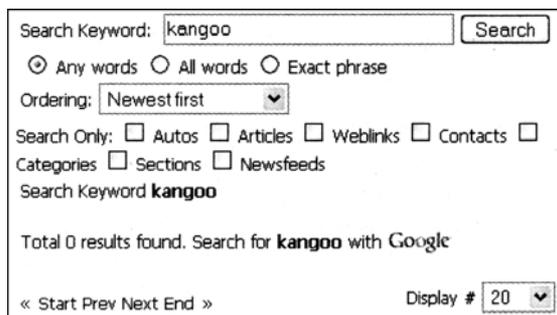
auto.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<install version="1.5" type="plugin" group="search">
<name>Search - Auto</name>
<author>Hagen Graf</author>
<creationDate>November 2007</creationDate>
<copyright>(C) 2007 cocoate.com. All rights reserved.</copyright>
<license>GNU/GPL</license>
<authorEmail>hagen.graf@gmail.com</authorEmail>
<authorUrl>www.cocoate.com</authorUrl>
<version>0.1</version>
<description>search plugin for the auto component</description>
<files>
<filename plugin="auto">auto.php</filename>
</files>
</install>

```

После инсталляции подключаемого модуля и активизации его через меню Extensions⇒Plugin Manager (Расширения⇒Диспетчер подключаемых модулей) ваш список будет включен в путь поиска, инициируемого из поля поиска веб-сайта. После ввода искомого фрагмента поля `text` и `manufacturer` базы данных будут участвовать в поиске, и результат отобразится на общей странице поиска (рис. 15.6).



The screenshot shows a search interface with the following elements:

- Search Keyword: kangoo
- Search button
- Search options: Any words, All words, Exact phrase
- Ordering: Newest first
- Search Only: Autos, Articles, Weblinks, Contacts, Categories, Sections, Newsfeeds
- Search Keyword: kangoo
- Total 0 results found. Search for kangoo with Google
- Navigation: « Start Prev Next End »
- Display #: 20

Рис. 15.6. Страница поиска

Этот подключаемый модуль поиска мы постарались сделать как можно проще. Ссылка на индивидуальное представление элемента списка должна быть помещена в позицию, где находятся результаты поиска, чтобы тот, кто запускал поиск, мог перейти туда. Но поскольку мы не построили индивидуального представления для нашего компонента, то не можем здесь поместить ссылку на него.

Резюме

Целью этой главы было дать вам представление о создании компонентов, модулей и подключаемых модулей.

На основе предложенных примеров вы можете легко в дальнейшем разрабатывать собственные компоненты. Наш компонент `auto`, например, имеет одно табличное представление. Посмотрите на компонент с индивидуальным представлением, как, например, `com_contact`, и пополните `auto` аналогичной функциональностью.

То же самое справедливо и в отношении назначения параметров в модулях. Найдите пример и создайте собственный модуль.

Вещи, которые кажутся на первый взгляд сложными, станут совершенно прозрачными, если вы взглянете на них еще раз.

Плодотворных исследований!

16

Построение веб-сайта с помощью Joomla!

Возможно, вы внимательно прочитали всю книгу вплоть до этой главы. Вы видели десятки административных страниц. Вы забивали себе мозги информацией о связи между веб-технологиями и структурами Joomla!. Вы познакомились с массой аббревиатур вроде HTTP, HTML, CSS, SQL, PHP, SEO, SEE DIV, MVC и т.п.

Но все, что вам было нужно — это веб-сайт! И потому, может быть, раскрыв книгу, вы сразу перешли к этой главе из практических соображений.

Как бы то ни было, добро пожаловать к рассмотрению конкретного примера. В этой главе описано построение веб-сайта от идеи до реализации.

Идея

Сайт, который мы опишем здесь, будет посвящен семье виноделов Бертран. Паскаль Бертран, владелец винного производства — представитель третьего поколения этой семьи, занимающегося этим бизнесом. Продукты, предлагаемые им, включают вина из различных сортов винограда.

До настоящего времени он поставляет свою продукцию оптовым торговцам вином, а летом продает ее непосредственно потребителям. Теперь мосье Бертран хочет продавать свое вино также через Интернет и, конечно, желает делать это с помощью Joomla!.

Разрабатываемый веб-сайт должен решать следующие задачи.

- Представлять семейный бизнес Бертранов.
- Распространять информацию о виноградниках и вине.
- Принимать онлайн-заказы.
- Предоставить мосье Бертрану возможность рекламировать новый продукт на веб-сайте.
- Включать галерею с изображениями и видео о сборе винограда и прочих событиях.
- Содержать внутреннюю область, где зарегистрированные пользователи могут получать специальные предложения и новости.
- Предоставлять возможность контакта посетителей с виноделом.
- Позволять вставку новых лент новостей о виноделии.

И все это надо сделать за два дня.

Подготовка

Прежде чем приступить к выполнению всех этих требований, необходимо провести некоторую подготовку.

Логотип и внешний вид

Семья Бертранов встретилась с Рут Прантц (Ruth Prantz) – другом и дизайнером, – чтобы обсудить компоновку и контент, который должен быть представлен на веб-сайте. Рут спросила, имеется ли логотип или какие-нибудь печатные материалы. Под доступными печатными материалами понимаются буклеты, флайеры, фирменные бланки и магазинная вывеска. Оказалось, что есть логотип, нарисованный бабушкой много лет назад (рис. 16.1).



Рис. 16.1. Эскиз логотипа

Этот логотип использовался семьей по разным поводам снова и снова. Однако полноценного изображения не было. Логотип печатался в разных цветах, с разными шрифтами и в разных стилях изображения и графики.

Рут также захотела узнать, каковы цели веб-сайта, и какой целевой группе он должен быть адресован.

Винодельня Бертран – экологически чистая компания, уделяющая особое внимание качеству производимого вина и использующая современные технологии. Бертран хотел бы продавать 5% своей продукции через веб-сайт. Целевая группа – любители вина и их друзья, в возрасте от 35 лет и старше.

Рут считает, что логотип, цвета и шрифты должны быть обновлены для соответствия современным требованиям, чтобы можно было легко отличить компанию от конкурентов.

Первый набросок, который появился на ее мониторе полчаса спустя, выглядел так, как показано на рис. 16.2.

Для дизайна логотипа Рут использовала Photoshop Elements. Множество провайдеров хостинга включают эту программу в качестве бесплатного компонента в пакет хостинга. Разумеется, вы можете применять другую графическую программу, такую как программа GIMP с открытым исходным кодом.

Семье понравился эскиз, и на этом Рут откланялась. У нее появилась основа для разработки шаблона Joomla! на базе согласованного наброска с выбранными цветами. А между тем, семья Бертранов занялась сбором материалов для веб-сайта и определением структуры их будущей системы Joomla!.

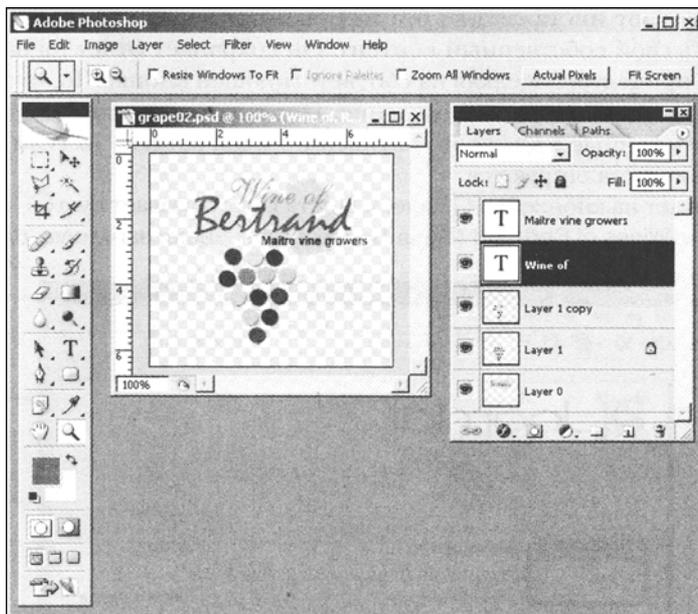


Рис. 16.2. Набросок логотипа в Photoshop

Фотографии

Для создания впечатления о вине, области, винограднике, работе и семье необходимы фотографии. Дидье и Марлен — сын и дочь владельца — получили задание подобрать в обширном семейном архиве фотографии и видеофрагменты для сайта, а также взять свои камеры на сегодняшний фестиваль вина, чтобы снять новые фотографии и видео о компании.

Тексты

Мосье Бертран решил лично заняться текстом, который появится на веб-сайте. Он уже писал в прошлом тексты бесчисленных рекламных материалов, а также статей для местной прессы и торговых изданий. У него также есть многочисленные документы о его владении и его вине, которые он отсканировал и предоставил для сайта в виде PDF.

Технические преобразования

Пока дети заняты сбором материалов, он занялся подготовкой структуры веб-сайта.

Локальная инсталляция

Он установил локальную систему Joomla! в среде XAMPP Lite, как описано в главе 3. Вместо подкаталога [Путь_к_Joomla!]/Joomla150, он использует [Путь_к_Joomla!]/bertrand. Информация для подключения к серверу MySQL выглядит так:

- имя хоста: localhost
- имя пользователя: root
- пароль: (оставлен пустым)
- имя базы данных: bertrand

Бертран запускает инсталляцию без загрузки примеров данных, поскольку собирается создавать свой собственный контент. Он сохраняет отчет на шестом шаге инсталляционной процедуры, щелкая на соответствующей кнопке. То, что отображается на экране, несколько вводит в заблуждение. Переключатель возле Install sample data (Инсталлировать данные примеров) активен по умолчанию. Но если не щелкать на нем, данные и не инсталлируются.

Бертран щелкает на кнопке Next (Далее). Инсталлятор создает пустой сайт Joomla!. Он уже может видеть Wines of Bertrand (Вина Бертрана) в панели заголовка окна (рис. 16.3).



Рис. 16.3. Черновой вид сайта после инсталляции Joomla!

Однако веб-сайт не совсем пуст. Мосье Бертран теперь щелкает на пункте меню Extensions⇒Module Manager (Расширения⇒Диспетчер модулей) в административной области. Здесь он может видеть активизированный модуль по имени Main Menu (рис. 16.4).



Рис. 16.4. Список модулей в Module Manager

Мосье Бертран тут же удивляется: "А где же все остальные системные модули, например, logon?". Краткий взгляд на пиктограмму New (Создать) его несколько успокаивает. Необходимые модули могут быть созданы через эту пиктограмму (рис. 16.5).

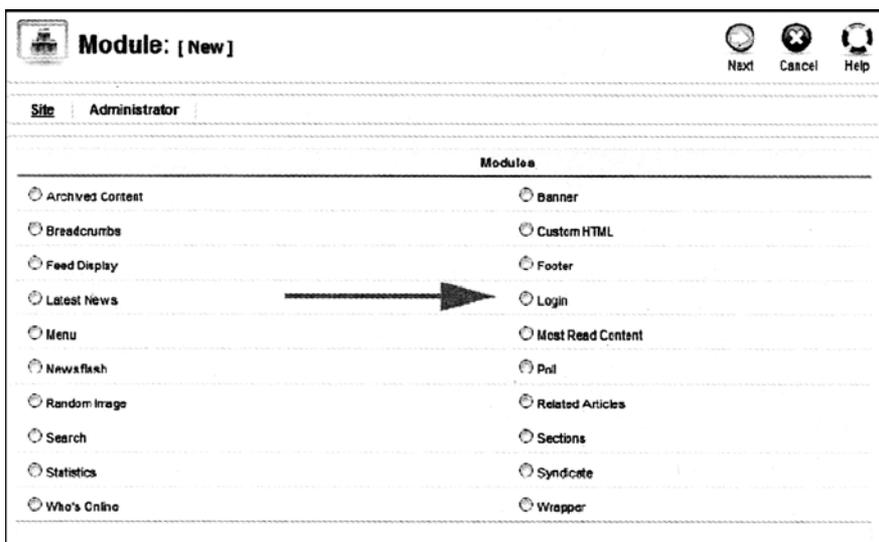


Рис. 16.5. Создание нового модуля

Первые несколько статей

Самое время заняться структурой. Мосье Бертран хотел бы видеть новости в форме блога на стартовой странице веб-сайта. Сначала он хочет установить три сообщения о новостях, чтобы посмотреть, как это будет выглядеть. Новости не должны быть категоризированы. Он заходит в меню Content⇒Article⇒New (Контент⇒Статья⇒Создать) и устанавливает три статьи. Он позиционирует курсор в место, куда желает поместить ссылку Read more (Читать далее), и щелкает на кнопке Read more под редактором; вставляется красная линия. В качестве Section (Раздел) и Category (Категория) остаются uncategorized (некатегоризировано); кроме того, выбирается переключатель Yes (Да) рядом с Front Page (Титульная страница).

Три кусочка новостей (три статьи) отобразились на веб-сайте и в пункте меню Content Start Page Articles (Статьи начальной страницы контента).

Затем мосье Бертран активизирует шаблон Beez на вкладке Templates (Шаблоны) диспетчера шаблонов.

Мосье Бертран желает видеть только дату¹ создания под заголовком фрагмента новостей. Он хочет скрыть имя и автора, но обновить дату. Поскольку он будет единственным, кто станет добавлять статьи на веб-сайт, ему нужно изменить установки в поле Author Name (Имя автора) и в двух полях дат ниже него.

Титульная страница теперь выглядит так, как он хотел. Joomla! автоматически создает наполнение новостей RSS для этих элементов новостей и отображает его в Internet Explorer версии 7 и выше, а также в браузере Firefox в виде оранжевого символа в поле URL.

