

OpenOffice

Inkscape

версия документа: 0.5.0.5
версия Drupal: 5.15 / 6.9
страниц: 113
последняя редакция: 05.03.09 17:14
автор: Lynx
проект: <http://drupal.bz>
собрано для: slavic Drupal community



DRUPAL

.BZ DRUPAL HAND BOOK

Drupal is a registered trademark of Dries Buytaert.

Оглавление

Немного об этой книге.....	5
Глава 1: CMS * – введение.....	10
1.1 Введение.....	11
1.2 Что такое CMS.....	14
1.3 Классификация CMS.....	16
1.4 Сравнение CMS или «Открытые проекты и заказчик».....	19
1.4.1 Функциональные отличия.....	19
1.4.2 Другие отличия.....	21
1.4.3 Подведем итог.....	26
Глава 2: CMS DRUPAL – основы.....	29
2.1 Кратко о системе в целом.....	30
2.1.1 Почему именно Drupal, или Выбор открытых CMS.....	30
2.1.2 История создания.....	32
2.2 Установка системы.....	34
2.2.1 База для функционирования CMS Drupal.....	34
2.2.2 Установка Drupal.....	38
2.3 Настройка.....	50
2.3.1 Укрепление безопасности на начальном этапе.....	50
2.3.2 Локализация.....	50
2.3.3 Общая информация о проекте.....	52
2.3.4 Установка даты и времени.....	52
2.3.5 Выбор административной и общей темы (оформления).....	52
2.3.6 Настройка производительности.....	52
2.3.7 Загружаемые изображения.....	52
2.3.8 Файловая система.....	52
2.3.9 Форматы ввода.....	52
2.3.10 Чистые ссылки.....	53
2.3.11 Настройка системы оповещения об ошибках.....	53
2.3.12 Закрыто.....	53
2.4 Поддержка.....	54
2.4.1 Обновление системы.....	54
2.5 Пользователи и доступ.....	64
2.5.1 Работа с учетными записями пользователей.....	64
2.5.2 Роли.....	64
2.5.3 Правила доступа.....	64
2.5.4 Глобальные пользовательские установки.....	64
2.6 Системные журналы (отчеты).....	66
2.6.1 Отчет «Последние записи (события)».....	66
2.6.2 Отчет «Ошибки обращения».....	66
2.6.3 Отчет «Отказ в доступе».....	66
2.6.4 Отчет «О состоянии...».....	66
2.6.5 Типичные ошибки.....	67
2.7 Внутренняя справка.....	68
2.8 Создание материала.....	69
2.8.1 Разметка страницы (блоки).....	69
2.8.2 Типы материалов.....	69
2.8.3 Содержание (контент).....	69
2.8.4 Категории.....	69
2.8.5 Меню.....	69
2.8.6 Комментарии.....	69
2.8.7 Настройка лент RSS.....	69
2.8.8 Настройка публикаций.....	69
2.9 Модули.....	70
2.9.1 Модуль Aggregator.....	70
2.9.2 Модуль Blog.....	70
2.9.3 Модуль Blog API.....	70

2.9.4 Модуль Book.....	70
2.9.5 Модуль Color.....	70
2.9.6 Модуль Comment.....	70
2.9.7 Модуль Contact.....	70
2.9.8 Модуль Drupal.....	70
2.9.9 Модуль Forum.....	70
2.9.10 Модуль Help.....	70
2.9.11 Модуль Legacy.....	70
2.9.12 Модуль Locale.....	71
2.9.13 Модуль Menu.....	71
2.9.14 Модуль Path.....	71
2.9.15 Модуль Ping.....	71
2.9.16 Модуль Poll.....	71
2.9.17 Модуль Profile.....	71
2.9.18 Модуль Search.....	71
2.9.19 Модуль Statistics.....	71
2.9.20 Модуль Taxonomy.....	71
2.9.21 Модуль Throttle.....	72
2.9.22 Модуль Tracker.....	72
2.9.23 Модуль Upload.....	72
Глава 3: CMS Drupal – «погружение».....	73
3.1 Интеграция CMS.....	74
Последнее слово.....	77
Авторский раздел.....	79
Полезные источники.....	80
Определения = А.....	82
Приложение А: Выбор хостинга.....	88
А.1 Классификация хостинг-операторов.....	88
А.2 Поиск, отбор, тестирование, приобретение.....	90
А.3 Доменные имена.....	96
Приложение Б: Дистрибутивы.....	99
Б.1 Получение.....	99
Б.2 Установка.....	101



В этой книге: слов 40582

символов 296513

абзацев 1468

[просто статистические возможности
OpenOffice.org Writer и ничего более]

Открытые решения и продукты с каждым годом становятся все более популярными. Многие пользователи уже успели оценить достоинства таких известных продуктов как OpenOffice, Firefox и многих др. Наибольшее же распространение открытые решения получили в среде веб-технологий – Apache, PHP, MySQL и Unix / Linux системы уже давно доминируют в этой области. К сожалению, большинство открытых решений пока не сопровождаются подробной русской документацией. К ним относятся и системы управления содержанием (или как их еще принято называть – CMS от англ. «Content Management System»), ставшие весьма популярными в последние годы.

Системы управления содержанием | контентом (CMS) – это мощный «инструмент», позволяющий значительно упростить процесс создания и поддержки веб-проектов (сайтов, порталов и сервисов). Среди подобных CMS есть и такая инновационная система управления как Drupal. Именно она и станет целью исследования данной книги.



Немного об этой книге

Целевая аудитория, или «для кого это книга»

Следует изначально подчеркнуть, что данная книга не является ни справочником, ни тем более энциклопедией по системе Drupal. Она занимает свою специфическую нишу.

Эта книга ориентированна в первую очередь на тех, кто желает самостоятельно «построить» собственный сайт или даже портал в минимальные сроки без привлечения высокооплачиваемых веб-программистов и дизайнеров на базе готовой системы управления. С этой стороны данная книга напоминает классический учебник (т.е. цельный и связанный объем информации), располагая которым, пользователь больше не нуждался бы в поиске ответов на типичные вопросы. В книге приводятся различные опытно-экспериментальные заключения («выбор хостинга» и д.р. – интересно, что подобные темы практически не освещаются где-либо еще, хотя для начинающих эта информация носит существенное значение).

Также она окажется полезной и для тех, кто уже имеет установленную систему Drupal и желает ознакомиться с ней более детально (т.е. клиентам веб-студий и другим лицам). Однако книга (а точнее ее автор) ориентируется не только на потребителей систем управления, но и на интеграторов – исполнителей (веб-студий и свободных профессионалов, внедряющий систему управления Drupal). Здесь для них приводятся аргументы, которые они могут представить будущему заказчику и графические «наброски»¹, на основе которых можно быстрее и проще подготовить промо-изделия² «Интегратора CMS/CMF Drupal».

Цели и задачи книги

Первой задачей является объективное рассмотрение достоинств и недостатков³ открытых систем в общем и систем управления в частности. Эта задача реализуется в первой главе, где проводится классификация и сравнение систем управления.

В последующих главах реализуется вторая задача – рассмотрение системы Drupal: принципов и инновационных подходов, положенных в ее основу.

Объективность изложения и Ошибки

Многие авторы, создавая материал, очень часто руководствуются собственными коммерческими интересами (либо лоббируют интересы третьих сторон), также немало и идейных, одержимых авторов – быть беспристрастным очень сложно. Однако, несмотря на все это, создавая данный материал, автор старался взять за основу реальные факты, а не субъективные домыслы.

Любой материал содержит ошибки (даже проверенный и исправленный). Эта книга не исключение. С каждым прочтением в ней будут обнаруживаться все новые неточности и недочеты. Если вы нашли нечто подобное, то вы можете сообщить об ошибках, и в последующих реализациях книги они будут устранены.

Благодарности

Высказываю огромную благодарность всем тем, кто оказал посильную поддержку при подготовке данного материала (все они перечисляются в разделе «Последнее слово») – благодаря их критике и советам, результат стал еще более понятным, точным и объективным. Также хотелось бы упомянуть и издательский дом «Питер», представитель которого сразу согласился «работать» с автором данной книги (однако автор все же принял решение опубликовать данный материал в глобальной сети).

1 Наброски предоставляются как вариант для развития идеи, а не в виде законченного решения!

2 Буклеты, брошюры, каталоги, календари, листовки, флаеры, бренд-буки и т.д.

3 Главное не наличие недостатков, а их доля среди достоинств, а также возможность их последующего устранения.

Вознаграждения, поощрения

Эта книга является результатом долгой и непростой работы. Если она понравилась вам, и вы желаете поспособствовать ее дальнейшему развитию, вы можете сделать следующее:

1. сообщить о найденных ошибках;
2. оказать посильный финансовый вклад¹ сообществу данной системы;
3. но самое главное – это то, что вы можете «передать эстафету» следующим участникам: друзьям, коллегам и другим заинтересованным пользователям, рассказав им о системе управления Drupal.

Авторские права и Условия распространения

Некоторые авторы, да и значительная часть интернет-ресурсов прибегают к заимствованию чужих материалов без согласия владельцев этих материалов (т.е. банальной краже). По мнению автора такое заимствование может лишь свидетельствовать о бездарности подобных «создателей». Нигде в тексте этой книги вы не встретите ни предложений, ни тем более целых абзацев из «чужих» произведений (кроме общедоступных спецификаций и исторических справок и цитат).

То же касается и оформления. Автор не стал заимствовать изображения или стиль из других источников, в том числе и таких как **openclipart**, вместо этого он постарался создать собственное оригинальное оформление.

Вы можете бесплатно распространять электронный вариант этой книги в неизменном виде (не смешивая ее текст с каким либо иным, не изымая и из книги и не добавляя в нее предложения, абзацы, главы и т.д.). Вы также можете в частном порядке распечатать книгу и бесплатно распространять ее в неизменном виде. Вы не можете выдавать этот труд за свой собственный.

Любая попытка использования данного материала в коммерческих целях без письменного согласия ее автора (например, издание и последующая продажа, продажа в электронном виде, распространение с взиманием платы за материал или же взиманием оплаты за возможность получения доступа к загрузке материала и т.д.), а также изменение материала и распространение его в модифицированном виде считается незаконной.

Данная книга подписана электронной цифровой подписью [ru.wikipedia.org/wiki/ЭЦП] – открытый ключ в сертификате должен иметь следующее значение:

```
30 82 01 22 30 0D 06 09 2A 86 48 86 F7 0D 01 01 01 05 00 03 82 01 0F 00 30 82 01 0A 02 82 01 01 00 CB CC A0 21 4C F8 9C D3 0B 67 73 A1 84 47 51 4A 4A
70 28 F6 58 A0 FA 7B 62 DF 2D 1B 9B 09 7F A1 2C 43 2D 8C D4 3E 2F 31 70 98 29 09 2B 35 D9 1F 02 21 59 5E 5F 60 DB 11 77 CA F8 B2 38 B1 25 E8 2E C5 15
79 39 C0 60 DD FC 55 4A 38 96 7C 42 4F 46 6E CC 1C DA 58 71 F7 F6 21 15 1E E0 90 65 20 D5 77 F9 71 4C DD 8C 4A 9A E9 58 FD 2D B5 B4 D4 33 06 2E 0A FA
8D AE 1D F3 0F 89 8B 35 87 8E A4 29 B6 04 44 F6 47 23 98 96 2E CC AC 84 4E 35 A0 22 BB 3D 73 1B 4E AA D3 C2 C6 1A 31 00 6D CD 47 23 D0 C1 BB 24 98 84
4A 65 E4 49 90 0C 87 14 6F B6 D7 8A C7 5B 08 BE 3B DA 74 90 41 89 3C B3 B9 69 A1 BA 13 9F 11 07 01 FC 27 D4 4C 97 E2 BA 58 F6 CA 6D 81 22 E9 21 FA A3
61 6D A9 9C 37 B1 E8 96 2F CC 3C 04 B9 DC 62 B6 E1 0F F9 D7 F1 41 CB 0C 5B BE A4 CD 47 F2 89 10 27 4C 5A AD BD B7 F9 02 03 01 00 01
```

Условные обозначения и Стиль оформления

Символы:

- Символ «|» разделяет синонимы.
- Символ «/» представляет собой замену слова «или».
- Содержание простых скобок: «(» и «)», уточняет ранее сказанное.
- Содержание квадратных скобок «[» и «]» служит опциональным продолжением фразы.
- Фигурные скобки «{» и «}» не несут дополнительного смысла.
- Символом «*» обозначается любое возможное значение в данном контексте. Например, при поиске файлов, выражение (маска) «*.jpg» означает «файлы формата jpg с любым допустимым именем».
- Символ «...» обозначает продолжение чего-либо, а также может использоваться как замена фразы «и т.д.».
- Символ «≈» обозначает примерное равенство.

Написание:

- На подчеркнутых словах и фразах следует заострить больше внимания, т.к. это могут быть не только новые термины, но и важные части текста.
- **Полужирным** шрифтом указываются самостоятельные объекты («имена собственные»), которые в данный момент активно участвуют в обсуждении.

Цвет:

- Ссылки, выделенные синим цветом, указывают на внешние ресурсы (расположенные в глобальной сети).

¹ Подробнее на <http://drupal.bz>.

- [Коричневые ссылки](#) указывают на внутренние разделы данной книги. В печатной версии (черно-белой) данной книги эффект от подобного цветоделения полностью исчезнет, однако это не существенно т.к. сделать переход по ссылке в такой версии все равно невозможно.

Знаки:

- Знаком **5** отображается 5 версия CMS/CMF Drupal.
- Знаком **6** отображается 6 версия CMS/CMF Drupal.
- Знаком **A** отмечены термины из словаря определений, [расположенного в конце книги](#).
- Значок **:)** обозначает классический «смайл» в символьном исполнении – к некоторым вещам в жизни стоит относиться с юмором **:)**.

В данной книге пути к каталогам в файловой системе могут описываться псевдонимами. Так, например, каталог `C:\Program Files1` будет иметь псевдоним `%program_files%`. Если в этом каталоге располагается подкаталог, то тогда путь подкаталогу с применением псевдонима будет иметь следующий вид `%program_files%имя подкаталога`.

Изначально такое описание путей может вызвать сложности, однако в последующем вы поймете, что это довольно удобно. Хочется также отметить, что термин «каталог» здесь употребляется как синоним к термину «директория». Хотя с профессиональной точки зрения эти термины не являются синонимами, употребление первого все же более удобно.

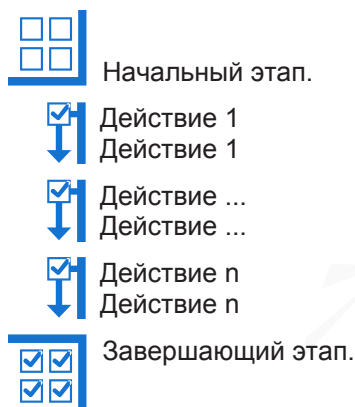
Следует также оговорить и названия некоторых элементов интерфейса. Безусловно, многие знают как называются большинство таких элементов, однако не будет лишним упомянуть эти названия еще раз.

ВИД	ОПИСАНИЕ
<input checked="" type="checkbox"/> <input type="checkbox"/>	<u>Флажок</u> <u>чекбокс</u> , может быть <u>пустым</u> (галочка снята), либо <u>установленным</u> (с галочкой), также имеются и другие состояния
<input type="radio"/> <input type="radio"/>	<u>Переключатель</u>
<input type="text"/>	<u>Поле</u>
<input type="text"/> ▾	<u>Список</u>

Программный код в книге отображается моноширинным шрифтом в специальном блоке. Вот пример оформления такого блока:

```
1 строка кода 1
2 строка кода 2
3 строка кода 3
```

Прохождение [мастера](#)² и последовательные действия отображаются вот таким оригинальным способом:



1 Обратите внимание, что путь записан без завершающего обратного слэша, т.е. «C:\Program Files», а не «C:\Program Files\
2 Последовательные обращения / запросы со стороны программы.

Описание полей форм (и способа их заполнения) может выделяться следующим способом.

- 1 Поле 1: Описание
- 1 Поле 1: Способ заполнения
- 1 Поле ... : ...
- 1 Поле ... : ...
- 1 Поле n: Описание
- 1 Поле n: Способ заполнения

Структура книги и Стиль изложения

Эта книга, как и любая другая, имеет свои неповторимые и уникальные «черты»: оригинальную структуру, авторское оформление, семантику и другие особенности.

Структуру данной книги было решено сделать минимально вложенной, в результате чего в книге количество заглавных уровней сократилось до 4. Такой подход напоминает разработку меню в веб-проектах – меню должно быть спроектировано таким образом, чтобы его вложенность, а точнее вложенность его блоков не была больше 3-4 уровней, а количество пунктов в каждом блоке не было больше 5-10 единиц (т.е. выбрать золотую середину между количеством таких пунктов и степенью их вложенности), только тогда пользователю будет удобно ориентироваться в подобном меню (это хорошее правило из «Информационной архитектуры»). Выделенные 4 уровня имеют следующие области действия:

- Уровень первый – глава – заголовок наивысшего уровня. Глава представляет логически завершенную часть книги (Глава 1, Глава 2 ...).
- Уровень второй и третий – это обычные заголовки с разной степенью детализации содержания (например 1.1 и 1.1.1).
- И четвертый уровень – это тезисный заголовок, включает в себе набор информации на определенную тему (нумерации не применяется).

Теперь о стиле изложения. Сухой научный стиль в виде скучных наставлений и констатаций стал бы здесь не лучшим вариантом, однако и веселое повествование здесь также было бы не к месту, поэтому автор выбрал в данном случае компромиссный вариант – повествование в нестрогом, но и не жестко-официальном стиле.

Последовательность описания в данной книге также имеет свою специфику – она несколько отличается от классических приемов, но линия повествования все же ведет читателя от общего к частному. В начале книги идет обсуждение всего класса систем управления содержанием. В подобный класс входит множество всех систем управления. Затем происходит переход от общего класса всех систем к конкретной системе – системе Drupal.

Теперь о делении книги на части. Как было сказано ранее, каждой части книги соответствует своя глава. Перечислим краткое описание каждой главы:

Глава 1: «CMS *¹ – введение».

Глава первая знакомит читателя с такими понятиями как «Система управления содержанием» и «открытые проекты». Этот раздел начинается долгий рассказ о нелегком деле создания интернет-проектов и посвящается в первую очередь пользователям, сталкивающимся с необходимостью проектированием таких проектов в общем и с системами управления в частности. Также она окажется очень полезна и тем, кто желает убедить свое руководство в необходимости использования открытых систем (именно тут приводятся все аргументы «ЗА» и «ПРОТИВ»).

Глава 2: «CMS Drupal – основы».

Во второй главе происходит переход от обсуждения всех систем управления в общем к конкретной системе управления – системе Drupal. В этой главе рассматриваются различные тонкости, связанные с системой, и обсуждается работа с ней. Все эти части будут интересны и заказчикам проекта, и его пользователям. Здесь также будут приводиться аргументы «ЗА» и «ПРОТИВ», но только уже на примере Drupal.

1 Символ «*» означает множество, т.е. не конкретную систему, а все системы в целом.

Глава 3: «CMS DRUPAL – погружение».

Третья глава посвящается архитектуре веб-проектов на базе CMS Drupal.

Глава 4: «The Last Part».

The Last Part – это не глава – это последняя «часть» книги в которой приводятся различные вспомогательные структуры (например, словарь определений, приложения и д.р).





CMS

*У пользователей проприетарного ПО есть только два преимущества – платить и еще раз платить. Все остальное отвергается в EULA.
[автор не найден]*

1.1 Введение

С момента появления Всемирной паутины прошло всего 15 лет, однако за этот период времени она вызвала революцию в информационных технологиях и кардинально изменила интернет-индустрию. Сейчас интернет и веб стали настолько рядовыми, что многие пользователи уже даже перестали различать такие понятия как «Интернет» и «Всемирная паутина», а при общении эти понятия даже употребляются как синонимы..

Немного из истории: официальным годом рождения Всемирной паутины считается 1989 год, однако свой «настоящий облик» она обрела лишь начиная с 93-94 года, когда окончательно утвердились все базовые протоколы верхнего уровня и появился w3c («Консорциум Всемирной паутины»). Сам же Интернет, как глобальная физическая сеть, появился гораздо раньше – в конце 60-ых, начале 70-ых.

Изначально веб не вызывал всеобщий интерес т.к. являлся лишь собранием научной и технической литературы. Но постепенно к веб-среде начали присоединяться и рядовые пользователи. Они стали создавать собственные проекты и все больше общаться посредством сети. Конечно, в начале 90-х не все могли разглядеть огромный потенциал, заложенный в этой, на первый взгляд, малопривлекательной среде – скептики даже и не предполагали, что рядовые пользователи постепенно преобразят монотонную веб-среду и раскроют ее возможности. Со временем пользователей становилось все больше и больше – их число экспоненциально увеличивалось. Постепенно к пользователям присоединились и коммерческие организации, которые отреагировали на бурный рост интернет-аудитории, а интернет-пользователи стали рассматриваться в качестве потенциальных клиентов – постепенно начала зарождаться электронная коммерция, появлялись первые электронные магазины, которые поначалу представляли собой веб-витрину (т.е. каталоги товаров) и не могли стать полноценными субъектами электронной коммерции – только спустя некоторое время, после появления новых (e-Gold, PayPal) и интеграции существующих электронных платежных систем (Visa, MasterCard), подобные магазины смогли перейти на полноценный электронный расчет. Коммерция также привнесла свой вклад в развитие сети – появились службы, сервисы, реклама и спам. Чуть позже на веб обратили внимание и другие предприятия, не связанные напрямую с электронной коммерцией, – они были заинтересованы в организации собственных «веб-представительств», на которых они могли бы оказывать поддержку своим клиентам. В глобальном масштабе сеть стремительно развивалась, однако в начале нового тысячелетия произошел уже новый рывок в ее развитии – начали появляться концептуально новые веб-проекты, ориентированные на сообщества. Среди них было и много образовательных (в качестве примера можно привести проект «Википедия»). На смену форумам и чатам пришли «социальные сети»... И хотя с начала появления веб-пространства прошло относительно немного времени (по техническим меркам), это не мешало ему в итоге стать глобальным информационным пространством.

Сейчас веб уже прочно «интегрировался» в различные научные отрасли и стал неотъемлемой частью жизни многих людей. Его влияние с каждым годом становится все более ощутимо. Что же будет с Всемирной паутиной дальше? В настоящее время наметились две тенденции в ее развитии – это семантическая паутина и социальная паутина (глобальная сеть становится все больше ориентированной на пользователя), а также совместно с этим ведется параллельная «осознание» концепции Веб 2.0. Необходимо также отметить, что современная глобальная сеть является еще превосходным «аккумулятором» инновационных идей и решений – интернет, как общедоступная глобальная информационная система, «с жадностью впитывает» все новое (к сожалению, «все новое» – это далеко не «все лучшее»), а новые и модернизированные поисковые системы без затруднений справляются с огромным потоком постоянно прибывающей информации.

Веб-проекты

Однако все это сугубо потребительский взгляд на всю паутину в целом. Если провести анализ, то можно обнаружить интересный факт: не смотря на то, что с каждым днем в сети появляется все большее число веб-проектов, это не снижает спрос на разработку новых проектов (а на общем фоне спрос даже продолжает расти). Довольно сложно спрогнозировать к чему это приведет –

переизбыток уже ощутим – многие авторы новых проектов желают занять существующие ниши, а в итоге вместо одного хорошего социального или коммерческого проекта появляются тысячи весьма посредственных. Другое вопрос, когда организация или частное лицо желают создать собственный вспомогательный проект, а не очередную интерпретацию «проекта для всех». В этом случае разработка нового проекта вполне целесообразна – в современных условиях любой серьезной организации просто необходимо иметь свой собственный «сетевой облик» в виде веб-сайта или даже целого портала. Даже общественные и государственные организации и предприятия уже организовали собственные веб-ресурсы, через которые они осуществляют поддержку граждан, своих работников и т.д.

Бесспорно, что во многих случаях создание собственного веб-ресурса оправданно, другой вопрос, какого масштаба проект следует создавать. Здесь необходимо рассмотреть типичные варианты. Вариант первый – это обычная веб-страница. По форме организации информации страница является наименьшей логической информационной единицей сети. Для обычного пользователя такой страницы может оказаться вполне достаточно, однако такой вариант не подходит для более серьезных целей, ведь страница, как логическая единица, сама по себе малоинформативна, это аналогично вырванному листу из книги. Другой вариант, когда страницы объединяются в группу и связываются между собой, превращаясь в веб-сайт. Веб-сайты – это уже более серьезная, базовая форма веб-проекта, подходящая для большинства организаций. Но и сайты – это еще не предел, дальнейшее его развитие может возвести сайт в ранг портала, но для этого потребуются внедрить дополнительные сервисы или службы и немало поработать над его позиционированием. Многие организации уже открыли собственные веб-проекты (веб-сайты или веб-порталы).

Теперь от вопроса рациональности следует перейти к инструментальным средствам разработки, т.е. системам, на базе которых и при помощи которых создаются современные веб-проекты.

В последние годы стали популярны системы управления содержанием (или как их еще принято называть CMS – от англ. «Content Management System»). С помощью подобных систем опытный пользователь может при необходимости легко создать базовый веб-проект (без дополнительных трудозатрат), а веб-интегратор получает возможность развернуть масштабный проект в приемлемые сроки (а модульность подобных систем расширяет круг задач, решаемых на базе таких систем). Раньше подобных систем практически не было и разработка очередного веб-проекта была сопряжена с тяжелым программированием, но современные системы управления снимают необходимость постоянного программирования – сейчас достаточно выбрать готовый модуль, из тысячи ранее созданных и протестированных, и интегрировать его в систему, после чего с минимальными затратами требуемая функциональность будет получена.

Чтобы оценить достоинство систем управления следует немного заглянуть в прошлое и посмотреть, как и с помощью каких инструментальных средств создавались веб-проекты до появления подобных систем, и как они создаются сегодня.

Начало развития веб-среды можно обозначить серединой 90-х г. Этот период характеризовался относительно невысоким уровнем развития веб технологий, а многие динамические веб-проекты (т.е. способные на интерактивное взаимодействие) могли создаваться только программистами. Вся динамика реализовывалась через CGI и другие сложные технологии, поэтому обычной пользователь, создающий собственный проект, мог рассчитывать только на статичный базовый проект масштаба сайта и не более того. Такой сайт обычно представлял собой набор статичных html страниц, подготовленных в WYSIWYG¹ html редакторах, которые начали появляться примерно с середины 1995 года. После набора, страницы объединялись ссылками (для осуществления возможности межстраничных переходов) и размещались на сервере. Вся работа по обновлению информации и проверке работоспособности проекта перекладывалась на его автора. Так, например, если требовалось изменить ссылку на странице, автор должен был найти эту страницу среди остальных, затем внести в нее изменения и снова загрузить страницу на сервер. Если же требовалось не просто изменить существующую страницу, а добавить новую, то приходилось также решать вопросы, связанные с логическим и физическим внедрением последней в весь проект. И, если объем страниц возрастал, то «справляться» с ними становилось еще сложнее. А изменять и расширять динамические проекты было еще более проблематично. Систем автоматизации всех этих процессов на тот момент практически не было, а, точнее, не было доступных и легких систем, не было выбора между открытыми и коммерческими системами, а популярность немногочислен-

1 WYSIWYG (сокращение от What You See Is What You Get, англ. «что видишь, то и получишь») – способ редактирования, при котором редактируемый материал в процессе редактирования выглядит в точности так же, как и конечный результат.

ных коммерческих систем была такова, что об их существовании практически никто не знал¹. Однако такое положение длилось относительно недолго, и в дополнение к статичным html страницам и CGI программированию появились более «дружественные» технологии – Asp (конец 1996г.), ColdFusion (июнь 1995г.), а позже и PHP (2-ая версия вышла в 1997г.). Новые технологии позволили совместить разметку html страниц и несложный программный код, сделав тем самым пассивные html страницы активными. Активность последних позволила легко организовать интерактивное взаимодействие с пользователями, ведь каждый раз при обращении к одной и той же активной странице пользователь мог получать новые данные. В то же время стали создаваться активные страницы для автоматизации определенных действий, например, процесса загрузки файлов на сервер или же процесса создания новых страниц. Такие полезные страницы постепенно собирались во вспомогательные пакеты. Подобные пакеты применялись в типовых задачах, однако их возможностей не всегда хватало, и многие клиенты предпочитали заказывать индивидуальные системы управления под свой собственный проект. Эти заказные системы изначально не были универсальными – когда требовалось изменять либо расширить их функциональность, заказчику приходилось снова обращаться к разработчикам. Вскоре и сами разработчики пришли к решению о необходимости создания универсальных систем. Таким образом и появились первые универсальные коммерческие | проприетарные системы управления. Утвердилась и аббревиатура CMS, которая была заимствована и русскоязычными пользователями. Позже появились и открытые системы управления. Дальнейшая эволюция этих систем привела всю область к такому уровню развития, что нынешнее словосочетание «система управления содержанием» уже не совсем актуально, т.к. функциональность современных систем значительно расширилась, теперь они способны работать с темами, модулями, а также управлять другими элементами. Современные системы уже можно смело именовать «системами управления веб-проектом (класса n)».

Сейчас системы управления это не только удобная оболочка-менеджер для пользователя, но и мощный инструмент для веб-разработчика (последнее справедливо далеко не для всех систем). Благодаря таким системам, все реже возникает необходимость в разработке веб-проектов «с нуля» – подготовленному пользователю или интегратору достаточно просто выбрать, установить и настроить существующую систему, чтобы быстро и без дополнительных финансовых затрат получить приемлемый результат профессионального уровня.

О перспективах

К сожалению, некоторая часть существующих веб-сайтов создана без систем управления. Это так называемые «пустышки». Почему так происходит? Многие клиенты попросту не знают о существовании подобных систем, а разработчики не утруждают себя их внедрением. К чему это приведет? Если у заказчика веб-проекта возникнет необходимость во внесении корректировок или изменений в проект, то ему придется самостоятельно выполнять сложные манипуляции с гипертекстом и программным кодом, либо потребуются обратиться к разработчикам проекта и за дополнительную оплату решать вопросы, с которыми он и сам бы мог справиться, имея систему управления (а во многих случаях разработчики даже заинтересованы в подобной зависимости собственных клиентов). По объективным оценкам на данный момент существует пока немалая часть веб-проектов без систем управления, которые в ближайшее время будут переводиться на подобные системы.

• • •

Следует также отметить, что в конце данной книги имеется раздел «[Определения](#)».

1 Первая система управления появилась в 1995 году – это была система под названием Vignette. [Это не проверенная информация!]


1.2 Что такое CMS

Кратко о термине «CMS»

Теперь немного о термине / аббревиатуре «CMS». Как было сказано ранее, данная аббревиатура происходит от английского Content Management System и переводится как «Система управления содержанием | содержимым | контентом». К подобным системам существует и русскоязычное название – «Программное средство управления интернет-ресурсом», однако ни это название, ни производная от него аббревиатура массово не употребляются, во многих статьях обычно можно встретить только аббревиатуру «CMS» (которая произносится как «СизМэС» или «ЦМС») или фразу «Система управления» (а иногда и просто «Система»). Теперь следует разобраться в таком понятии как «содержание | содержимое | контент», а также в смысле слова «управление».

Содержание | содержимое | контент, что это такое? В контексте систем управления «содержание» – это совокупность всей информации, размещаемой в веб-проекте (на сайте / портале и т.д.). Вместо слова «содержание» также очень часто употребляются слова «содержимое» или «контент» (от английского Content). У многих возникнет вопрос, почему употребляются слово «содержание», а не «содержимое». На него нет однозначного ответа, вполне возможно, что это просто разногласия в переводе. Например, [Google Translate](#) переводит Content как «Содержимое», другие же переводчики как «Содержание». Также можно предположить, что это связано с тем, что благозвучность первого выше.