Щелчок на этом символе в Firefox добавляет динамическую закладку (рис. 16.6).

Как только мосье Бертран пишет новую статью для титульной страницы, она немедленно отображается в его браузере. Разумеется, это работает и в других программах чтения наполнений. В последнее время эта концепция стала популярной даже в мобильных терминалах, таких как телефоны и PDA. Мосье Бертран надеется, что множество клиентов подпишутся на его службу и ознакомятся с предлагаемыми им продуктами.

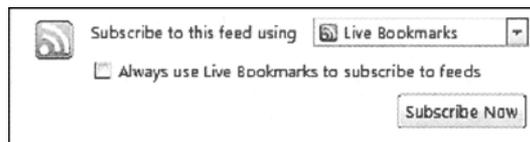


Рис. 16.6. Подписание на ленту новостей в Firefox

Masthead

Титульные данные (masthead) содержат различные части информации: посетитель веб-сайта должен иметь возможность видеть, кто отвечает за контент. В данном случае титульные данные выглядят так, как показано ниже.

Responsibility for the content of this site:

Fa. Bertrand

Examplestraße 1

12345 Exampleville

Exampleland

E-Mail: bertrand@cocoate.com

Telephone: 0123 4567-89

Поскольку этот сайт используется для коммерции, он также должен содержать идентификационный номер налогоплательщика.

Мосье Бертран также устанавливает другой некатегоризированный элемент контента через меню Contents⇒Articles⇒New. На этот раз он не публикует его на титульной странице. Ссылка на титульные данные теперь должна быть помещена в Main Menu. Мосье Бертран выбирает в меню Menus⇒Main Menu⇒New (Меню⇒Главное меню⇒Создать). Запускается диспетчер пунктов меню. Он выбирает Articles (Статьи) в первом окне и затем выбирает Standard Article Layout (Стандартный шаблон статьи).

Мосье Бертран переключается на экран редактирования нового пункта меню, вводит имя ссылки (Masthead (Титульные данные)) и теперь выбирает желаемую статью в области Parameters (Параметры).

После щелчка на кнопке Select (Выбрать) появляется другое окно выбора со всеми статьями.

Мосье Бертран щелкает на статье Masthead, окно выбора закрывается, и пункт меню готов (рис. 16.7). Финальный щелчок на пиктограмме Save (Сохранить) приводит к появлению ссылки в меню сайта.

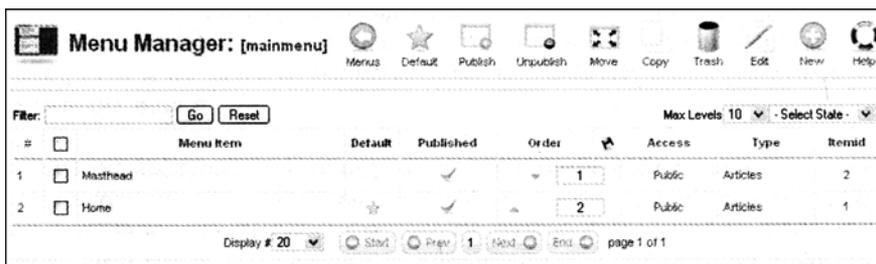


Рис. 16.7. Диспетчер меню с пунктом меню Masthead

Теперь Бертрану осталось убедиться, что ссылка активизирована (галочка зеленого цвета). Публикация здесь ведет просто к ссылке меню, но не самой статье. Ссылка теперь может быть позиционирована кнопками со стрелками зеленого цвета.

Сейчас Бертран желает скрыть дату создания и пиктограммы печати, PDF и электронной почты. Чтобы сделать это, он вызывает статью Masthead еще раз, выбрав в меню Content⇒Articles⇒Masthead, и изменяет значения в расширенных параметрах пиктограмм PDF, email (Электронная почта) и print (Печать) на Hide (Скрыть),

На этом титульные данные готовы, и мосье Бертран начинает работу над структурой ссылок меню для остальных страниц.

Структура меню

Мосье Бертран желает иметь три меню: главное меню, горизонтальное верхнее меню и пользовательское меню для зарегистрированных пользователей.

Структура главного меню

Полная структура веб-сайта должна быть показана в главном меню.

- Start Page (Новости)
- The wines (Общий текст)
- Wine 1 (Описание вина с условиями приобретения)
- Wine 2 (Описание вина с условиями приобретения)
- Wine 3 (Описание вина с условиями приобретения)
- The vineyard (Общий текст)
- Vineyard 1 (Описание виноградника со ссылкой на вино)
- Vineyard 2 (Описание виноградника со ссылкой на вино)
- Vineyard 3 (Описание виноградника со ссылкой на вино)
- Impressions (Компонент-галерея с фотографиями сбора урожая винограда)
- Offers (Специальные предложения в пользовательской области)
- Contacts (Контакты)
- Masthead (Титульная информация)

Структура верхнего меню

Верхнее меню должно обеспечивать быстрый доступ к четырем центральным темам. Если кто-то щелкает на ссылке в верхнем меню, то ссылки, относящиеся к ней, также открываются в главном меню.

- The wines (Общий текст)
- The vineyard (Общий текст)
- Impressions (Компонент-галерея с фотографиями сбора урожая винограда)
- Contacts (Контакты)

Структура пользовательского меню

После регистрации и успешного входа через модуль login зарегистрированные пользователи должны иметь возможность воспользоваться преимуществами специ-

альных предложений. Цель регистрации – выстраивание долгосрочных отношений с заказчиками. Регулярно обновляемая страница должна отображаться с привлекательными предложениями о продаже вина и информацией о том, как осуществить покупку. Мосье Бертран обеспечит доступ к сканированным документам для зарегистрированных пользователей внутри ссылки Download (Загрузить).

- Offers (Предложения)
- Download (Загрузить)
- Log Out (Выход)

Настройка текстов и ссылок в главном меню

Все тексты, которые Бертран установил в меню Content⇒Article Manager (Контент⇒Диспетчер статей), за исключением формы контактов и галереи, пока представляют собой некатегоризированный контент. Он убедился, что контент не отображается на титульной странице. Все статьи должны отображать пиктограммы PDF, e-mail и print, но не должны показывать дату создания.

После настройки всех статей он переключается на создание ссылок меню. Таким образом, чтобы Wine 1, Wine 2 и Wine 3 появились на втором уровне меню, должен быть выбран пункт The Wine.

В меню Menus⇒Main Menu находится древовидное меню.

Ссылка Contact

Мосье Бертран желает использовать интегрированный компонент contact, который поставляется с Joomla! для организации формы контактов. Сначала ему нужна категория по имени *Fa. Bertrand*, которая устанавливается через пункт меню Components1⇒Contacts⇒Categories (Компоненты⇒Контакты⇒Категории), как показано на рис. 16.8.



Рис. 16.8. Создание категории в диспетчере категорий

На вкладке Contacts (Контакты) он вводит информацию о себе как контактной персоне. Со временем сюда, возможно, будут добавлены многие контакты. А пока центральной формы должно быть достаточно.

Теперь он добавляет ссылку на контакт в главное меню. Для этого он щелкает на Menus⇒Main Menu⇒New, затем выбирает Contacts⇒Standard Contact Layout (Контакты⇒Стандартный шаблон контактов) в диспетчере пунктов меню и заполняет экран редактирования нового пункта меню. Доступные контакты теперь показаны в разделе параметров. На данный момент он – единственный контакт. Мосье Бертран помечает свое имя и щелкает на Save (Сохранить).

Теперь ему еще осталось опубликовать ссылку на контакт в `Menus⇒Main Menu` и расположить ее в правильном месте.

Информация о мосье Бертрane вместе с формой, которую пользователь может применять, чтобы задать вопрос, теперь присутствует в ссылке на контакты на веб-сайте.

Верхнее меню

За исключением интеграции компонента-галереи, главное меню готово. Теперь мосье Бертран создает верхнее меню (Top Menu). Он выбирает `Menus` в области меню и щелкает на пиктограмме `New` (Создать). Новое меню он называет `Top Menu`.

Это меню должно быть помещено в позицию `user3` и активировано. Мосье Бертран не приходится беспокоиться о заголовке, поскольку обычно в горизонтальных меню он не отображается.

Необходимые ссылки можно скопировать из `Main Menu` в `Top Menu`. Для этого он заходит в `Menus⇒Main Menu`, отмечает ссылки `The Wines`, `The Vineyards` и `Contact` и затем щелкает на пиктограмме `Copy` (Копировать), как показано на рис. 16.9.

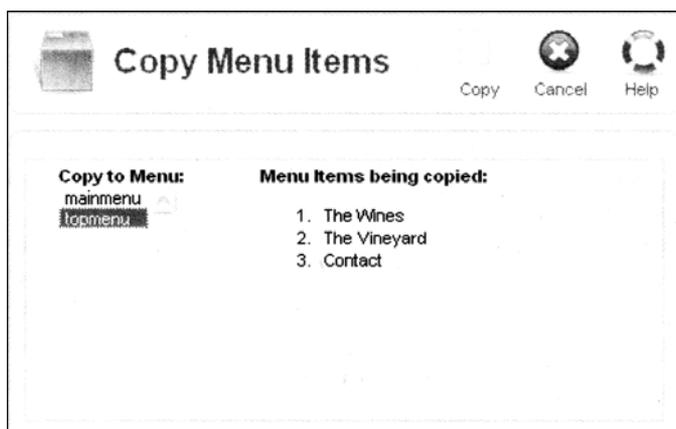


Рис. 16.9. Копирование пунктов меню

Ссылки скопированы. Мосье Бертран быстро проверяет результаты на веб-сайте. Ссылки действительно появляются на веб-сайте и отображаются в верхнем меню. При щелчке на них отображаются нужные статьи. Однако в главном меню не открывается правильное подменю. Когда он проверяет их, то замечает другие значения `ItemID` в URL. Когда он берет значение `ItemID` из главного меню и изменяет его вручную в URL в браузере, то ссылка ведет себя как надо, и подчиненные ссылки становятся видны в главном меню.

После некоторых размышлений мосье Бертран решает актуализировать ссылки верхнего меню как URL-адреса. Он хочет перенести две ссылки – `The Wines` и `The Vineyards` – из `Main Menu` в `Top Menu`.

Чтобы сделать это, он отправляется в `Menus⇒Top Menu⇒The Wines` и щелкает на кнопке `Change Type` (Изменить тип) на странице редактирования шаблона, а затем щелкает на `External Link` (Внешняя ссылка) на появившейся странице диспетчера пунктов меню.

Поле `Link` (Ссылка) отныне не отображается серым цветом, а становится доступным для ввода. Он вводит соответствующие ссылки. В данном случае это будут:

- ссылка `index.php?option=com_content&view=article&id=5&Itemid=4` для The Wines:
- ссылка `index.php?option=com_content&view=article&id=9&Itemid=8` для The Vineyards.

Перед `index.php` не должно быть слэша (`/`). В нашем случае это должно указывать на каталог документов веб-сервера Apache, и тогда будет отображена стартовая страница XAMPP Lite. Относительная ссылка не содержит полного объявления ссылки, а всегда рассматривается дополнением к соответствующему адресу вызова. Например, если некто переходит в другой домен, все относительные ссылки останутся работоспособными. Если же этот некто записывает все ссылки как абсолютные (с начальным слэшем), то для перехода в другой домен они все должны быть сначала изменены. Ваши `ItemID`, конечно же, будут иметь другие значения, чем у Бертрана.

Магазин

Чтобы пользователи могли платить в онлайн-режиме, на веб-сайте должна быть установлена система электронного магазина. Пока компонента электронного магазина, который бы работал с Joomla!, нет, и мосье Бертран решил установить как можно меньше дополнительных компонентов, чтобы избежать проблем с сопровождением и для минимизации необходимых обновлений.

Его сын Дидье посоветовал интегрировать электронный магазин PayPal, которая представляет собой крупнейшую в мире систему онлайн-платежей, принадлежащую eBay. Она предлагает многочисленные инструменты торговли, среди которых и систему электронного магазина. Мосье Бертран зарегистрировал пользовательскую учетную запись PayPal. Другие системы, такие как WorldPay и т.п., предлагают аналогичные средства.

На данный момент на продажу представлены три продукта:

- белое вино по 10 евро за бутылку;
- розовое вино по 10 евро за бутылку;
- красное вино по 12 евро за бутылку.