В любом случае популярность «содержания» выше, вы можете и сами в этом убедиться: зайдите на Google и проведите поиск сначала по выражению «система управления содержанием», а затем по «система управления содержимым». В первом случае будет найдено 2 500 000 упоминаний (страниц), а во втором – 2 040 000. Другие популярные поисковые системы показывают аналогичное соотношение.

Теперь о слове «управление». Как было подмечено ранее, современные CMS расширили значение данного слова в своем контексте. Смысл управления уже не сводится только к информационному наполнению, современные системы способны и на многое другое, например, на работу с [аккаунтами](#) , модулями, темами, языками и т.д.

CMS, взгляд со стороны

Теперь следует рассмотреть, как системы управления представляются различным пользователям. Кто такие пользователи? Фактически к пользователям систем управления можно причислить всех тех, кто хоть как-то с ними взаимодействует. К примеру, пользователями обычно становятся следующие лица:

- будущие и настоящие посетители разрабатываемого / разработанного веб-проекта (гости, зарегистрированные и незарегистрированные пользователи), в корпоративных проектах в эту группу войдут и сотрудники предприятия;
- контент-менеджеры, модераторы, редакторы, администраторы (владельцы);
- программисты и интеграторы (пользователи CMS «как продукта»).

Посетители – это самая многочисленная и самая значимая группа пользователей. Без них любой проект перестал бы развиваться (а возможно и вообще существовать), поэтому за их внимание и ведется столь жесткая борьба. Посетитель, он никогда не вникает в технические подробности – ему абсолютно безразлично, на базе какой системы управления функционирует разрабатываемый проект, ему важен результат: информация и способ ее получения. Систему управления посетитель не видит в таком виде, в каком ее видят доверенные участники вашего проекта, вся система для него – это страницы с материалами, навигационные элементы, формы обратной связи, подсказки и т.д, но не более того. Посетитель – это типичный потребитель, и чем проще и быстрее интересная информация «извлекается» из вашего проекта, тем меньше он задумывается о других, альтернативных проектах. Поэтому недостаточно создать удобный и хорошо настроенный проект, необходимо также позаботиться и о его содержании – оно должно быть интересным и актуальным, так вы сможете привлечь больше посетителей.

Контент-менеджеры, модераторы, редакторы, администраторы – это доверенные лица, которым делегирован набор дополнительных прав. Пользователям такого уровня система управления представляется в расширенном виде – к интерфейсу системы добавляются дополнительные элементы управления или целые административные разделы. В таких разделах администратор может легко и быстро добавлять, например, новых пользователей, а модератор может вносить правки в оставленные посетителями сообщения и делать другие разрешенные администратором действия.



1.3 Классификация CMS

Для лучшего понимания предметной области, а также установления порядка, была создана такая методика как классификация. Классификация – это разделение общего множества на частные по выбранным критериям, т.е. попросту деление на группы по общим признакам. Классификация также требуется и в области систем управления, где она должна сделать четкие разграничения между системами.

На данном этапе будут рассмотрены два наиболее очевидных вида классификации:

1. классификация по «степени открытости»;
2. классификация по «разработчику системы».

По «степени открытости»

Что такое степень открытости? Думаю, многие догадываются, что речь здесь идет об открытом программном обеспечении, ставшем очень популярным в последнее время. Многие уже успели попробовать такие продукты как OpenOffice, Firefox, Linux, а также другие известные программы и системы, которые распространяются не просто бесплатно (*free*), но и с открытыми исходными кодами (в рамках движений **OpenSource**, **GNU**, **FSF** и других¹). Последнее означает, что каждый желающий может вносить изменения в исходный код таких программ и даже распространять их в модифицированном виде². Однако открытые проекты – это не просто программное решение и набор вседозволяющих лицензий – это еще и социальное явление, базирующееся на принципах взаимопомощи, а также движение, объединяющее своих участников по идейным интересам.

Какие уникальные возможности дают открытые проекты различным людям, и какими преимуществами обладают открытые продукты³? Рассмотрим наиболее очевидные преимущества и возможности:

1. Главным преимуществом открытых продуктов является их доступность – открытый продукт можно не только свободно получать, но и быстро обновлять, ведь новые версии таких продуктов выходят с завидной регулярностью, а исправление найденных ошибок не выливается в очередное ожидание заплаток или сервиспаков, как это обычно происходит с коммерческими решениями. Открытые продукты предоставляют свободу выбора и действий, а также дают возможность с минимальными издержками осуществлять миграцию с одного решения на другое, если существующее не удовлетворяет требованиям заказчика.
2. Открытые продукты благодаря заимствованиям стороннего открытого кода, а также взаимоподдержке развиваются порой быстрее своих коммерческих конкурентов. Благодаря свежим идеям и неустанной поддержке со стороны сообщества, открытые продукты способны не только конкурировать с коммерческими решениями, но и занимать доминирующие позиции на рынке. Примечательно, что сейчас уже и коммерческие разработчики присоединились к подобной инициативе – многие из них уже включают в состав своих продуктов открытые решения.
3. Любой желающий, от простого пользователя до программиста, может стать участником открытого проекта, ведь большим интернациональным открытым проектам всегда требуются локализаторы (переводчики), тестировщики и другие специалисты.
4. Программист / организация, выбирая за основу открытые решения, избавляет себя от необходимости подстраивания под чужой корпоративный «черный ящик». Также из открытого продукта можно позаимствовать новые и/или интересные идеи, что не только не запрещается, а даже приветствуется. Открытые проекты и продукты практически всегда ориентируются на стандарты, а не идут вразрез с ними. Это значительно упрощает процесс доработки продукта, а также гарантирует совместимость с другими продуктами, поддерживающими стандарты.
5. Любой специалист в своей области, получая исходные тексты открытых программ, имеет возможность провести анализ продукта и вынести свое заключение относительно его надежности, безопасности, совместимости и т.д.

1 Эти движения всячески поддерживают разработчиков открытого программного обеспечения (в том числе и в юридических вопросах, когда дело доходит до «дележа» интеллектуальной собственности).

2 Если лицензионные соглашения этих программ не накладывают ограничения на подобные действия.

3 Понятие «программный продукт» намного шире, нежели понятие «программное средство».

Анализ кода открытого продукта позволяет удостовериться в отсутствии в нем лазеек. Например, никто не может гарантировать того, что применение не аттестованной или не сертифицированной версии ОС Windows окажется безопасным в сфере гос. управления.

6. Открытый продукт можно «пересобрать» с учетом своих собственных требований, и тем самым, получить максимальный результат.

Возвращаясь к классификации по «степени открытости» все продукты можно разделить на две группы. В первую группу следует отнести все открытые продукты, во вторую – остальные (к остальным относятся коммерческие | проприетарные и бесплатно распространяемые продукты).

По «разработчику системы»

Разработчик системы, как правило, является автором, и сопроводителем системы. В роли разработчика может выступать как открытое сообщество со своими идейными лидерами, так и целая коммерческая организация с серьезным штатом сотрудников. Для удобства отнесем проекты сообществ к первой группе, а проекты организаций, соответственно, ко второй. Для того чтобы понять, чьи продукты все же лучше, следует рассмотреть типичные сценарии появления новых CMS (хотя это актуально и не только для CMS).

Начнем с систем управления первой группы. Как появляются сообщества и как появляются продукты сообщества? Это довольно интересный вопрос.

В мире существует довольно много свободных профессионалов (высококласных программистов, дизайнеров и т.д.), ценящих свой труд и свободу. Они никогда не соглашаются на роль эксплуатируемого звена, они предпочитают самостоятельно находить заказы и работать без посредников (а для больших проектов даже собирать целые команды) – и, как результат, получать за свой труд достойное вознаграждение. В своем большинстве это идейные лидеры, которые не желают мириться с текущим положением вещей¹. Так или иначе, но в своей практике им приходится применять различные «решения» (готовые идеи и продукты). Со временем они понимают, что существующие решения обладают рядом недостатков (ограниченная функциональность, закрытость и т.д.) либо малопригодны в среде новых технологий, и тогда они берут на себя смелость создать нечто новое и более доступное. Так появляется новая идея. После воплощения задуманного, на свет появляется и «решение», пока еще, возможно, и сырое, но уже перспективное. Далее автор принимает решение сделать все наработки общедоступными – так открывается новый проект. Это приводит к тому, что в скором времени, у проекта, появляются и свои последователи – образуется сообщество. Именно благодаря сообществу и его мощной поддержке в дальнейшем, все последующие «решения» будут эволюционировать (идея при этом останется неизменной). Как итог: уже через несколько лет полностью готовое и стабильное «решение» начнет вытеснять коммерческие продукты, а новый проект превратится в достояние сообщества.

Теперь о CMS из второй группы. Ко второй группе были отнесены коммерческие системы. Системы такого класса разрабатываются преимущественно веб-студиями. Следует также заметить, что веб-студии бывают разные – на роль веб-студий могут претендовать как команды из 2-3 человек, так и вполне серьезные организации. Причем в первом случае такая веб-студия может представлять собой самый примитивный «сайто-строительный» полигон² и не более того. Теперь вернемся к коммерческим CMS. Когда они стали появляться? Многие из ныне существующих коммерческих систем, т.е. систем второго потока, «зародились» в периоде между 1999г. и 2003г. – именно тогда начали стремительно появляться все популярные на данный момент веб-студии, которые в первые годы своего существования и стали создавать собственные системы управления. И здесь может возникнуть вопрос: «Почему веб-студии пошли на такой шаг, вместо того, чтобы использовать готовые открытые системы?»³. Самый простой ответ: на тот период (2000г.) еще просто не существовало альтернатив – многие открытые системы либо находились в проекте (**Drupal**), либо в начальной стадии развития (**Plone**), либо вообще были непригодны для массового использования.

Конечно, и сейчас продолжают открываться новые веб студии (спрос рождает предложения), однако станут ли они разрабатывать систему управления «с нуля»? Возможно... Но тем, кто все же решиться на это, стоит попробовать ответить на следующие вопросы:

1 Как правило такие профессионалы обладают более широким мировоззрением, нежели их корпоративные коллеги.

2 Однако со временем такая незначительная организация все же может превратиться в нечто серьезное.

3 Использование открытых решений в коммерческих целях не запрещено и даже не обязывает выплачивать какие либо авторских отчисления.

- Какую пользу (и кому) принесет такое создание? Что выиграет от этого заказчик и какую выгоду преследует ее автор?
- Кто будет поддерживать эту систему в будущем и есть ли вообще у системы будущее?
- Сколько дополнений (модулей / тем / сервисов) и какую базовую функциональность автор успеет реализовать за приемлемый срок?
- Как и кем будет тестироваться система?

Это довольно серьезные вопросы, но как ни странно, многие будущие создатели новых систем управления даже не задумываются об этом.

Хочется отметить, что описанные выше сценарии являются сугубо показательными – в реальной жизни существует немало открытых решений, создаваемых коммерческими организациями.

К системам управления первого потока, можно условно отнести все системы, появившиеся до момента выхода PHP3-их и 4-ых версий (98-00г.). Второй поток – это системы появившиеся после. Большая часть систем второго потока «базируется» на PHP v3, v4 и v5.

Такое разделение сугубо условно и ни в коем случае не претендует на правило.

Стоит заметить, что, несмотря на то, что многие коммерческие системы появились все же раньше открытых, это практически ничего не изменило – даже имея «преимущество на старте» многие из них уже давно «отстали» от своих открытых конкурентов¹ – в то время как открытые системы наращивали свой функционал, коммерческих занимались самооблагораживанием. Хотя подобного нельзя сказать обо всех коммерческих системах, т.к. и среди них существует довольно много качественных решений, однако их цена (**300€-10000€**) заставляет задуматься о целесообразности приобретения таких систем (хотя в некоторых случаях действительно есть смысл задействовать коммерческие, не столь дешевые решения).

Подведем итог классификации

Мы выяснили, что бывают открытые и закрытые проекты, которые могут создаваться как открытыми сообществами, так и коммерческими организациями. Поскольку открытые сообщества не создают закрытых решений, то при данной классификации можно получить лишь 3 класса систем:

1. открытые системы сообществ;
2. открытые системы коммерческих организаций;
3. закрытые системы коммерческих организаций.

Безусловно, первый вариант наиболее привлекателен, но и второй вариант также имеет право на существование, однако для того, чтобы окончательно поставить все точки над «i», автору придется провести небольшое межклассовое сравнение систем, а также показать проблемы, возникающие при выборе того или иного класса.

¹ Здесь не берутся в расчет системы первого потока, перешедшие в разряд корпоративных решений.

1.4 Сравнение CMS или «Открытые проекты и заказчик»

В данном разделе производится сравнение открытых и коммерческих систем управления. Также здесь приводятся описания последствий, которые могут возникнуть при непродуманном выборе системы того или иного класса (в будущем эти последствия могут затронуть и возможности заказчика и, что немаловажно, оказать влияние на успех всего проекта в целом). Если раньше в книге аргументы «ЗА» и «ПРОТИВ» обсуждались сугубо на абстрактных понятиях, то теперь это делается на реальных примерах.

Также интеграторы могут позаимствовать (после переработки) многие тезисы данного раздела (например, при создании ознакомительной брошюры).

1.4.1 Функциональные отличия

Каждая система реализует определенный набор функциональных возможностей (как уникальных, так и типичных) и многие из этих возможностей уже успели стать частью «минимально необходимого» функционального набора (например, модули и темы). Рассмотрим эти возможности более подробно.

Модули



Любая современная система управления содержанием построена по модульному принципу (если вы пользуетесь веб-браузером [Firefox](#), то вам должно быть это знакомо). Каждый модуль вносит свою функциональность. Модули могут добавляться и извлекаться – например, вы можете установить дополнительный модуль работы с покупательской корзиной, и при этой вам не придется менять всю систему в целом, и тем более нет никакой необходимости в написании программного кода – благодаря модулям все делается быстро и просто.

Современная модульная CMS чем-то напоминает детский конструктор – изначально имеется набор кубиков, из которого можно собрать нечто минимально необходимое, если же требуется новый кубик (новая функциональность), то он берется из стандартного набора (дистрибутива) или из других источников (например, с официального сайта разработчика системы).

Что происходит, когда клиенту вдруг понадобилась новая функция (например, он хочет поставить модуль опросов или модуль рассылки писем)? Если клиент использует коммерческую (в данном случае не имеет значения открытую или закрытую) CMS, то вначале он обращается к ее разработчику с запросом нового модуля. И, как правило, ему отвечают следующее: «Зайдите на наш официальный сайт и в разделе модулей поищите готовый. Если такой имеется, то мы готовы выслать вам прейскурант. После оплаты **n€** вы сможете воспользоваться новым модулем. Если готового модуля нет, то мы готовы создать его персонально для вас за **n*m€** и через неделю он появится в разделе модулей, откуда вы и сможете его благополучно получить, предварительно оплатив его активацию». Однако если быть полностью объективным, то необходимо подчеркнуть, что не все коммерческие системы работают подобным образом – жесточайшая конкуренция заставляет идти на уступки и делать модули общедоступными.

Если же вам вдруг понадобится модуль к закрытой коммерческой системе из открытых или бесплатных источников, то скорее всего ваша идея не получит дальнейшего развития – ни один уважающий себя сторонний программист не напишет бесплатно ни строчки кода под коммерческую систему (одним словом, не станет «развивать чужой бизнес»).

В противоположность всему этому открытые системы поддерживаются мощным сообществом. Эти сообщества неустанно генерируют свежие решения. Количество таких решений (модулей, тем и т.д.) на текущий момент без преувеличения превышает цифру с тремя нулями, так что им уже даже в пору создавать поиск по подобным дополнениям, как это сделано, например, на официальном сайте веб-браузера Firefox.

Темы: оформление | дизайн

Большинство систем управления позволяют также работать и с графическими темами, определяющими внешний облик проекта¹. Благодаря такой замечательной возможности, процесс разработки проекта может значительно упроститься и ускориться, ведь для открытых систем в глобальной сети существует уже значительное количество готовых тем. Если клиент (заказчик) согласен использовать одну из таких тем, то все затраты, связанные с дизайном, можно убрать из сметы проекта (а в какую сумму выльется работа хорошего дизайнера и интегратора объяснять думаю не нужно). Если же заказчика не устраивают существующие темы (не подходят цвета, структура макета и т.д.), то и в данном случае можно прибегнуть к компромиссу – среди множества готовых тем можно выбрать такую, которая наиболее соответствует требуемому результату, а затем, с минимальными затратами доработать ее: изменить цвет и / или расположение элементов. Если же требуется создать эксклюзивное оформление, то и здесь открытые системы не оставят веб-разработчика «в беде» – встроенные механизмы тем большинства открытых систем имеют известный задокументированный интерфейс, что дает возможность разработчикам создать собственную тему.

При покупке коммерческой системы вы столкнетесь с определенными трудностями в вопросе выбора тем. Не все коммерческие системы поддерживают механизм работы с темами (что не удивительно). Если же система поддерживает возможность смены тем, то это еще не означает успех в деле будущего «Переоформления» вашего проекта. Найти готовые темы ко многим закрытым системам довольно сложно². Однако, если разработчик системы управления проявит инициативу, тогда у вас все же может появиться неплохая возможность сменить оформление с минимальными затратами, но сделать это бесплатно у вас все равно не получится. В качестве примера можно взять **CMS Bitrix**. Это довольно популярная коммерческая система (и даже открытая, если ее приобретать за четырехзначную сумму). Многие веб-студии предпочитают внедрять ее своим клиентам. Самая дешевая версия системы из линейки Bitrix стоит **200\$**, самая дорогая – ровно **10000\$**. Почему автор заговорил здесь об этой системе. Дело в том что, это система просто «уничтожает» всю монотонность статистики. В то время когда остальные коммерческие системы продолжают и дальше разочаровывать количеством собственных тем, разработчики Bitrix умудрились создать **4900** шаблонов (по данным Bitrix). Каждый шаблон стоит порядка **80** долларов. Если вы желаете заказать эксклюзивный шаблон, то он вам обойдется в **3400\$ ± 200\$** (вы не перепутали, все верно – **3400**, а не **340**). Бесплатных или более доступных шаблонов здесь вообще нет³ (и это притом, что в глобальной сети средняя цена на типичный шаблон колеблется в пределах **15\$** – вы можете немного доработать такой шаблон, после чего он станет уникальным) – в любом случае расходы на создание уникального шаблона в реальной ситуации даже не превысят и **1000\$** (например, если вы заказываете проект «под ключ» в веб-студии). Вывод: разработчики системы Bitrix смогли поставить шаблонную работу на широкий «конвейер», но подобное «количество» никак не повлияло на стоимость⁴ таких шаблонов

1 А «активные» темы могут даже задавать и новое поведение.

2 Но к открытым коммерческим все же проще.

3 Кроме демонстрационного.

4 Цена и стоимость – в контексте этой книги данные понятия употребляются как синонимы.

1.4.2 Другие отличия

Доступность модулей и тем – это еще не все существенные отличия открытых и коммерческих систем, имеются также другие немаловажные аспекты, например, поддержка системы и пользователей.

Поддержка



Понятие поддержки довольно обширное, но в данном случае имеется в виду поддержка заказчиков и пользователей и поддержка системы управления как продукта (т.е. перспектива развития, исправление ошибок и т.д.). Начнем с первого варианта.

Кто такие **заказчики**? Заказчики – это, как правило, клиенты веб-студий (организации или частные лица), а **пользователи** – это те, кто решил самостоятельно «развернуть» систему управления (конечно, заказчики – это тоже пользователи, но для удобства здесь и проводится подобное разграничение). Рассмотрим случаи запроса поддержки заказчиком и пользователем.

Случай первый: заказчик обратился в веб-студию и поручил «создать» персональный веб-проект, т.е. «систему под ключ». В данном случае если клиент сделал именно так, а не обратился к частному мастеру, то с поддержкой у него не возникнет каких-либо проблем – клиент всегда и вовремя получит помощь¹. Причем относительно вопроса поддержки не важно, какую именно систему управления «выберет» исполнитель: открытую или же коммерческую, так как сопровождением системы будет в первую очередь заниматься непосредственно исполнитель.

Случай второй: пользователь самостоятельно внедрил систему управления. Поскольку пользователь не является клиентом веб-студии, не размещал заказ и не оплачивал его исполнение (здесь не берется в расчет цена на коробочные версии коммерческих систем), то, на первый взгляд, можно было бы подумать, что он вообще не должен получить какой-либо поддержки, однако это не совсем так, пользователь всегда может обратиться непосредственно к разработчику системы (любого типа), где ему и будет оказана поддержка, однако эта поддержка будет все же отличаться от описанной выше.

Теперь от заказчиков и пользователей перейдем непосредственно к системам управления и рассмотрим варианты того, как осуществляется поддержка коммерческих систем и как осуществляется поддержка открытых.

Коммерческие системы предлагают живую поддержку на своем официальном сайте и довольно редко где-либо еще. В чем выражается эта поддержка? Как правило, здесь имеется форум с соответствующим разделом, где разработчики системы отвечают на вопросы заказчиков и пользователей. Если на подобном форуме пользователь обратится с вопросом, то он легко получит ответ на свой вопрос (правда не стоит забывать, что обращаться он будет всего лишь к представителю немногочисленной команды разработчиков этой системы). Также на официальном сайте могут располагаться и различного рода документы (справочники, учебники, ответы на часто задаваемые вопросы и д.т.), но автор не берет их в расчет, т.к. это сугубо пассивная и не оперативная поддержка. В общем случае такой поддержкой все и ограничится, при возникновении проблем разработчик системы не станет персонально отвлекаться на каждого клиента, он лишь порекомендует возможные пути решения.

С открытыми системами ситуация похожая, только «свободы» выбора и действий здесь будет значительно больше. Дело в том, что открытые системы очень популярны – для них создаются целые разделы на различных форумах, где имеются ответы на всевозможные вопросы (правда, это все также пассивная помощь). На открытые системы имеется на порядок больше документации, нежели на коммерческие. В плане профессиональной помощи здесь, конечно же, все иначе – в открытых системах несколько другой масштаб – это уже не десяток специалистов, как было ранее, а сотни серьезных программистов и тысячи «уверенных» пользователей. Конечно же, здесь также никто не станет разбираться в системе клиента, однако если на форуме клиент сумеет грамотно описать суть своей проблемы, то помощь ему будет оказана не хуже, чем на «коммерческих» форумах, а даже и лучше (более оперативно и с большим числом вариантов решений), а возможно, возникшую проблему даже станут обсуждать участники сообщества.

¹ В разумных пределах.

В любом случае поддержку всегда можно приобрести, причем как поддержку коммерческих, так и открытых систем (например, это очень популярное решение в *nix системах)².

Из всего этого можно сделать вывод: получается, что обычный пользователь коммерческой системы ничем не отличается от пользователя открытой; вопрос: «Тогда, за что платит клиент?»...

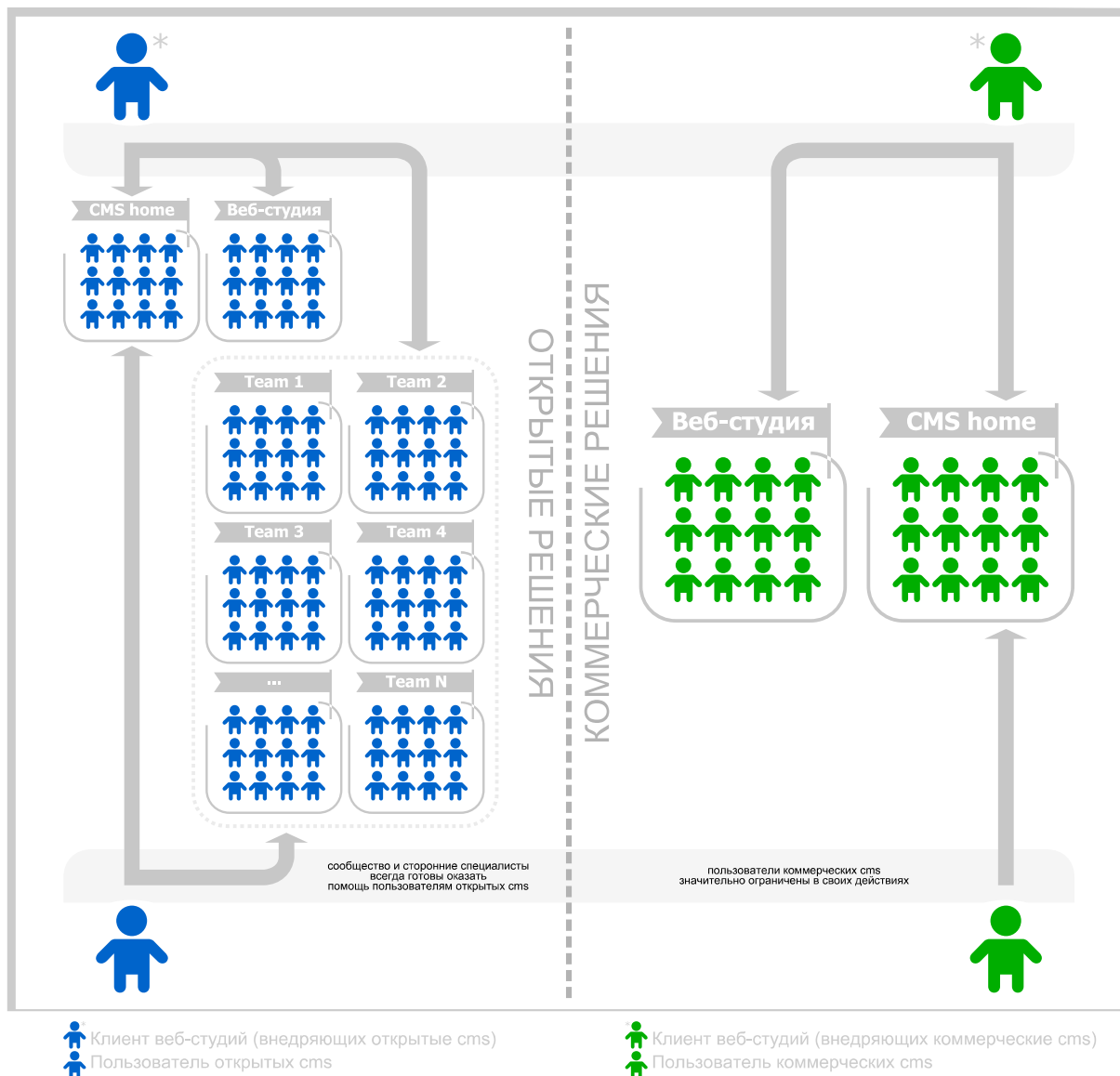


Рис. 1: «Отличия в поддержке коммерческих и открытых проектов» (концепция части буклета).

Теперь о поддержке самого продукта (в нашем случае в качестве продукта естественно выступает система управления содержанием). Здесь под поддержкой подразумевается качество (т.е. поиск и исправление ошибок), перспектива дальнейшего развития и некоторые другие факторы. О качестве и перспективах развития вы можете прочесть далее.

2 В частности и по системе Drupal может осуществляться коммерческая поддержка. За рубежом уже работают компании, специализирующиеся на поддержке такого класса (например acquia.com). В странах СНГ коммерческой поддержкой пользователей Drupal в основном занимаются веб-студии.

Качество, надежность



Многие пользователи полагают, что только коммерческий продукт может быть качественным и надежным. В принципе, с их стороны это было бы логично, но на практике все складывается иначе. Для доказательства этого достаточно двигаться от противного. Возьмем, к примеру, пакет Microsoft Office. Это типичный закрытый коммерческий продукт, и, следовательно, «он не должен иметь изъянов» – так рассуждает пользователь. Однако спустя некоторое время после его выпуска в недрах этого пакета продолжают находить все новые и новые ошибки, дыры в защите (о защищенности сказано ниже). А очередная установка дополнительной заплатки (обновления) не исправляет все огрехи – последнее обновление лишь оттянет время до того момента, когда будет обнаружена новая «порция» ошибок, исправлением которых «займется» очередная заплатка. Так происходит снова и снова. Так что же получается – умудренная опытом, высокооплачиваемая¹ и хорошо организованная команда программистов снова напортичила?...

Что же насчет открытых проектов? Они тоже содержат ошибки и будут их содержать, и так будет продолжаться до тех пор, пока программы будут разрабатываться людьми, а не искусственным интеллектом.

Хотя даже в совершенстве искусственного интеллекта можно будет усомниться, ведь он также является продуктом человеческой деятельности.



Теперь отвлечемся от продуктов и рассмотрим самих разработчиков. Люди – именно они разрабатывают все, что мы так быстро и с интересом потребляем. В первом случае, когда речь идет о свободном программном обеспечении, продукт разрабатывает профессионал-энтузиаст, либо профессионалы-энтузиасты. Он / они вкладывает в свое детище все силы. На практике ему / им, конечно же, помогает сообщество (и возможно спонсоры).

В тоже время коммерческие решения разрабатываются группой профессионалов. В их работе все поставлено очень грамотно – технические задания, менеджмент, отделы тестирования и т.д., однако здесь исполнителями все же движет не энтузиазм, а скорее возможность заработать и, в меньшей степени, возможность самореализоваться (в данном случае для командного программиста разрабатываемый продукт – это всего лишь очередной «безликий» заказ). В итоге получается практически равный результат – все имеют право на ошибку и все могут ее совершить.

В заключении этого раздела хотелось бы отметить следующий факт: 70% (а к этому моменту возможно уже и все 80%) серверов в глобальной сети функционирует на открытых операционных системах (Linux, BSD и д.р.), и этот показатель продолжает увеличиваться.



Защищенность

«Защищенность продукта» – это довольно интересная тема. Защищенность не зависит напрямую от качества и надежности. Даже самый отработанный и отлаженный продукт может при определенных обстоятельствах превратиться в самое уязвимое решение. Возьмем, к примеру, веб-сервер Apache. Уже давно многие признали его качество (точнее, качество его работы) и его надежность. При «правильном обращении» этот веб-сервер является действительно самым лучшим решением, но если этим же веб-сервером попытается воспользоваться неподготовленный (обычный) пользователь, то безусловно, надежность такого решения и всей системы в целом окажется под большим вопросом.

В реальности и открытые и закрытые продукты могут на равных использоваться для создания защищенных систем, а многие открытые системы даже ни в чем не уступают своим коммерческим собратьям.

Удобство

Можно сказать много хорошего об коммерческих продуктах, ориентированных, в первую очередь на пользователя. В этом аспекте значительная их часть, действительно, дает больше «комфорта» нежели открытые решения. Автор не станет лукавить и признается, что и сам использует коммер-

¹ Достоверно известно, что средняя заработная плата программиста в США составляет от 5000\$ в месяц. В крупных корпорациях она может достигать до 10000-15000\$.

ческие системы, например, ОС Windows XP¹ – самой удобной на данный момент (2003-2008гг.) операционной системой (по мнению автора).

Поскольку в данной книге обсуждаются главным образом не популярные пользовательские системы наподобие MS Office, OpenOffice и т.д., а серверные, в которых фактор эргономики играет не самую главную роль (а иногда даже и не берется в расчет), то аргументы этого раздела за коммерческие продукты теряют свою актуальность. К тому же сравнение систем управления – это не сравнение обычных программ, существует довольно значительно количество удобных как открытых систем, так и коммерческих закрытых (и этого нельзя отрицать).

Перспектива дальнейшего развития, окупаемость



Как известно все коммерческое существует лишь до тех пор, пока оно приносит доход, или, по-другому выражаясь, ровно до того момента, пока проект остается самоокупаемым. Как только окупаемость проекта падает, интерес к нему со стороны разработчиков и инвесторов постепенно угасает, и они переключаются на другое более прибыльное дело – типичный капитализм в действии. В свое время из-за такого подхода очень много интересных проектов «закрылось». Сейчас многие коммерческие системы управления находятся в шатком положении, можно даже сказать, балансируют на грани «закрытия». Правда всегда есть шанс спасти такой проект (если, конечно, он представляет какой-либо интерес для сообщества разработчиков) – его можно сделать открытым. Многие разработчики так в свое время и поступили и, благодаря их дальновидности, угасающие в прошлом решения, сегодня обрели второе рождение.

Открытые проекты в своем большинстве защищены от такой угрозы. Конечно, бывают моменты, когда автора открытого проекта одолевает алчность, и он продает его или начинает брать за свое решение мзду, в результате такой проект начинает постепенно угасать – интерес к нему со стороны других разработчиков, да и пользователей постепенно утрачивается. Радует однако то, что встретить такое доводится очень редко. Еще интересный факт: как только возникает подобная ситуация, на свет тут же появляется форк, т.е. альтернативный открытый проект (клон).

Миграция (переход на другое решение)

Под миграцией понимает смена решения, т.е. переход с продукта А на продукт Б. Это довольно интересная тема. Дело в том, что в миграции существует два понятия, «цена перехода» и «цена решения». Теперь немного об этих понятиях.

Цена перехода. Сам переход несет в себе определенные человеческие трудозатраты, поэтому переход с одного решения на другое (открытое или закрытое решения – не важно) предполагает оплату подобной работы, если, конечно, пользователь не возьмется за нее самостоятельно. Поэтому здесь мы не станем обсуждать цену перехода, а обратимся к вопросу «цена решения и миграционные последствия».

Цена решения. Многие заказчики вряд ли особо задумываются над этим вопросом, а многие даже и не предполагают, что в будущем им придется столкнуться с миграцией. Рано или поздно возникает ситуация, когда вас как клиента / заказчика / пользователя перестанет устраивать существующее решение (и это будет абсолютно нормально), и вы пожелаете совершить переход на другое решение.

Если вы используете коммерческую систему и желаете перейти также на коммерческую, то вам придется забыть о тех финансовых вливаниях в вашу прошлую систему, которые вы уже совершили. Далее вам придется совершить новый «транш» очередному автору-разработчику за его «детские», от которого, возможно, в будущем вы снова откажетесь. Многие клиенты полагают, что их затраты будут разовыми, но действительность оказывается суровее. Ситуацию перехода даже можно описать на простом примере: клиент приобрел ранее офисный пакет MS Office 2000 (скажем, за **250-300\$**), прошло время и теперь клиент хочет приобрести MS Office 2007 – нет проблем, ему всего лишь необходимо заплатить за него очередную сумму (примерно **300-500\$**). Аналогичная ситуация возникнет и с будущим переходом с пока еще нового MS Office 2007 на последующую версию этого продукта.

Теперь об открытых решениях. Здесь все очень просто, при выборе нового открытого решения клиент не теряет ни копейки (подчеркну еще раз, что цена перехода здесь не обсуждается). На бытовом уровне это можно описать так: к примеру, вы желаете заменить **KOffice** новым **OpenOffice**,

¹ Хотя и пытается отказаться от нее в пользу ОС Linux **Ubuntu**.

вы просто скачиваете последний дистрибутив этого продукта и используете его – это все. Похожая ситуация и с корпоративными решениями – вместо **MySQL Community Server** вы можете использовать **PostgreSQL**.

Теперь о смешанном переходе. Смешанный переход, «с коммерческого решения на открытое» или «с открытого на коммерческое», как вы уже и сами, наверное, догадываетесь, предполагает лишь частичную потерю ваших средств.

Существуют бесплатные (но очень ограниченные) версии таких коммерческих продуктов как СУРБД **Oracle** и **MSSQL Server** (например, MSSQL 2005 *Express*). Эти потребительские версии появились сравнительно недавно и их наличие, в общем, никак не противоречит всему вышесказанному, ведь переход осуществляется при нехватке функциональных возможностей, а вышеупомянутые версии продуктов обладают лишь минимальными возможностями. Выпускаются такие продукты с определенным подтекстом (по мнению автора) – предполагается, что в будущем вы, как пользователь, такого решения откажетесь от бесплатной базовой версии и мигрируете на более продвинутую – платную.

Лицензирование и ответственность

Никто не желает брать на себя ответственность (мнение автора)...

Многие пользователи не читают лицензионные соглашения – бытует даже шутка о том, что по статистике скорости прочтения лицензионного соглашения русские пользователи самые быстро читающие в мире, ведь мало кому еще удастся прочитать стандартное 20 страничное лицензионное соглашение за 2 секунды :).

Начнем с открытых решений. На такие решения, как правило, применяются лицензии **Gpl** и им подобные. Эти лицензии дают неограниченные возможности: в них говорится, что любой желающий имеет право копировать, либо изменять это программное обеспеченное и даже распространять его под собственным названием (если в неизменном виде, то со ссылкой на первоисточник) и т.д. – в общем, фактически с продуктом можно делать все что угодно. И здесь же для всех открыто будет сказано следующее: «Автор не несет какой либо ответственности за причиненный ущерб, порчу и т.д. и т.п.». Т.е. никакого обмана. И к этому небольшой факт для размышлений: несмотря на полное прозрачное отсутствие ответственности Gpl-подобных лицензий, 70% серверов глобальной сети функционируют на Gpl-лицензированных продуктах – Unix, Apache, PHP, MySQL.

Теперь немного о коммерческих решениях. Любое такое решение сопровождается собственной проприетарной лицензией. Соглашаясь с ней юридически (и, как следствие, и физически), пользователь обязуется установить продукт не более чем на один компьютер и использовать только по прямому назначению. И все, ничего более он не имеет право делать – т.е. фактически пользователя ставят перед фактом «собственного бесправия». При всем при этом разработчик также снимает с себя всю ответственность. Если же эксплуатация программы повлекла за собой потерю ценной информации, то попытки обращения в суд с целью выигрыша дела против автора / производителя данной программы ни к чему не приведут – в проприетарной лицензии присутствует аналогичная строка «Автор не несет каких либо обязательств и д.т. и т.п.», только написана она, как правило, мелким шрифтом на иностранном языке и расположена в конце соглашения.

Хочу сразу разубедить пользователей, которые думает, что «дистрибутив любой программы стоит ≈2\$» и «радио рынок – это официальный центр их продаж». Это не так! Автор полагает, что многие пользуются пиратским программным обеспечением (ПО). Конечно, пиратство не тема данной книги, но ради интереса вы даже можете и сами подсчитать, какую сумму вы не доплатили разработчикам ПО. ОС Windows Vista + ≈250-500\$, MS Office 2007 + ≈300\$, Photoshop CS3 + ≈700\$ (а он есть у многих – правда, автор не понимает, зачем для простой коррекции фотографий использовать пакет профессиональной уровня), Corel Graphics Suite + ≈250\$, Nero, Flash, Dreamweaver, игры (каждая 20-50\$) – по самым скромным подсчетам, стоимость вашего компьютера окажется в два раза ниже стоимости всего ПО, установленного на нем.

P.S. Все цены периода 2007 года.

Ориентация на потребителя

Открытые проекты развивается сугубо в интересах потребителей таких систем. Возможно, это и сказано громко, но этим словам есть подтверждение. Конечно, авторы таких проектов ведут свою идеологическую линию (они понимают, какой должна стать система, как сделать ее более удобной, как направить помощь извне в правильное русло и т.д.) и с этим никто не станет спорить, од-

нако они не занимают авторитарную позицию. Авторы часто советуются с пользователями в разных аспектах, организуют обсуждения на форумах, и если идея не находит всеобщего одобрения, то она просто не принимается.

В коммерческих же проектах решение принимается где-то внутри. Причем обоснованность многих таких решений можно поставить под сомнение. Очень часто коммерческие системы усложняют то, что, по определению, должно быть простым. С пользователями, как правило, никто не советуется и порой кажется, что их мнение вообще не учитывается. Новые решения внедряются по принципу, «а почему бы не реализовать эту функцию, которую потом можно будет позиционировать как новую уникальную возможность».

1.4.3 Подведем итог

Безусловно, коммерческие продукты и системы создаются, в первую очередь, с целью получения дохода, и с этим вряд ли кто-то станет спорить. К сожалению, и, как следствие, такие продукты не предоставляют каких-либо уступок ни пользователям, ни сторонним разработчикам. Скрытие кода, лицензионные ограничения, отсутствие многих «интересных» и конкурентных возможностей – все это основные черты коммерческих систем управления. В противоположность этому можно поставить открытые системы управления, разрабатываемые открытыми сообществами в условиях полного или частичного отсутствия инвестиционных вложений¹, которые способны составить конкуренцию проприетарным решениям.

Еще один интересный факт: в последнее время наблюдается тенденция, когда закрытые коммерческие проекты становятся открытыми. Делается это исключительно для «возврата» себе некогда былой славы могущественных «брендов». Например, Sun уже открыла часть своей Java платформы.

Итак, автор надеется, что приведенные доводы позволят вам более реально смотреть на данный вопрос, ведь лучше иметь свое объективное мнение, нежели делать «слепой» выбор, основываясь на чужих советах.

• • •

И, напоследок, автор хотел бы привести вам реальный пример. В глобальной сети уже появилось довольно много статей-сборок на тему систем управления. Одной из такихборок стала сборка cmsarticles.narod.ru. Обнаружил ее автор не случайно – ссылка на нее находилась на [Википедии](#) в конце описания термина «CMS». На момент обращения к сборке в ней имелось 18 статей. Сейчас хотелось бы обсудить только статью под номером 3 – «Чем платная CMS лучше бесплатной?»². Это довольно интересная статья, правда, интересна она не своей объективностью (которой там практически нет), а ссылкой на первоисточник. Ссылка ведет на официальный сайт **CMS System** (реальное имя опускается и, при желании, вы можете и сами узнать о какой системе управления здесь идет речь). Почему именно эта система? На самом деле вопрос, конечно же, не конкретно в этой CMS, при желании на ее место можно было бы поставить любую другую коммерческую систему управления. Теперь об этой системе в целом, не особо углубляясь в ее суть, а начнем с ее разработчика. Разработчиком данной системы является веб-студия (данная система расположена на отдельном от официального сайта разработчика веб-проекте).

Первым делом автор посетил официальный сайт разработчика этой системы³. Сайт выполнен довольно неплохо, однако внизу автор обнаружил на тот момент ошибку (изображена на рисунке ниже), далее о ней будет еще одно упоминание. Удивила цена, цитата: «Стоимость разработки корпоративного сайта в нашей студии – от 5000USD», и это при тех факторах, которые будут обсуждаться далее.

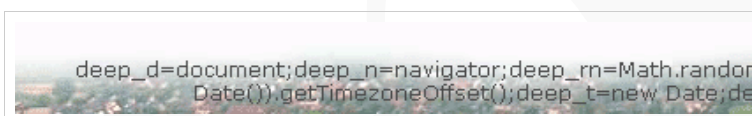


Рис. 2: Вырезка из официального сайта разработчика – что-то не досмотрели 1.

- 1 Не берется в расчет мощная спонсорская поддержка известных открытых решений
- 2 После публикации книги, данная сборка и ссылка на нее странным образом исчезли, однако данную статью можно легко найти в поисковой системе по ее названию.
- 3 Что примечательного? Название веб-студии, если перевести его с белорусского языка на русский, то получится слово «Потеря», а если с украинского то «Казнь» (в качестве украинского использовался переводчик translate.meta.ua). Название же системы управления – «Сумерки» (дословный перевод с английского).

Далее автор перешел на официальный сайт системы управления. Первое впечатление было таким – очень привлекательный дизайн (пример-вырезка на рисунке ниже), даже подсознательно возникло желание приобрести эту коробочную систему только ради того, чтобы посмотреть, что же там внутри.

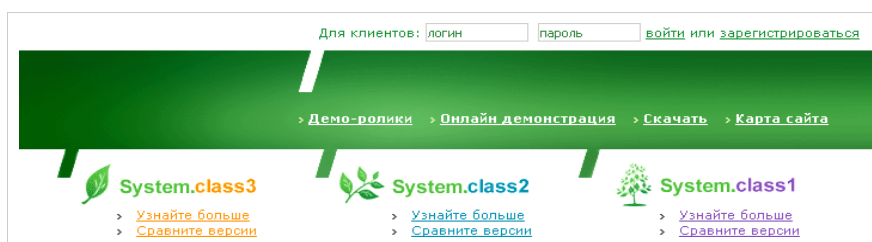


Рис. 3: Вырезка из официального сайта CMS проекта.

Далее шло описание этой системы, по которому о ней можно было судить как о лучшей системе в России. В конце страницы появилась такая же ошибка, что и на сайте разработчика (изображена на рисунке ниже), которая испортила не только компоновку страницы, но и первое впечатление. Интересно, что эти ошибки не были исправлены даже в течение дня.

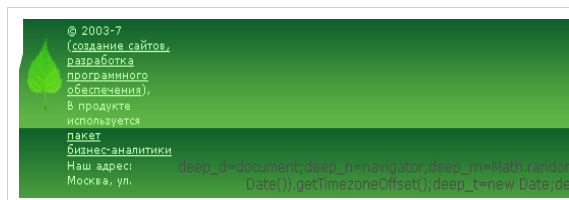


Рис. 4: Вырезка из официального сайта CMS проекта – что-то не досмотрели 2.

Далее автор попробовал скачать базовую версию продукта, которая, как оказалось, предоставляется бесплатно. Для ее получения необходимо было заполнить форму, после чего нажать кнопку «Отправить». К сожалению, эти действия ни к чему не привели – в качестве результат автор получил сообщение об ошибке.

1

Откуда Вы узнали о System CMS?

Я хочу получать информацию о нововведениях в System CMS

```
deep_d=document;deep_n=navigator;deep_m=Math.random();deep_n_n=(deep_n.appName.substring(0,3)=='Date()').getTimezoneOffset();deep_t=new Date;deep_t_m=deep_t.getTime();deep_t.setTime(deep_t_m+315360000);deep_uid=""{};deep_uid=deep_t_m+"|"+Math.round(Math.random()*1000000);}deep_d.cookie="stra="+deep
```

2 Form data must be send from this site only.

Рис. 5: Форма для получения дистрибутива – что-то не досмотрели 3.

Безусловно, у всех случаются неприятности, и автор решил не останавливаться на достигнутом – в надежде узнать о системе подробнее, в адрес разработчиков было отправлено письмо следующего содержания:

вопрос:

Добрый день.

Меня заинтересовала ваша CMS, однако у меня есть несколько вопросов:

- 1) Насколько я понял для работы в административном режиме в вашей системе требуется Internet Explorer. Возможно ли использование других браузеров?
- 2) Поддерживается ли у вас модульная технология? Сколько свободный модулей у вас имеется (например, имеется ли модуль для работы с корзиной)?

ответ:

Добрый день.

1. Нет, с другими браузерами система не работает. "System CMS" – это приложение, которое использует IE в качестве оболочки, а не чистая web-based система. Это дает массу возможностей, которых нет у web-based систем. Также, IE стоит абсолютно на любой Windows машине, поэтому никаких проблем это не вызывает.

2. Понятие "модуль" в нашей системе отсутствует, используется близкое понятие "сервис". Перечень возможностей вы можете посмотреть на странице ...

Ответ пришел не так быстро. Какой из него можно было сделать вывод? Если с отсутствием кросс-браузерности еще можно было смириться, то с отсутствием модулей уж точно нет (причем на сайте разработчика автор не нашел ни одного модуля). Создавалось впечатление, что за 5000\$ данная веб-студия для каждого заказчика создает систему управления «с нуля». Также оказалось, что базовая версия лишь условно-бесплатна – коммерческое использование такой версии, предоставляющей лишь минимальные возможности, обойдется клиентам в **125€** (следующие версии линейки данной CMS стоили соответственно 400€ и 1000€).

Такой ответ породил желание создать еще один запрос, и в адрес разработчика было отправлено уже второе письмо со следующим содержанием:

вопрос:

Добрый день.

У меня еще есть к вам вопрос. У вас есть партнерская программа и это замечательно. Также вы предлагаете свой продукт в виде коробочной версии. Насколько я понял базовая версия вашего решения стоит 125\$, однако хотелось бы узнать какую поддержку получает легальный потребитель вашей системы (т.е. уплативший n-ую сумму, а не использующий бесплатную лицензию).

Ответ на это письмо пришел лишь через двое суток со второй попытки, вот его содержание:

ответ:

Базовая версия в РФ стоит 3750 р. Коммерческий пользователь получает гарантированную техническую поддержку (бесплатные версии обеспечиваются только поддержкой через форум). Поддержка осуществляется через email или форум. Служба поддержки помогает разобраться в вопросах, которые не освещены, либо недостаточно освещены в документации.

Стоимость базовой версии в ответе оказалась даже выше, чем на официальном сайте разработчика. Из письма в принципе стало понятно, что же все-таки получают «платные» пользователи, однако фраза «служба поддержки помогает разобраться в вопросах, которые не освещены, либо недостаточно освещены в документации» означает, что вся поддержка ограничиться советом.

Внимание: представленная здесь информация о коммерческой CMS была актуальна только на момент создания данного раздела, также эта информация является исключительно мнением автора (его опытом) и в данный момент может не иметь ничего общего с действительностью !!!



DRUPAL

2.1 Кратко о системе в целом

CMS/CMF Drupal (на русском произносится как «Друпал») – это открытая инновационная и перспективная CMS¹ (система управления контентом). Однако Drupal – это не только система управления контентом, но также и мощный прогрессивный инструмент, завоевавший уважение многих веб-разработчиков и здесь Drupal выступает уже в качестве CMF, т.е. в качестве «инструмента для построения других систем».

В контексте CMS, Drupal представляется готовым «ориентированным» продуктом. Базовый пакет CMS Drupal идеально подходит для организации социальных проектов, таких как блоги, wiki-проекты и др., однако, благодаря своей каркасности, а также огромному числу уже готовых и доступных модулей и тем², данную систему можно настроить практически под любую задачу (в том числе и электронную коммерцию). Конечно, CMS Drupal – это не идеально-универсальная система (таких, впрочем, и не существует, кто бы там этого не утверждал) и для некоторых задач разумнее использовать специализированные системы (например, для организации крупного форума специализированная система phpBB³ будет более уместна, нежели универсальная⁴). Поскольку уже существуют средства интеграции разнородных систем (например, форумного движка и универсальной CMS), разработчикам приходится все реже делать бескомпромиссный выбор между системами различного назначения.

О возможностях системы. У CMS Drupal действительно много базовых возможностей, а с добавлением внешних модулей их становится еще больше. В дистрибуцию системы входит набор модулей, дающих следующие возможности:

- сбор информационных лент (RSS, RDF, Atom);
- ведение блогов, подшивок и форумов;
- комментирование контента (содержания);
- создание форм для отправки сообщений;
- локализация системы;
- переименование ссылок (назначение ссылкам понятных и удобных псевдонимов);
- проведение опросов;
- настраиваемые пользовательские профили;
- поиск по содержанию (содержанием считается и сообщения на форумах, и страницы подшивок, и любые другие назначенные элементы);
- ведение журнала статистики (посещаемости);
- таксономия (упорядочивание материала по категориям) – очень «ценная» возможность;
- включение механизма авторегулятора контроля нагрузки и другие.

Некоторые возможности этой системы довольно необычны в сравнении с другими CMS, например, таксономия, поэтому их описание будет даваться в соответствующих разделах.

Drupal, как CMS, предоставляет возможность в относительно короткие сроки развернуть собственный базовый⁵ веб-проект, без необходимости привлечения «дорогих» веб-программистов, дизайнеров и других специалистов. В тоже время, Drupal как CMF (как каркасная система), дает возможность построения других, уже более сложных систем. И в данной роли, система Drupal выступает в качестве мощного инструмента для программистов и веб-интеграторов. Благодаря открытости, функциональности, простоте, а также принадлежностью как к классу CMS, так и к классу CMF эта система и получила столь широкое распространение.

2.1.1 Почему именно Drupal, или Выбор открытых CMS

В мире очень много систем управления (более 500), и их число продолжает расти. Такое их количество привело к появлению различных cms-каталогизаторов и cms-рейтингов (top100). Подобные рейтинги и каталогизаторы могут оказаться полезными при поиске новой системы. В целом, рейтинги – это довольно неплохая вещь. Они позволяют составить общую картину рынка всех систем управления. Однако доверять рейтингам не стоит, особенно отечественным, т.к. их объектив-

1 И в данной книге эта система рассматривается в первую очередь в качестве именно такой системы.

2 Не включенных в базовый дистрибутив, но всегда доступных на официальном сайте системы.

3 Популярность данной системы в последнее время значительно снизилась.


4 Третья версия этого продукта заметно преобразилась.


5 Надо объективно оценивать ситуацию – без участия подобных специалистов обычный пользователь сможет создать только лишь базовый проект и не более того – это справедливо и для любой другой системы управления. Для внедрения серьезных проектов потребуются консультации и/или помощь соответствующих специалистов.

ность порой вызывает сомнение. Каталогизаторы (или просто каталоги) также весьма полезны. Они позволяют вести выборочный поиск систем и сравнивать их по определенным критериям. Однако каталогизаторы, также как и рейтинги – дело не надежное (порой они могут предоставлять абсолютно недостоверную информацию), поэтому обращаться с ними, как и с рейтингами, нужно осторожно. Все системы уникальны, и попытка ограничить их возможности рамками каталогов – дело бесполезное и порой даже вредное.

Среди множества открытых систем можно выделит следующие наиболее интересные проекты: **Joomla**, **Typo3**, **Wordpress** и, конечно же, **Drupal**. Практически все эти системы входят в первую десятку top100 CMS-рейтингов (кроме еще набирающей популярность Wordpress). Теперь немного рассмотрим эти системы (исключая CMS Drupal, т.к. его достоинства и недостатки, а также его уникальные особенности будут обсуждаться далее):

Joomla – это ответвление от системы **Mambo**. Joomla является довольно интересной и привлекательной, но все же несколько громоздкой системой. По мнению аналитиков, сейчас Joomla прибывает в состоянии «рецессии»¹ (в последнее время этот экономический термин стал узнаваемым), но автор не может это как-либо прокомментировать. На базе Joomla можно создать проекты различного назначения.

Wordpress – система для публикаций (блогов, новостных ресурсов). Эта система пока не входит в десятку top100, однако ее «наступление» уже ощутимо. С виду, по своей простоте, эта система чем-то даже напоминает Drupal, хотя проводить аналогии между столь разнородными системами не совсем правильно. В системе реализованы виджеты (блоки с информацией или элементами управления), которые работают в режиме drag-and-drop² – простое перетаскивание виджетов в разные области приводит к изменению числа и положения реальных элементов в проекте. Это сильно напоминает новую функциональность Drupal , где блоки и пункты меню также можно перемещать мышью. В комментариях к системе утверждается, что Wordpress может быть задействован и в других задачах, так это или нет автору достоверно неизвестно.

Typo3 – это «большая» система для «больших» проектов. Она является прямым конкурентом больших коммерческих систем и, по словам немецкого IT-журнала iX, «TYPO3 не нуждается в том, чтобы прятаться за спины коммерческих конкурентов»³. Эта система довольно громоздка, она считается полноценной **СМЕ**  и использовать такую для большинства классических веб-проектов просто расточительно.

Каждая из этих систем достойна внимания, однако среди большинства систем управления только CMS Drupal имеет полный набор следующих преимуществ:

1. Первое, довольно значительное преимущество, это принадлежность CMS Drupal к классу открытых интернациональных систем. Конечно, можно довольно долго вести дискуссию относительно того, какие системы лучше, коммерческие или же открытые, причем каждая из сторон обязательно приведет свои доводы, однако в данном случае можно обойтись и без долгих сравнений, тем более что эта тема поднималась ранее.
2. Официальный сайт проекта (сайт разработчика) – это в первую очередь «лицо» такого проекта. Официальным сайтом проекта Drupal является ресурс Drupal.org. Drupal.org – это действительно качественный проект, выполненный в респектабельном стиле. Однако внешнего вида не достаточно, оформления – это лишь косвенный признак того, насколько серьезно разработчики относятся к своему проекту. Другой немаловажный фактор – это содержание проекта – официальный сайт любой серьезной CMS должен размещать не только свежие дистрибутивы систем (с последними исправлениями), но и минимальное количество готовых модулей и тем, а возможно и локализаций. Drupal.org полностью соответствует этим требованиям – здесь имеются последние дистрибутивы, модули (на данный момент в количестве ≈ 1500 единиц), ≈ 190 тем, ≈ 40 переводов, новости, форум, скриншоты, демо-версию продукта (любой желающий, не устанавливая систему, может просмотреть все ее возможности), книга (handbook) и много документации по api⁴ системы. Есть также и

1 Рецессия (лат. Recessus – отступление) – в экономике термин обозначает спад производства...

2 Drag-and-drop – в переводе с английского означает буквально: тащи-и-бросай. Действие выполняется путем оперирования видимыми на экране компьютера объектами при помощи мыши. Суть действий заключается в перемещении виртуальных объектов относительно друг друга и таким образом выполнении каких-либо действий в приложении. [ru.wikipedia.org/wiki/Drag-and-drop]

3 Цитата с ru.wikipedia.org/wiki/Typo3

4 API – интерфейс программирования приложения – набор известных функций и возможностей определенного приложения, зная которые становится возможной работа с ним на программном уровне: взаимодействие с ним и

еще один немаловажный фактор – это организация всего вышеназванного. Глядя на Drupal.org – сразу становится понятно, что разработчики системы создали не только самый перспективный продукт, но и позаботились о его поддержке и поддержке его пользователей. Бесспорно Drupal.org, ровно как и CMS Drupal, соответствует многим требованиям, предъявляемым к массовым проектам такого класса.

Конечно, количество не всегда означает качество. На официальном сайте проекта Drupal имеется порядка ≈1500 модулей (это не считая модулей из других источников). Однако если брать в расчет только самые первоклассные модули (например, их доля из общего числа будет составлять лишь 15%), то и здесь их количество остается весьма значительным (т.е. более 200 модулей).

3. Большим плюсом открытого проекта является наличие у него национальных сообществ и региональных представительств. Причем подобные сообщества «объединяют» не только рядовых пользователей, но и хороших специалистов. У проекта Drupal кроме всех прочих также имеются русское, белорусское и украинское сообщество (в «полезных источниках», в конце книги на них имеются ссылки). Благодаря им Drupal становится «ближе», ведь теперь всю необходимую поддержку, локализацию и часть документации к нему можно получить на родном языке.
4. Простота и функциональность – это не взаимоисключающие понятия. К сожалению, многие функциональные продукты достаточно сложны, а простые обычно предоставляют только базовую функциональность – золотую середину найти достаточно сложно. Однако участники проекта Drupal все же смогли создать довольно простой, но в тоже время очень функциональный продукт. Дистрибутив этой системы состоит всего из 277 файлов (версия 5.7, размер дистрибутива 2,17mb), в то время как Joomla из 2201 (версия 1.0.12ru, размер дистрибутива 7,87mb), а одна из частей Туро3 из 3225 (версия 4.1.1, размер дистрибутива 19,5mb). Причем следует также отметить, что код данной системы признан самым задокументированным, поэтому разобраться в нем программисту будет намного проще, нежели с кодом других систем. Также не следует забывать и о том, что сложность сказывается на надежности и защищенности – в масштабных проектах участвуют сотни разработчиков одновременно и их работу довольно сложно контролировать.
5. Инновационность. Разработчики CMS Drupal постоянно вносят в свой продукт «свежие» решения. Конечно, раньше такой подход сказывался на переносимости в общем – отказ от более старых технологий и переход к новым ставил под сомнение возможность легкой миграции ранее созданных проектов на новую версию продукта, а также приводил к необходимости переписывания существующих модулей и тем. Однако разработчикам в последующих версиях (5-ые и 6-ые версии) удалось избежать подобных недостатков, теперь новые возможности в своем большинстве дополняют ранее реализованные, а не полностью исключают их. Другие же системы в погоне за обратной совместимостью продолжают и далее эксплуатировать уже не самые перспективные решения.

Многие убеждены, что для каждого конкретного случая требуется своя система. Для форума одна, для электронного магазина другая. И отчасти они правы. Есть вещи сугубо специализированные, где универсальные системы будут малоэффективны. Но Drupal – это универсальная система, которая способная к интеграции с другими более специализированными системами, например, в разнородных системах Drupal можно использовать в качестве центрального объединяющего звена. В конечном счете только вам решать, какой системе отдавать предпочтения, однако если вы дочитали до этой строки, значит, скорее всего, вы уже сделали свой выбор в пользу Drupal.

2.1.2 История создания

История создания Drupal начинается в 2000 году в кругах антверпенского университета. Для общения пользователей в одной из сетей местного общешития Дрис Байтаерт создал внутренний сайт и программу, на которой этот сайт функционировал. Программа на тот момент еще не имела названия. Когда Дрис Байтаерт собирался зарегистрировать доменное имя для этого сайта, он сделал опечатку и вместо имени dogr.org (от голландского dogr – деревня), зарегистрировал dgor.org (на английском – капля). В 2001 году Дрис опубликовал исходные коды dgor.org под названием «Drupal». После этого система «переехала» на домен Drupal.org, который приобрел статус официального сайта этого проекта. Автор и создатель системы Дрис Байтаерт и поныне является его ру-

ководителем. Эту историю в оригинале вы можете прочесть здесь: <http://drupal.org/node/769>. Также посетив веб-архив (ссылки расположены под изображением ниже), вы можете посмотреть, как выглядела первая реализация проекта Drupal.

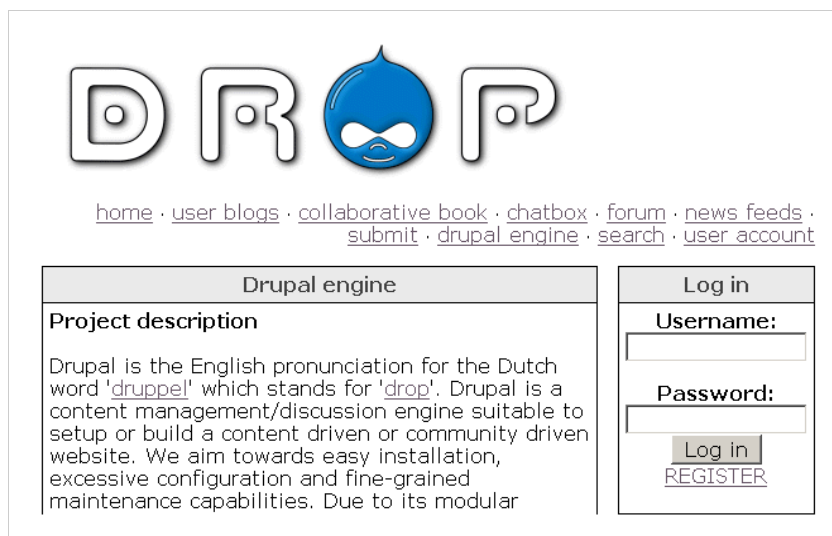



Рис. 6: http://web.archive.org/web/*/http://drupal.org
<http://web.archive.org/web/20011214140315/www.drop.org/node.php?id=411>

По прошествии времени к проекту Drupal стали подключаться новые люди. Это были и веб-разработчики, и другие профессионалы. В данное время над ядром системы трудятся с десяток программистов. Есть много и сторонних разработчиков, также оказывающих огромную поддержку (например, присылающих свои исправления). Система постоянно развивается: разрабатываются новые модули и темы.

2.2 Установка системы

Практически любая система требует установки и настройки – Drupal и вся база, на которой он функционирует, не исключение. Если вы приобрели систему «под ключ» (целиком сайт или портал), то устанавливать систему вам не потребуется, в этом случае рекомендую сразу перейти к следующим разделам, где описаны процессы настройки и наполнения системы.