PayPal имеет опцию создания кнопок, которые связаны с определенными продуктами. Когда вы щелкаете на такой кнопке, соответствующий элемент помещается в корзину для покупок PayPal. Также могут быть добавлены затраты на доставку и необходимые налоги.

Мосье Бертран встроил эти кнопки в статьи Wine 1, Wine 2 и Wine 3. Он решил установить собственный модуль. Для этого он щелкнул на пиктограмме New (Создать) в Extensions⇒Module Manager (Расширения⇒Диспетчер модулей) и выбрал модуль HTML, после чего вставил код PayPal в представление HTML редактора (кнопку HTML). Чтобы несколько облегчить эту работу, он отключил редактор через меню Site⇒Global Configuration (Сайт⇒Глобальная конфигурация). Затем он поместил модуль слева под главным меню и выбрал страницы, которые должны отображать его.

В случае PayPal адрес электронной почты Бертрана должен быть помещен в исходный код. Стандартный подключаемый модуль Email Cloacking, который устанавливает Joomla!, маскирует адрес электронной почты, и потому PayPal не может найти правильную корзину для покупок. Мосье Бертран отключил этот подключаемый модуль через меню Extensions⇒Plugin Manager (Расширения⇒Диспетчер подключаемых модулей).

Теперь, если пользователь щелкает на одной из кнопок, появляется корзина покупок PayPal. Пользователь может поместить несколько элементов в корзину и изменить количество каждого наименования.

Если у заказчика есть учетная запись PayPal, он может использовать ее для немедленной оплаты покупки. Людям, покупающим вино в онлайн-режиме, имеет смысл обзавестись учетной записью PayPal, и эта тенденция быстро растет.

Impressions

Мосье Бертран хотел бы интегрировать на сайте изображения и видео, чтобы создать у посетителя ощущение, что он находится в сельской местности. Он установил альбом *Grave Harvest* (Урожай винограда) с вложенным альбомом 2007 г., затем альбом *Wine Festival* (Фестиваль вина) с вложенным альбомом 2007 г., а также альбом *Production* (Продукция) с несколькими фотографиями. Было принято решение, что его сын Дидье постепенно наполнит их картинками.

Выбрав *Menus*⇒*Main Menu*⇒*New*, он вставляет в меню ссылку на компонент *Exposé*. Затем он задает ссылку на заголовок *Impressions* (Впечатления) на экране редактирования ссылки меню. После первоначального сохранения он может поместить ее в правую позицию.

Пользовательский раздел

В разделе для зарегистрированных пользователей должна быть настроена страница с несколькими предложениями и областью загрузки статей в формате PDF, сертификатов, аэрофотоснимков и тому подобного.

Пользовательское меню

Мосье Бертран устанавливает некатегоризированную статью под заголовком *Offers* (Предложения). Статья публикуется, но не отображается на титульной странице, а доступна только зарегистрированным пользователям.

Он настраивает пользовательское меню в диспетчере меню.

Он позиционирует пользовательское меню слева, активизирует его и устанавливает права для зарегистрированных пользователей.

Он добавляет статью в новое пользовательское меню и указывает в качестве права доступа *Registered* (Зарегистрированные).

Мосье Бертран теперь может войти на веб-сайт. Он использует свою пользовательскую учетную запись *admin* и потому, рядом с каждым заголовком, в дополнение к вновь созданному- пользовательскому меню, может видеть пиктограмму с изображением карандаша.

Щелкая на пиктограмме с карандашом, он вызывает шаблон редактирования одной из статей. Здесь он может изменять текст статьи, параметры и даже метаданные. Работая со статьей, он замечает, что кнопка *Logout* (Выход) из модуля входа (*logon*) отображается только на титульной странице. Он заходит в *Extensions*⇒*Modules* и изменяет представление на *All* (Все) в разделе *Pages/Articles* (Страницы статьи модуля входа).

Раздел загрузки

И, наконец, мосье Бертран желает иметь раздел загрузки своих сканированных документов. Он хотел бы вывести небольшое изображение сертификатов в описании

своего виноградника, а также полную документацию в разделе для зарегистрированных пользователей. Для этого он использует компонент Weblink из Joomla!.

Он переключается на страницу Extensions⇒Web Links⇒Categories (Расширения⇒Веб-ссылки⇒Категории) и устанавливает три категории:

- Announcements (Анонсы)
- Certificates (Сертификаты)
- Ground surveys (Аэрофотосъемка)

Он добавляет маленькие описания к каждой категории и позднее планирует добавить изображения. Эти категории будут видимы только зарегистрированным пользователям.

PDF-файлы должны будут храниться в специальном каталоге в разделе медиа (Site⇒Media Manager (Сайт⇒Диспетчер медиа)). Он вводит имя каталога pdf в поле рядом с путем и щелкает на кнопке Create Folder (Создать папку).

Теперь он может загрузить все свои PDF-файлы. Он может также скопировать файлы в локальный каталог C:/xampplite/htdocs/bertrand/images/pdf с помощью проводника, или же загрузить их позднее на сервер посредством FTP.

URL-адреса PDF-файлов по-прежнему должны вводиться в виде ссылок в меню Components⇒Web Links⇒New.

Он также выбирает категорию, и одним щелчком определяет целью для PDF-файла New Window Without Browser Navigation (Новое окно без элементов навигации). Поскольку мосье Бертран работает в данный момент локально, URL-адреса ссылаются на http://localhost/bertrand/. Когда Бертран переместит этот сайт на свой веб-сервер, он должен будет модифицировать их в соответствии с будущим доменным именем.

После небольшого набора мосье Бертран добавляет несколько ссылок на различные PDF-файлы.

Эта коллекция ссылок еще должна быть помещена в пользовательское меню. Мосье Бертран выбирает Menus⇒User Menu⇒New, указывает компонент Web Links в диспетчере пунктов меню и затем выбирает Layout category list (Список категорий).

Он еще вводит заголовок для раздела (Download Section (Раздел загрузки)) в последующем редакторе шаблона. Доступ в раздел будут иметь только зарегистрированные пользователи. Он сохраняет ссылку и помещает ее ниже ссылки Offers. Ссылка появляется на веб-сайте, как и планировалось. Три категории отображены (рис. 16.10).

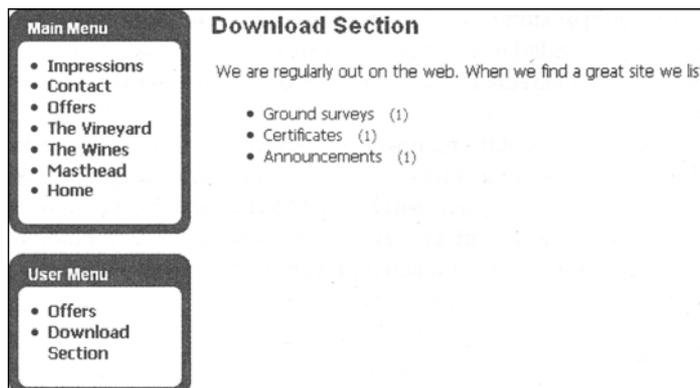


Рис. 16.10. Раздел Download Section

После щелчка на одной из категорий отображается таблица со списком PDF-файлов.

Мосье Бертран удовлетворен. Хотя это еще не настоящее администрирование документов, его вполне достаточно для достижения поставленных целей. Даже частота щелчков на индивидуальных документах отображается в таблице. Вид описания и всей таблицы может быть также определен через Menu⇒User Menu⇒Download⇒Parameters (Меню⇒Пользовательское меню⇒Загрузка⇒Параметры).

Подробности о пользователе

Мосье Бертран помнит, что он видел ссылку в пользовательском меню в примерах данных Joomla!, через которую пользователь может редактировать информацию о себе (адрес электронной почты, пароль и т.д.).

Ему также хочется иметь такую ссылку в пользовательском меню на его веб-сайте. Он выбирает Users⇒User Details⇒View (Пользователи⇒Подробности о пользователе⇒Показать) в диспетчере пунктов меню. Теперь пользовательский раздел имеет соответствующую ссылку и каждый пользователи может изменять свою информацию.

Первые результаты

Мосье Бертран приятно удивлен тем, насколько быстро у него все получается. Он звонит Рут и рассказывает о прогрессе в работе. Рут несколько растеряна, поскольку все шаблоны, которые она строила до сих пор, были ориентированы на Joomla! 1.0.x, а для новой версии многое придется менять. Но она заверяет, что сможет показать им шаблоны на следующий день.

Она тут же отправляет два экранных снимка своей работы по электронной почте (рис. 16.11).

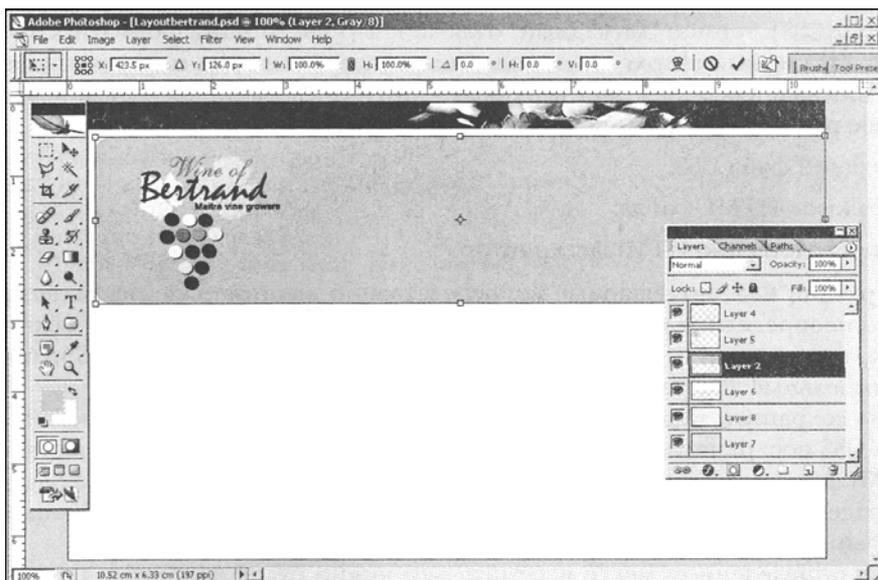


Рис. 16.11.Набросок шаблона Рут

Занимаясь интеграцией Joomla! 1.5, Рут быстро обнаруживает, что многие классы CSS изменили имена со времен версии Joomla! 1.0.

Чтобы получить обзор классов CSS и остальных параметров. Рут использует Firefox Web Developer (файл `web_developer-1.0.2-fx+fl.xpi`).



Если в вашей системе уже инсталлирован браузер Firefox, просто дважды щелкните на имени файла для его инсталляции.

Рут и Бертран соглашаются с тем, что мосье Бертран может уже инсталлировать веб-сайт на своем сервере. Она добавит шаблоны позднее.

Мосье Бертран изучает CSS

Мосье Бертран уже неплохо разбирается в HTML, и хотел бы узнать, как работают шаблоны.

Длительная беседа...

Рут объясняет ему, что внешний вид шаблонов определяется двумя файлами CSS, которые сообщают браузеру, как следует отображать веб-сайт. Например, должен фон быть красным или коричневым, должны ли вставляться графические элементы, и многие другие вещи.

Эти инструкции записываются командами каскадных таблиц стилей и помещаются в два файла CSS — `template.css` и `joomlastuff.css`.

Рут оптимизирует эти файлы для наиболее распространенных браузеров Internet Explorer 6.0 и Firefox 1.5 под Windows, а также для Firefox 1.06 и Safari 2.03 для Mac OS X 10.4.

Мосье Бертран желает научиться понимать CSS, чтобы уметь изменять шаблоны или создавать собственные.

Рут объясняет ему принципы организации каскадных таблиц стилей (Cascading Stylesheets — CSS): термин "каскадный" означает, что инструкции форматирования CSS читаются браузерами сверху вниз, подобно каскаду. Если одна инструкция написана дважды, нижняя переопределяет верхнюю — другими словами, последняя переопределяет более ранние. Инструкции CSS могут помещаться в следующие места:

- внешний файл CSS;
- заголовок HTML-файла;
- непосредственно HTML-дескриптор.

Инструкции CSS, помещаемые непосредственно в соответствующий дескриптор HTML, переопределяют форматирование из внешних CSS-файлов. Это важно знать, что поскольку элементы Joomla! иногда включают форматирование CSS непосредственно в отдельные дескрипторы HTML.

Чтобы сохранить таблицу стилей шаблона для мосье Бертрана простой. Рут использует CSS иерархически. Например, она определяет информацию о шрифте в общих HTML-дескрипторах: `body`, `table`, `div`, `p` и `form`. Таким образом, она определяет, что все содержимое этих элементов должно отображаться в гарнитуре Verdana, с определенным размером шрифта и серым цветом.

Если в определенном месте веб-сайта это нужно будет изменить, то соответствующим HTML-дескрипторам необходимо будет назначить класс; например, класс `greytable` с такими свойствами:

```
{ background-color: #333; }
```

Он интегрируется в каркас HTML следующим образом:

```
<table class="greyclable">
```

В результате индивидуальные элементы могут контролироваться по-разному классами CSS.

В Joomla! есть множество predefined классов, посредством которых любой может легко программировать свои собственные шаблоны. Взглянем, например, на навигацию по сайту), показанную на рис. 16.12.

Навигация по сайту использует три модуля Joomla!:

- Main Menu (Главное меню)
- User Menu (Пользовательское меню)
- User Login (Вход пользователей)

Рут определила компоновку навигации сайта с помощью следующих классов.

HTML-код навигации по страницам:

```
...
<div class="moduletable_menu">
<h3>main menu</h3>
<ul class="menu">
<li id="current" class="active item1">
<a href="http://localhost/bertrand/">front page</a> </li>
<li class="parent item4">
<a href="index.php?option=com_content...">The Wines</a>
</li>
<li class="parent item8">
<a href="index.php?option=com_content...">Wineyard </a> </li>
...
</ul>
</div>
<div class="moduletable">
<h3>Announcement</h3>
<form action="index.php" method="post" name="login" id="form-login">
<fieldset class="input">
<p id="form-login-username">
<label for="modlgn_username">user name</label><br>
<input id="modlgn_username" name="username" class="inputbox"
alt="username" size="18" type="text">
</p>
...
</fieldset>
<ul>
<li>
<a href="index.php?option=com_user...">Forgot your password?</a>
</li>
...
</ul>
<input name="option" value="com_user" type="hidden">
</form>
</div>
<div class="clear"></div>
...
```

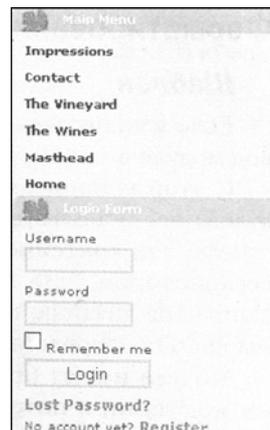


Рис. 16.12. Навигация по сайту

Форматирование шаг за шагом

Шаблон

Если взглянуть на код HTML, в нем легко распознать шаблон. Рут определила вывод модуля в `index.php` конструкцией `style="xhtml"`.

С этой установкой, а также с установкой параметров модуля (в диспетчере модулей) все используемые здесь модули выводятся в "списках" внутри дескрипторов `<div>`. Эти контейнеры `<div>` содержат заголовок и списки, определенные HTML-дескриптором `<h3>`. Ссылки располагаются в списках. При этом доступно множество вариантов дизайна, поскольку все дескрипторы HTML могут быть запрограммированы либо непосредственно, либо через классы Joomla!

Модули имеют приятный глазу графический фон, который помещается туда всеми модулями, и потому был определен во включающем дескрипторе HTML. Все дескрипторы, которые придуманы Рут для ее структуры HTML, находятся в файле `template.css`. Это также верно для повсеместной страничной навигации.

CSS-код `.naviside`:

```
.naviside {
background: url (../images/naviside_top.jpg) top left; /*фоновое изображение*/
width: 210px; /*ширина*/
vertical-align: top; /*контент выравнивается по верхнему краю*/
border: 0px; /*рамка*/
padding: 0px; /*заполнение*/
margin: 0px; /*границы*/
background-color: #fff; /*цвет фона*/
text-align: left; /*выравнивание по левой рамке*/
float: left; /*общее выравнивание внутри HTML-структуры*/
}
```

На следующем шаге Рут назначает свойства дескрипторам `<div>`, которые активизируются здесь классами `moduletable_menu` и `moduletable` (форма `login`) и должны быть действительны для всех меню: ширина, расстояние между соседними пунктами и цвет. Для этого ей нужно знать только два класса — `moduletable` и `moduletable_menu`.

Классы `moduletable` и `moduletable_menu` — это классы по умолчанию Joomla!. Для просмотра Рут сохранила классы, определенные ею, отдельно от стандартных классов Joomla! в CSS-файлах `joomla_stuff.css` и `template.css`.

Код CSS:

```
.naviside .moduletable, .moduletable_menu
{
width: 210px; /*ширина*/
background: none; /*фон*/
margin-top: 18px; /*поле сверху от модуля*/
text-align: left; /*выравнивание влево*/
padding: 0px; /*никакого заполнения от края до контента*/
}
```

Заголовок модуля

Легко создать заголовок, который будет выглядеть одинаково для всех модулей, указав непосредственно заголовок таблицы в дескрипторе `<h3>` внутри модуля.

Для этого потребуются сначала опять установить класс в CSS, в данном случае — `.navitop`. С классом `.navitop` все модули находятся внутри этого `<div>`. Это дает

тот эффект, что только оператора заголовка внутри раздела `.navitop` форматируется этими командами CSS. И теперь добавлен ключевой `<h3>`. Рут установила заголовок с фоновым изображением, чтобы иметь больше разнообразия в меню.

Заголовок модуля:

```
*/Определение заголовка модуля слева*/
.navitop h3 {
font: 18px/18px Times New Roman, serif; /*размер/тип шрифта*/
font-weight: normal; /*вес шрифта*/
color: #A49A66; /*цвет шрифта*/
background: url(..images/blatt.gif) top left; /*фоновое изображение*/
background-repeat: no-repeat; /*повторяющееся фоновое изображение*/
padding-left: 25px; /*дополнение слева*/
height: 25px; /*высота div*/
width: 185px; /*ширина*/
}
```

Пункты меню

После того, как с заголовком покончено, должны быть сформатированы ссылки, устанавливаемые в списках HTML (рис. 16.13).

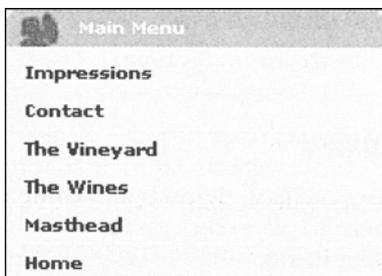


Рис. 16.13. Главное меню с измененным стилем элементов

Рут не хотела иметь никаких точек перед ссылками, а также избавиться от обычного сдвига элементов списка. Она использовала для этого предопределенные в Joomla! классы CSS `.menu` и `.moduletable`.

Форматирование списка в модулях:

```
ul.menu, .naviside .moduletable ul
{
list-style: none; /*список без точек*/
display: block; /*блок элементов, сформатированный в виде прямоугольника*/
width: 210px; /*ширина*/
margin: 0px; /*границы*/
border: 0px; /*рамка*/
padding: 0px; /*заполнение*/
}
```

Линии-разделители

Рут хотела отобразить индивидуальные ссылки, разделенные линиями. Для этого она использовала дескрипторы `id` для списков специальным образом:

```
ul.menu li {
border-bottom: 1px dotted #ccc; /*нижняя граница*/
}
```

Подзаголовки ссылок могут обрабатываться по отдельности; в данном примере они сдвинуты слева.

Подзаголовки:

```
/*форматирование подзаголовков списков*/
ul.menu ul {
/*поле служит для сдвига текста*/
margin-left: 10px;
background-color: #EEE; /*другой цвет фона*/
width: 200px; /*меньшая ширина*/
}
```

Подзаголовки получают совершенно иначе форматированную нижнюю границу которая заполняется цветом фона.

```
ul.menu ul li {
border-bottom: 1px solid #fff; /*нижняя граница*/
}
```

Каждый элемент списка может быть сформатирован отдельно, поскольку каждый дескриптор `li` имеет свой собственный класс.

```
.level1 item1, .level1 item2 parent, .level1 item3 parent,
.level1 item4
```

Действительные ссылки

Теперь перейдем к самим ссылкам. Форматирование касается дескриптора, который позиционируется в списке `ul` классом `.menu`.

Все ссылки должны выглядеть так:

```
ul.mainmenu a {?
text-decoration: none; /*ссылка без подчеркивания*/
color: #6B5E588; /*цвет шрифта*/
font-size: 11px; /*размер шрифта*/
padding-left: 10px; /*расстояние от текста до рамки*/
}
```

Наведение курсора мыши изменяет фон и цвет шрифта ссылки. Рут настроила это следующим образом:

```
ul.menu a:hover {
color: #CAC303;
text-decoration: none;
}
```

И, наконец, ссылка на текущую активную страницу специально подсвечивается с помощью следующего класса:

```
.menu li#current a {
color: #B22819; /*цвет шрифта*/
}
```

Теперь все готово.

Мосье Бертран доволен, и теперь, после объяснение, находит форматирование CSS намного более простым. Его голова слегка кружится, но Рут также дала ему ссылку www.w3schools.com/css/. Этот сайт содержит хорошее описание всех атрибутов, которые могут быть определены в CSS.

Изменения для Joomla! версии 1.5

Если вы уже строили шаблоны с Joomla! 1.0.x, для вас есть хорошая новость: Joomla! 1.5 обратно совместима в унаследованном (legacy) режиме.

Существует, однако, несколько нюансов, о которых следует знать. Ранее использованные команды `mosload()` в коде PHP и старые классы CSS и в `index.php` шаблона в ряде мест не работают, поскольку изменилось некоторое форматирование.

Изменения в меню

По сравнению с Joomla! 1.0, появилось много опций для форматирования модулей и компонентов. Вы могли управлять модулями разными способами, назначая им их собственные расширения классов CSS; теперь их HTML-вывод может также контролироваться следующими предложениями в собственном `index.php` шаблона.

Ниже перечислены опции, из которых можно выбирать.

- `table` (по умолчанию). Модуль отображается в колонке таблицы.
- `horz`. Модуль отображается горизонтально и выводится в ячейку окружающей таблицы.
- `xhtml`. Модуль выводится в простой элемент `div`.
- `rounded`. Вывод выполняется в формате со скругленными углами; при этом имя `<div>` изменяется с `moduletable` на `module`.
- `none`. Модуль выводится без какого-либо форматирования.

Отлично! Все это позволяет проектировать дизайн сайта Joomla! без каких-либо таблиц и организовывать контент логически (в соответствии с требованием **свободы от барьеров**).

Используя дескрипторы `<div>`, вы можете проектировать с меньшим количеством ограничений, чем с таблицами HTML, которые все равно никогда не предназначались для этой цели.

Инсталляция на веб-сервере

Локальный веб-сайт мосье Бертрана теперь выглядит примерно так, как мы представляли. В течение нескольких недель он искал подходящего хостинг-провайдера для своего веб-сайта, и после долгих поисков решил стать клиентом небольшой компании, расположенной по соседству. Эта компания предложила ему SSL-защищенный административный интерфейс по имени Plesk для настройки его веб-сервера, базы данных, конфигурации электронной почты и т.п.

PHP выполняется в безопасном режиме (Safe Mode). База данных может администрироваться инсталлированным средством phpMyAdmin. Мосье Бертран хотел иметь веб-ориентированное администрирование своего сервера, потому что не желал изучать громоздкие конфигурационные файлы.



Существует множество других интерфейсов, среди которых Confixx, Visas и Webmin. Многие провайдеры, такие как 1&1, Strata, Hetzner и другие, используют свои собственные разработки. Чуть ниже мы поговорим кратко о Plesk, как о примере. Другие интерфейсы работают аналогично.

Как выполняется инсталляция

Мосье Бертран воспользовался веб-инсталлятором Joomla! для установки программы в своей локальной среде. Затем он потратил утро на ввод данных и не хотел бы потерять эти данные в рабочей инсталляции. Здесь у него, как минимум, два варианта выбора.

- Можно загрузить все файлы Joomla! по FTP на сервер провайдера, изменить конфш урацнонный файлы вручную и импортировать базу данных посредством phpMyAdmin.
- Можно загрузить файл Joomla! на сервер провайдера по FTP, инсталлировать его с помощью инсталлятора Joomla!, а затем импортировать данные базы посредством phpMyAdmin.

Мосье Бертран отдал предпочтение первому варианту. Чтобы иметь возможность приступить к инсталляции, ему нужен доступ к данным по FTP, СУБД MySQL и, конечно, доменное имя.

В своем административном интерфейсе он находит следующий доступ к данным.

FTP

Имя хоста: `bertrand.cocoate.com`

Имя пользователя: `fusfusfus`

Пароль: `pwрwрw`

MySQL

Имя хоста: `localhost`

Имя пользователя: `dbusdbus`

Пароль: `pwрwрwрw`

Имя базы данных: `dbdbdb`

Инсталляция Joomla! у провайдера

Мосье Бертран запускает программу FTP и вводит необходимую информацию. Он получил программу FTP от своего провайдера бесплатно (вы можете найти также бесплатную программу FTP на www.filezilla.sourceforge.net/).

Он переносит существующую инсталляцию из подкаталога `c:\xampplite\htdocs\bertrand`, используя FTP, в подкаталог `httpdocs` сервера провайдера.

Он делает резервную копию файла `configuration.php` на своем собственном ПК и открывает его в текстовом редакторе.

Чтобы локальная версия `configuration.php` могла работать также и на сервере, потребуется изменить несколько значений:

```
/* Настройки базы данных */  
...  
var $host = 'localhost'; // обычно устанавливается в localhost
```

```

var $user = ''; // имя пользователя MySQL
var $password = ''; // пароль MySQL
var $db = ''; // имя базы данных MySQL

```

После проведения этих изменений он загружает файл посредством FTP в каталог `htdocs` на сервере и назначает ему права `chmod 0777`. Программы FTP могут выполнять эту команду и с ее помощью назначать права доступа к каталогу. Joomla! требует как минимум `chmod 755`; для файлов достаточно `chmod 644`.

Импортирование данных

Для того чтобы иметь возможность импортировать данные в базу MySQL у провайдера, они должны быть сначала экспортированы из локальной версии XAMPP Lite. Мосье Бертран обращается по адресу `http://localhost/phpmyadmin` в браузере, выбирает базу данных `bertrand` и щелкает на вкладке `Export` (Экспорт). Далее он щелкает на `Select All` (Выбрать все), чтобы выбрать все таблицы для экспорта.

- В разделе `Structure` (Структура) он выбирает `Structure` (Структура) и `Add DROP TABLE / DROP VIEW` (Добавить DROP TABLE / DROP VIEW). Это приводит к вставке дополнительной команды SQL в файл экспорта, которая гарантирует, что при последующем импорте данных существующие одноименные таблицы из базы данных будут удалены. Это важно для мосье Бертрана, поскольку в результате работы веб-инсталлятора все таблицы уже находятся там.
- В разделе `Data` (Данные) он выбирает `Data` (Данные) и `Extended Inserts` (Расширенные вставки).
- В разделе `Compression` (Сжатие) он выбирает переключатель `Zip-Compressed` (Сжатие Zip), потому что это немного ускорит передачу.
- Он щелкает на кнопке `Go` (Выполнить) для получения загруженных данных в форме файла.

Затем он щелкает на кнопке `OK`, и файл `bertrand.sql` готов к загрузке. Вкладка экспортирования `phpMyAdmin` показана на рис. 16.14.

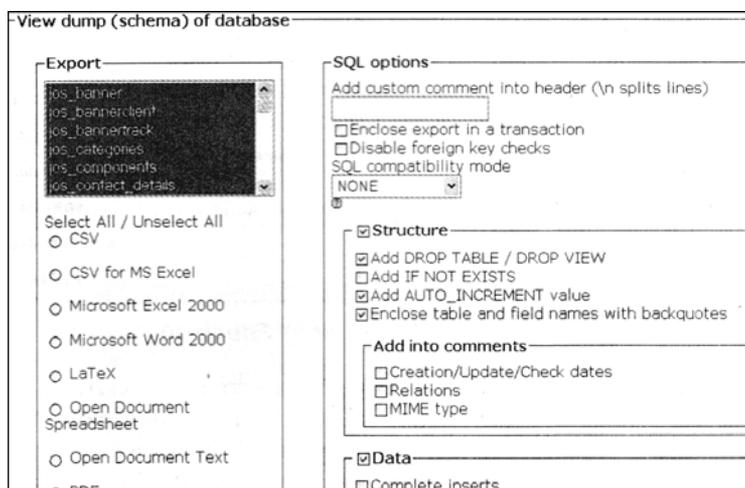


Рис. 16.14. Вкладка экспортирования

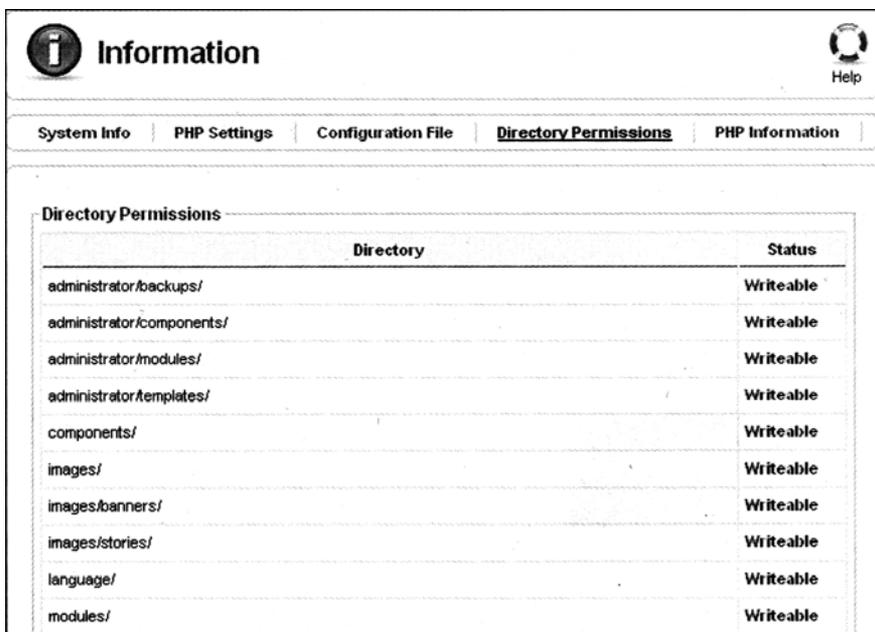
В этом файле сохранено все содержимое базы данных. Файл располагается где-то за пределами каталога Joomla! и после распаковки получается файл `bertrand.sql`. Файл содержит команды SQL, которые в точности воспроизводят состояние локальной инсталляции и, выполненные последовательно, воссоздадут ее в точном виде на другой системе MySQL.

Именно это и нужно мосье Бертрану, поэтому он открывает phpMyAdmin на сервере своего провайдера. Затем он щелкает на вкладке Import (Импорт) в своей базе данных (которая также называется `coco_bertrand`). На странице он щелкает на кнопке Search (Поиск) и выбирает локальный файл `bertrand.sql`. Один щелчок на кнопке OK – и файл загружен. Команды SQL выполняются последовательно, и его веб-сайт на сервере теперь представляет собой копию локальной версии.

Веб-сайт на сервере теперь должен быть готов, и в нашем случае выглядит, как <http://bertrand.cocoate.com>.

Права доступа к файлам и каталогам

Чтобы гарантировать нормальную работу Joomla!, следует подкорректировать права доступа к каталогам. Мосье Бертран проверяет настройки через меню Help⇒System Info (Справка⇒Информация о системе). Вкладка Directory Permission (Права доступа к каталогам) показана на рис. 16.15.



Directory	Status
<code>administrator/backups/</code>	Writeable
<code>administrator/components/</code>	Writeable
<code>administrator/modules/</code>	Writeable
<code>administrator/templates/</code>	Writeable
<code>components/</code>	Writeable
<code>images/</code>	Writeable
<code>images/banners/</code>	Writeable
<code>images/stories/</code>	Writeable
<code>language/</code>	Writeable
<code>modules/</code>	Writeable

Рис. 16.15. Вкладка Directory Permission

Сервер провайдера работает под Linux. В Linux есть пользователи и группы пользователей. С мосье Бертраном связан пользователь, и с веб-сервером Apache также связан пользователь. В зависимости от конфигурации, они оба могут принадлежать к одной или же к разным группам пользователей. Если они находятся в разных группах, то права доступа к каталогу должны быть установлены как `chmod 0777`.

Мосье Бертран также изменяет пароль для галереи Exposé в административном разделе. Сразу после инсталляции он установлен в manager.

Механизмы поиска

Существуют два подхода к включению веб-сайта мосье Бертрана в механизмы поиска.

SEF URL

Выбор SEF URL также зависит от провайдера. Провайдер мосье Бертрана разрешает такие URL, и Бертран активизирует это средство через меню Site⇒Configuration⇒SEO. Затем он переименовывает файл htaccess.txt в .htaccess. Галерея Exposé, например, теперь доступна по URL-адресу <http://bertrand.cocoate.com/Impressions>.

Такое переименование URL работает только с веб-сервером Apache.

Мета-дескрипторы

Мосье усердно вводит краткие описания и ключевые слова в свои статьи. Дидье и Марлен подобрали массу фотографий. Мосье Бертран создал веб-сайт, как ему нравится, и даже интегрировал в него PDF-файлы.

Соединение PayPal работает.

Все они встретились за завтраком, а Рут между тем закончила с шаблоном. Она принесла готовый компонент на USB-флешке и целый час рассказывала о своем опыте ереноса шаблона на новую версию Joomla!.

Инсталляция шаблона

Мосье Бертран скопировал файлы шаблона в каталоги /httpddocs/templates/tmpl_bertrand и /httpddocs/templates/tmpl_bertrand_shop посредством FTP. В меню Extensions⇒Templates (Расширения⇒Шаблоны) он определил новый шаблон tmpl_bertrand в качестве шаблона по умолчанию (рис. 16.16).

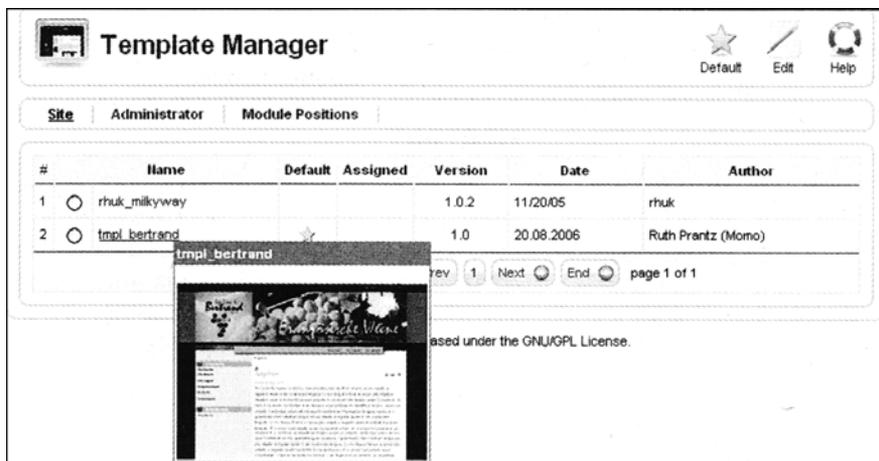


Рис. 16.16. Определение нового шаблона по умолчанию

Он назначает шаблон `tmpl_bertrand_shop` для страниц вина в режиме редактирования. Таким образом, заказчик будет знать, что он находится в отделе магазина.

Шаблон

Рут создала шаблон в соответствии с описанными ниже соображениями.

Сопровождение

Поскольку мосье Бертран будет поддерживать веб-сайт самостоятельно, Рут использовала простой и ясно организованный базовый шаблон сетки. Мосье Бертран, когда пожелает, может легко заменять изображения в верхнем разделе новыми.

Компоновка

Рут объяснила свой выбор следующим образом.

Существует много способов творческой реализации веб-сайта.

Мосье Бертран хочет продавать вино.

Это значит, что его веб-сайт нуждается в визуальной презентации традиционного качественного производителя, у которого заказчик с удовольствием будет покупать хорошее вино.

Презентация в стиле "хай-тек" с металлическим видом была бы неподходящей, даже несмотря на то, что выглядела бы более современно.

Поэтому вино, виноград и земля должны определять внешность и поведение веб-сайта, что достигается выбором цвета и фотографий.

Визуальное представление состоит из нескольких важных элементов. К ним относятся: логотип, цвета, стиль фотографий и графики (под графикой она подразумевает все изобразительные элементы, не являющиеся фотографиями, включая кнопки и символы), а также гарнитуры шрифтов.

Деление пространства и пропорции также оказывают влияние, и все эти элементы должны работать вместе на согласованное визуальное впечатление. Лучше всего использовать только одну гарнитуру шрифта. Различия легко можно подчеркнуть полужирной и курсивной версиями того же самого шрифта.

В шаблоне Бертрана она не использовала слишком оригинальной гарнитуры для заголовков, а предпочла классическую, что создает впечатление надежности. Дизайн не всегда должен быть слишком оригинальным; достаточно, чтобы он был подходящим.



Рис. 16.12. *Набросок Рут*

Портрет

За завтраком и во время телефонных переговоров, когда Рут объясняла технологию CSS мосье Бертрану, она сделала небольшой набросок. Этот набросок она отправила мосье Бертрану по электронной почте, и он включил его в свою галерею изображений (рис. 16.17).

Благополучное завершение

Сайт готов и доступен. На все ушло два дня.

Мосье Бертран потратил следующие две недели на вставку фотографий со сбор урожая винограда и отшлифовку текста. Он связался со всеми знакомыми и пригласил их зарегистрироваться в качестве пользователей на сайте, и даже получил первый заказ через новую систему.

Провайдер обсудил с Бертраном проблемы обновления Joomla!, и через некоторое время был подписан договор на обслуживание. Провайдер взял на себя обязательства проводить обновление при каждом выходе новой версии Joomla!.

Такой сайт будет относительно несложно поддерживать в актуальном состоянии, так как он использует стандартную версию Joomla! и поскольку галерея Exposé работает в HTML-элементе `iframe`, другими словами – в компоненте-оболочке.

Мосье Бертран доволен, что он перепоручил работу над обновлениями и шаблоном.

Как это работает в действительности?

Огромное число веб-сайтов мелких компаний создаются примерно так же, как в описанной истории с мосье Бертраном. Разработка крупных сайтов чаще всего начинается аналогично, хотя компании не любят это признавать.

В более крупных проектах процесс, в принципе, тот же. Графическое агентство выполняет дизайн, а техническая компания – разработку и программирование. Часто привлекаются профессиональные подразделения, которые занимаются текстом, корпоративной идентичностью и безопасностью веб-сайта. Хостинг также должен быть утвержден.

Однако в остальном все делается так, как мы описали!

Резюме

Наконец-то вы можете приступить к созданию собственного веб-сайта. В следующей главе мы обсудим несколько бесплатных шаблонов-бонусов.

17

Шаблоны-бонусы