Если же вы самостоятельно приобрели ХОСТИНГ , то видимо, вам не удастся избежать установки этой системы собственными силами. Хотелось бы отметить, что в сети имеется немало желающих произвести такую установку за вас (за небольшое вознаграждение). Довольно трудно понять, кто эти помощники. Возможно вы найдете известного интегратора с серьезного ресурса – в этом случае вам не придется опасаться за безопасности своего проекта – риск будет минимален. Но возможен и другой исход – вам может попасться и неблагонадежный интегратор (в этом случае в вашу систему могут попутно интегрироваться троянские скрипты, чужая реклама и т.д.). Только вам решать, устанавливать ли систему самостоятельно или же к этому вопросу стоит привлечь сторонних специалистов. В первом случае, устанавливая систему самостоятельно, вы сможете получить дополнительный опыт, и в будущем у вас будет возникать все меньше вопросов в работе с данной CMS. Во втором же случае автор рекомендует более обстоятельно подходить к вопросу выбора специалистов.

Прежде чем развернуть систему Drupal, необходимо выяснить ее функциональные требования к среде исполнения – понять какая «минимальная база» требуется для ее работы. Эта тема обсуждается далее.

2.2.1 База для функционирования CMS Drupal

Drupal сам по себе является надстройкой, которая функционирует на определенной «базе». К такой базе обычно относят трехзвенную связку программ **Apache**[1] + **PHP**[2] + **MySQL**[3]¹ – данная связка является основой полноценного веб-сервера. Опишем немного каждое звено этой базы.

Звено 1: Веб-сервер Apache

Apache – это открытое популярное приложение²-веб-сервер ³. Приложение-сервер – это, как правило, резидентная (находящаяся постоянно в памяти) программа, обслуживающая запросы клиентов. В качестве клиента может выступать пользователь или другая программа.

Программа-веб-сервер Apache устанавливается на удаленную сервер-машину, на которой Apache постоянно находится в активном состоянии, ожидая входящие запросы. При поступлении запроса Apache производит его обработку, после чего результат обработки отправляет запросившей стороне. Фактически Apache выступает «посредником» между вашим браузером и внутренними ресурсами сервера – когда вы нажимаете на веб-ссылку, браузер посылает запрос удаленной машине-серверу, на которой «работает» Apache, Apache принимает запрос и генерирует ответ (например, в виде html страницы), передавая его обратно, вашему браузеру.

1 В приобретаемом хостинге она уже имеется.

2 Если не вдаваться в формальности, то можно условно считать, что приложение и программа – это одно и то же.

3 Конечно Drupal поддерживает и некоторые другие веб-сервера, однако последний остается вне конкуренции.

Звено 2: Интерпретатор PHP

PHP – это приложение-транспирующий-интерпретатор. В его задачу входит подготовка (трансляция) программного кода и его исполнение (**интерпретация** **A**). Здесь в качестве программного кода выступает Drupal, который по сути и является набором **кода**. В принципе, PHP это самостоятельная¹ программа, равно как Apache и MySQL, однако только в связке друг с другом они могут стать полноценным веб-сервером.

В чистом виде **программный код** представляет собой набор предписаний в **текстовом** виде. Код может преобразовываться в настоящую **программу** (*.exe и *.com файлы), т.е. **компилироваться**, после чего ее исполнением занимается **операционная система**, сокращенно **ОС** (хотя обычному пользователю всегда кажется, что программа работает независимо от ОС). На рисунке ниже этой последовательности соответствует левая часть. Код также может оставаться и в «открытом», т.е. не преобразованном виде, либо в частично преобразованном (в подготовленном виде). Тогда его исполнением занимается специальная **программа-интерпретатор** (эта программа является «связкой» между кодом и вашей ОС), а для операционной системы этот код продолжает оставаться просто текстом. На рисунке ниже этой последовательности соответствует правая часть. PHP является интерпретатором, а не компилятором, поэтому и код на php исполняется непосредственно PHP (некоторые вместо выражения «php код» предпочитают использовать «**php-скрипт**»). PHP – это не только интерпретатор, но также и **название языка**.



Многие пользователи не понимают, чем отличается **компиляция** от **трансляции**, поэтому часто называют PHP компилятором, что абсолютно неверно. Компиляция – это преобразование программного кода к конечным машинным командам. Трансляция, в сущности, не преобразует код к машинным командам, а лишь делает его более «понятным» для интерпретатора.

Также хочется вернуться к вопросу закрытых CMS и PHP, а, точнее, каким образом можно спрятать заведомо открытый php код. Самый надежный способ сделать это – скомпилировать его. После компиляции код станет набором машинных команд и уже вряд ли будет кому-нибудь интересен. Однако, как уже было сказано выше, PHP не делает компиляцию, он лишь транслирует код (причем в классической версии делает он это в момент обращения к php-скрипту, а сам скрипт при этом никак не защищен и располагается на сервере в виде открытого текста). Однако и интерпретируемый открытый код также можно преобразовать – его можно оттранслировать, после чего код превратиться в набор предписаний для интерпретатора (непонятных человеку). Подобную трансляцию позволяет выполнить такая программа как Zend Guard (есть и альтернативные программы). Она не только транслирует код, но и позволяет сделать его обфускацию – процедуру, искажающую / усложняющую код настолько, насколько это возможно. Запустить оттранслированный код можно только при использовании расширения Zend Optimizer.

¹ Благодаря этому он может быть использован как подобие java машины – phpGtk.

Звено 3: MySQL

MySQL – это СРБД | СУРБД (система управления реляционными базами данных), или как его еще называют пользователи, «SQL-сервер» или «база данных». MySQL – это также серверное приложение. В его задачу входит работа с данными (хранение, выдача и т.д.). Однако MySQL – это не единственный SQL-сервер, с которым может работать Drupal, так, например, вместо MySQL можно использовать **Postgre SQL** – сервер, который менее популярен у хостинг-операторов. Drupal использует все СУРБД только как хранилище данных (data store), не задействуя как-либо специфических функций, поэтому выбор между различными СУРБД основан, как правило, на сравнении их скоростных показателей (при этом возможности СУРБД не являются определяющим фактором) – MySQL более «быстрый» и не имеет каких-либо функциональных недостатков, поэтому предпочтение отдается именно ему.

Сделать СУРБД MySQL «более быстрым» можно только после его соответствующей настройки. Так, например, за счет использования таблиц типа MyISAM можно добиться значительного ускорения (жертвую при этом надежностью, однако во многих веб-задачах решающим фактором является не надежность, как это не парадоксально, а скорость). В разделе **Первая настройка** → **Локализация** об этом говорится более подробно.

Как работают все звенья?

В сильной абстракции картину взаимодействия клиента и веб-сервера (на котором установлено вышеописанное ПО) можно отобразить одной иерархической схемой «запрос-ответ»:

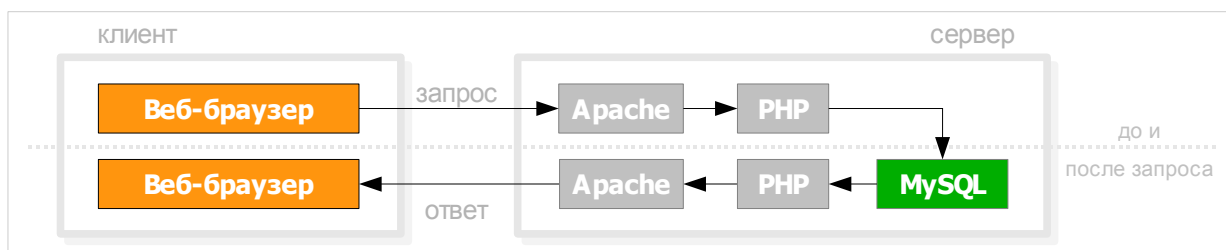


Рис. 7: Схема взаимодействия клиента и веб-сервера.

Эту схему можно объяснить следующим образом:



Клиент (т.е. пользователь) в браузере нажимает на ссылку.

- ✓ Браузер посылает запрос на удаленный сервер (на удаленном сервере работает веб-сервер Apache, которому и передаются все веб-запросы – в нашем случае этим запросом является ссылка).
- ✓ Apache анализирует запрос (ссылку). Если ссылка указывает на статичный html-документ, то в ответ на запрос отдается этот документ (в неизменном виде), после чего запрос считается выполненным. Когда же ссылка указывает на php-скрипт (файл с расширением php / php5 / php4 и др.), Apache запускает php-модуль¹ (т.е. PHP) и передает ему все параметры своего запроса (ссылку).
- ✓ После этого в работу вступает PHP. PHP исполняет скрипт, на который ему была передана ссылка. Если в коде скрипта имеются обращения (запросы) к СУРБД, то PHP инициирует такое обращение – подключается к серверу СУРБД и передает необходимые запросы.
- ✓ Далее СУРБД выполняет свою часть работы, после чего направляет результат обратно PHP (результатом работы СУРБД, например, может стать выборка данных).
- ✓ После обработки php-скрипта, PHP возвращает результат Apache. Результатом к примеру может стать сгенерированный документ, в который также может включаться и результат выборки из базы

1 Или php через cgi.



На последнем этапе Apache отправляется результат клиенту.

В описанной схеме выше Drupal занимает промежуточное место – логически между PHP и MySQL (см. на схеме ниже) и физически «в среде PHP».

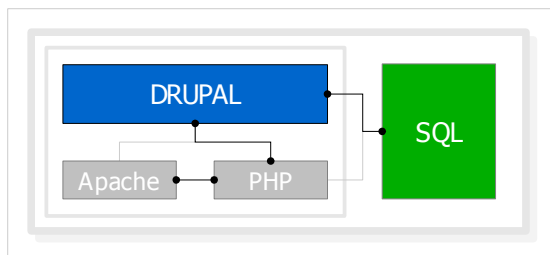


Рис. 8: Drupal и окружение.

Если вы желаете узнать больше об Apache, PHP и MySQL (истории создания, возможности разных версий), то рекомендую посетить следующие ссылки:

1. <http://ru.wikipedia.org/wiki/Apache>
Википедия об «Apache»
2. <http://ru.wikipedia.org/wiki/PHP>
Википедия о «PHP»
3. <http://ru.wikipedia.org/wiki/MySQL>
Википедия о «MySQL»
4. <http://ru.wikipedia.org/wiki/Drupal>
Википедия о «Drupal»

2.2.2 Установка Drupal

Установить CMS Drupal можно как локально, так и удаленно. Далее будут обсуждаться эти два варианта, но сначала следует оговорить один момент: обсуждение локальной установки будет описываться здесь в рамках установки под ОС Windows 200x/XP/Vista. Почему именно под эти операционные системы? В данном случае ответ простой: если вы работаете под *nix подобной системой (Unix, Linux) и при этом сумели самостоятельно его установить и настроить, значит, вы уже и сами готовы (морально и интеллектуально) к поиску всей необходимой документации и последующей установки Drupal и его функциональной базы на данную платформу.

Однако из этого списка есть и исключение: ОС Mac OS X является предустановленной полноценной Unix системой на компьютерах Apple. Приобретая такой компьютер, вы получаете его именно с этой системой (теперь на машины от Apple через Boot Camp можно устанавливать и ОС Windows). Поскольку под ОС Mac OS X имеется много удобных «сборок», которые позволяют произвести установку всех необходимых систем без как-либо «экссессов», обсуждать здесь эту платформу также не имеет смысла.

Под локальным компьютером подразумевается свой собственный, соответственно, локальная установка, есть установка на свой собственный компьютер. У многих, конечно, может возникнуть вопрос: «А зачем устанавливать на собственном компьютере такую систему, ведь он не является сервером и не будет обслуживать пользователей?» Вопрос хороший. В принципе, можно обойтись и без такой установки, однако установив Drupal локально, вы можете легко экспериментировать с ним (устанавливать новые модули и т.д.) не опасаясь за потерю информации, ведь это будет не реально работающая общедоступная система, а локальная. Для установки Drupal на локальный компьютер вам понадобятся описанные выше программы, т.е. Apache, PHP и MySQL. О том, как получить их дистрибутивы, как их установить и связать, описано в приложении Б.

Удаленная установка – это установка на удаленный сервер. Под удаленным сервером подразумевается сервер, на котором хостинг-оператор вам официально предоставил аккаунт. Выбор хостинг оператора подробно описан в Приложении А. (рекомендую ознакомиться с этим приложением). Если вы приобрели «виртуальный хостинг», то вопрос установки Apache, PHP и MySQL вас уже не коснется, т.к. в подобных тарифных планах все это уже предустановлено и остается лишь перейти сразу к установке самой Drupal. Если же вы приобрели целый виртуальный сервер, со всем необходимым ПО, то здесь возникает аналогичная ситуация как и с «виртуальным хостингом», только возможностей по настройке сервера в данном случае будет намного больше.

Функциональные требования

Для работы Drupal необходимо, функциональная база удовлетворяющая следующим условиям:

Apache:

версия: 1, 2, 2.2

дополнительно:

- + поддержка .htaccess
- + активация модуля mod_rewrite

PHP:

версия: 5-ая (4-ая)

модули:

- + mysql / mysqli
- + gd2
- + mbstring
- + xml (в 5-ой версии по умолчанию включен)

не должны быть отключены стандартные модули (bcmath, dom, filter, json, pcre, xml и др.)

MySQL:

версия: 5-ая (4-ая)

Настройка Apache

Раздел установки веб-сервера Apache, его первой настройки и связке с PHP, был вынесен в [приложение Б: Установка Apache](#) (данный вопрос актуален только в случае локальной установки веб-сервера).

Для установки и работы системы Drupal, необходима поддержка веб-сервером Apache следующих функций:

- 1) Возможность работы с .htaccess (обязательное условие);
- 2) Активность модуля mod_rewrite (данный модуль необходим для работы «Чистых ссылок» / «Clean URLs»).

Htaccess. Вся конфигурация (настройка) веб-сервера Apache располагается в соответствующих конфигурационных файлах. Одним из таких файлов является **httpd.conf** – это главный конфигурационный файл Apache. Также возможность конфигурации (ее малая часть) может быть передана и пользователю хостинг-аккаунта, реализуется такая возможность через файлы **htaccess**. Drupal активно использует возможности htaccess, поэтому «включение htaccess» является необходимым условием для работы Drupal.

Mod_rewrite. Системе Drupal данный модуль необходим для работы с «Чистыми ссылками». Что такое чистые ссылки? В данном случае в качестве наглядного примера можно использовать ссылку на пользовательский раздел системы: вот так эта ссылка выглядит в обычном режиме: <localhost/?q=user/1>, а так она станет выглядеть, после включения mod_rewrite и включения опции «Clean URLs» в Drupal: <localhost/user/1> – с первого взгляда отличия не существенны, однако все преимущества данного решения вы сможете оценить лишь на практике.

Настройка Apache: Локальная настройка

В настройках веб-сервера по умолчанию заблокирована возможность работы с .htaccess, поэтому, если вы произвели новую установку Apache, вам необходимо будет активизировать эту возможность. Для этого откройте главный конфигурационный файл Apache (httpd.conf) и замените в нем каждое выражение «AllowOverride None» на «AllowOverride All»¹, после чего перезагрузите веб-сервер. Работа с httpd.conf была подробно описана в [приложении Б: Установка PHP](#).

По умолчанию модуль mod_rewrite также не активен. Для его включения вам необходимо открыть файл httpd.conf, найти строку «#LoadModule rewrite_module modules/mod_rewrite.so» и разкомментировать ее, убрав символ «#». После этого следует перезагрузить веб-сервер.

Настройка Apache: Удаленная настройка

На удаленном сервере возможность работы с htaccess будет уже включена (во всяком случае такая функциональность должна предоставляться хостинг-оператор как базовая; если возможность работы с htaccess отсутствует, значит вам необходимо обратиться в службу поддержки оператора и запросить ее², в случае же отказа оператора от ее предоставления, вам придется «переехать» на другой хостинг). Аналогичная ситуация и с mod_rewrite – это также базовая возможность любого серьезного хостинг-оператора.

1 Возможны и другие опции.

2 «Виртуальный хостинг» не дает возможностей производить такие действия самостоятельно,

Настройка PHP

Раздел установки и первой настройки PHP, был вынесен в [приложение Б: Установка PHP](#).

Для полноценной работы Drupal рекомендуется 5-ая версия PHP со следующими, дополнительно подключаемыми модулями: mysql / mysqli, gd2, mbstring и xml.

Настройка PHP: Локальная настройка

Подключение модуля mysql подробно описывалось в [приложении Б: Установка PHP](#). По аналогичной схеме производится подключение и остальных вышеупомянутых модулей, кратко опишем эту схему еще раз: в начале следует открыть файл php.ini – это главный конфигурационный файл PHP; затем, в нем, необходимо найти каждую директиву включению нужного модуля; после чего найденную директиву необходимо сделать активной, т.е. попросту разкомментировать ее, убрав символ «;». После внесенных всех изменений php.ini следует сохранить, а веб-сервер перезагрузить.

Настройка PHP: Удаленная настройка

На удаленном сервере PHP будет уже настроен так, как требуется. Все описанные выше модули являются частью дистрибутива данного продукта.



Настройка MySQL

Раздел установки и первой настройки MySQL был вынесен в [приложение Б: Установка MySQL](#).

Drupal использует MySQL (или PostgreSQL и т.д.) в качестве хранилища данных (именно здесь и будут храниться все страницы, комментарии и т.д.), поэтому для Drupal на SQL-сервере требуется одна существующая база. Также имя этой базы будет запрошено инсталлятором на первом шаге установки системы Drupal, поэтому следует заранее создать такую базу. Далее рассматриваются способы создания новой базы для локальной и удаленной установки.

Также вместо MySQL или PostgreSQL вы можете использовать другие СУРБД или даже файловую систему, но для этого вам потребуется найти соответствующие библиотеки (абстрактные слои). Однако следует помнить, что большинство сторонних модулей поддерживают только MySQL и PostgreSQL.

Настройка MySQL: Настройка локального mysql сервера

Создать новую базу можно, минимум, двумя способами: при помощи **MySQL Query Browser / MySQL Administrator** и при помощи **mysqladmin**.

Способ первый – самый очевидный. Если ранее, по совету автора, вы произвели установку **MySQL Gui Tool**, то уже сейчас вы можете получить за это небольшое вознаграждение. Откройте **Query Browser** и пройдите аутентификацию (об аутентификации было написано в [Приложении Б](#), в части установки MySQL сервера). Далее, оказавшись в среде управления SQL-сервером, наведите курсор на блок **Schemata** и вызовите контекстное меню (правая клавиша мыши), в появившемся меню выберите «Create New Schema», и во вновь появившемся одноименном окне в поле «Schema name» введите название новой базы (на английском). Все, нажмите **ОК**, и новая база сразу будет создана (а также появиться в блоке «Schemata»).

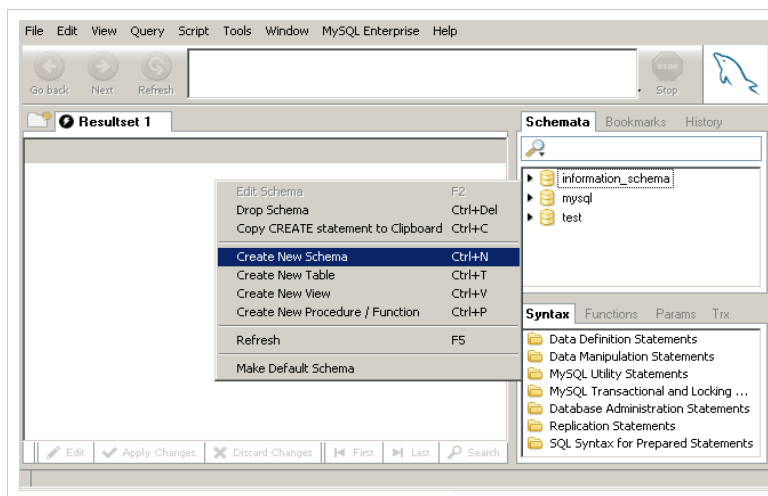


Рис. 9: Query Browser в действии.

Этот способ будет работать только на локальной машине – многие хостинг-операторы для безопасности отключают возможность удаленного подключения к своим SQL-серверам, поэтому вы не сможете подключиться к ним при помощи MySQL Query Browser или MySQL Administrator.

Теперь о самом легком способе. Зайдите в каталог **data** MySQL сервера. Создайте в этом каталоге новый подкаталог (имя подкаталога должно быть таким же, как и имя создаваемой базы). Все. Этот способ будет работать только на локальной машине т.к. на удаленном сервере с тарифным планом «Виртуальный хостинг» проделать нечто подобное практически невозможно.

Способ второй – для тех, кто не ищет легких путей. В [Приложении Б](#) автор уже упоминал о том, что к MySQL серверу прилагаются дополнительные утилиты (они устанавливаются вместе с ним). Располагаются эти утилиты в каталоге **bin**, и многие из них работают только в командной строке. Командную строку очень часто называют **консолью**, и наоборот.

Командная строка располагается в **Пуск → Программы → Стандартные → Командная строка (Start → Programs → Accessories → Command Prompt)**. Запустив командную строку, вы можете пря-

мо в ней набрать имя **mysqladmin** и нажать Ввод – в результате должна появиться длинная подсказка с параметрами mysqladmin (пример ниже).

```
Administration program for the mysqld daemon.
Usage: mysqladmin [OPTIONS] command command...
-c, --count=#      Number of iterations to make. This works with -i
                   <--sleep> only.
-#, --debug[=name] Output debug log. Often this is 'd:t:o,filename'.
-f, --force        Don't ask for confirmation on drop database; with
                   multiple commands, continue even if an error occurs.
...
ping              Check if mysqld is alive
processlist       Show list of active threads in server
reload            Reload grant tables
refresh           Flush all tables and close and open logfiles
shutdown         Take server down
status           Gives a short status message from the server
start-slave       Start slave
stop-slave        Stop slave
variables         Prints variables available
version           Get version info from server

C:\>_
```

Рис. 10: Запуск mysqladmin из командной строки.

Если при установке MySQL сервера вы не включили bin директорию в переменную окружения Path, то в результате вы можете получить ошибку, показанную ниже. Тогда для запуска mysqladmin вам придется перейти непосредственно в каталог bin (посредством команды «CD путь_k_bin»).

```
C:\>mysqladmin
'mysqladmin' is not recognized as an internal or external command,
operable program or batch file.

C:\>_
```

Рис. 11: В Path нет bin директории.

Теперь в командной строке наберите «**mysqladmin --user=root --pass=*** create base_name**», где base_name – это то имя, которое вы хотели бы задать вновь создаваемой базе, а «***» – это пароль к MySQL серверу. Вот и все.

Настройка MySQL: Настройка удаленного mysql сервера

Создание базы на удаленном сервере производится через панель управления собственного хостинг-аккаунта. После того как вы договорились с хостинг-оператором (приобрели аккаунт¹ или за-требовали демо-аккаунт), на ваш электронный почтовый ящик должно придти письмо с реквизитами нового аккаунта. Это письмо может выглядеть примерно так:

Domain:	example.com	POP Server:	mail.example.com
Control panel:	https://example.com:2222/	SMTP Server:	mail.example.com
Username:	User	FTP accounts:	1
Password:	0000	Anonymous FTP:	OFF
FTP:	ftp://example.com	NS1:	ns1.example.com
Bandwidth:	unlimited Megabytes	NS2:	ns2.example.com
Disk Space:	50 Megabytes	MySQL Databases:	1
Virtual Domains:	1	Domain Pointers:	2
Subdomains:	2	SSH Access:	OFF
POP Email Accounts:	2	Secure Socket Layer:	ON
Email Forwarders:	unlimited	CGI:	ON
Email Autoresponders:	unlimited	PHP:	ON
Email Mailing Lists:	2	DNS control:	OFF

Рис. 12: Письмо с реквизитами аккаунта.

Это письмо следует сохранить локально (эти данные желательно поместить в файл-архив, защищенный паролем, а письмо удалить из почтового ящика, не перемещая в корзину).

Если при работе с почтой вы используете внешние почтовые программы², то рекомендуется в их настройках (в настройках параметров сервера) устанавливать **SSL** протокол – тогда транслируемая от сервера почта будет передаваться в защищенном режиме.

Теперь рассмотрим это письмо более подробно. Поле **Domain** – это имя вашего домена (проекта),

¹ Надеюсь, что вы поступили разумно и приобрели аккаунт не сразу на год, а только на месяц.

² Например Outlook, Thunderbird и д.т.

в демо-аккаунте это будет имя примерно такого формата `http://demoAccount1.hostname.domain`. Если в строке адреса браузера вы наберете это имя, то попадете на главную (пока еще пустую) страницу своего будущего проекта. **Control panel** – это адрес панели управления вашего аккаунта (вместо слов «Control panel» может быть указано название самой панели управления, например **DirectAdmin** или **ISP manager**). Далее идут **Username** и **Password** – это имя (логин) и пароль, которые необходимо будет указать при «входе» в панель управления. Остальные поля в данный момент пока не представляют интереса, поэтому сейчас они не будут рассматриваться. Далее на примере панели управления **DirectAdmin** будет рассмотрена возможность создания базы.

Узнать больше о DirectAdmin можно прямо на его официальном сайте <http://directadmin.com> (англ.). Здесь также вам предоставляется возможность попробовать его в действии – выбрав раздел «demo», вы можете запустить демо-версию данной панели управления.

Существует и множество других панелей управления (например, **ISP manager**). Практически все популярные панели предоставляют базовые возможности управления СУРБД.

Для того чтобы попасть в панель управления, введите в строке адреса браузера адрес из письма (рисунок 12), указанный в поле «Control panel». В результате в браузере должно появиться аутентификационное приглашение, изображенное на рисунке ниже.

Рис. 13: DirectAdmin: Приглашение.

После входа в панель вы попадаете на ее главную страницу (или на страницу выбора доменов, если у вас на одном аккаунте зарегистрировано сразу несколько доменов). По умолчанию весь интерфейс панели может быть на английском языке. Для выбора русского языка зайдите в раздел **Site Summary / Statistics / Logs** и в поле **Language** выберите **ru**, а затем нажмите **save**. Сразу после этого весь интерфейс панели управления станет русскоязычным. Главная страница панели управления DirectAdmin (ее верхняя часть) показана на рисунке 14.



Рис. 14: DirectAdmin: Часть главной страницы.

На главной странице имеется блок **Ваш аккаунт**¹. В нем отражены возможности (точнее, их небольшая часть). Также в правой части страницы располагается блок, отражающий состояние ресурсов аккаунта (объем диска, канал, email адреса, ftp аккаунты и базы данных). Обратите внимание на поле **Базы данных - Макс.**, если оно нулевое или же значение **Текущее** равно значению **Макс.**, то это значит, что вы не сможете **больше** (или **вообще**) добавить новую базу. Пример этого блока показан ниже.

¹ Слово «аккаунт» принято записывать с двумя «к», однако этого придерживаются не все.

Ваш аккаунт		
Диск		
	Текущее	Макс.
Диск (МБ)	0.0605	50
Канал (ТБ)	0.0021	unlimited
E-mail адреса	1	2
Фр аккаунты	1	1
Базы данных	0	1

Рис. 15: DirectAdmin: Состояние ресурсов аккаунта.

Для создания новой базы перейдите в раздел [MySQL управление](#). Затем выберите ссылку [Создать новую базу данных](#). Теперь располагаясь в разделе «Создать новую базу данных» заполните следующие поля:

Имя базы данных – составьте небольшое имя (например, «drupal» или «myProject»), обратите внимание, что к этому имени добавляется префикс «Логин_». В имени (и в других параметрах) допускаются только символы латиницы;

Имя пользователя базы данных – придумайте и впишите имя пользователя (на его длину имеется ограничение);

Пароль пользователя и **Подтвердить пароль**: в эти поля следует вписать пароль для пользователя базы данных (пароль должен быть довольно сложным – вам все равно не придется его запоминать).

Далее следует нажать кнопку [Создать](#), после чего должно произойти создание новой базы (после ее создание появится небольшая таблица с ранее указанными параметрами, эти параметры необходимо будет указать в процессе установки Drupal).

Database:	prefix_basename
Host:	localhost
Username:	prefix_usernamr
Password:	...

Рис. 16: База создана, ее параметры следующие...

Установка Drupal

Итак, если вы дошли до этой части, значит «минимальная база» для установки у вас уже подготовлена. Автор надеется, что приобретенный вами хостинг поддерживает всю необходимую функциональность (требования были описаны в [Приложении А – «Выбор хостинга»](#)), иначе неминуемо возникнут трудности, которые придется преодолевать.

Установка Drupal: Локальная установка

Войдите в каталог `%htdocs%` (он же `%apache%/htdocs`, где `%apache%` – это тот каталог, в который вы произвели установку Apache). Удалите все его содержимое. Распакуйте дистрибутив системы Drupal (это будет файл `drupal-5.X.tar.gz` или `drupal-6.X.tar.gz`, который можно распаковать любым способом архиватором, например, WinRAR'ом). Далее переместите все распакованное содержимое в `%htdocs%` (распакованное содержимое – это папка, в которой находятся такие файлы как `index.php`, `install.php` и др., а также каталоги).

Теперь, если вы устанавливаете Drupal **5**, создайте в `%htdocs%` пустую папку с именем «`files`». В папке `files` будут находиться загружаемые вами в систему файлы, а также файлы, сгенерированные непосредственно системой. Задать путь к директории `files` можно и прямо через интерфейс Drupal, определив в раздел *Управление сайтом* → *Настройка сайта* → *Файловая система* [[url: admin/settings/file-system](url:admin/settings/file-system)] параметр «Путь файловой системы» для Drupal **5** и тот же параметр для Drupal **6** но в разделе *Управление* → *Настройка* → *Файловая система* (в случае с Drupal **6** нет необходимости создавать папку `files` вручную, по умолчанию система создаст ее при установке – [ф-путь: `sites/default/files`]).

Для Drupal **6** также необходимо в директории `sites/default` создать копию файла `default.settings.php` с новым названием `settings.php`. При этом `settings.php` станет основным конфигурационным файлом Drupal (кроме всего прочего, в этом файле будут храниться и параметры подключения к собственной базе). В ранних **6**-ых версиях Drupal делал подобное «клонирование» самостоятельно, но теперь эти действия приходится выполнять пользователям.

Автор полагает, что разделение на `settings.php` и `default.settings.php` сделано для удобства восстановления файла/файлов конфигурации, а также для удобства обновления всей системы в целом.

Далее вам следует перейти к пункту «*Общие шаги*».

Установка Drupal: Удаленная установка

Далее предполагается, что после создания базы, описанной выше, вы все еще находитесь в панели управления хостинг-аккаунта, если нет, то войдите в нее повторно.

Оказавшись на главной странице панели управления, перейдите по ссылке «*Файловый менеджер*» → «*domains*» → *Имя_домена* → «*public_html*», после чего вы окажетесь в корневой директории веб-сервера. Теперь необходимо загрузить в нее дистрибутив системы Drupal, для этого нажмите кнопку «Загрузить файлы в текущую директорию». Попав на страницу загрузки файлов (рис. 17), укажите файл дистрибутива системы Drupal (`drupal-5.X.tar.gz` или `drupal-6.X.tar.gz`), а затем, нажав «Загрузить файлы», произведите его загрузку на сервер.

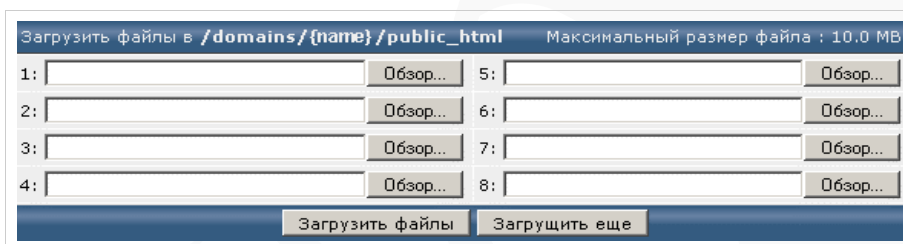


Рис. 17: DirectAdmin: Страница загрузки файлов.

Если загрузка прошла успешно, то у вас должно получиться все так, как показано на рисунке 18.




Type	Name	Size	Perm.	Action
	../ Up a Level	0.50k	755	
	cgi-bin	2.50k	755	Protect Rename Copy
	drupal-X.X.tar.gz	0.720M	755	Extract Rename Copy

Рис. 18: DirectAdmin: Загрузка прошла успешно.

Теперь необходимо распаковать загруженный архив. В панели DirectAdmin, в списке файлов у любого архива появляется дополнительная ссылка «Extract». Далее, выбирая эту ссылку, вы перейдете в окно «Извлечения файлов», показанное на рисунке 19.

DirectAdmin не считает архивом zip файлы, однако вы можете переименовать zip архив в архив tar.gz (т.е. в полном названии файла изменить только расширение), после чего попробовать распаковать переименованный архив. Сделать аналогичные действия с архивами rar уже не получится.

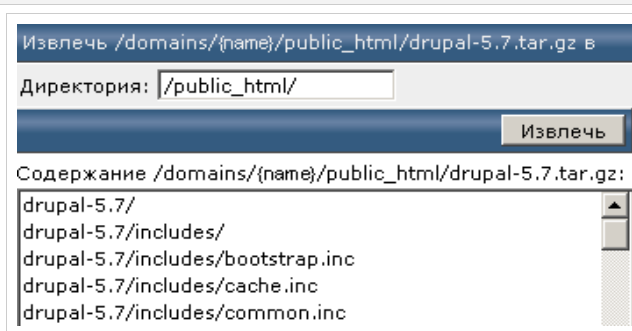


Рис. 19: DirectAdmin: Извлечения файлов из архива.

Нажмите кнопку «Извлечь». После этого рядом с архивом появится одноименная распакованная папка. Теперь необходимо войти в нее. Внутри вы увидите все те же файлы дистрибутива системы (среди них будут [index.php](#) и [install.php](#)). Вам следует выделить все файлы при помощи ссылки [Select](#) вверху списка файлов, а затем при помощи кнопки «Add to Clipboard» добавить все содержимое в буфер обмена. После этого вам нужно подняться на один уровень выше («../Up a Level») и нажать кнопку «Move Clipboard Files here». После этих действий все файлы окажутся там, где им и положено быть ([index.php](#) будет теперь доступен через следующий путь [имя_домена / index.php](#), а не через длинный [имя_домена / drupal-X.X / index.php](#)). И, напоследок, вы можете удалить пустую папку [drupal-X.X](#).

Также создайте в корневой папке веб-сервера пустую папку с именем «files» и скопируйте и переименуйте конфигурационный файл [settings.php](#). Более подробно об этом вы можете прочесть в разделе выше – «Установка Drupal: Локальная установка», 2-ой и 3-ий абзац.

Далее вам следует перейти к пункту «Общие шаги».

Установка Drupal: Общие шаги

Теперь следует открыть веб-браузер и в строке адреса ввести **localhost** (или **127.0.0.1**, если не работает первое имя) в случае локальной установки, либо имя вашего домена, на котором вы и производите установку, в случае удаленной установки. В результате вы должны увидеть изображения, показанные ниже (изображение слева – это 5 версия системы, справа – 6).

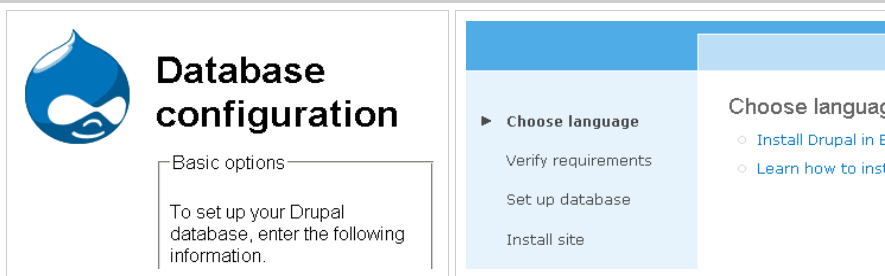


Рис. 20: Установка: начало (5, 6).

Всю установку можно разбить на следующие этапы (перед каждым действием стоит знак 5 и/или 6, это означает, что данное действие будет выполняться только в указанной версии Drupal):



6 «Choose language»

На первом этапе вам предстоит выбрать язык. По умолчанию здесь нет альтернативы английскому языку, поэтому следует выбрать ссылку «[Install Drupal in English](#)».



«Verify requirements»

Далее инсталлятор производит проверку конфигурации среды исполнения, т.е. попросту проверяет, удовлетворяют ли минимальным требованиям программы и их настройки. Проверка выполняется в фоновом режиме (скрытно), и о ее результатах (а, точнее, о несоответствиях, если таковые имелись) вы узнаете на следующем этапе.



5 «Database configuration»

6 «Set up database: Database configuration»

Теперь вы должны оказаться на этапе задания параметров СУБД (базы данных). Если на предыдущем этапе инсталлятор выявил несоответствия минимальным требованиям конфигурации, то сейчас вы будете уведомлены об этих несоответствиях. При обнаружении критических несоответствий дальнейшая установка системы станет возможной до тех пор, пока эти несоответствия не будут устранены. Если же все прошло успешно, тогда инсталлятор предложит вам заполнить следующий набор полей и переключателей:

Переключатель «**Database type**». Он позволяет выбрать один из 3 возможных вариантов расширения баз данных: `mysql`, `mysqli` или `pgsql`¹ (т.е. попросту, с какими базами данных может производиться работа). Данный список отображает лишь доступные (рабочие) расширения. Теперь перейдем к выбору. В нашем случае не рассматривается СУБД PostgreSQL, поэтому и вариант выбора расширения `pgsql` здесь также не рассматривается. Остаются два варианта: `mysql` и `mysqli`. Расширение `mysqli` является стандартной возможностью PHP5² – это новое³ «более быстрое» расширение, способное работать с новыми возможностями MySQL сервера 5-ой версии (храняемыми процедурами и функциями и т.д.), поэтому всегда нужно отдавать предпочтение именно этому расширению. Если же данное расширение недоступно, тогда вам придется остановиться на «устаревшем» `mysql`. К этому также стоит добавить, что Drupal 6 считает расширение `mysqli` более приоритетным в сравнении с `mysql`, поэтому когда оба расширения доступны, инсталлятор не показывает расширение `mysql`.

Поле «**Database name**». В данное поле необходимо вписать имя базы. В случае удаленной установке подобную базу вы уже создавали ранее посредством панели управления (DirectAdmin). На рисунке 16 показан отчет о создании, там же имеется и поле «Database:», рядом с которым и указано имя базы. В случае локальной установки имя базы вам будет достоверно известно.

Поле «**Database username**». В случае локальной установки это будет администратор MySQL сервера – «root» (для подключения к локальному серверу ранее мы не создавали отдельную учетную запись). В случае удаленной установки это будет значение поля «Username» из отчета о создании базы (пример – рисунок 16).

1 Существуют и другие варианты, например, возможна установка Drupal совместно с MSSQL, однако здесь такая возможность (как и любые другие) здесь не рассматривается.
 2 Установка расширения `mysqli` на PHP4 сопряжена с определенными сложностями, поэтому на 4-ую версию PHP практически никто не устанавливает это расширение.
 3 Относительно новое, т.к. ему на смену уже готовится следующее решение.

Поле «Database password». В случае локальной установки это будет пароль к учетной записи «root», т.е. пароль администратора MySQL сервера. В случае удаленной установки это будет значение поля «Password» из отчета о создании базы, пример – рисунок 16 (это поле видимое, но на рисунке вместо пароля стоит троеточие).

Поле «Database host». В случае локальной установки это будет «localhost». В случае удаленной установки это будет значение поля «Host» из отчета о создании базы, пример – рисунок 16 (большая вероятность, что это также будет «localhost»).

Поле «Database port». Если MySQL сервер занимает не порт по умолчанию (не порт 3306), тогда вам следует указать иной порт. Это поле практически никогда не меняется.

Поле «Table prefix». Вот этот пункт уже представляет интерес. Если вы укажете в данном поле какое-либо значение, тогда это значение будет прибавляться к именам каждой таблицы в базе, создаваемой данной копией системы Drupal. Это очень полезная возможность в случае, когда одна база используется различными системами (например, несколько копий Drupal работают с одной базой¹). Благодаря добавлению такого префикса пересечения в именах таблиц разных проектов будут отсутствовать. К размещению всех таблиц в одной базе обычно прибегают в случае, когда возможности (ресурсы) хостинга ограничены (например, данный тарифный план предоставляет только 1 базу и не более). Если вас интересует подобная возможность, тогда автор рекомендует указать в этом поле префикс (например «Drupal_» – после чего все таблицы, создаваемые данной копией системы будут именоваться «Drupal_имяТаблицы»).



Если инсталляция прошла успешно, то в результате вы должны в 5-ой версии увидеть сообщение «Drupal installation complete». В 6-ой же версии, после инсталляции, вы окажетесь в разделе «Configure site».

Однако это не все, сразу же после установки вы должны приступить к конфигурированию веб-проекта (созданию административного аккаунта), иначе пройденная инсталляция окажется бесполезной и тогда вам придется проводить повторную установку системы² (при этом удаляя все созданное ранее).

Установка Drupal: Создание административного аккаунта: Drupal 5

В 5 версии, после инсталляции, вы должны увидеть сообщение «Drupal installation complete», также внизу вы увидите ссылку «...to [your new site](#)», вам следует перейти по этой ссылке, после чего вы окажетесь на главной странице веб-проекта, на которой имеется еще одна ссылка «To begin, [create the first account...](#)», перейдите и по этой ссылке. После проделанных действий вы окажетесь на странице «[User account](#)».

На странице «[User account](#)» вы увидите 2 поля: [Username](#) и [E-mail address](#). Заполните их: в поле [Username](#) можно вписать любое допустимое имя (например, вы можете выбрать имя «root» или «admin»); в поле [E-mail address](#) вы можете указать адрес своего электронного почтового ящика. Далее вы должны нажать «Create new account», после чего система создаст новый аккаунт, в раздел настройки которого вы и будете затем перенаправлены.

Оказавшись в разделе настройки собственного аккаунта, вы должны задать новый пароль к своей учетной записи. Также вы увидите, что система уже сгенерировала для вас пароль – это сообщение «Your password is xxxxxxxxxx», если он устраивает вас, тогда менять пароль нет необходимости.

Теперь установку системы можно считать завершенной полностью.

Установка Drupal: Создание административного аккаунта: Drupal 6

В 6-ой версии конфигурирование происходит на последнем этапе инсталляции. Если все этапы ранее прошли без эксцессов, то сейчас вы должны оказаться на странице «Configure site».

На странице «Configure site» вам предстоит заполнить 5 полей и выбрать некоторые опции.

¹ Хотя для этих целей Drupal реализует возможность мультисайтинга.

² Конечно, в 6 версии можно попытаться инициализировать учетную запись «placeholder-for-uid-1», однако все эти действия являются неформальными – в итоге все это может привести к печальным последствиям.

Первых два поля – это поля Site name и Site e-mail address. В поле Site name вы должны указать название вашего веб-проекта. В поле Site e-mail address вы должны указать адрес электронного почтового ящика либо (свой либо специально для этого зарегистрированный).

Следующие поля: Username, E-mail address и Password. В поле Username можно вписать любое допустимое имя (например, вы можете выбрать имя «root» или «admin»); в поле E-mail address вы можете указать адрес своего электронного почтового ящика. В поле Password указывается пароль к создаваемой учетной записи.

И последнее – опции Default time zone, Clean URLs и Update notifications. Опция Default time zone позволяет указать часовой пояс, в котором вы находитесь. Опция Clean URLs, в отличие от Drupal 5, может быть определена прямо при установке. Если данная опция доступна (т.е. вы можете выбрать «Enabled»), тогда вы можете смело ее активизировать. Опция Update notifications активизирует извещение о появлении новых версий Drupal.

Теперь установку системы можно считать завершенной полностью.



2.3 Настройка

Раздел в разработке

2.3.1 Укрепление безопасности на начальном этапе

Раздел в разработке

2.3.2 Локализация

Теперь необходимо произвести локализацию системы, т.е. перевести ее интерфейс на понятный язык. Для этих целей в Drupal имеется специальный модуль **Locale**. Он уже имеется в системе в неактивном состоянии. Модуль Locale довольно простой, но, несмотря на это, вполне функциональный. Этот модуль позволяет локализовать систему двумя способами. Способ первый – это импорт готового перевода и способ второй – «ручной перевод». В данном случае автор кратко опишет первый способ (как самый эффективный в данном случае). Боле детальное обсуждение модуля Locale будет вестись в соответствующем разделе.

Итак, способ первый – импорт готового перевода. Для импорта перевода вам понадобится файл или файлы с переводом. Что из себя представляет любой из таких файлов? Внешне это обычный файл с расширением «po», например, файл русского перевода для Drupal **5** имеет название «ru drupal5x v2.po». Если вы откроете его текстовым редактором, то внутри вы обнаружите набор сопоставлений в формате «иностранная фраза» = «переведенная фраза»¹. Файлы перевода также отличаются по области действия – одни способны переводить всю систему целиком (весь перевод в одном файле), другие же предназначены для перевода каждого модуля в отдельности (их, конечно, можно использовать и для перевода всей системы в целом, но работы у вас тогда значительно прибавится – естественно, для того, чтобы не создавать себе дополнительных трудностей с большим количеством файлов, мы прибегнем к переводу системы «одним файлом»).

Далее скачайте файл перевода (найти его не составит особого труда – достаточно обратиться к ресурсам из раздела «Полезные источники»). Обычно перевод выполняет Вадим Барсуков [**Vadbars**]² (причем делает это довольно качественно). После того как вы нашли и закатали нужный вам файл, вы можете приступить к локализации всей системы. Ранее уже было отмечено, что по умолчанию модуль Locale не является активным, а локализацию начинают именно с него, поэтому далее автор опишет процесс его активизации.



Необходимо зайти в систему под Администратором, если раньше вы не сделали этого.



Затем перейдите в раздел *Administer* → *Site building* → *Modules*. Этому разделу будет соответствовать ссылка: <localhost/?q=admin/build/modules> **5****6**, где localhost это **localhost**, в случае когда вы устанавливаете систему локально и какое-либо иное (имя домена), при работе с проектом в глобальной сети.



После этого вы попадаете в раздел модулей системы. Найдите описание модуля Locale и поставьте галочку напротив его названия, затем нажмите кнопку «Save configuration».



Если все прошло успешно, то в меню должен появиться новый пункт или пункты. В **5** версии появиться один пункт – Localization. Он будет расположен по пути: *Administer* → *Site configuration* [<localhost/q=admin/settings/locale>], а для **6** версии их будет уже 2: Translate interface и Languages. Но из этих двух нам понадобится только первый пункт, располагающийся по пути: *Administer* → *Site building* [<localhost/?q=admin/build/translate>].

В **6** версии модуль локализации несколько изменил свое положение в меню. Теперь здесь вместо одного пункта Localization появилось целых два: Languages и Translate interface, причем находятся они в разных ветвях меню. Пункт Languages отвечает за «ручной перевод» языковых фраз (это второй способ локализации, упомянутый выше), а пункт Translate interface отвечает за импорт готовых переводов.

Теперь необходимо приступить к настройке модуля Locale. Переходим в раздел модуля Locale, т.е.

1 Естественно, в реальности формат несколько иной.
2 Один из участников проекта Drupal.ru

переходим по вышеуказанной ссылке (для разных версий системы они различные) и далее:



Переходим в раздел Import.



В поле Language file выбираем файл перевода (с расширением .po). В списке Import into выбираем язык перевода, другие опции оставляем как есть (по умолчанию).



Нажимаем Import и ждем выполнения операции «внедрения» нового языка – она может длиться от 15 секунд до целой минуты (в заметке ниже описывается этот момент подробнее). После успешного выполнения операции вы будете перенаправлены в раздел List. Здесь список языков должен пополниться новым, только что добавленным вами языком.



Теперь вам нужно активизировать свой язык. Выставьте переключатель Default в строку с вашим языком и нажмите Save Configuration.



После этого весь интерфейс системы управления должен стать для вас «родным».

Теперь о скорости. При установке процесс внедрения языка может затянуться настолько, что в браузере может появиться сообщение об ошибке: «Fatal error: Maximum execution time of 30 seconds exceeded in includes/database.mysql.inc». Так бывает, когда скрипт (в нашем случае, это сам Drupal) слишком долго выполняется. В файле `php.ini` (главном конфигурационном файле PHP) есть параметр `max_execution_time=30`, он то и задает максимальное время выполнения. Что делать, если произойдет такая ошибка? Для решения этой проблемы нет необходимости в изменении конфигурации PHP (т.е. в изменении непосредственно переменной `max_execution_time`), вполне достаточно просто обновить страницу в браузере (клавиша F5) – разработчики Drupal предусмотрели эту ситуацию и при повторной загрузке Drupal «доделывает» то, что не успел сделать в первый раз.

Есть возможность ускорить Drupal, если он использует в качестве хранилища данных (*data store*) СУБД MySQL. MySQL поддерживает разные типы таблиц. По умолчанию в нем используются таблицы типа InnoDB. Это медленные, но надежные таблицы. Вы можете изменить этот порядок – «выставить» по умолчанию другой тип, более быстрый, например, MyISAM. Благодаря этому вы можете получить отличный прирост производительности (особенно ощутимый на операциях записи), но в ущерб надежности (конечно же все относительно, впрочем, как и надежность, надежных InnoDB). Автор проделал небольшой эксперимент: сначала установил Drupal с таблицами типа InnoDB и провел описанную выше локализацию – по времени она прошла за 125 ± 3 сек., затем установил Drupal с таблицами типа MyISAM – локализация прошла за 14 ± 1 сек. – разница весьма ощутима. **Вывод:** если вы планируете создавать сугубо потребительский проект (например, проект по размещению объявлений и д.р.), то использование в нем менее надежных таблиц оправданно, т.к. потеря одного объявления или пользователя (при очень неудачном стечении обстоятельств) ничего не изменит, но в тоже время, если вы планируете создать проект, связанный с электронной коммерцией, тогда такое ускорение уже может обернуться для вас крупными неприятностями.

P.S. Изменить тип таблиц на локальном MySQL-сервере можно через MySQL Administrator¹. Изменения вступают в силу после перезагрузки. Однако такой способ не изменит тип ранее созданных таблиц. Изменить тип существующих таблиц можно выполнив определенные sql запросы.

После локализации могут остаться не переведенные строки. В переводе «ru drupal5x v2.po» находится 2523 выражения (в системе же их 2524) – одна фраза остается не переведенной. Вы можете попробовать самостоятельно ее перевести (в данном случае это хороший повод для того, чтобы рассмотреть эту возможность). Для перевода фразы выполните следующие действия:



В главном меню **5** версии пройдите следующий путь: Управление сайтом → Настройка → Переводы [localhost/?q=admin/settings/locale] (Обратите внимание, что имена пунктов меню уже на русском – в дальнейшем все пути будут именно на русском и только иногда на английском).



Далее необходимо перейти в раздел Управлять строками². Теперь в группе переключателей «Поиск в:» сделайте активным пункт «Только не переведенные строки», после чего нажмите Поиск.

¹ Он был описан в приложении Б «Дистрибутивы»

² Если бы вы решились изначально вводить все выражения самостоятельно, то после добавления аббревиатуры языка вы посетили бы именно это раздел. Однако нам не нужно вводить все строки, нам лишь нужно изменить одну переведенную.



В результате должна появиться одна строка (немного выше). Ее следует изменить – пройдите по соответствующей ссылке. Теперь в разделе «Изменить строку» в поле «Russian:» впишите ее перевод, а затем нажмите «Сохранить переводы». Вот и все. Подобным образом вы в дальнейшем самостоятельно сможете перевести не переведенное выражение или же изменить переведенное.

2.3.3 Общая информация о проекте

Раздел в разработке

2.3.4 Установка даты и времени

Раздел в разработке

2.3.5 Выбор административной и общей темы (оформления)

Раздел в разработке

Включение темы

Раздел в разработке

Общие настройки

Раздел в разработке

Настройка тем *bluemarine / chameleon / garland / marvin / minnelli*

Раздел в разработке

2.3.6 Настройка производительности

Раздел в разработке

Кэш страницы

Раздел в разработке

Оптимизация пропускной способности

Раздел в разработке

2.3.7 Загружаемые изображения

Раздел в разработке

2.3.8 Файловая система

Раздел в разработке

2.3.9 Форматы ввода

Раздел в разработке

Создание нового

Раздел в разработке

Редактирование существующего

Раздел в разработке

Удаление

Раздел в разработке

2.3.10 Чистые ссылки

Раздел в разработке

2.3.11 Настройка системы оповещения об ошибках

Раздел в разработке


2.3.12 Закрыто...

Раздел в разработке



2.4 Поддержка

2.4.1 Обновление системы

Даже не смотря на то, что система Drupal бесконечно совершенствуется, в ней периодически находят ошибки. Последнее вовсе не означает, что разработчики не уделяли должного внимания своей работе, просто невозможно учесть абсолютно все варианты. Идеальных систем в реальном мире не существует (во всяком случае автору с такими системами не довелось встретиться). Любые системы, в том числе и коммерческие, содержат неизвестное количество потенциальных ошибок, и N-ое количество уже выявленных¹. Каждая погрешность в системе в свою очередь может влиять на ее надежность и/или ее защищенность. К системе Drupal своевременно выпускаются заплатки (патчи )², а с накоплением N-ого количества заплаток (патчей), либо после нахождения существенной уязвимости, выходит и очередная версия системы. В принципе, все вышесказанное справедливо и для модулей Drupal.

Как часто следует производить обновление непосредственно самой системы? На этот вопрос можно ответить довольно просто – это следует делать с каждым новым выходом патча, либо с каждым новым выходом версии системы (или версии модуля). Однако осуществлять обновления по такому графику довольно сложно, т.к. частота выхода патчей высока, а их «важность» («критичность») порой незначительна, поэтому многие ограничиваются лишь обновлением «с выходом новой системы» (т.е. фактически после выявления критической уязвимости).

Когда не следует проводить обновление? Есть случаи когда обновление может привести к краху. Иногда разработчики совершают грубейшую ошибку – они собственноручно вносят правки в ядро системы. Обновление такой «модифицированной» системы может привести к ее полному «развалу». Подобные ситуации обычно разрешаются следующим образом: разработчики (интеграторы) записывают свои изменения базовой системы в виде патчей, а при каждом обновлении системы эти патчи «накладываются поверх» обновленной системы (в результате все изменения, сделанные разработчиками веб-проекта на этапе его создания, снова «возвращаются», но уже в новую версию системы).² Также имеет немаловажное значение и файловая структура вашего проекта. Если вы не владеете полной информацией о том, содержатся ли в вашем веб-проекте дополнительные файлы (предполагается, что дополнительные файлы могли вноситься разработчиками веб-проекта), тогда при обновлении вы можете нарушить что либо³. Если же вы являетесь владельцем «правильного» Drupal-проекта (т.е. выполненного в рамках api, концепций и правил системы Drupal, а также, без хаков, модификаций ядра и т.д.) тогда обновление у вас пройдет легко и быстро.

Для обновления системы Drupal вам необходим последний дистрибутив CMS/CMF Drupal и новые версии дополнительных модулей⁴ (сторонних⁵ или пользовательских⁶). Взять последний дистрибутив системы Drupal вы можете с официального сайта этого проекта (<http://drupal.org>). Посмотреть наличие новых версий модулей и загрузить новые модули вы можете там же, посетив раздел Modules.

Ниже приводится переработанная инструкция по обновлению системы Drupal из файла UPGRADE.txt (он имеется в дистрибутиве системы). В действительности, эта инструкция не столь «прозрачна» как того хотелось бы (тех, кто не выполнял обновление ранее, она может разозлить своей пунктуальностью⁷).

- 1 Причем каждое нововведение лишь увеличивает количество потенциальных ошибок.
- 2 Конечно такой подход – не более чем «костыли» для такой системы (патчи будут «накладываться правильно» только в том случае, если фрагмент кода, который предполагается пропатчить, остался прежним и в новой версии).
- 3 Хорошим примером дополнительных файлов может стать, например, файл-пустышка типа google_n*.html, который необходим Google для подтверждения того, что используемый домен является действительно вашим (например при использовании сервиса Google Analytics).
- 4 Если вы используете дополнительные модули, хотя вряд ли можно найти проекты, в которых бы они не использовались.
- 5 Сторонние модули – здесь модули которые не входят в основной дистрибутив системы Drupal. К ним могут относиться как модули с проекта Drupal.org, так и модули с других проектов. Предполагается, что на эти модули распространяется поддержка (условно), а также осуществляется выпуск новых версий этих модулей.
- 6 К пользовательским модулям здесь сносятся модули «собственного производства», либо сторонние модифицированные модули. Предполагается, что на пользовательские модули не распространяется поддержка, а также не осуществляется выпуск новых версий таких модулей.
- 7 Всегда хочется, чтобы обновление проходило в автоматическом режиме (без дополнительных трудозатрат).

Существует также более простой но экстремальный способ обновления – это простая замена файлов (в большинстве случаев такое обновление проходит успешно, т.к. новый дистрибутив зачастую отличается от старого лишь парой строчек кода). Но этим способом можно пользоваться только, если вы владеете необходимой информацией – к примеру требуется точно знать, что обновлять базу не потребуются. И все-равно такой способ лучше не применять на практике – обновление, выполненное не за 5 минут, а за 15 хоть и затратит чуть больше вашего времени, зато «пройдет» без последствий.

Получить дополнительную информацию о процессе обновления вы можете из книги «Drupal handbook» – адрес: <http://drupal.org/upgrade> (английская версия).



Обновление Drupal 5

Процесс обновления по инструкции выполняется следующим образом:

Шаг 1-ый:

Сделайте полную резервную копию базы данных и файлов обновляемого веб-проекта, в особенности копию директории `Sites` В `sites` размещаются конфигурационные файлы вашей системы [ф-путь: «`sites/default`»]. Также в `sites` могут размещаться дополнительные модули [ф-путь: «`sites/all/modules`»] и дополнительные темы [ф-путь: «`sites/all/themes`»]¹. Если же вы разместили дополнительные темы непосредственно в конечной директории `themes`, а дополнительные модули в корневой директории `modules`, тогда вам стоит переместить их согласно рекомендациям системы Drupal. Также следует сделать резервную копию директории `files`. (в ней могут находиться файлы, загруженные вами ранее в систему, либо сгенерированные непосредственно системой). По умолчанию `files` располагается в корневой директории, хотя может быть размещен и по любому другому пути.² Также, если вы модифицировали файлы «`.htaccess`»³ и «`robots.txt`»⁴ (они располагаются в корне), их тоже следует сохранить. В последующем, все дополнительное содержание, упомянутое выше, необходимо будет «вернуть» в уже новую систему. В общем случае сделайте полную резервную копию всего веб-проекта.



В случае простого веб-проекта (не мультисайтинг-проекта) конфигурационным файлом является файл `settings.php` [ф-путь: `sites/default/settings.php`]. Файл `default.settings.php` является чистым образцом файла конфигурации и используется для восстановления испорченного конфигурационного файла. При использовании мультисайтинга, файл конфигурации может располагаться в следующих директориях:

[ф-путь: `sites/default/settings.php`]
 [ф-путь: `sites/example.com/settings.php`]
 [ф-путь: `sites/sub.example.com/settings.php`]
 [ф-путь: `sites/sub.example.com.path/settings.php`]

Получить дополнительную информацию о мультисайтинге вы можете из файла `INSTALL.txt`.



Шаг 2-ой:

Войдете в систему под основным административным аккаунтом (т.е. под аккаунтом, ID которого равен 1⁵). Аккаунт с ID 1 – это первый создаваемый аккаунт в системе Drupal (при установке), который также является основным административным аккаунтом. Только пользователь с ID равным 1 сможет получить доступ к файлу `update.php` (на 9-ом шаге). Не закрывайте окно браузера до тех пор, пока 10-ый шаг не завершится.

- 1 В случае использования мультисайтинга, вместо `All` может быть задействовано иное имя.
- 2 При использовании приватного метода загрузки, `files` могут размещать вне веб-директорий. В настройках системы вы можете самостоятельно задать путь к `files` [url: [admin/settings/file-system](#)].
- 3 `Htaccess` обычно редактируют, когда в него необходимо включить параметры, необходимые для работы на удаленном сервере, т.е. под определенный хостинг.
- 4 `Robots.txt` редактируют довольно часто – с помощью этого файла можно «настроить» работу поисковых роботов.
- 5 Этому аккаунту соответствует путь [user/1](#)

Шаг 3-ий:
 Переключите ваш проект в режим «Off-line» («На обслуживании»), для того чтобы скрыть от посетителей последствия процесса обновления (ошибки и т.д.), а также для того, чтобы обновление базы системы проходило без «обрыва». Для Drupal **5** такое переключение можно выполнить в разделе *Управление сайтом* → *Настройка* → *Обслуживание* [url: admin/settings/site-maintenance]. После того, как сайт будет переведен в «неактивный» режим, доступ к нему ограничивается, а его внешний вид координально изменится – вместо привычного содержания, посетитель теперь увидит сообщение «Сайт на техническом обслуживании»⁶.

Шаг 4-ый:
 Если вы используете дополнительные модули, тогда перейдите в раздел «Модули» [url: admin/build/modules] и отключите все дополнительные (сторонние или пользовательские) модули. Если вы используете дополнительные темы (не **Garland** или **Bluemarine**), тогда перейдите в раздел «Оформление», [url: admin/build/themes] и переключитесь на тему Garland или Bluemarine.

Шаг 5-ый:
 Удалите все старые файлы и каталоги из установочной директории (главной директории веб-проекта). В действительности нужно удалить следующие файлы: временные или ненужные файлы, файлы старого дистрибутива (например старые файлы в includes и т.д.). Если вы не понимаете какие файлы можно удалять, а какие нет, тогда лучше воздержаться от удаления.

Разработчики системы Drupal не могут знать, что именно располагается в вашей веб-директории и если следовать их рекомендациям, то можно легко удалить нужные файлы, которые были размещены там веб-разработчиками.

Шаг 6-ый:
 Распакуйте новые файлы дистрибутива Drupal в директорию установки. При этом, если на шаге 5 вы не производили удаление файлов, тогда новые файлы должны заменить старые, т.е. вы должны скопировать новые файлы поверх имеющихся.

Шаг 7-ой:
 После этого восстановите ранее зарезервированные (на 1-ом шаге) директории files и sites, т.е. скопируйте их, в каталог установки системы. Если ранее вы модифицировали файлы «.htaccess» или «robots.txt», их также необходимо восстановить (но лучше все же их пересоздать используя новые версии, т.е. внести необходимые правки в новые версии, а не заменить старые файлы новыми)². Проверьте конфигурационный файл, чтобы быть уверенным в том, что он содержит корректную и актуальную информацию (как правило, в типичных проектах таковым является файл settings.php)³.

Шаг 8-ой:
 Далее переустановите дополнительные модули. Выше говорилось о том, что вам следует проверить имеются ли новые версии сторонних модулей. Убедитесь в том, что новые версии модулей совпадают с новой версией Drupal [url: <http://drupal.org/project/Modules>]. Модули предыдущих версий могут быть не совместимы с новой версией системы. Пользовательские модули (т.е. несвязанные с проектом drupal.org) возможно придется конвертировать⁴, более подробно об этом можно узнать по адресу <http://drupal.org/update/modules>.

6 При этом доступ к сайту теперь могут получить лишь те, у кого имеется право «Управление настройками сайта», а для того чтобы пользователь с таким правом смог войти на сайт, ему необходимо в строке адреса браузера набрать «<http://localhost/?q=user>» – войти классическим способом он не сможет – форма входа (на главной странице), в таком режиме не отображается.

2 Для сравнения содержания файлов можно использовать редактор Notepad++.

3 В данном файле находится переменная \$db_url, ее следует проверить в первую очередь.

4 Под конвертированием в данном случае понимается корректировка кода модулей. Если у вас имеются собственные модули, тогда вам придется проверить, нуждаются ли последние в корректировке. К примеру, если вы проводите переход с версии 6.0 или 6.1 на версию 6.2, вам придется проверить работу hook_menu в этих модулях.