Можно поспорить, что, ознакомившись с опытом мосье Бертрана и инструкциями по созданию шаблонов в главе 16, у вас появится желание разработать свой собственный шаблон. Благодаря Алексу Кемпкенсу (Alex Kempkens), который организовал разработку этих шаблонов, мы можем дать вам несколько идей и предложить два готовых шаблона, чтобы вы могли отправиться в свое путешествие.



Обратите внимание, что вы можете свободно использовать эти шаблоны в своих собственных проектах либо в качестве основы дизайна -роектов для ваших заказчиков, а также частных инсталляций Joomla!. Но есть два ограничения: вам не разрешено продавать эти шаблоны как таковые (только авторы этих шаблонов имеют такое право), а также вы должны соблюдать некоторые ограничения, связанные с использованием контента (подробности перечислены в соответствующих сайтах сопровождающих каждый шаблон).

Выбирая эти шаблоны, мы хотели представить вам разнообразные веб-сайты из разных областей и от разных дизайнеров.

Вот шаблоны, которые мы выбрали, и которые вы можете свободно использовать:

- шаблон от Тома Бохачека (Tom Bohacek) для негосударственных организаций;
- шаблон от Энди Миллера (Andy Miller) для коммерческого предприятия.

Во время написания этой книги глава о виноделении мосье Бертране и его амбициях в области электронной коммерции все росла и росла, и в результате появились еще два дополнительных шаблона, которые не входили в первоначальные планы:

- два шаблона от Рут Прантц (Rut Prantz) для винодельческой компании среднего размера в Лангедоке, Франция.

Порядок создания этих шаблонов и актуализации веб-сайта был изложен в главе 16.

NGO

Термин *негосударственная организация* (non-governmental organization — NGO) ссылается на любое организованное объединение людей, которое самостоятельно принимает решение о начале некоторой деятельности на определенный период времени, без намерения получить прибыль, и организованное независимо от каких-либо государственных структур. Примерами негосударственных организаций являются ассоциации работодателей, профсоюзы, спортивные клубы и ассоциации по защите животных.

Любая ассоциация, которая основана в соответствии с законом своей страны в общественных интересах, является потенциально негосударственной организацией.

Том Бохачек

Вместе со своим агентством дизайнер Том Бохачек поддерживает другие агентства и предприятия в разработке их решений, связанных с коммуникациями. Он работает с Joomla! по легко понятной причине:

"Эта система дает мне максимальную свободу в реализации моих идей."

Почитайте, что Том говорит о своих шаблонах.

Шаблон World Knowledge

Идея, положенная в основу, состоит в разработке шаблона, который может удовлетворить требования к меняющемуся состоянию информации из многочисленных источников NGO.

Пример воображаемой негосударственной организации **World Knowledge**, которая провозглашает своей целью сохранение культурного наследия и знаний, был придуман (рис. 17.1).

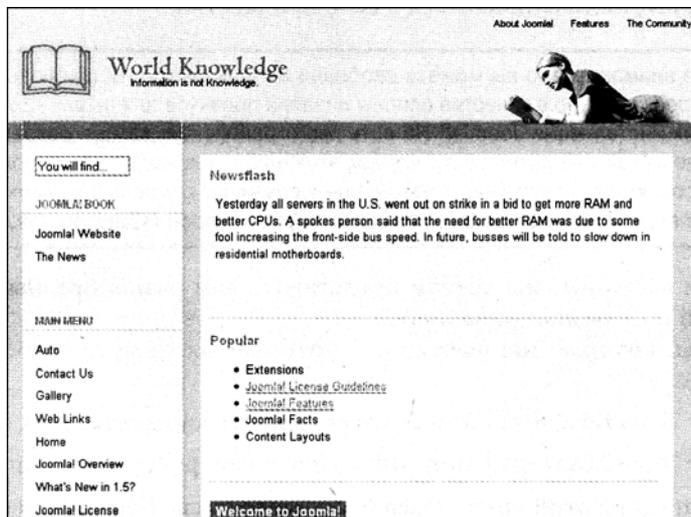


Рис. 17.1. Сайт World Knowledge

Творческий подход

"Каждая теория сера, мой дорогой друг!" — на основе этой цитаты Гете была разработана дружественная цветовая схема, напоминающая о пергаменте и библиотеках, что ассоциируется со знаниями. Для выделения важных элементов и помощи в ориентации был добавлен светло-бирюзовый цвет. Этот цвет, выпадающий из компоновки, символизирует знание как таковое, и использован в заголовке.

Поскольку ХГО часто работают в мультикультурной среде, компоновка выбрана ясная и нейтральная. Это гарантирует, что люди с разных континентов смогут в ней быстро сориентироваться.

Семейство шрифтов Helvetica, которое приобрело популярность после появления Web 2.0 Нуре, было выбрано не только из-за его популярности, но также потому, что оно символизирует компетентность и ясность, а также потому, что оно инсталлировано почти на всех системах.

Структурный подход

Компоновка веб-сайта делится на четыре части — раздел заголовка, панель навигации и раздел контента, состоящий из двух подчастей. Раздел контента включает статьи и дополнительный раздел с изображениями и прочей информацией, такой как ссылки на другие сайты, связанные с основным контентом.

Навигация слева *не* содержит второго уровня навигации. Она может, однако, быть выведена из нескольких разделов навигации (например, Present (Настоящее) и Future (Будущее)). Для того чтобы связаться с организацией непосредственно, предусмотрен;! прямая ссылка на страниц); которая ведет глубже в пункт меню — к статье или блогу. Вся более глубокая информация показывается либо рядом со статьей в виде контекстно-зависимой ссылки, либо в виде непосредственной ссылки на статью. Это гарантирует, что навигация предложит согласованную структуру, в то время как индивидуальные статьи могут быть представлены более динамично.

Деловое предприятие

Деловое предприятие (business establishment) — это юридическая, экономическая и финансово независимая хозяйственная единица со своим собственным бизнес-управлением. Целью делового предприятия, однако, всегда является финансовая прибыль. Поэтому большие усилия прилагаются к тому, чтобы завоевать доверие заказчиков. В зависимости от продукта, репутация фирмы должна быть отражена при первом взгляде на веб-сайт, и это впечатление должно быть сразу закреплено в сознании заказчика. Необходимость переноса этой репутации на веб-сайты обеспечивается работой многочисленных агентства сегодня, и, вероятно, положение дел не изменится на протяжении еще многих лет.

Вы можете найти один из подходов к этой задаче в шаблоне Энди Миллера.

Энди Миллер

Энди Миллер — это программист и веб-разработчик с более чем 10-летним опытом профессиональной веб-разработки. Он является членом команды разработчиков Joomla!. а ранее состоял в команде разработчиков Mambo. В разработке Joomla! он отвечает за пользовательский интерфейс, дизайн и свободу от барьеров.

Энди имеет богатый опыт в Joomla! — в частности, в области разработки шаблонов. Все административные и интерфейсные шаблоны, которые поставлялись с Mambo и Joomla!, были (и есть) разработаны им (имя пользователя rhuk: http://dev.joomla.org/component/option,com_jd-wp/Itemid,33/cat,8/).

Энди также является основателем RocketTheme (<http://www.rockettheme.com/>). RocketTheme ежемесячно предлагает новые шаблоны и организует сообщество для сопровождения и непрерывной разработки существующих шаблонов на основе бизнес-модели "подписки". Форумы доступны для чтения всем, а не только подписчикам. На рис. 17.2 показан его шаблон Versality II.



Рис. 17.2. Шаблон Versality II

Шаблон Versality II представляет собой продолжение разработки шаблона Versality и технически весьма совершенен. Так, например, он предлагает пятнадцать позиций модулей, каждая из которых отображается на веб-сайте только в случае доступности соответствующего модуля. Такое деление пространства сайта делает этот шаблон исключительно гибким.

Шаблон имеет четыре опции меню:

- SplitMenu
- SuckerFish
- раскрывающееся меню
- меню SuperSucker

Вдобавок, этот механизм позволяет интегрировать ваши собственные системы меню или системы от независимых поставщиков. Ширина и размер шрифта в шаблоне Versality II могут изменяться динамически посетителями веб-сайта за три шага.

Резюме

В этой заключительной главе были представлены шаблоны-бонусы, которые могут помочь в работе с шаблонами и при создании ваших собственных веб-сайтов.



Онлайновые ресурсы

Joomla! – проект с открытым кодом, который аккумулирует в себе идеи и опыт сотен тысяч людей, объединяет их вместе, развивает его дальше, и предлагается всем желающим в виде продукта, доступного в онлайн-режиме. Вы всегда можете найти в сети самую свежую информацию о нем.

В англоязычном мире самое лучшее место для начала знакомства с продуктом – сайт <http://www.joomla.org/> (в русскоязычном – это <http://www.joomla.ru/>).

Он состоит из трех разделов:

- <http://help.joomla.org>
- <http://forum.joomla.org>
- <http://dev.joomla.org>
- <http://extensions.joomla.org>

Здесь вы действительно найдете все необходимое для успешного использования Joomla!.

На сайте присутствует хорошо сопровождаемый обзор текущего момента – http://help.joomla.org/component/option,com_easyfaq/task,view/id,167/Itemid,268/.

Б

Модули шаблонов

Элемент `<jdoc>` предоставляет возможность вставки динамических разделов в шаблон, не используя ни единой команды PHP (табл. Б.1).

Таблица Б.1. Коды шаблона

Код шаблона	Эффект
<pre><jdoc:comment> Комментарий </jdoc:comment></pre>	Строка комментария
<pre><?php echo '<?xml version="1.0" encoding="utf-8"?.'.'.'; ?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/dtd/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo \$this->language; ?>" lang="<?php echo \$this->language; ?>" dir="<?php echo \$this->direction; ?>" ></pre>	Объявление типа документа (заголовок XHTML)
<pre>HEAD <meta http-equiv="Content-Type" content="text/html; <?php echo _ISO; ?>" /></pre>	Объявление корректного типа контента
<pre><link rel="stylesheet" href="<?php echo \$this->baseurl ?>/templates/system/css/system.css" type="text/css" /> <link rel="stylesheet" href="<?php echo \$this->baseurl ?>/templates/system/css/general.css" type="text/css" /> <link rel="stylesheet" href="<?php echo \$this->baseurl ?>/templates/<DIRECTORY_OF_THE_TEMPLATE_/css/template.css" type="text/css" /></pre>	Установка CSS-файла
<pre><?php if(\$this->direction == 'rtl') : ?>, <link href="<?php echo \$this->baseurl ?>/templates/rhuk_milkyway/css/template_rtl.css" rel="stylesheet" type="text/css" /> <?php endif; ?></pre>	Установка CSS-поддержки языков, читаемых справа налево
<pre>BODY <jdoc: include type="message" /></pre>	Отображение системных сообщений Joomla!
<pre><jdoc:include type="modules" name=" [позиция]" style=" [стиль]" /></pre>	Загрузка модуля [позиция],[стиль]: см. главу 13
<pre><jdoc:include type=component /></pre>	Отображение компонента. Имя компонента выводится из URL

В

Как изменить изображение (логотип) в шаблоне

Перед изменением логотипа вы должны исследовать исходный код соответствующего шаблона. Графика и изображения могут определяться в HTML и в структуре CSS. Размер графики обычно соответствует шаблону.

Если вы хотите заменить одно из изображений, вы можете сделать это разными методами.

Метод 1

1. Создать изображение с таким разрешением и размерами, чтобы оно заполнило нужное место.
2. Загрузить изображение с помощью диспетчера! медиа (Media Manager) Joomla!.
3. Щелкнуть на изображении, чтобы получить ссылку на него.
4. Изменить соответствующий дескриптор `` в исходном коде шаблона на новое изображение.

Метод 2

1. Создать новое изображение и присвоить ему то же имя, что указано в шаблоне.
2. Просто перезаписать старый файл с изображением новым.

Г

Joomla! API

Интерфейс прикладного программирования (Application Program Interface – API) определяет то, какие функции/методы в каких местах исходного кода находятся и что делают. Он состоит из операторов, которые генерируются автоматически из исходного кода – каждый с примером. Команда Joomla! открыла доступ к этой информации на английском языке по адресу http://api.joomla.org/li_Joomla-Framework.html (рис. Г.1).

Joomla! 1.5 API Reference

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 Joomla! Content Management System (CMS) is an application built on top of the Joomla! Framework. This website documents the API of the Joomla! Framework and so does not include code from the Joomla! CMS. In particular, the legacy API is now implemented in a plug-in which is part of the CMS and so is no longer documented here.

This is what you will find here:

- The Joomla! Framework API, called simply Joomla-Framework.
- 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.

Third-party libraries are documented here for completeness and because they contain valuable information not available elsewhere. However, to get the best out of the Joomla! Framework, please try to use the Joomla! API calls in preference to calling the libraries directly.

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.

This website covers the latest official release of the Joomla! Framework. The equivalent to this site for the latest SVN code is available in the svn subdirectory.

Symbol	Description	Symbol	Description	Symbol	Description
	Class		Method		Variable
	Private class		Private method		Private variable

These pages were generated from Joomla! 1.5 Stable (22 January 2008)

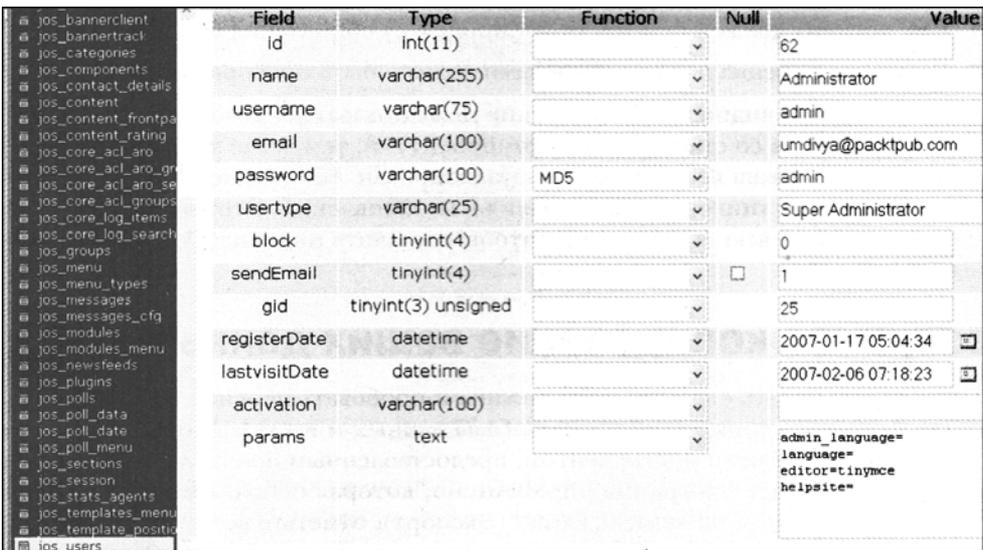
Рис. Г.1. Сайт с описанием Joomla! API

Д

Забыли пароль администратора?

Если вы забыли пароль администратора, то можете изменить его непосредственно в базе данных. Для этого вам понадобится такой инструмент, как phpMyAdmin. В среде XAMPP Lite вы вызываете его в браузере по URL-адресу <http://localhost/phpmyadmin>.

Выберите используемую базу данных. В ней должна присутствовать таблица `jos_users`. Найдите в ней запись о пользователе `admin`. Пароль зашифрован процедурой MD5. Вы можете изменить его, выбрав пункт MD5 в списке опций слева от поля и введя новый пароль в виде простого текста (рис. Д.1).



Field	Type	Function	Null	Value
id	int(11)		62	
name	varchar(255)		Administrator	
username	varchar(75)		admin	
email	varchar(100)		umdiya@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=

Рис. Д.1. Изменение пароля `admin` в `phpMyAdmin`

Средство `phpMyAdmin` для администрирования базы данных предлагают многие провайдеры.

Е

Переход от Joomla! 1.0.x к Joomla! 1.5

Многие из вас, вероятно, уже работали с веб-сайтами Joomla! 1.0.x или Mambo 4.5.x, и теперь требуется совершить переход на Joomla! 1.5.

При этом вы столкнетесь с рядом основных проблем.

- Нужно выполнить резервное копирование всего веб-сайта.
- Сгенерированные файлы исходного кода должны быть обновлены.
- Дополнительные компоненты, модули и подключаемые модули/мамботы должны быть обновлены.
- Любые изменения, которые вы проводили в программах, должны быть обновлены.
- Ваш контент должен перенести эту процедуру обновления без повреждений.
- Контент со специальными символами должен быть преобразован в UTF-8.

Из этого списка видно, что обновление может оказаться довольно сложным. Чем больше вы работали со стандартными компонентами, тем легче этот процесс; но чем в большей степени ваш веб-сайт индивидуализирован, тем труднее будет обновление. Пока не существует универсальных правил для обновлений. Это значит, что сначала вы должны собрать всю информацию, чтобы составить индивидуальный план обновления вашего сайта.

Резервное копирование ваших данных

Прежде чем начать строить любые планы и пробовать перенос компонентов, выполните резервное копирование вашей базы данных и всех файлов. Создайте дамп MySQL соответствующим инструментом, предоставленным провайдером. В большинстве случаев это будет программа phpMyAdmin, которая используется также в среде XAMPP Lite. Щелкните на вкладке Export (Экспорт), отметьте все таблицы и все поля в разделе Structure (Структура). В разделе Data (Данные) выберите Complete inserts (Полные вставки). Вы должны отметить флажок Save as file (Сохранить в файле) в нижнем разделе и также указать нужный формат. Если провайдер допускает это, используйте ZIP-сжатие. Сжатые файлы базы данных могут оказаться на 95% меньше, чем нормальные. Подтвердите выбор, щелкнув на кнопке Go (Выполнить). Страница экспорта базы данных показана на рис. Е.1.

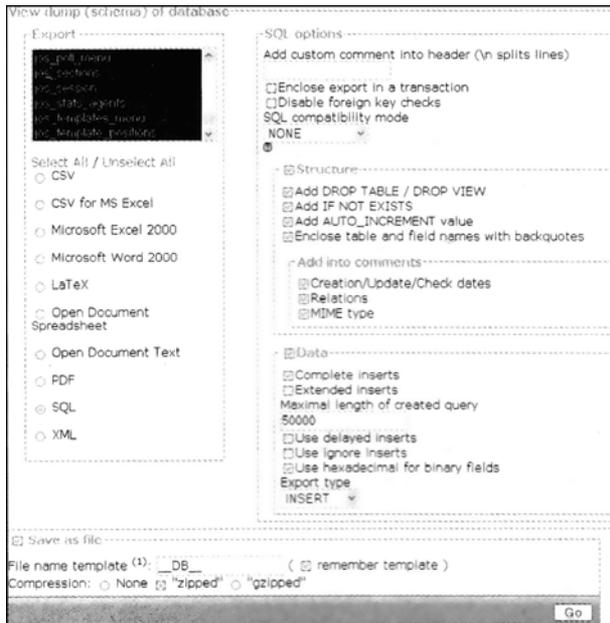


Рис. Е.1. Экспорт базы данных из phpMyAdmin

Содержимое базы данных извлекается и становится доступным для загрузки (рис. Е.2). Файл содержит все команды SQL, необходимые для создания таблиц в другой базе данных с вашим содержимым. Это дополнительный способ резервирования ваших файлов.

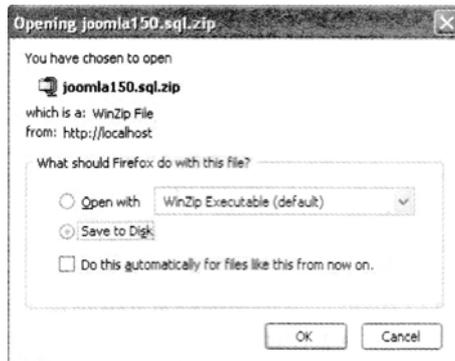


Рис. Е.2. Открытие ZIP-файла дампа базы данных

После этого можно воспользоваться phpMyAdmin для восстановления данных. Щелкните на ссылке SQL, выберите файл с резервированными данными и щелкните на кнопке ОК. Все команды SQL в файле будут выполнены, а база восстановлена.

Резервное копирование файлов

В дополнение к данным из базы также должен резервироваться исходный код. Воспользуйтесь предпочитаемой программой FTP и скопируйте всю папку Joomla!.

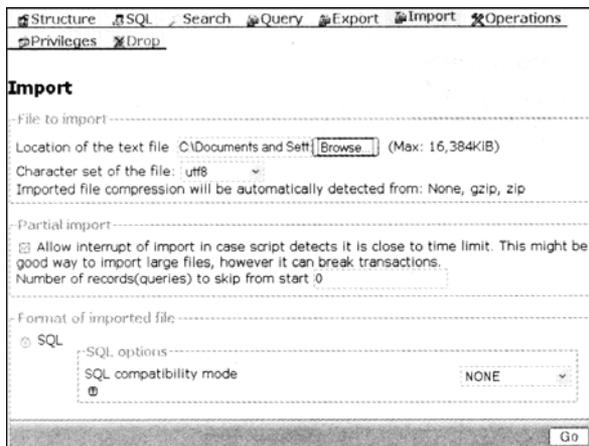


Рис. Е.3. Импорт базы данных из *phpMyAdmin*

Сценарий перехода

Существует компонент перехода, который должен быть установлен в старой версии Joomla! 1.0.x. После щелчка на пиктограмме Dump It (Дамп) этот компонент создаст три ZIP-файла:

- полная резервная копия всего веб-сайта;
- резервная копия центральных компонентов;
- резервная копия компонентов от независимых поставщиков.

Сохраните эти файлы в безопасном месте.

Эти файлы содержат оригинальные, не измененные данные вашего веб-сайта Joomla 1.0.x. Это касается, в частности, используемого набора символов.

Новая инсталляция Joomla! 1.5.0

Настройте новую папку и скопируйте в нее все файлы Joomla! 1.5.0.

Установите Joomla! 1.5.0 веб-инсталлятором вплоть до шага 6 (как было описано в главе 2).

Затем выберите переключатель Load Migration Script (Загрузить сценарий перехода). Введите старый префикс (обычно mos_ или jos_) и выберите старую кодировку. Если вы не уверены, загляните в старые файлы языка и найдите там переменную _ISO.

Теперь загрузите главный экспортный файл компонента миграции и щелкните на кнопке Upload and Execute (Загрузить и выполнить).

Что произойдет дальше, зависит от ваших данных. Обычно вы получите сообщение об успехе – значит, ваши данные трансформированы. Но если вы получите ошибку базы данных, это также не составит большой проблемы. Разберитесь, о чем говорит сообщение об ошибке, и исправьте ошибку вручную либо в *phpMyAdmin*.



Я знаю, что это звучит не слишком вдохновляющее, но, может быть, подготовит вас к тому, что обновление веб-сайта не пройдет гладко с первой попытки. Я избавляю вас от рассказов о проблемах обновления других веб-сайтов, со всеми дополнительными компонентами, надеясь на будущее совершенствование инсталлятора.

Что теперь?

Все центральные компоненты, такие как `com_content`, `com_weblinks`, `com_banner`, `com_contact`, `com_newsfeed`, `com_poll` и `com_users`, обновлены.

Ссылки меню основных компонентов обновлены. Модули установлены как Unpublished (Не опубликовано). Позиции отображения модулей и их параметры должны быть переработаны.

Никаких компонентов от сторонних разработчиков не переносилось, равно как и их файлов (баннеры, изображения, файлы PDF и т.п.).

Ручной метод

Сначала плохие новости. Изменение таблиц не работает правильно в phpMyAdmin, потому что расширенные символы интерпретируются некорректно! Все данные импортированы, но расширенные символы ASCII (такие, как Ä, Ö, Ü, ß, ä, ö, ü, ... и многие другие) отображаются неверно.

Сначала вы должны экспортировать все данные и перевести их в формат UTF-8, а затем внести несколько изменений в структуры таблиц. Экспорт в phpMyAdmin работает без проблем.