**Шаг 9-ый:**

Запустите файл `update.php`, посетив адрес http://путь_к_веб_проекту/update.php. Этот шаг должен обновить базу данных (ниже представлен раздел «Обновление базы Drupal 5» – данный раздел актуален именно сейчас¹). Если вы не можете получить доступ к файлу `update.php`, сделайте следующее: откройте файл `update.php` в текстовом редакторе², в начале этого файла найдите строку `$_access_check = TRUE;` и замените ее строкой `$_access_check = FALSE;`. Попробуйте снова запустить `update.php`. После того, как скрипт выполнится вы должны вернуть `update.php` к прежнему (оригинальному) состоянию – измените «FALSE» на «TRUE» на той же строке (выполнить «возврат» необходимо будет после завершения всех шагов, в том числе не раньше того момента, когда будет выполнено обновления базы).

**Шаг 10-ый:**

В завершении переведите проект в рабочее состояние, т.е. отключите режим «Off-line», включите необходимое оформление (тему) – теперь ваши посетители смогут продолжить работу с проектом.

1 Процесс обновления базы был вынесен в отдельный раздел в связи с его громоздкостью.
2 На удаленном сервере это действие можно выполнить при помощи соответствующей функции панели управления хостингом.

Обновление Drupal 6

Перед обновлением системы вы должны выполнить следующие действия:

- проверить, удовлетворяет ли среда исполнения (Apache, PHP, MySQL и т.д.) минимальным требованиям системы Drupal [url: <http://drupal.org/requirements>];
- проверить, имеются ли у вас релевантные резервные копии всех данных (шаг 1-ый);
- проверить совместимость собственных и сторонних модулей (шаг 9-ый);
- проверить совместимость собственных и сторонних тем (шаг 10-ый);
- прочесть заранее всю нижеизложенную последовательность выполняемых действий.

Процесс обновления по инструкции выполняется следующим образом:



Шаг 1-ый:

Является аналогичным шагу 1 раздела «Обновление Drupal 5». Обратитесь к упомянутому разделу для получения информации.



Шаг 2-ой:

Если возможно, войдете в систему под основным административным аккаунтом (т.е. под аккаунтом, ID которого равен 1¹). Аккаунт с ID 1 – это первый создаваемый аккаунт в системе Drupal (при установке), который также является основным административным аккаунтом. Пользователь с ID равным 1 сможет автоматически получить доступ к файлу `update.php` (на 8-ом шаге). На 8-ом шаге содержатся специальные инструкции для тех, кто не имеет возможности войти в аккаунт с ID равным 1. Не закрывайте окно браузера до тех пор, пока последний шаг не завершится.



Шаг 3-ий:

Переключите ваш проект в режим «Off-line» («На обслуживании»), для того чтобы скрыть от посетителей последствия процесса обновления (ошибки и т.д.), а также для того, чтобы обновление базы системы проходило без «обрыва». Для Drupal 6 такое переключение можно выполнить в разделе *Управление* → *Настройка сайта* → *Сайт на обслуживании* [url: <admin/settings/site-maintenance>]. После того, как сайт будет переведен в «неактивный» режим, доступ к нему ограничивается, а его внешний вид координально изменится – вместо привычного содержания, посетитель теперь увидит сообщение «Сайт на техническом обслуживании».



Шаг 4-ый:

Если вы используете дополнительные темы (не **Garland** или **Bluemarine**), тогда перейдите в раздел «Темы оформления», [url: <admin/build/themes>] и переключитесь на тему **Garland** или **Bluemarine**. Если вы используете дополнительные модули (сторонние или пользовательские), тогда перейдите в раздел «Модули» [url: <admin/build/modules>] и отключите их.



Шаг 5-ый:

Удалите все старые файлы и каталоги из установочной директории (главной директории веб-проекта). В действительности нужно удалить следующие файлы: временные или ненужные файлы, файлы старого дистрибутива (например старые файлы в includes и т.д.). Если вы не понимаете какие файлы можно удалять, а какие нет, тогда лучше воздержаться от удаления.

Разработчики системы Drupal не могут знать, что именно располагается в вашей веб-директории и если следовать их рекомендациям, то можно легко удалить нужные файлы, которые были размещены там веб-разработчиками.



Шаг 6-ый:

Распакуйте новые файлы дистрибутива Drupal в директорию установки. При этом, если на шаге 5 вы не производили удаление файлов, тогда новые файлы должны заменить старые, т.е. вы должны скопировать новые файлы поверх имеющихся.

1 Этому аккаунту соответствует url <user/1>

**Шаг 7-ой:**

После этого восстановите ранее зарезервированные (на 1-ом шаге) директории `files` и `sites`, т.е. скопируйте их, в каталог установки системы. Если ранее вы модифицировали файлы «`.htaccess`» или «`robots.txt`», их также необходимо восстановить (но лучше все же их пересоздать используя новые версии, т.е. внести необходимые правки в новые версии, а не заменить старые файлы новыми)¹. Проверьте конфигурационный файл, чтобы быть уверенным в том, что он содержит корректную и актуальную информацию (как правило, в типичных проектах таковым является файл `settings.php`)².

**Шаг 8-ой:**

Запустите файл `update.php`, посетив адрес http://путь_к_веб_проекту/update.php. Этот шаг должен обновить базу данных (ниже представлен раздел «Обновление базы Drupal 6» – данный раздел актуален именно сейчас³). Если вы не можете получить доступ к файлу `update.php`, проделайте следующее: откройте файл `settings.php` в текстовом редакторе⁴, в середине этого файла найдите строку «`$update_free_access = FALSE;`» и замените ее строкой «`$update_free_access = TRUE;`». Попробуйте снова запустить `update.php`. После того, как скрипт выполнится вы должны вернуть `settings.php` к прежнему (оригинальному) состоянию – измените «TRUE» на «FALSE» на той же строке (выполнить «возврат» необходимо будет после завершения всех шагов, в том числе не раньше того момента, когда будет выполнено обновления базы).

**Шаг 9-ый:**

Далее переустановите дополнительные модули. Выше говорилось о том, что вам следует проверить имеются ли новые версии сторонних модулей. Убедитесь в том, что новые версии модулей совпадают с новой версией Drupal [url: <http://drupal.org/project/Modules>]. Модули предыдущих версий могут быть не совместимы с новой версией системы. Пользовательские модули возможно придется конвертировать⁵, более подробно об этом можно узнать по адресу <http://drupal.org/update/modules>.

**Шаг 10-ый:**

Включите необходимое оформление (тему). Если вы используете дополнительные темы, вам следует убедиться в их совместимости с новой версией системы. Дополнительную информацию о совместимости сторонних тем вы можете получить по адресу <http://drupal.org/project/themes>, пользовательских тем – <http://drupal.org/update/theme>.

**Шаг 11-ый:**

В завершении переведите проект в рабочее состояние, т.е. отключите режим «Off-line» – теперь ваши посетители смогут продолжить работу с проектом.

1 Для сравнения содержания файлов можно использовать редактор Notepad++.

2 В данном файле находится переменная `$db_url`, ее следует проверить в первую очередь.

3 Процесс обновления базы был вынесен в отдельный раздел в связи с его громоздкостью.

4 На удаленном сервере это действие можно выполнить при помощи соответствующей функции панели управления хостингом.

5 Под конвертированием в данном случае понимается корректировка кода модулей. Если у вас имеются собственные модули, тогда вам придется проверить, нуждаются ли последние в корректировке. К примеру, если вы проводите переход с версии 6.0 или 6.1 на версию 6.2, вам придется проверить работу `hook_menu` в них.

Обновление базы Drupal 5

Данное обновление является частью раздела «Обновление Drupal 5». Инструкции этого раздела нельзя использовать отдельно от основного обновления!

Для того чтобы произвести обновление базы системы, необходимо запустите файл `update.php` [url: [update.php](#)], после чего перейти по ссылке [run the database upgrade script](#). Далее, необходимо развернуть список «Select versions» и убедиться, что в нем присутствует хотя бы одно доступное обновление (пример на рисунке 21).

The version of Drupal you are updating from has been automatically detected. You can select a different version, but you should not need to.
Click Update to start the update process.

▼ [Select versions](#)

system module:
1022

comment module:
No updates available

Update

Рис. 21: Обновление базы Drupal 5: доступные обновления.

Если обновление имеется, тогда можно приступить к обновлению базы: для этого достаточно нажать на кнопку «Update». В процессе обновления вы будете наблюдать (не долго) индикатор обновления, изображенный на рисунке ниже.

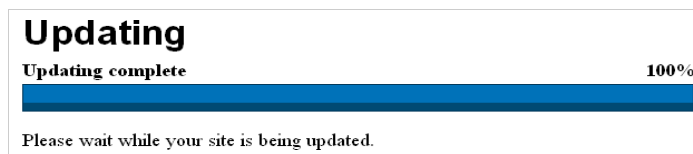


Рис. 22: Обновление базы Drupal 5: индикатор процесса обновления.

После обновления вам будет представлен отчет о «проделанной работе» (пример на рисунке 23)

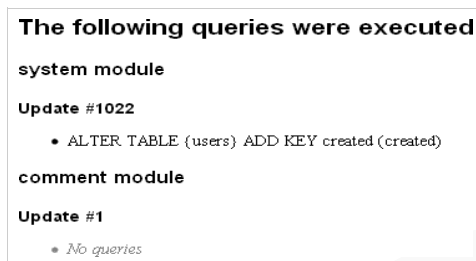


Рис. 23: Обновление базы Drupal 5: отчет.

Теперь вернитесь в раздел *Отчет о состоянии* и убедитесь, что обновление прошло успешно (пример на рисунке 24).

Существует также еще один способ определения необходимости обновления базы. Для того чтобы понять необходимо ли вам обновлять базу системы или нет, достаточно зайти в раздел *Управление сайтом* → *Системные журналы* → *Отчет о состоянии* [url: [admin/logs/status](#)] и посмотреть на отчет «Database schema». Если обновление базы не требуется, тогда отчет примет вид рисунка 24, если же требуется, тогда 25.

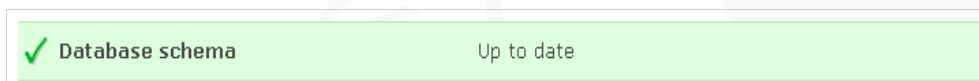


Рис. 24: Обновление базы Drupal 5: обновление базы не требуется.

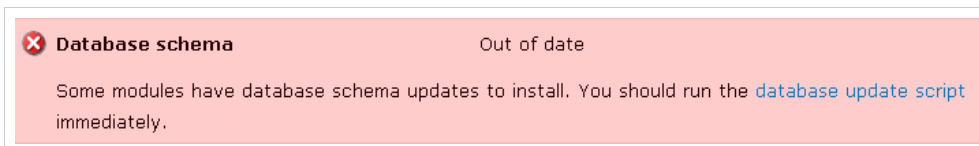


Рис. 25: Обновление базы Drupal 5: требуется обновление базы.

Однако нет никаких гарантий, что после первой стадии обновления (файловой) система сможет функционировать нормально, поэтому вышеописанный отчет вы можете так и не увидеть, вместо этого, в браузере вы можете получить «белый экран» с сообщением об ошибке. Так обычно получается, когда в новой версии изменилась одна или несколько системных таблиц.

Обновление базы Drupal 6

Данное обновление является частью раздела «Обновление Drupal 6». Инструкции этого раздела нельзя использовать отдельно от основного обновления!

Для того чтобы произвести обновление базы системы, необходимо запустите файл `update.php` [[url: update.php](#)], после чего пройти 4 стадии процесса обновления.

На первой стадии обновления вам предстоит прочитать небольшое введение (изображено на рисунке 26). Далее приводиться его перевод:

«Используйте эту утилиту для обновления базы данных, когда установлен новый релиз Drupal или новый модуль. Более подробную информацию смотрите в разделе справочника «Установка и обновление» [[url: http://drupal.org/node/258](http://drupal.org/node/258)]. Если вы не понимаете, что означают эти термины, вы, вероятно, должны, связаться с вашим хостинг-провайдером. 1) Сделайте резервное копирование базы данных. Этот процесс изменяет значения в вашей базе и в случае возникновения проблем вам потребуются резервная копия для восстановления. 2) Сделайте резервное копирование кода. Замечание: когда вы сделаете резервную копию модулей, не оставляйте эту копию в директории `modules` или `sites//modules`, поскольку она может запутать auto-discovery механизм системы Drupal. 3) Переведите ваш сайт в режим Off-line. 4) Разместите новые файлы по соответствующим локациям, как это описано в руководстве. Если вы выполнили все вышеописанные действия, вы можете продолжить.»*

По существу ни одно предложение из приведенного введения не нуждается в комментариях. Все вышеописанные рекомендации вы должны были выполнить на начальной стадии, поэтому на этой стадии вам лишь остается нажать кнопку «Continue».

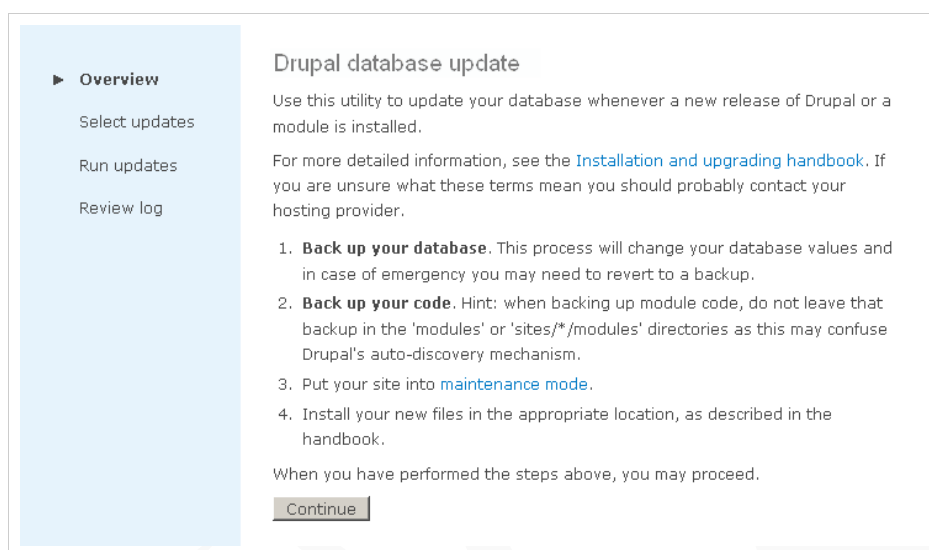


Рис. 26: Обновление базы Drupal 6: стадия 1.

На второй стадии предстоит выбрать версию обновления, однако здесь вам также не придется принимать каких либо решений т.к. система автоматически сделает «сверку версий» и определит (укажет), обновление каких частей требует система. В блоке «Select versions» (если вы его развер-

нете), вы можете увидеть список модулей участвующих в обновлении (если обновление не требуется, тогда в списке будет присутствовать значение «No updates available»). Ни одно значение из этих списков не следует изменять. Вам необходимо просто нажать на кнопку «Update». На рисунке 27 показан примерный вид второй стадии обновления.

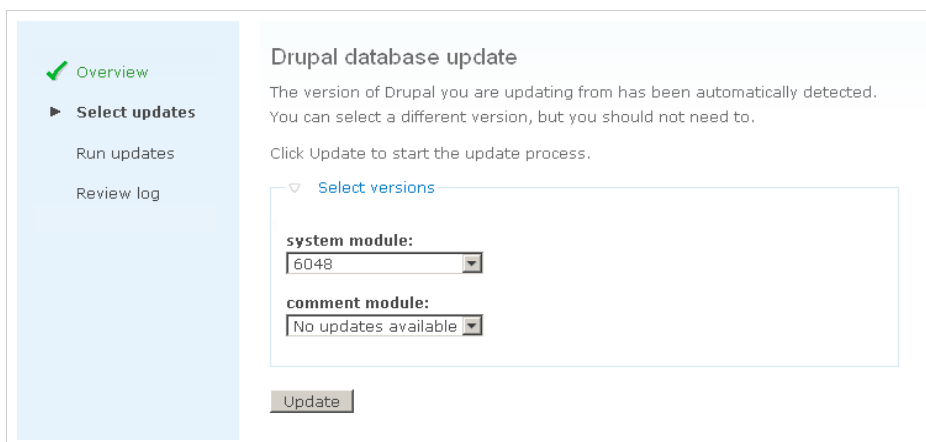


Рис. 27: Обновление базы Drupal 6: стадия 2.

На третьей стадии система произведет обновление базы (рисунок 28).

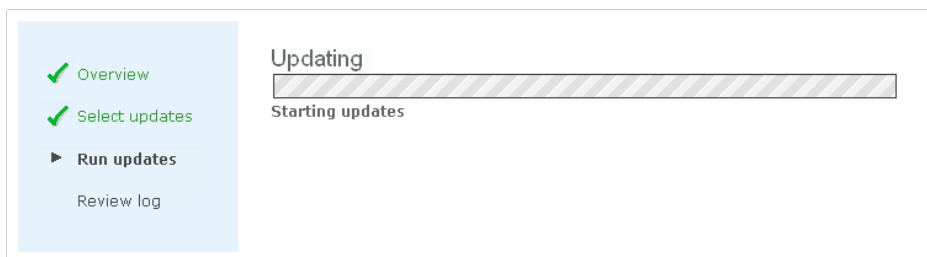


Рис. 28: Обновление базы Drupal 6: стадия 3.

После обновления, на последней, четвертой стадии, вам будет предоставлен отчет о «проделанной работе». Если все прошло успешно, тогда вы можете приступить к работе (а точнее вернуться к 9-ому шагу раздела обновления). На рисунке 29 изображен «положительный» отчет (пример обновления с Drupal 6.1 до Drupal 6.9). Если возникли ошибки, вы будете проинформированы о них здесь же. Все ошибки регистрируются. В случае возникновения серьезных ошибок вам придется прибегнуть к ручному обновлению базы.

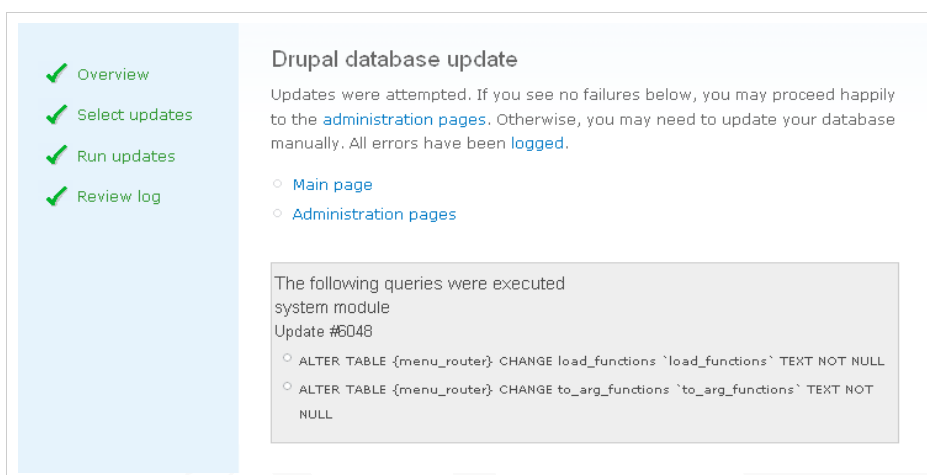


Рис. 29: Обновление базы Drupal 6: стадия 4.

После обновления зайдите в раздел *Управление* → *Отчеты* → *Отчет о состоянии* [url: admin/reports/status] и убедитесь в том, что обновление базы прошло успешно – посмотрите на отчет «Database schema». Если оно прошло успешно, тогда отчет примет вид рисунка 30, если нет, тогда рисунка 31.

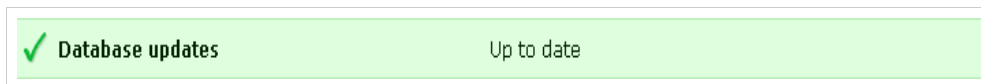


Рис. 30: Обновление базы Drupal 6: обновление базы не требуется.

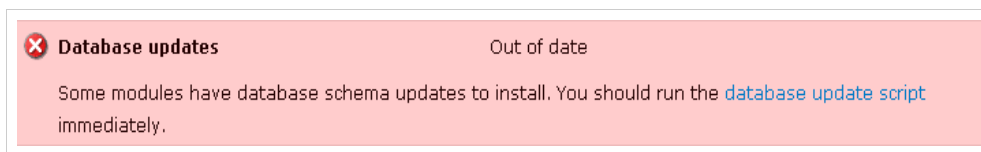


Рис. 31: Обновление базы Drupal 6: требуется обновление базы.

Действия после обновления

После обновления автор также рекомендует очистить кэш и запустить вручную `cron`, после чего «пройтись» по всем страницам проекта (если это возможно), дабы убедиться в том, что обновление прошло успешно.

В Drupal 6 очистить кэш можно используя интерфейс системы, для этого достаточно зайти в раздел *Управление сайтом* → *Настройка сайта* → *Производительность* и нажать кнопку «Очистить кэш данных». В Drupal 5 нет такой кнопки, поэтому очистить кэш здесь можно либо задействуя сторонние модули, либо сделав это вручную, очистив соответствующие таблицы (к примеру, через phpMyAdmin).

Запустить `cron` можно используя интерфейс системы. Для этого достаточно кликнуть по ссылке «запустить выполнение регулярных процедур» в разделе *Управление сайтом* → *Системные журналы* → *Отчет о состоянии* для Drupal 5 [url: admin/logs/status] или в разделе *Управление* → *Отчеты* → *Отчет о состоянии* для Drupal 6 [url: admin/reports/status].

Вовремя реагировать...

Узнать обо всех обновлениях системы Drupal и ее модулей вы можете в разделе **Security** официального сайта проекта Drupal [url: drupal.org/security].

Также этот раздел вы можете получить в виде новостей если подпишетесь на них по адресу <http://drupal.org/security/rss.xml>.

Для того, чтобы получать уведомления о новых обновлениях по электронной почте, вам необходимо зарегистрироваться на проекте drupal.org, после чего в своем пользовательском профиле (ссылка *My account*), в разделе *Edit*, подразделе *My newsletters*, поставить галочку напротив *Security announcements*. После этого проект drupal.org будет присылать вам письма в следующем формате:

```

--- SA-CONTRIB-2009-007 - ADVERTISEMENT CROSS-SITE SCRIPTING ---
* Advisory ID: DRUPAL-SA-CONTRIB-2009-007
* Project: Advertisement module (third-party module)
* Versions: 5.x, 6.x
* Date: 2009 February 11
* Security risk: Less critical
* Exploitable from: Remote
* Vulnerability: Cross-site scripting (XSS)
  
```

где в поле *Project* будет указываться имя обновляемого проекта (к примеру, имя модуля), в поле *Versions* – версия, а в поле *Security risk* – степень уязвимости. Если в своем веб-проекте вы не используете такого модуля такой версии, тогда вам не о чем волноваться.

2.5 Пользователи и доступ

Раздел в разработке

2.5.1 Работа с учетными записями пользователей

Раздел в разработке

Создание нового пользователя

Раздел в разработке

Удаление существующего пользователя

Раздел в разработке

Блокировка пользователя

Раздел в разработке

Пароль: подбор, смена

Раздел в разработке

«Назначение ролей» как способ глобального контроля

Раздел в разработке

2.5.2 Роли

Раздел в разработке

Создание ролей

Раздел в разработке

Удаление ролей

Раздел в разработке

Блокировка ролей

Раздел в разработке

Роли и их возможности – контроль доступа

Раздел в разработке

2.5.3 Правила доступа

Раздел в разработке

2.5.4 Глобальные пользовательские установки

Раздел в разработке

Способы регистрации

Раздел в разработке

Параметры электронной почты

Раздел в разработке

Параметры загружаемых изображений

Раздел в разработке



2.6 Системные журналы (отчеты)

Система Drupal, как «ответственная» система, ведет журналирование своей деятельности. В специальные таблицы (в базе данных) система заносит информацию о своей работе: информацию о выполненных действиях, информацию о возникших ошибках и т.д. Пользователям же информация из этих журналов предоставляется системой в виде отчетов, просмотреть которые он может в разделе *Управление сайтом* → *Системные журналы* [url: [admin/logs](#)] в 5-ой версии системы и в разделе *Управление* → *Отчеты* [url: [admin/reports](#)] в 6-ой версии системы.

Качество и количество отчетов зависит от версии системы, а также от количества установленных в системе модулей. Кроме отчетов система также информирует пользователя с помощью сообщений: сообщений об ошибках, уведомительных сообщений и т.д. Следует позаботиться о том, чтобы сообщения об ошибках не показывались посетителям вашего веб-проекта, т.к. это может сказаться на его безопасности. Для этого вам необходимо зайти в раздел *Управление сайтом* → *Настройка* → *Об ошибках* [url: [admin/settings/error-reporting](#)] в 5-ой версии системы и в раздел *Управление* → *Настройка сайта* → *Сообщения об ошибках* [url: [admin/settings/error-reporting](#)] в 6-ой версии системы, где установить поле «Об ошибках» или «Сообщения об ошибках» в положение «Записывать ошибки в системный журнал» (т.е. не выводить на экран). Вам не стоит беспокоиться о подобном «безмолвии» системы – все сообщения будут всегда заноситься в системный журнал, и вы их никогда не пропустите, если будете регулярно просматривать отчеты.

Важно регулярно просматривать отчеты системы, чтобы успевать своевременно реагировать на определенные ситуации. !

Также следует настроить «актуальность» отчетов. В Drupal 5 можно указать как долго следует хранить информацию в журнале. В Drupal 6 эта опция выполнена более грамотно – здесь нет привязки ко времени, вместо этого система руководствуется количественным показателем – сколько записей следует хранить. Для того чтобы задать «параметр актуальности», перейдите в раздел *Управление сайтом* → *Настройка* → *Об ошибках* [url: [admin/settings/error-reporting](#)] в 5-ой версии системы и задайте требуемое значение для поля «Очищать записи в системном журнале старше, чем...» или перейдите в раздел *Управление* → *Настройка сайта* → *Отчеты и сообщения* → *Отчеты в базе данных* [url: [admin/settings/logging/dblog](#)] в 6-ой версии системы и задайте требуемое значение для поля «Удалять старые записи из журнала, если их становится больше, чем...». После этого все «старые» записи будут удаляться автоматически при запуске cron.

2.6.1 Отчет «Последние записи (события)»

Раздел в разработке

2.6.2 Отчет «Ошибки обращения»

Раздел в разработке

2.6.3 Отчет «Отказ в доступе»

Раздел в разработке

2.6.4 Отчет «О состоянии...»

Раздел в разработке

2.6.5 Типичные ошибки

Не одна ошибка не возникает беспричинно. Одни ошибки могут являться следствием «неправильной» конфигурации системы Drupal либо его функциональной базы (Apache, PHP, MySQL), другие могут быть следствием нарушения целостности данных и т.д.

Нарушение целостности данных – самая коварная ошибка. Она встречается очень редко. Эту ошибку сложно идентифицировать, т.к. она в своем большинстве, оказывает лишь косвенное влияние на работу всей системы в целом. Эта ошибка возникает при определенных обстоятельствах и повторить ее не так то просто.

Версия системы: 5**Тип: PHP****Сообщение:**

```
Duplicate entry '119' for key 1 query: INSERT INTO users (name, mail, pass, status, init, uid, created, access) VALUES ('user', 'user@localhost', '202cb962ac59075b964b07152d234b70', 1, 'user@localhost', 119, 1235397273, 1235397273) в файле /includes/database.mysql.inc в строке 156.
```

Описание:

Возникает, когда в таблицу, имеющую ключевое поле, производится вставка строки с не уникальным значением ключевого поля. Значения в ключевых полях должны быть уникальными (например, 1, 2, 3, 4, 5 и т.д.) – ключевые поле не могут иметь 2 одинаковых значения (например 1, 2, 3, 3 и т.д.). Ключевые поля используются для хранения уникальных идентификаторов. В примере выше производилась добавление нового пользователя, а поскольку пользователь с ID = 119 уже имелся в таблице users, добавление нового пользователя (вставка новой строки) не состоялось.

Решение:

Drupal использует дополнительную таблицу sequences, в которой хранится список используемых ключей. Необходимо, используя MySQL Query Browser (для локальной установки) или phpMyAdmin (для установки на удаленном сервере) скорректировать значения так, чтобы в sequences содержались только последние значения ключей. В примере выше необходимо было заменить значение 119 на 120 т.к. в работе системы произошла десинхронизация и в таблице users последним был идентификатор со значением 120, а не 119. Данная ошибка в данном примере привела к невозможности регистрации новых пользователей.

2.7 Внутренняя справка

Раздел в разработке



2.8 Создание материала

Прежде чем приступать к информационному наполнению развернутой системы, необходимо рассмотреть понятия и концепции Drupal.

Node (по русски произносится как «Нода») – это страница. Ноду можно представить в виде информационного контейнера. Нодой может стать как обычная текстовая страница, так единая «информационная» сборка – в последнем случае нода используется в качестве обертки к такого рода «страницам».

2.8.1 Разметка страницы (блоки)

Раздел в разработке

2.8.2 Типы материалов

Раздел в разработке

2.8.3 Содержание (контент)

Раздел в разработке

2.8.4 Категории

Раздел в разработке.

2.8.5 Меню

Раздел в разработке

2.8.6 Комментарии

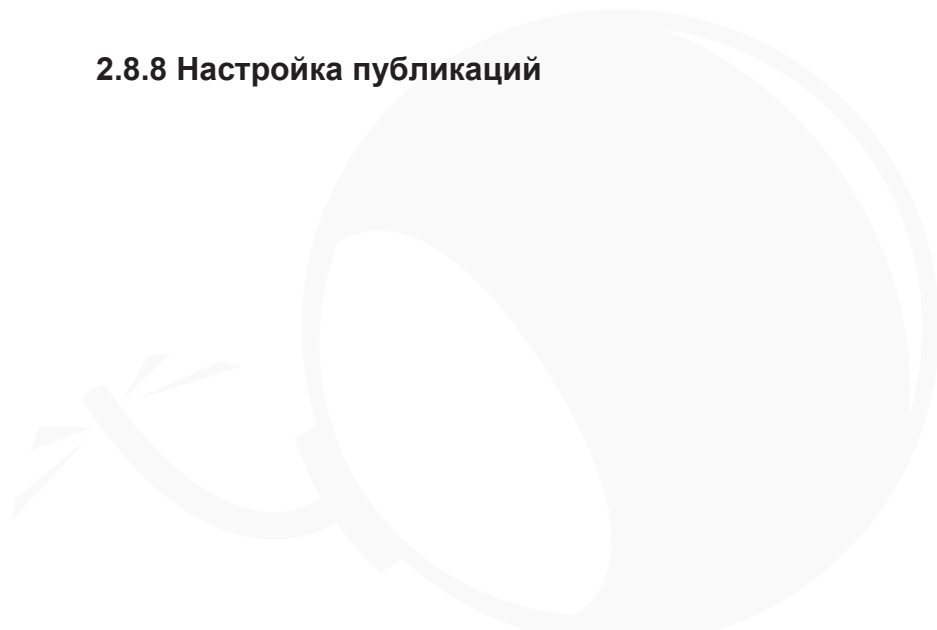
Раздел в разработке

2.8.7 Настройка лент RSS

Раздел в разработке

2.8.8 Настройка публикаций

Раздел в разработке



2.9 Модули

Раздел в разработке

Установка нового

Раздел в разработке

Удаление существующего

Раздел в разработке

2.9.1 Модуль Aggregator

Раздел в разработке

2.9.2 Модуль Blog

Раздел в разработке

2.9.3 Модуль Blog API

Раздел в разработке

2.9.4 Модуль Book

Раздел в разработке

2.9.5 Модуль Color

Раздел в разработке

2.9.6 Модуль Comment

Раздел в разработке

2.9.7 Модуль Contact

Раздел в разработке

2.9.8 Модуль Drupal

Раздел в разработке

2.9.9 Модуль Forum

Раздел в разработке

2.9.10 Модуль Help

Раздел в разработке

2.9.11 Модуль Legacy

Раздел в разработке

2.9.12 Модуль Locale

Раздел в разработке

2.9.13 Модуль Menu

Раздел в разработке

2.9.14 Модуль Path

Раздел в разработке

2.9.15 Модуль Ping

Раздел в разработке

2.9.16 Модуль Poll

Раздел в разработке

2.9.17 Модуль Profile

Раздел в разработке

2.9.18 Модуль Search

Раздел в разработке

2.9.19 Модуль Statistics

Раздел в разработке

2.9.20 Модуль Taxonomy

Модуль **Taxonomy** реализует возможности таксономии в системе. Что такое таксономия? Таксономия – это иерархическая классификация объектов (от общего класса к частному). Хорошим примером может стать биологическая классификация, где «Отряд», это более общий класс по отношению к «Семейству», «Семейство» к «Роду» и т.д.

Таксономия является частью концепции Drupal. На физическом уровне таксономия позволяет произвести классификацию элементов содержания (например, классификацию страниц). На логическом же уровне, благодаря такой классификации, появляется возможность выстраивания множественных¹ логических связей между элементами содержания, что, в свою очередь, позволяет абстрагироваться от физической реализации веб-проекта.

Таксономия в Drupal реализуется следующим способом:

- в употребление вводится новый элемент – таксономический словарь со списком категорий и подкатегорий (терминов);
- а в каждый документ (страницу | ноду) добавляется новый атрибут – поле, в котором указывается имя категории (или категорий) из упомянутого выше словаря.

На рисунке 32 показан схематический пример классической таксономии. Этот пример показывает упрощенную логическую структуру реально существующего туристического портала. На данном примере было выделено 3 группы – это множество всех изображений, множество всех статей и таксономический словарь с набором терминов (в данном случае в качестве терминов используются названия городов). Каждая из этих статей или изображений получает дополнительный атрибут «Таксономическая категория» (на данной схеме этот атрибут показан абстрактно – в виде точки). Изменяя данный атрибут, а также модифицируя таксономический словарь, разработчики и пользо-

1 И даже иерархических.

ватели могут «изменять логические отношения» между связанными элементами, причем элементы, в зависимости от способа организации таксономического словаря, могут связываться как напрямую, так и косвенно (например, через общую родительскую категорию). В частности, в описываемом примере некоторые изображения при помощи такого дополнительного атрибута были отнесены к подкатегории «Мельбурн» и косвенно к категории «Города».

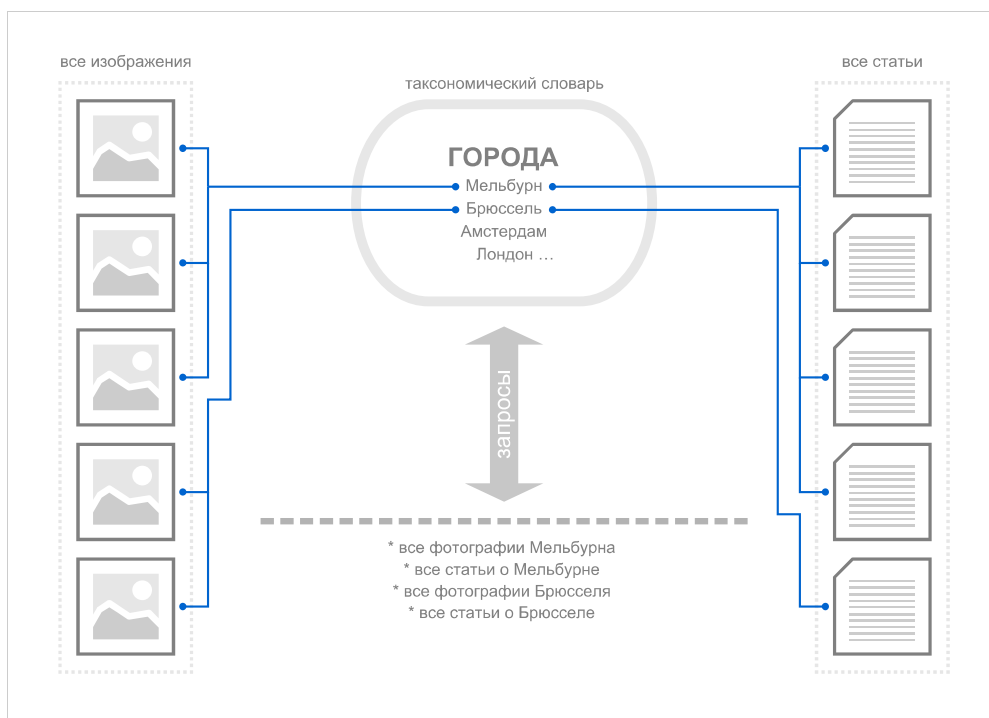


Рис. 32: Пример таксономической автоматизации.

Важным преимуществом таксономии является возможность произведения запросов на выборку целого класса связанных элементов, например, можно сделать запрос на «выборку всех изображений, города Мельбурн» или «выборку всех страниц с описанием городов»¹. Благодаря этому у разработчиков появляются возможности создания автоформирующихся меню или галерей, а также других элементов на базе таксономической автоматизации.

Однако таксономия, как часть концепции проекта Drupal, также позволяет учитывать и будущее разрабатываемой системы. Многие разработчики даже не уделяют внимание такому вопросу как последующее развитие проекта, в итоге это приводит к печальным последствиям. К примеру, система Joomla. Данная система не столь универсальна как Drupal, поэтому желание расширить веб-проект (построенный на базе Joomla), может в итоге привести к необходимости его полной переработки. Концепция же Drupal в данном аспекте дает больше преимуществ – при правильном проектировании (в рамках и на основе концепций Drupal и с применением таксономии), результатом может стать действительно масштабируемый веб-ресурс.

2.9.21 Модуль Throttle

Раздел в разработке

2.9.22 Модуль Tracker

Раздел в разработке

2.9.23 Модуль Upload

Раздел в разработке

¹ Модуль **View** частично реализует подобные запросы.



DRUPAL

3

Погружение ...

3.1 Интеграция CMS

Что такое интеграция? Обычно это понятие употребляют в качестве синонима к термину «объединение», но популярно и употребление «интеграции» в контексте термина «внедрение». И в первом, и во втором случае употребление действительно верное – все зависит от того, по отношению к чему проводится интеграция. В данном случае (и в книге в общем) понятие интеграции употребляется как в смысле объединения, так и в смысле внедрения. При этом интеграцию системы управления (интеграцию веб-проекта в систему управления) – это процесс создания веб-проекта на базе выбранной системы управления с учетом ее особенностей и в рамках концепций и правил, принятых в данной системе.

В реальной практике обычно можно выделить 3 степени интеграции:

- **Полная интеграция** – веб-проект разработан с учетом особенностей и в рамках концепций и правил выбранной системы управления и построен (внедрен) на базе выбранной системы управления.
- **Средняя степень** (неполная интеграция) – веб-проект разработан с учетом особенностей, но вне рамок концепций и правил выбранной системы управления, однако построен на базе выбранной системы управления – физически веб-проект построен на базе системы, однако **1)** логически он является надстройкой над системой, или **2)** не все элементы разрабатываемого проекта учтены, либо не все элементы достаточно проработаны (имеется в виду оформление, а не логика).
- **Низкая степень** интеграции, либо **ее отсутствие** – веб-проект разработан без учета особенностей, правил и концепций выбранной системы управления и не работает на базе выбранной системы – фактически весь веб-проект в целом представляет собой связку двух физически и логически независимых систем, при этом выбранная система выступает в качестве «хранилища данных» и «обработчика запросов» (хорошим примером может стать сравнение с СУРБД). Связь этих систем может осуществляться через специальные интерфейсы, каналы, средства сопряжения и т.д.¹

Теперь необходимо рассмотреть каждую степень интеграции более подробно и выяснить какие именно концепции и правила здесь упоминались.

Полная интеграция

Полная интеграция, с логической точки зрения – это интеграция в рамках концепций, правил и особенностей выбранной системы управления. К примеру, в Drupal одной из таких концепций является таксономия. При разработке проекта на базе Drupal можно игнорировать таксономию и создавать веб-проект по своим собственным принципам, а можно воспользоваться возможностями таксономии в полной мере. Аналогично и с другими концепциями, понятиями и правилами. В физическом же смысле, полная интеграция – это полная проработка всех возможностей стандартного пакета системы: описание всех стилей, прорисовка всех элементов каждого базового модуля и т.д.

Теперь о плюсах и минусах полной интеграции. Существенным плюсом полной интеграции является действительная «настоящая» расширяемость будущего веб-проекта². В свою очередь, это означает (в идеале), что **1)** установка любого нового модуля будет проходить без необходимости доработки этого модуля и ранее разработанного проекта (т.к. уже будет достигнута логическая совместимость), при этом **2)** в таком проекте новый модуль, его интерфейс и внешний вид будут максимально адаптироваться («вписаться») в существующий веб-проект.

Минусом же полной интеграции являются большие трудозатраты и соответственно большая стоимость веб-проекта³. В чем же выразятся эти трудозатраты? В случае с полной интеграцией необходимо не только полностью проработать все возможности стандартного пакета системы (о чем уже было сказано ранее), но и учесть все будущие изменения (новые модули и возможности их

1 Крайне нежелательным и даже опасным является вмешательство одной системы в структуру другой (например, работа через СУРБД с таблицам другой рабочей системы).

2 Подобная расширяемость означает, что в целом степень расширяемости разрабатываемого веб-проекта будет приближаться к степени расширяемости самой системы управления в ее базовом виде.

3 Если интегратор, в лице веб-студии, работает только с типичными веб-проектами, отказываясь при этом от нетривиальных заказов, а схема его работы налажена и отработана, как конвейерный механизм, тогда данный недостаток перестает быть значимым.

отображения и т.д.)¹, что в принципе не так уж и сложно сделать, имея соответствующие наработки.

Теперь о полной интеграции при переходе с одной системы на другую (процесс миграции) – довольно непросто перенести и полностью интегрировать уже созданные ранее большие сложные проекты (хотя заказчики думают иначе), ведь для этого необходимо перенести существующую логику в новый контекст (иначе это уже будет не полная интеграция), а для этого потребуется описать существующую логику новыми понятиями, либо полностью ее переработать.

Средняя степень интеграции

Средняя степень интеграции (или неполная интеграция) – это степень интеграции проекта, который разрабатывался с учетом особенностей, но вне рамок концепций и правил выбранной системы. В этом случае такой проект является надстройкой над системой и реализует свою логику самостоятельно, т.е. такой проект представляет из себя независимую логическую подсистему в основной системе. Этот так называемые «запрограммированные проекты». Чаще всего такие проекты являются следствием неопытности интегратора (запрограммировать работу порой значительно проще, нежели пытаться сделать все стандартными средствами, либо средствами сторонних модулей), и реже – следствием сложной, либо нетривиальной логической структуры проекта. Как результат, полная, либо частичная потеря расширяемости (вносить изменения в логику таких проектов могут только сами интеграторы).

Также средне интегрированным проектом иногда можно считать функционально не проработанный проект. В таком проекте не все элементы интерфейса переопределены и стилизованы (например, новые подключаемые модули работают без сбоев, однако оформление интерфейса этих модулей и их вывод сильно отличается по цветам или расположению элементов от всего проекта). Это выгодный подход, если заказчику необходимо срочно все выполнить, тогда всю интеграцию можно разбить на два и более этапа: этап 1) разрабатывается основная функционально-достаточная часть проекта, которая затем запускается в работу; этап 2) ведется доработка оставшихся стандартных функций (при этом параллельно исправляются ошибки уже работающего проекта и учитываются пожелания заказчика); этап 3) расширение – по желанию заказчика внедряется и дорабатывается новая функциональность. Типичным примером такого подхода может стать использование одновременно двух тем в системе Drupal, когда в административном режиме используется полнофункциональная тема (например, **garland**), а во всех остальных – разработанная тема, которую и увидят посетители ресурса.

Также если заказчику не требуется все стандартные функции системы, например, он не собирается открывать общественный форум и т.д., тогда он может отказаться от проработки остальных частей (например, элементов все того же форума), в результате клиент сможет сократить свои расходы. Хотя, по мнению автора, оформление неиспользованных элементов можно просто позаимствовать из любой другой темы, например, оформление таблиц, элементов форм и др. можно взять с соответствующими цветами из темы **garland**.

Если вопрос интеграции не был согласован (оговорен) с заказчиком веб-проекта, значит, такие недоделки можно уже смело назвать «халтурой», а не слабой интеграцией. Когда ничего не подозревающий заказчик вдруг включает новый модуль и обнаруживает, что все новые блоки и другие элементы отличаются от цветовой схемы проекта, он, в последствии, начинает сожалеть, что обратился именно в эту веб-студию.

Низкая степень интеграции

Низкая степень интеграции либо полное ее отсутствие. Встречается довольно редко. Обычно применяется в случаях, когда уже имеется большая рабочая система (с высокой посещаемостью) и необходимо внедрить в нее новую функциональность, но при этом подобную функциональность не реализует ни один модуль к подобной системе.

В этом случае разумным выходом из такого положения, конечно же является решение создать такой модуль, однако не все принимают подобные решения – время, трудозатраты и т.д. могут помешать такой инициативе

¹ Необходимо будет разработать макет и стили таким образом, чтобы сторонние модули могли беспрепятственно использовать общие стили (оформление), либо чтобы существующие стили не перекрывали стили сторонних модулей.



the **Last Part**...

Последнее слово

«Инструменты»

Эта книга целиком и полностью была набрана и подготовлена средствами пакета **OpenOffice.org (Writer, Draw, Calc)**. Автоматическая нумерация заголовков и страниц, логотипы на полях и т.д. – все это возможности **OpenOffice.org Writer**. Векторное оформление данной книги создано в среде **Inkscape**¹ (на рисунке 33 вы можете увидеть пример векторной графики). Также была использована открытая виртуальная машина **VirtualBox**.



Рис. 33: Даже после увеличения векторные изображения продолжают «выглядеть» безупречно.

В создании книги приняли участие:

Хочу поблагодарить всех тех, кто оказал посильную помощь в деле создания данной книги, и в первую очередь: **Данилову Галину Владимировну, Сыроежко Дениса Александровича**, а также всех инициативных пользователей, приславших исправления и/или свои рекомендации: **Петра Рыбина, Kotsura Denis Alekseevich, Анну [Sa_Bo_Nim], Katsovich Evgene, Alex Netkachov** и многих других пользователей.

С чего все начиналось

В свое время (это был 2006-2007г.) автор (т.е. я) пытался создать собственную систему управления, однако через определенное время (примерно месяцев пять) я пришел к выводу, что такая работа окажется не востребовавшей: она не принесет пользы ни мне, как разработчику, ни моим заказчикам, ни другим пользователям. Естественно, я хотел сделать лучшее решение: использовать исключительно PHP5 (возможности ООП); использовать расширение mysqli и работать с MySQL только посредством хранимых процедур и функций; делать поиск по внутренней структуре, а не использовать сторонние поисковые системы; организовать информацию в виде блоков, где каждый блок мог содержать в себе сколь угодно много аналогичных блоков и т.д. В общем, идея была хорошая ровно до того момента, пока я не понял, что наиболее правильным решением станет не создание очередного неизвестного и сырого продукта, а выбор существующего надежного решения с готовой инфраструктурой (модулями, темами и т.д.).

Отказавшись от собственной идеи, я начал поиск такого открытого решения – закрытые коммерческие продукты меня не интересовали в принципе, я не мог понять (да и сейчас не могу понять), как можно выбирать системы, которые предлагаются и позиционируются по следующей схеме:

*Мы предлагаем три класса нашей системы управления: **Mini**, **Standart** и **Expert**. Класс **Expert** предоставляет вам «все» возможности за **1000*n€** и рассчитан, в первую очередь, на большие веб-проекты. Класс **Standart** рассчитан на рядовые веб-проекты, данный класс обойдется вам в **100*n€** (но, как покажет время, функциональных возможностей данного класса вам окажется недостаточно, поэтому в любом случае вам придется перейти на наш класс **Expert**). Класс **Mini**, предоставляется бесплатно (для некоммерческого использования) – это сильно урезанная версия класса **Standart** (мы, разработчики, сами не понимаем, как на базе данного класса можно построить проект даже начального уровня, однако мы надеемся, что это все же выполнимая задача)...²*

В качестве критериев отбора я выбрал следующие: разработчик и его поддержка, возможности системы и некоторые другие качества. После пересмотра большого числа вариантов я остановил свое внимание на довольно необычном, на мой взгляд, решении – это был Drupal. После недолгого знакомства с системой я понял, что передо мной была не очередная заурядная CMS, а система, которая через несколько лет может стать лидером в своей области (что, собственно сейчас и

1 Переработанный логотип Drupal, который вы можете увидеть на обложке книги также сделан в Inkscape, причем он составлен всего из 5 элементов.

2 Довольно скептически, но суть отражает верно.

происходит), кроме всего прочего, это была система, которая удовлетворяла всем моим критериям отбора. На тот период Drupal еще не был столь популярен в России как сейчас, а в Белоруссии Drupal практически никто не внедрял. Однако последнее меня нисколько не смущало. Смущало только отсутствие русскоязычной документации к данной системе, что в принципе было рядовым явлением – на тот момент ситуация складывалась таким образом, что ни по одной CMS не было отечественных книг, ни в электронном, ни в печатном виде¹, и это несмотря на то, что первые системы управления появились довольно давно (а в последнее время их число значительно возросло). Конечно в глобальной сети к тому времени уже появились обзоры систем управления и даже небольшие сборки статей на данную тему, однако этого было явно недостаточно.

После недолгого знакомства с Drupal я принял решение создать небольшую книгу для будущих пользователей этой системы (т.е. для тех кто ранее не «встречался» с подобными системами вообще). Таким образом и появилась эта книга – «Первая оригинальная русскоязычная книга о Drupal»², начало которой было положено весной 2007 года. На какой то период времени работа над данной книгой была приостановлена, поэтому первый результат все смогли увидеть только весной 2008 года.

В заключении

Хочется верить, что вы остались довольны своим выбором – выбором системы управления Drupal. Не забывайте, что Drupal это не просто система управления, но и инструмент, поэтому если в вашем собственном веб-проекте (построенном на базе Drupal) вас что-то не устраивает, то вы можете легко это исправить – главное, не останавливать на достигнутом.

Надеюсь, что эта книга окажется для вас полезной и время, потраченное автором на ее создание, не пройдет даром.

1 Из этого числа есть и исключение – в свое время была выпущена книга по PHPNuke, о которой уже никто и не вспомнит. Также разработка CMS вскользь обсуждалась в некоторых книгах по PHP-программированию, но создание CMS – это прерогатива программистов, а не пользователей.

2 Цитата Вадима Барсукова.

Авторский раздел

В данном разделе автор хочет выразить собственное мнение по поводу «положения некоторых вещей в природе». Все сказанное здесь является сугубо личным мнением автора, имеет довольно посредственное отношение к теме книги и не преследует цель оскорбления кого-либо.

От себя лично ...

Хотелось бы отметить одну печальную тенденцию: автор заметил, что будущие программисты (речь идет о студентах), как правило, идут по пути наименьшего сопротивления, начиная изучать легкодоступную литературу или же литературу с подачи (советов) доброжелателей – типичная ситуация «Ученик нашел учителя»¹. А в результате «на свет» появляется очередной невостребованный специалист.

Хочется, конечно, чтобы таких ситуаций становилось все меньше и меньше, а качественной литературы по свободным открытым технологиям хватало всем.

Радуют однако открытые движения и проекты – их влияние на софтверные корпорации (которые постепенно начинают сдавать свои позиции). Сейчас уже практически не приходится выбирать между открытыми решениями и коммерческими – открытых решений стало больше, их качество стало лучше (надежность увеличилась). Linux стал все больше продвигаться в массы². Apache, PHP, MySQL с каждым годом лишь усиливают свои позиции, но, главное, что и обычные пользователи стали узнавать открытые решения.

Также впечатляет развитие социальных сетей – появилось еще больше не только интересных, но и реально полезных ресурсов (например [Wikipedia](http://ru.wikipedia.org) и д.р.).

Теперь о ситуации с поддержкой пользователей. Мир открытых проектов функционирует, в первую очередь, благодаря идеям и стремлению сделать что-то лучше и проще. Разрабатывая открытые решения, авторы своей инициативной и бурной деятельностью поддерживают пользователей таких решений – пользователи постоянно и свободно получают новые версии продукта, а исправление ошибок не затягивается на недели. При этом на западе такая помощь не является односторонней – благодарные пользователи в ответ готовы добровольно поддержать авторов открытых решений (скромное спонсирование). Однако не только пользователи, но и крупные компании готовы поддержать авторов (причем среди последних есть и Sun и Novell и Apple и другие³). Получается некий симбиоз на добровольной основе, без каких-либо обязательств.

На отечественном же рынке открытых решений все складывается совсем иначе – пользователь, по непонятным причинам, и сейчас продолжает оставаться обиденным потребителем. Хочется пожелать, чтобы в будущем пользователи не отстранялись от открытых проектов, а оказывали малую и посильную помощь, благодаря которой авторам проектов не приходилось бы выбирать, например, между «дешевым хостингом» и «чуть подороже», а авторам лучших проектов такая помощь давала бы возможность уделять больше времени любимому открытому делу, не давая поводов «отвлекаться» на «банальное зарабатывание средств ради самообеспечения».

В общем случае ситуация складывается довольно оптимистично...

1 Соловье В.П: «Русская рулетка».

2 Интересно, что несмотря на то, что сам ОС Linux и смартфоны появились уже довольно давно, только сейчас (с 2007-2008 года) на них стали устанавливать эту операционную систему.

3 И не важно с какой целью они это делают: ради собственного выживания или же с целью реально помочь в развитии открытых проектов – в любом случае такая помощь важна.

Полезные источники

1. <http://drupal.bz>
BZ – это проект, частью которого является данная книга. Этот проект создан, в первую очередь, для потребителей системы Drupal. Сейчас Drupal – это не только готовая система, но и целое движение / интернациональное сообщество. Такая популярность системы Drupal не может не радовать, однако есть у проекта Drupal и минусы – данная системы пока довольно слабо представлена в России, на Украине или в Белоруссии. Безусловно, уже многие веб-интеграторы успели «взять Drupal на вооружение», но пока еще никто не занимается позиционированием и продвижением данной системы. Также для Drupal в наших странах еще не успела сформироваться и потребительская инфраструктура – всю информацию (литературу, переводы), дистрибутивы, наборы необходимых «мелочей» и т.д. необходимо искать, так как располагается все это довольно разрознено. Практически все отечественные ресурсы (Drupal.ru, Drupal.by и др.), посвященные Drupal, ориентируются, в первую очередь, на разработчиков (программистов) – на потребителя пока никто не обращает внимания, а ведь потребитель – это и будущий пользователь, который возможно уже «присматривается» к данной системе, но благодаря такой организации Drupal-инфраструктуры, в конечном счете, вряд ли выберет именно Drupal, ведь в сравнении с ним коммерческие продукты в «красивой обертке» выглядят гораздо привлекательней.
2. <http://drupal.org>
Это официальный сайт проекта Drupal и интернационального Drupal сообщества. С этого ресурса можно получить не только самую последнюю версию дистрибутива системы, но и дополнительные модули, темы и переводы. Также здесь представлен и форум разработчиков Drupal, на котором можно узнать все необходимое о Drupal (общение ведется на английском).
3. <http://drupal.ru>
Drupal.ru – это попытка представления данной системы в Рунете¹. Этот проект смог консолидировать русскоязычное Drupal-сообщество. Здесь представлено достаточное количество полезной технической информации, однако для ее поиска приходится прилагать усилия, т.к. организация данного проекта не самая приемлемая («страдает» юзабилити ^А²). На «живом» форуме этого проекта можно задать интересующий вопрос и довольно быстро получить ответ.
4. <http://drupal.by>
Нельзя не упомянуть и других национальных сообществ этой системы. Белоруссия – Drupal.by довольно молодой проект (в сравнении с Drupal.ru), но не смотря на это, его уже можно считать состоявшимся. Разработчики и пользователи Drupal.by довольно быстро реагируют на все изменения. Общение здесь ведется, как правило, на русском языке. Технической документации на проекте не много, но ее заменяют (образно) инициативные пользователи, которые всегда готовы придти на помощь.
5. <http://drupal.kiev1.org> и <http://drupal.kiev.ua>
Украинские проекты. Первый проект довольно слабый, возникает ощущение, что он уже давно перестал развиваться. Второй выглядит более перспективно, но содержит информацию сугубо на украинском и английском, поэтому не представляет особого интереса для не украинцев. К сожалению, украинское Drupal-сообщество не столь едино, отсюда и результаты.

Интересно то, что все Украинские Drupal-проекты зарегистрированы в доменах третьего, а не второго уровня. Как оказалось, «согласно действующим Правилам домена .UA, доменные имена делегируются только владельцам прав на соответствующие торговые марки», поэтому Drupal.ua появиться видимо не скоро. Также небольшая заметка о других доменах Drupal: небольшой тест показал, что основная часть Drupal-доменов уже занята. К сожалению, многие из них используются не по назначению либо вообще не используются.

6. <http://setegnom.com>
Setegnom.com – очень интересный ресурс. На нем приводится довольно много актуальной разносторонней информации (в основном, это переведенные статьи с Drupal.org), имеется интересная аналитика (например, здесь представлен рейтинг «разрастания кода» различных систем управления) и много другой информации. С каждым днем объем ресурса увеличивается, однако статьи здесь не большие – трудно сказать, хорошо это или плохо.

1 Рунет – русскоязычная часть сети Интернет

2 На взгляд автора, идеальный юзабилити имеет проект php.net

7. <http://drupal.org>

Drupal.org – сервер переводов. Этот проект можно назвать общественным – с его помощью любой желающий может внести свой вклад в дело локализации системы Drupal¹. Данный проект позиционируется как интернациональный сервер локализации. Кроме русского на проекте представлены также следующие языки: белорусский, болгарский, эстонский, немецкий, казахский, латвийский, литовский, польский, русский, сербский, украинский. По умолчанию для всех незарегистрированных (и не прошедших аутентификацию) пользователей весь интерфейс проекта представляется на английском языке. Для того чтобы переключить язык интерфейса пользователю необходимо зарегистрироваться², после чего в настройках своего профиля (ссылка [Мои учетные данные](#)) задать необходимый язык. На данном проекте ведется локализация не только непосредственно системы Drupal (ее стандартных модулей), но и множества других модулей. Получить перевод с данного проекта не сложно, к примеру, для того чтобы получить перевод непосредственно системы Drupal, вам необходимо выполнить следующие действия:




На главной странице перейти по ссылке [Экспорт \[ru\]](#), после чего, оказавшись на странице «Export Russian translations», в поле [Choose a project](#) (Выберите проект) необходимо вписать фразу «drupal», а затем нажать кнопку [Choose project](#).



После этого, на той же странице, появится предупреждающее сообщение «You should choose a release». Теперь необходимо заполнить новые поля, которых раньше не было: список [Release](#) (Релиз) – выберите интересующую вас версию системы; переключатель [Type of export](#) (Тип экспорта) – по умолчанию будет выбрано значение «Translation» («Перевод»), эту опцию следует оставить без изменений; переключатель [Format used](#) (Используемый формат) – если вы делаете локализацию системы на примере данной книги, тогда рекомендуется выбрать значение «All in one file» («Все в один файл»). После проделанных действий следует нажать кнопку [Export](#) (Экспорт).



В завершении всего процесса вам будет предложено скачать файл перевода (в данном случае таковым являлся файл «[drupal-5.X-ru.po](#)»). К сожалению, этот сервис работает с большим числом ошибок, поэтому описана выше последовательность не всегда проходит именно таким образом. Если у вас что то не получилось – автор рекомендует очистить [кэш](#)  веб-браузера и попробовать пройти все эти этапы заново.

1 Однако в своей работе пользователь должен использовать общепринятую терминологию. Хотя унификация терминологии и приводит к незначительному «отдалению от контекста» (например, во многих случаях можно было применить более дружественные и понятные описания), такая унификация все же способствует систематизации процесса локализации.

2 Либо использовать логин с drupal.org.

Определения = А

В этом разделе даются определения / понятия и раскрывается их смысл. Все определения взяты с ресурса Википедия.

У многих возникает вопрос: «Почему все понятия взяты именно с Википедии? Ведь логичнее давать ссылки на определения именно с того ресурса, где основной темой и являются эти определения». Этому есть вполне логичное объяснение – «Википедия» – это объективный общественный информационный ресурс, который, возможно, «переживает» любую стороннюю статью, размещенную на специализированном форуме или ресурсе. Также размещение ссылок на множество других проектов – это бессмысленная реклама таких проектов.

Фонд Викимедиа

<http://ru.wikipedia.org/wiki/Викимедиа>

Фонд «Викимедиа» (англ. Wikimedia Foundation Inc) – некоммерческая организация, обеспечивающая материальную основу для ряда интернет-сообществ, создающих свободно распространяемую литературу, и выполняющая в них функции управления. Деятельность Фонда осуществляется благодаря добровольным пожертвованиям. Саму же работу по написанию «свободной» литературы делают добровольцы со всего мира.

На данный момент Фонд Викимедиа объединяет следующие ресурсы:

название	адрес	описание
Википедия	wikipedia.org	Свободная энциклопедия
Викисловарь	wiktionary.org	Толковый словарь и словарь синонимов
Викицитатник	wikiquote.org	Архив знаменитых цитат
Викитека	wikisource.org	Свободный архив источников
Викиверситет	wikiversity.org	Коллективное дистанционное обучение
Викиучебник	wikibooks.org	Свободные учебная литература
Викиновости	wikinews.org	Свободное информационное агентство
Викисклад	commons.wikimedia.org	Общее хранилище файлов мультимедиа
Викивиды	species.wikimedia.org	Справочник по таксономии биологических видов

Википедия (Википедия о понятии «Википедия»)

<http://ru.wikipedia.org/wiki/Википедия>

Википедия (англ. Wikipedia) – многоязычная общедоступная свободно распространяемая энциклопедия, публикуемая в Интернете. Создается на многих языках мира коллективным трудом добровольных авторов, использующих технологию вики. С момента зарождения в начале 2001 года и поныне Википедия неуклонно растет и набирает популярность у пользователей Сети.

Авторизация

<http://ru.wikipedia.org/wiki/Авторизация>

Авторизация – процесс, а также результат процесса проверки необходимых параметров и предоставление определенных полномочий лицу или группе лиц (прав доступа) на выполнение некоторых действий в различных системах с ограниченным доступом.

Аккаунт (Учетная запись)

<http://ru.wikipedia.org/wiki/Аккаунт>

Учетная запись – запись, содержащая сведения, которые пользователь сообщает о себе некоторой компьютерной системе. Как синонимы в обиходе могут использоваться сленговые термины «аккаунт» и «экзаунт» (от англ. account – учетная запись, личный счет).

Аптайм (Uptime)

<http://ru.wikipedia.org/wiki/uptime>

Аптайм вычислительной системы (от англ. Uptime) – время непрерывной работы вычислительной системы или её части. Измеряется с момента загрузки и до момента прекращения работы (зависания, перезагрузки, выключения, прекращения работы анализируемого приложения). В случае, если система всё ещё функционирует на момент обсуждения, аптайм считается с момента загрузки по текущий момент.

Аутентификация

<http://ru.wikipedia.org/wiki/Аутентификация>

Аутентификация (англ. Authentication) или подтверждение подлинности – процедура проверки соответствия субъекта и того, за кого он пытается себя выдать, с помощью некой уникальной информации, в простейшем случае, с помощью имени и пароля. Данную процедуру следует отличать от идентификации (опознавания субъекта информационного взаимодействия) и авторизации (проверки прав доступа к ресурсам системы).