Отметьте флажок Complete inserts. Теперь вы должны сохранить экспортированные данные в соответствующем редакторе с использованием формата UTF-8. (Это можно сделать даже в Windows Notepad, если дампы не слишком большой.)



Будьте осторожны!

Убедитесь, что экспортируете только данные, но не структуры таблиц.

Модификация схемы базы данных Joomla!

Есть всего два поля, которые должны быть переименованы.

- В таблице `jos_core_acl_aro` поле `aro_id` переименовывается в `id`.
- В таблице `jos_core_acl_aro_groups` поле `group_id` переименовывается в `id`.

Импорт таблиц

Перечисленные ниже таблицы могут быть импортированы:

- `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

Таблица jos_usertypes в Joomla! 1.5 больше не используется.

Меню и модули могут быть заново сконфигурированы либо с помощью phpMyAdmin, либо через административный интерфейс.



При наличии некоторого опыта работы с phpMyAdmin и MySQL и/или языком SQL, вы сможете обновить основные компоненты без особых проблем.

Перенос компонентов от независимых разработчиков может оказаться более проблематичной. Однако я надеюсь, что вскоре будут доступны надежные сценарии миграции.

И последнее — по порядку, но не по значению — это то, что помимо набора символов в таблицах изменилось не так много. Большая часть работы разработчиков компонентов приходится на исходный код, который должен быть обновлен до новой версии.

Руководство по миграции доступно на сайте Joomla.org. Обязательно прочтите его.

Ж

Безопасность без глобальных переменных

Колес старые версии Mambo и Joomla! требуют стандартной настройки PHP-`register_globals = on`. Этот переключатель обрабатывает видимость глобальных переменных, которые могут, помимо прочего, запускать внешние программы, используя формы или строки `get`.

Эта установка чревата проблемами для безопасности, поэтому лучше использовать `register_globals = off`. Даже если этот переключатель не защитит от всех проблем с безопасностью, он весьма поможет повысить ее уровень.

Эту установку можно было применять и в старых версиях Mambo и Joomla!. Начиная с Joomla! 1.0.11, появлялось явное предупреждение об установке `register_globals = off` в административном интерфейсе с рекомендацией использовать вариант `off`.

Joomla! 1.5 также имеет настройку `register_globals = off` по умолчанию, как это было и в Joomla! 1.0.11. С этой настройкой ядро Joomla! будет работать безопасно и стабильно.

На веб-сайте Joomla.org вы найдете и другие советы по безопасности. Доступен документ *Administrator's Security Checklist* (Список мероприятий безопасности для администратора) для Joomla! 1.0, который также полностью действителен и для Joomla! 1.5.



Когда речь идет о ваших расширениях, вы должны проверить, будут ли они работать этими настройками. На данный момент некоторые расширения еще используют глобальные переменные. Однако, начиная Joomla! 1.0.11, многие заказчики также осведомлены об этой проблеме, и те расширения, которые это затрагивает, скорее всего, уже модифицированы.

Предметный указатель

А

Application Programming Interface (API), 24;
284

С

Cascading Style Sheets (CSS), 192
Content Management System (CMS), 17
Customer Relationship Management System
(CRM), 17

Д

DOCman, 174
Document Management System (DMS), 17

Е

Enterprise Content Management System
(ECMS), 17
Enterprise Resource Planning System (ERP), 17
Extended Markup Language (XML), 193

Ф

File Transfer Protocol (FTP), 91
Fireboard Forum, 170

Н

Human Resource Management System
(HRM), 17
Hyper Text Markup Language (HTML), 18

Ж

Joomla!, 24
API, 284
веб-инсталлятор, 45
инсталляция, 35; 43
на виртуальном сервере в сети, 42
список версий, 26
технические требования, 36

Л

Linux, Apache, MySQL и PHP (LAMP), 19

М

Mambo Open Source (MOS), 20

W

Web Content Management Systems (WCMS), 17

Х

XML Remote Procedure Call (XML-RPC), 28

Б

Баннер, 59; 69; 129
графический, 131
управление баннерами, 130
Безопасность, 291

В

Веб-инсталлятор Joomla!, 45
Веб-ссылки, 141
Видеоролик, 184
Всплывающая подсказка (speech balloons), 7

Г

Галерея изображений, 180

Д

Диалоги, 70
Диспетчер
альбомов, 185
баннеров, 69
веб-ссылок, 141
категорий, 126; 137; 142
контактов, 133
меню, 256
модулей, 144
подключаемых, 157
с элементами фильтрации, 69
пунктов меню, 101
разделов, 123
расширенный Joomla!, 64
статей, 112
титulyных страниц, 127
шаблонов, 162
языков, 64; 164

И

Инсталляция
Joomla!, 35
Joomla! 1.5.0, 288
языка, 64

К

Каскадные таблицы стилей (CSS), 192
Каталог
создание нового каталога, 80
Категории, 137; 142
Класс helper, 246
Клиент, 130

Клиентская часть, 22
Компонент, 23
 пример компонента, 232
Контакты, 133
Контент, 22; 56
 глобальная регистрация контента, 167
Корзина для статей, 123
Кэш, 89

Л

Ленты новостей (news feeds), 60; 137

М

Меню, 55

 Banners, 129
 Components, 129
 Content, 111
 DOCman, 175
 Extensions, 143
 Menus, 95
 Site, 73
 Tools, 165
 Trash, 105
 верхнее, 55
 главное, 55; 68
 горизонтальное, 95
 модификация имени меню, 65
 панель меню, 67
 подменю, 69
 справки, 71

Метаданные, 83

Модуль, 144

 Admin Menu, 156
 Admin Submenu, 157
 Advertisement, 152
 Archive, 150
 Authentication – GMail, 158
 Authentication–Joomla!, 157
 Authentication – LDAP, 158
 Authentication – OpenID, 158
 Banner, 147
 Breadcrumbs, 146
 Content - Code Highlighter (GeSHi), 159
 Content – Email Cloaking, 159
 Content – Image, 159
 Content – Load Module, 160
 Content – Pagebreak, 160
 Content – Page Navigation, 159
 Content – Rating, 159
 Content – SEF, 159
 CSS Admin Menu, 157
 Editor – No Editor, 160
 Editor – TinyMCE 2.0, 160

 Feed Display, 151
 Footer, 147; 156
 Latest News, 153
 Login, 156
 Login Form, 149
 Main Menus, 148
 Menu Stats, 156
 Newsflash, 153
 Online Users, 156
 Polls, 152
 Popular, 154; 156
 Random Image, 152
 Recent added Articles, 156
 Related Items, 15 1
 Search, 154
 Sections, 150
 Statistics, 149
 Syndication, 153
 System – Backlink, 161
 System – Cache, 161
 System – Debug, 161
 System – Legacy, 161
 System – Log, 161
 System – Remember Me, 161
 Title, 157
 Toolbar, 157
 Unread Messages, 156
 User Status, 137
 User–Joomla!, 161
 Who’s Online, 152
 Wrapper, 151
 XML-RPC – blogger API, 162
 XML-RPC – Joomla!, 162
 администратора, 155
 копирование модуля, 155
 подключаемый, 248
 пользователя, 161
 шаблона, 282

Н

Навигация с помощью указаний, 55

О

Опросы, 59; 139
Ошибка (bug), 88

П

Пакет

 DOCman, 174
 Fireboard, 170
 инсталляция, 170
 конфигурирование, 171

Панель инструментов. 68
 меню, 67
 управления, 70
Пароль администратора, 285
Подключаемый модуль (plug-in), 23; 248
Поиск, 60
 поле поиска, 60
 статистика поиска, 140
Права доступа, 22
Программа
 art, 42
Протокол
FTP, 91
LDAP, 158

Р

Рабочий поток (workflow), 24
Раздел
 диспетчер разделов, 123
 редактирование разделов, 123
Рассылка
 массовая, 166
Редактор TinyMCE, 78
Резервное копирование, 286

С

Серверная часть, 22; 61
Система
 FTP, 28
 клиент-сервер, 35
 планирования бизнес-ресурсов (ERP), 17
 управления веб-содержимым (WCMS), 17

управления взаимоотношениями
 с клиентами (CRM), 17
управления документооборотом (DMS), 17
управления персоналом (HRM), 17
управления содержимым (CMS), 17
управления содержимым масштаба
 предприятия (ECMS), 17

Списки, 69
Справка, 70

Т

Титульная страница (Frontpage), 127

Ф

Файл
 auto.xml, 244
 mod_auto.php, 246
 mod_auto.xml, 247
Функции, 59

Ш

Шаблон (template), 23
 Beez, 215
 Versality II, 280
 World Knowledge, 278
 бонусный, 277
 изменение шаблона для веб-сайта, 66
 модули шаблонов, 282

Я

Язык
 HTML., 18; 192
 XML, 193

Создание веб-сайтов с помощью Joomla! 1.5

Популярное учебное пособие по Joomla!, обновленное с учетом последней версии системы

Хаген Граф

Согласно описанию, Joomla! — это “эксклюзивная система управления содержимым и одна из наиболее мощных систем управления контентом с открытым исходным кодом в мире. Она используется для управления содержимым как простых домашних страниц, так и сложнейших корпоративных веб-сайтов. Система проста в инсталляции, управлении и очень надежна”.

Это учебное пособие позволит изучить основные возможности версии Joomla! 1.5. Вы научитесь устанавливать и настраивать Joomla! 1.5, конфигурировать административный интерфейс, создавать собственные шаблоны, а также расширять систему Joomla! новыми компонентами, модулями и подключаемыми модулями.

Об авторе

Хаген Граф родился в июле 1964 г. в Нижней Саксонии, Германия. Первое знакомство с компьютерами у него произошло в конце семидесятых с Radioshack TRS 80. Как специалист по продажам, он организовал свои данные о клиентах, самостоятельно разработав соответствующие приложения. Это обеспечило ему значительные преимущества перед другими продавцами. Намереваясь повысить свою квалификацию, он записался на вечерние курсы программирования и, таким образом, стал программистом. В настоящее время он работает в консалтинговой компании своей жены в качестве инструктора, консультанта и программиста (<http://www.cocoate.com>). Хаген Граф опубликовал в Германии ряд книг, посвященных веб-серверу Apache, проблемам безопасности Windows, продуктам Mambo и Drupal.

НА WEB-САЙТЕ

Исходные коды всех примеров, рассмотренных в книге, можно загрузить с Web-сайта издательства по адресу: <http://www.williamspublishing.com>.

Категория: Программирование для веб/CMS

Предмет рассмотрения: Joomla! 1.5

Уровень: Для начинающих и пользователей средней квалификации



Издательский дом “Вильямс”
www.williamspublishing.com

PACKT
PUBLISHING

www.packtpub.com