Веб-браузер

<http://ru.wikipedia.org/wiki/Браузер>

Веб-обозреватель или браузер (из англ. Web browser) – это программное обеспечение для просмотра веб-сайтов, то есть для запроса веб-страниц (преимущественно, из Паутины), для их обработки, вывода и перехода от одной страницы к другой.

Веб-сайт

<http://ru.wikipedia.org/wiki/Веб-сайт>

Веб-сайт или просто сайт (англ. website, от web – паутина, «веб» и site – «место») – это совокупность веб-страниц, доступных в интернете через протоколы HTTP/HTTPS; совокупность всех общедоступных веб-сайтов есть Всемирная Паутина. Страницы веб-сайта объединены общим корневым адресом, а также обычно темой, логической структурой, оформлением и/или авторством.

Ранее понятие смешивалось с физическим узлом сети – хостом, сервером (узлом). Но с ростом Интернета и технологическим улучшением серверов на одном компьютере стало возможно размещение множества сайтов и доменов.

Веб-сервер

<http://ru.wikipedia.org/wiki/Веб-сервер>

Веб-сервер – это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными. Веб-серверы – основа Всемирной паутины.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и компьютер, на котором это программное обеспечение работает.

Клиенты получают доступ к веб-серверу по URL адресу нужной им веб-страницы или другого ресурса.

Веб-страница

<http://ru.wikipedia.org/wiki/Веб-страница>

Веб-страница (англ. Web page) – гипертекстовой ресурс Всемирной паутины, обычно написанный на языке HTML. Веб-страница может содержать ссылки для быстрого перехода на другие страницы, а также изображения. Программа, демонстрирующая веб-страницу, называется веб-браузер.


Кроме текста и изображений веб-страница может содержать медиа файлы, например звуковые файлы и видео, апплеты, а также скрипты (JavaScript, VBScript), позволяющие делать содержание страницы динамическим.

Несколько веб-страниц, объединенных общей темой и дизайном, а также связанных между собой ссылками и обычно находящихся на одном веб-сервере, образуют веб-сайт.

Обычно веб-страница имеет расширение .html или .htm.

Всемирная паутина

http://ru.wikipedia.org/wiki/Всемирная_Паутина

Всемирная паутина (англ. World Wide Web) – глобальное информационное пространство, основанное на физической инфраструктуре Интернета  и протоколе передачи данных HTTP. Всемирная паутина вызвала настоящую революцию в информационных технологиях и бум в развитии Интернета. Часто, говоря об Интернете, имеют ввиду именно Всемирную паутину. Для обозначения Всемирной паутины также используют слово веб (англ. web) и аббревиатуру «WWW».

Датацентр

<http://ru.wikipedia.org/wiki/Датацентр>

Датацентр (от англ. data center) – специализированное здание (площадка) для размещения серверного и коммуникационного оборудования и подключения к каналам сети Интернет. Датацентры подключены к различным узлам связи (каналам). Именно качество и ширина каналов влияют на уровень предоставляемых услуг, так как основным критерием оценки качества работы любого датацентра является время доступности сервера (Uptime). В русскоязычной терминологии датацентр получил устойчивое название центр хранения и обработки данных (ЦХОД).


Интерпретация

<http://ru.wikipedia.org/wiki/Интерпретация>

Интерпретация (информатика) – процесс непосредственного покомандного выполнения программы без предварительной компиляции «на лету»; в большинстве случаев интерпретация намного медленнее работы уже скомпилированной программы, но не требует затрат на компиляцию, что в случае небольших программ может повышать общую производительность.

Интернет

<http://ru.wikipedia.org/wiki/Интернет>

Интернет (англ. Internet, пишется с прописной (заглавной) буквы, читается [интэрнэт]) – всемирная система добровольно объединенных компьютерных сетей, построенная на использовании протокола IP и маршрутизации пакетов данных. Интернет образует всемирную (единую) информационную среду – обитель оцифрованной информации. Служит физической основой для Всемирной [паутины](#) . Часто упоминается как Всемирная сеть и Глобальная сеть.

Контент

<http://ru.wikipedia.org/wiki/Контент>

Контент (англ. content) – информационное (содержательное) наполнение ресурса.

Кроссбраузерный

<http://ru.wikipedia.org/wiki/Кроссбраузерный>

Кроссбраузерность – свойство сайта отображаться и работать во всех популярных браузерах идентично.

Кроссплатформенный

<http://ru.wikipedia.org/wiki/Кроссплатформенный>

Кроссплатформенное программное обеспечение – программное обеспечение, работающее более чем на одной аппаратной платформе и/или операционной системе.

Кэш

<http://ru.wikipedia.org/wiki/Кеш>

Кэш или кеш (англ. Cache) – подборка данных, дублирующих оригинальные значения, сохраненные где-то или вычисленные ранее, когда оригинальные данные сравнительно труднодоступны для выборки (из-за большого времени доступа) или для вычисления по сравнению со скоростью чтения из кэша. Другими словами, кэш – это промежуточный буфер с быстрым доступом, который хранит в себе ту информацию, которая с наибольшей вероятностью может быть запрошена. Один раз сохраненные в кэше данные в будущем могут использоваться доступом к кэшированной копии – это будет быстрее, чем повторная выборка или повторное вычисление оригинальных данных, так что среднее время доступа будет короче. Поэтому кэш помогает ускорить доступ к данным, которые в противном случае ЦПУ нужно было брать из основной памяти.

Патч

<http://ru.wikipedia.org/wiki/Патч>

Заплата, или патч (англ. Patch – заплатка) – автоматизированное отдельно поставляемое программное средство, используемое для устранения проблем в программном обеспечении или изменения его функционала, а также сам процесс установки патча («пропатчивание»). Исправление может применяться к уже установленной программе, либо к ее исходным кодам. Сюда входит исправление ошибок, изменение внешнего вида, улучшение эргономичности или производительности программ, а также любые другие изменения, которые разработчик пожелал сделать.

В качестве синонима может использоваться термин «обновление» (англ. update). Однако под словом «патч» чаще понимают исправление каких-то ошибок, в то время как под обновлением – улучшение функционала и добавление новых возможностей.

Пиринг

<http://ru.wikipedia.org/wiki/Пиринг>

Пиринг (от англ. peering – соседство). Договор обмена интернет-трафиком между двумя и более сетями интернет-провайдеров. Он состоит из трех элементов:

1. физическое соединение сетей;
2. техническое взаимодействие между сетями, обмен маршрутами;
3. коммерческие и договорные пиринговые соглашения.

Интернет-провайдеры часто организуют точки обмена трафиком, т.е. помещения, в которых происходит физическое соединение сетей многих операторов.

Портал

<http://ru.wikipedia.org/wiki/Портал>

Портал (англ. Portal) – в Интернете: крупный сайт, объединяющий различные универсальные сервисы. (см. подробнее интернет-портал).

Хостинг

<http://ru.wikipedia.org/wiki/хостинг>

Хостинг – услуга по предоставлению дискового пространства для физического размещения файлов сайта на сервере, постоянно находящегося в сети (обычно Интернет). В услугу также может входить предоставление места для почтовой корреспонденции, баз данных, файлового хранилища и т.п., а также поддержка функционирования соответствующих сервисов.

Обычно предоставляется хостинговыми компаниями. Предоставлению хостинга, как правило, сопутствует услуга по регистрации домена.

Хостинг часто разделяется на платный и бесплатный. Обычно компания, предоставляющая бесплатный хостинг, зарабатывает путем показа рекламы на страницах, размещенных на нем. Частные лица для своих домашних страничек обычно используют бесплатный хостинг, а коммерческие организации – почти всегда платный хостинг. Общественные организации могут использовать как платный, так и бесплатный хостинг. Бесплатный хостинг, как правило, медленнее платного, предоставляет только базовые услуги и иногда ненадежен (то есть может закрыться).

Также можно разделить услуги хостинга по типу предоставляемого ресурса:

1. виртуальный хостинг – предоставляется место на диске и/или внешний трафик, среда исполнения веб-сервисов единая для многих пользователей.
2. виртуальный выделенный сервер (VPS или VDS) – предоставляется место на диске, внешний трафик, часть общей памяти, процессорное время сервера. Выглядит для пользователя как аренда целого сервера, но физически на одном реальном сервере располагается несколько виртуальных серверов. Обычно предназначен для проектов средней тяжести и реселлеров.
3. выделенный сервер – предоставляется целый сервер с определенным дисковым пространством, памятью, процессорами и внешним трафиком. Используется для «тяжелых» веб-проектов, которые не могут сосуществовать на одном сервере с другими проектами и требуют под себя все ресурсы сервера.

Юзабилити

<http://ru.wikipedia.org/wiki/Юзабилити>

Юзабилити (англ. usability – дословно «пользуемость», «возможность быть используемым») – понятие в микроэргономике, обозначающее общую степень удобства предмета при использовании; термин схож с термином «эргономичность», однако имеет иную область распространения и отличается нюансами определения. Применительно к компьютерной технике термином «юзабилити» называют концепцию разработки пользовательских интерфейсов программного обеспечения, ориентированную на максимальное психологическое и эстетическое удобство для пользователя.

Язык разметки

http://ru.wikipedia.org/wiki/Язык_разметки

Язык разметки (текста) в компьютерной терминологии – набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении. Принадлежит классу компьютерных языков. Текстовый документ, написанный с использованием языка разметки, содержит не только сам текст (как последовательность слов и знаков препинания), но и дополнительную информацию о различных его участках, например, указание на заголовки, выделения, списки и т.д. В более сложных случаях язык разметки позволяет вставлять в документ интерактивные элементы и содержание других документов.

CGI

<http://ru.wikipedia.org/wiki/CGI>

CGI (от англ. Common Gateway Interface – «общий интерфейс шлюза») – стандарт интерфейса, служащего для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия скрипт (сценарий) или CGI-программа.

CMF

<http://ru.wikipedia.org/wiki/CMF>

Content Management Framework (CMF) – каркасная система для управления содержимым. CMF – это инструмент для создания систем управления содержимым, а также веб-приложений вообще. Ряд CMS, предоставляющих API для расширения своей функциональности, претендуют на звание cmf, хотя провести четкую границу между CMS и cmf порой сложно. К примерам cmf, также являющимися готовыми CMS, можно отнести такие системы как Cairo CMS, Plone, MODx, Drupal, TYPO3.

CMS (Система управления содержимым)

http://ru.wikipedia.org/wiki/Система_управления_содержимым

Система управления содержимым | контентом (англ. Content management system, CMS) – компьютерная программа, используемая для управления содержимым чего-либо (обычно это содержимое рассматривается как неструктурированные данные предметной задачи в противоположность структурированным данным, обычно находящимся под управлением СУБД). Обычно такие системы используются для хранения и публикации большого количества документов, изображений, музыки или видео.

Частным случаем такого рода систем являются системы управления сайтами. Подобные CMS позволяет управлять текстовым и графическим наполнением веб-сайта, предоставляя пользователю удобные инструменты хранения и публикации информации.

Термин контент-менеджер обозначает род человеческой деятельности – редактор сайта.

Большая часть современных систем управления содержимым реализуется с помощью визуального (WYSIWYG) редактора – программы, которая создает HTML-код из специальной упрощенной разметки, позволяющей пользователю проще форматировать текст.

Localhost / 127.0.0.1

ссылки нет

В Википедии есть также и это понятие, однако его трактовка оказалась слишком «мудреной», поэтому автор попробовал сам объяснить, что это такое.

Представьте, что вы подключены к сети и «localhost» или «127.0.0.1» это имя (или адрес) вашего компьютера в такой сети. Когда вы набираете localhost в строке адреса браузера, ваш браузер обращается к сетевой машине с таким именем, а такой машиной и является ваш собственный компьютер. Если ранее вы не предприняли каких либо действий по организации веб-сервера из собственного компьютера, то, естественно, на запрос «localhost» никакого ответа не последует – браузер «скажет», что «адрес не найден», однако если вы все же запустили Apache или другой веб-сервер, то на такой запрос должен появиться ответ.

Md5

<http://ru.wikipedia.org/wiki/md5>

MD5 (Message Digest 5) – 128-битный алгоритм хэширования, разработанный профессором Рональдом Л. Ривестом в 1991 году. Предназначен для создания «отпечатков» или «дайджестов» сообщений произвольной длины.

XLIFF

<http://ru.wikipedia.org/wiki/XLIFF>

XML Localisation Interchange File Format – расширяемый платформенно-независимый стандарт обмена данными, подлежащими локализации, и сопутствующей информацией, определенный как словарь XML. В настоящее время распространены версии формата 1.1 и 1.2, последняя версия – 1.2 - была принята OASIS 24 июля 2007 года.



Приложение А: Выбор хостинга

В данном разделе рассматривается только платный хостинг, т.к. бесплатный хостинг пока не удовлетворяет даже минимальным требованиям любого серьезного веб-проекта. Также услуга бесплатного хостинга налагает на пользователя определенные обязательства (например, не препятствовать хостинг-оператору размещать на вашем проекте собственную рекламу). Минимальная стоимость¹ платного базового хостинг-аккаунта приближается к 1\$ в месяц, поэтому практически любой пользователь может себе позволить такую услугу.

Определения

Можно перефразировать понятие «Хостинга» из Википедии и дать ему следующее определение: хостинг – это услуга размещения и поддержки вашего веб-проекта в глобальной сети. Под размещением понимается предоставление ресурсов веб-сервера (дискового пространства и т.д.). К поддержке же обычно относят набор следующих действий: мониторинг активности серверной системы, резервное копирование, поддержка клиентов и некоторые другие услуги.

Лицо, предоставляющее услугу хостинга, именуется хостинг-оператором, также употребляются выражения «хостер» или «оператор». Каждый хостинг-оператор имеет свой набор тарифных планов, а длинное выражение «тарифный план хостинг-оператора» обычно употребляется как «хостинг-план» или просто «план». Лицо, приобретающее услугу хостинга, именуется клиентом или просто пользователем.

Хостинг и системы управления содержанием

Как взаимосвязаны между собой хостинг и система управления содержанием? В первую очередь, функционально – хостинг, а, точнее, тарифный план предоставляет определенный набор возможностей, а система управления потребляет их. Если система не получает минимально необходимый набор таких возможностей, то ее работа на таком хостинге становится невозможной, и тогда клиенту приходится выбирать другой тарифный план либо идти на крайние меры – менять хостинг оператора, если последний не желает «идти на встречу» (хотя по опыту автора многие хостинг-компании всегда готовы вести конструктивный диалог). Однако проблемы возникают лишь в том случае, если тарифный план уже приобретен и отказаться от него не представляется возможным. Следует помнить, что не система управления выбирается под хостинг, а хостинг под систему. Если кто-то пытается переубедить вас в обратном, то тогда посмотрите на следующую фразу: «Приобретая телевизор, вы должны заранее определиться с просматриваемыми каналами, т.к. каждая марка телевизора может показывать только определенные каналы!» – звучит довольно глупо, но именно так и получается, если выбор хостинга по подобным советам сравнить с покупкой телевизора.

В последнее время появилось немало тарифных планов, включающих предварительно установленную систему управления. К сожалению, многие хостинг-операторы в таких тарифных планах не всегда используют первоклассные открытые решения, поэтому приобретать такие планы довольно рискованно.

А.1 Классификация хостинг-операторов

Для того чтобы понять, чем один оператор отличается от другого, следует провести их классификацию. Перед началом классификации стоит рассмотреть все возможные предложения услуги хостинга. Итак, приступим.

Хостинг – это, в первую очередь, услуга, однако, чтобы оказать такую услугу, требуется наличие специализированного оборудования и технического персонала. К оборудованию, в первую очередь, относят сервера, именно на них портируются (размещаются) все веб-проекты. Сервера, как правило, размещаются на площадках в датацентрах **А**. Расположение самих датацентров не играет принципиальной роли², если информационная безопасность³ в них остается всегда на высо-

1 Цена и стоимость – в контексте этой книги данные понятия употребляются как синонимы.

2 Если вас не интересуют пиринговые **А** каналы и трафик не тарифицируется, то, расположение, действительно не играет решающей роли.

3 Конфиденциальность, целостность и доступность информации.

ком уровне. Для поддержания всего оборудования в рабочем состоянии в датацентре постоянно присутствует технический персонал. Сервера – это, конечно же, не все оборудование датацентра, также к оборудованию относятся брандмауэры, источники бесперебойного питания, многоканальная связь и средства оповещения. Серьезные датацентры одновременно размещают и обслуживают до тысячи серверов. Пример датацентра можно увидеть на рисунке 34. Датацентры не работают с частными лицами, а предлагают лишь массовую аренду серверов – этой услугой и пользуются хостинг-операторы.



Рис. 34: Вот так может выглядеть образцовый датацентр.

Также сервера могут размещаться и непосредственно у хостинг-оператора – последний может располагать своей собственной серверной площадкой¹, которая может быть организована как на местном уровне, так и интегрирована с площадкой местного интернет-провайдера. Хостинг-оператор в отличие от датацентров работает с любыми клиентами.

Практически все интернет-провайдеры имеют собственную серверную площадку. Часть такой площадки может быть отведена под хостинг. Интернет-провайдеры работают как с физическими, так и с юридическими лицами.

Крупные веб-студии также в состоянии обзавестись собственной серверной площадкой, хостинг на которых обычно предлагается клиентам таких веб-студий.

Также серверные площадки имеются и в других учреждениях: университетах, исследовательских центрах и других государственных и частных учреждениях. Аккаунты на таких площадках иногда распродают работниками этих учреждений, что, в принципе, запрещено делать.

Частные лица также могут организовать хостинг-площадку, ведь приобрести профессиональные сервера по приемлемой цене довольно просто – крупные провайдеры и операторы нередко проводят плановую модернизацию собственного оборудования, после чего устаревшие сервера распродают. Такие сервера могут быть куплены очередными горе-предпринимателями, решившими заняться хостингом – размещаются купленные сервера, как правило, в непригодных для этого помещениях², где за их работой никто не следит. Автор рекомендует держаться в стороне от подобного хостинга.

Все остальные операторы, как правило, не имеют собственного оборудования и являются реселлерами (перекупщиками). Реселлинг является абсолютно законной деятельностью (при соблюдении юридических норм).

Подводя итог к вышесказанному можно сделать следующий вывод: сервера по форме собственности могут быть как арендуемым оборудованием, так и собственностью хостинг-оператора, соответственно, возвращаясь к классификации, всех хостинг операторов можно условно разделить на два класса: 1) хостинг-операторы, имеющие свою собственную техническую базу (так называемый хостинг «из первых рук»), и 2) реселлеры.

Хостинг «из первых рук»

Итак, выбирая только самые серьезные категории операторов, мы можем выделить следующую группу компаний, владеющих серверным оборудованием – это:

- хостинг-операторы;
- интернет-провайдеры;
- крупные веб-студии.

Хостинг-оператор, владеющие собственным оборудованием – это хорошее решение, если не наилучшее, среди хостинга «из первых рук». Если при поиске хостинга вы найдете подобную компанию, то благополучное сотрудничество с таким оператором станет для вас наиболее вероятным исходом.

¹ Что встречается пока не часто: содержание собственной площадки – дело хлопотное.

² Автор знает случаи, когда сервера размещались даже в домашних условиях.

Хотя интернет-провайдеры и предоставляют хостинг из «первых рук», цены и качество таких услуг порой оставляют желать лучшего. Интернет-провайдер – это крупная частная или государственная компания, поэтому ей свойственны следующие недостатки:

- отсутствие гибких тарифных планов, а также не самая приемлемая стоимость этих планов (особенно у государственных провайдеров);
- медленное реагирование на динамически изменяющемся рынке хостинг-услуг;
- отсутствие индивидуального подхода к клиентам.

Теперь о хостинге веб-студий. Содержание серверного оборудования – это хлопотное и дорогое занятие (в отличие от веб-разработок), поэтому существует вероятность того, что в будущем веб-студия решит отказаться от подобных слуг. К тому же предлагаемый ими хостинг позиционируется скорее как дополнительная бонус к общему заказу (например, заказу на создания «проекта под ключ»), поэтому автор также не рекомендует приобретать подобный хостинг.

Хостинг от реселлера

Реселлеры (от англ. reseller) – это перекупщики. Кто они такие? В жизни ими могут оказаться как серьезная компания, так и частное лицо.

По сути реселлером может стать каждый – датацентры и более крупные реселлеры предлагают специальные тарифные планы, ориентированные на реселлинг, поэтому довольно часто в перепродаже хостинга участвуют и частные лица. Что можно сказать о подобных предприимчивых личностях? Покупая хостинг по минимальным ценам (0.5-0.9€ за базовый пакет), они пытаются продать его по цене несколько ниже средней (в пределах 2-3€ за базовый пакет). В лучшем случае под красивым образом подобного оператора может оказаться студент профильного вуза, поэтому, приобретая такой хостинг, вы не сможете рассчитывать на защищенность проекта от внешних сетевых атак, на качественную поддержку и на какие-либо уступки с их стороны (тарифные планы, приобретаемые ими, имеют много физических ограничений – фактически они являются обычными пользователями с небольшой долей дополнительных прав). Не стоит и удивляться, если завтра подобные «рога и копыта» перестанут существовать.

Другое дело, когда реселлингом пользуется серьезные хостинговые компании, которые арендуют сервера целиком на европейских или американских площадках. Такие компании имеют следующие преимущества:

- предоставляют весь спектр хостинг услуг;
- работают легально (заключают договора на оказание услуг с физическими и юридическими лицами);
- отвечают за свою деятельность;
- ценят своих клиентов (число которых может быть даже больше 1000).
- имеют хорошую репутацию.

Выбор такой компании – это вполне приемлемый вариант, а базовая цена от 3€ в месяц устроит многих пользователей.

A.2 Поиск, отбор, тестирование, приобретение

Поиск, отбор

Как найти достойного оператора и с чего начинать поиск? В принципе в качестве инструмента поиска можно задействовать мощные поисковые машины (например, Google), которые по фразе «хостинг» представят все отечественные предложения, а по фразе «hosting» – зарубежные (правда, отчасти, т.к. в последнем случае в результат поиска войдут и отечественные предложения). Какой же хостинг стоит выбрать: отечественный или зарубежный? Наверное, каждый для себя сам должен ответить на этот вопрос, однако следует также учесть следующее:

- 1) стоимость зарубежного хостинга выше (качество при этом остается таким же);
- 2) если вы отлично владеете иностранным языком, тогда проблема языкового барьера обойдет вас стороной, иначе вам придется ограничиться отечественными предложениями.

После того как вы произвели поиск, сразу можно приступать к рассмотрению каждого предложения. Следует отметить, что 95% предложений вы сможете отсеять на первом же этапе. Все предложения могут рассматриваются следующим образом: 1) посещение официального сайта ис-

следуемой компании; и 2) анализ перспективность ее предложений.

Ниже перечислены критерии (ограничения), по которым вы сразу можете определить подходит ли вам данный претендент (оператор и его хостинг) или нет.



Попробуйте определить, к какому типу относиться рассматриваемая вами компания. Если вас пока все устраивает, тогда можно переходить к следующему ограничению, иначе следует перейти к очередному претенденту.



Юридический адрес: как ни странно, но законно оформленная деятельность нисколько не сказывается на цене. Предложения реальных существующих организаций ничуть не хуже (в ценовом плане) эфемерных сетевых лавочек. Поэтому автор рекомендует рассматривать предложения только настоящих предприятий с классическим юридическим адресом, банковскими реквизитами, телефонами и т.д. – таким образом, вы сможете отсеять 50% всех неблагонадежных претендентов, при этом, если вы не желаете официально заключать договор, никто не заставит вас этого сделать.



Количество клиентов: чем их больше, тем лучше. Оптимальное число 1000-2000 клиентов, но большее количество уже не является показателем качества¹ – быть 5001 клиентом не так уж и приемлемо.



Поддержка: отправьте письмо с интересующим вас техническим вопросом (обычно это следующий адрес support@хостинг). Если ответ пришел быстро, значит, работа администраторов выполняется хорошо. Если ответ приходит лишь спустя несколько часов, либо сам ответ больше напоминает отписку, тогда рекомендую сразу отказаться от предложений данной компании. Если компания поддерживает общение с клиентами не только посредством электронной почты, но и менеджеров сообщений (ICQ, MSN) – это дополнительный плюс, т.к. вы получите возможность общения вживую (хотя во многих компаниях icq пока не используется). Здесь также следует обратить внимание и на количество клиентов компании, если у компании несколько тысяч клиентов, тогда ответ может обрабатываться и несколько часов, что в принципе нормально.



Современные технологии: оператор должен устанавливать современные приложения, а не обновленные прошлых лет. Так, например, вместо MySQL и PHP пятых версий довольно часто можно встретить устаревшую четвертую. Существует практика, когда компании приобретают «жестко прошитые» сервера, на которых имеется от 100 до 500 подобных аккаунтов (каждый стоимостью меньше 0.5€) – это так называемый «бросовый товар». Затем компании пытаются продать каждый аккаунт за 2-4€. Главная цель таких операторов – это получение прибыли, причем ни о какой поддержке даже и речи не идет, а если она и имеется, то только в минимальной форме. Определить таких операторов довольно просто: в таблице возможностей явно указан `php3`, `php4`, `mysql4` и т.д. Если версия не задана явно, например, просто PHP или MySQL, тогда вы можете написать письмо (актуально к пункту выше) с запросом о предоставлении результата-отчета функции `phpInfo` – этот результат даст возможность на начальном этапе проанализировать функциональность хостинга, не требуя демо-аккаунт. Если вам откажут в его предоставлении, значит, этот оператор имеет проблемы с защитой серверов – подобным образом (путем неэффективного сокрытия) оператор пытаются предотвратить будущие атаки, вместо того чтобы применять эффективные способы защиты. Полученный ранее результат-отчет функции `phpInfo` позволит также выяснить, проводит ли оператор обновление программного обеспечения.

¹ Возможно это заслуга отдела маркетинга.



Цена: средняя стоимость качественного хостинга составляет 4€ в месяц за базовый аккаунт¹. Базовый аккаунт включает в себя: дискового пространства **500mb**, **10** почтовых ящиков, **одну** MySQL базу (пятой версии), полноценный² PHP (пятой версии), поддержку большинства технологий (SSI, .htaccess и др.), имя в доменной зоне com / org / net при заказе хостинга на год, а также неограниченный трафик. Если вы собираетесь покупать более дорогой хостинг с такими же возможностями, значит либо вы переплачиваете, либо есть объективная причина такой стоимости. При заказе хостинга на 1 год / 2 года / 3 года вы также можете получить скидку в размере 10%, 20% и 30%. Однако не спешите приобретать хостинг сразу на длительный срок – начните с демо-аккаунта (он выдается как правило на несколько недель и этого времени вполне достаточно с тем, чтобы определиться подходит ли вам этот хостинг или нет).

Получение демо-аккаунта

Итак, вы отсеяли 95% бесперспективных предложений и теперь следует приступить к более детальному тестированию оставшихся вариантов (если вы просмотрели 40 предложений, значит, 5% – это всего 2 предложения). Далее вам предстоит получить демо-аккаунт.

Демо-аккаунт – это самый верный способ проверить выбранный хостинг и его поддержку. Для получения демо-аккаунта вам необходимо написать письмо в службу поддержки хостинг-оператора. Если вам откажут и предложат приобрести аккаунт на 1 месяц, значит, скорее всего, вам хотят предложить весьма посредственный хостинг. В сети существует много компаний, зарабатывающих подобным образом, схема их работы примерно такая: клиент оплачивает месяц работы аккаунта (на меньший срок аккаунты не продаются), затем через сутки он понимает, что такие условия его не устраивают, после чего он досрочно отказывается от хостинга (без возврата средств).

После получения демо-аккаунта может вдруг выясниться, что его возможностей окажется явно недостаточно, при этом оператор может вас убеждать в том, что это следствие ограниченности всех демо-аккаунтов (дословно: «мы ограничиваем возможности наших демо-аккаунтов т.к. хотим убедиться от внутренних атак, куплетный же аккаунт будет функционален на 100%»). В этом случае у вас также должны появиться подозрения, т.к. у нормального оператора демо-аккаунты будут предоставлять все возможности³ (автор не раз получал подобные демо-аккаунты и может утверждать это с уверенностью). В любом случае не стоит никому ничего доказывать, если вас не устраивает выбранный оператор или у вас возникают какие-либо подозрения – сразу отказывайтесь от таких услуг – в мире довольно много хостинг-компаний и лучше потратить время на поиск нового надежного партнера, чем потом жалеть о своем выборе.

Раздел в разработке

Проверка возможностей + phpInfo

Получив демо-аккаунт вам предстоит проверить его возможности. Сделать это будет несложно. Далее рассказывается как можно осуществить такую проверку.

Возможности хостинга – это, в первую очередь, возможности самих приложений (PHP, Apache и MySQL). Возможности каждого приложения в отдельности зависят от его версии и его настроек. Так, например, многие операторы при помощи настроек умышленно отключают стандартные возможности даже в новых версиях приложений. Это связано с неспособностью таких хостинг-операторов обеспечить достойную защиту своих серверов («нет возможности – нет проблемы»).

Раздел в разработке

1 Это пример тарифного плана Mini одного из белорусских хостинг-операторов.
 2 Полноценный – все модули по умолчанию включены, следовательно, нет функциональных ограничений.
 3 Любой нормальный оператор будет всегда заинтересован показать все свои возможности.

Тестирование отказоустойчивости

Для оценки «качества хостинга» довольно часто используют такой показатель как **Аптайм** **A** (**Uptime**). Что такое аптайм? Аптайм – это показатель доступности¹ вашего веб-проекта. Аптайм выражается в процентах. В большинстве случаев аптайм приближается к цифре 99%, которая означает, что ваш веб-проект в среднем недоступен не более 14,4 минут в сутки. Хорошим показателем аптайма является показатель в «три девятки», т.е. 99,9% (что составляет не более 1,4 минут в сутки). Очень хорошим – «четыре девятки», т.е. 99,99%.

Считается нормальным, если недоступность веб-проекта приходится на ночное время – именно ночью хостинг-оператор должен производить все плановые технические работы (поэтому в долгосрочном периоде, за месяц, квартал или год, аптайм не может быть равен 100%). Однако если ваш веб-проект недоступен в рабочее время, тогда это может весьма существенно отразиться на репутации вашей компании, либо ваша компания просто потеряет «хорошего» потенциального заказчика², ведь потенциальный заказчик не будет выяснять причину недоступности вашего веб-проекта, он просто посчитает вашу компанию несостоятельной и обратиться к конкуренту. Особенно взыскательно к таким моментам относятся крупные заказчики, ведь они считают, что их «серьезный» заказ достоин лучшей компании – лучшего исполнителя (и в этом они конечно же правы). Поэтому важно проводить мониторинг активности собственного веб-проекта.

Теперь о том, каким образом вычисляется аптайм веб-проекта. Аптайм – это статический показатель, а процесс его получения – это статистика в чистом виде, поэтому весь процесс вычисления аптайма можно условно разделить на 3 стадии: 1) статистическое наблюдение; 2) группировка и сводка результатов наблюдения; 3) вычисление показателя. На первой стадии необходимо провести наблюдение, в нашем случае это будет наблюдение за активностью (доступностью) всего веб-проекта в целом. Такое наблюдение следует проводить извне, с любой другой удаленной машины (т.е. не средствами самого веб-сервера³). В ходе этого наблюдения необходимо проводить опрос веб-сервера с заданной частотой в течении определенного периода времени. В простейшем виде такое наблюдение можно представить обычным интернет-серфингом – когда пользователь в браузере «перемещается» по веб-проекту (сайту) он как бы опрашивает его (правда при этом он не производит никаких записей, т.е. ничего не регистрирует). Поскольку человек не в состоянии в таком режиме «опрашивать» веб-проект и при этом регистрировать значения активности веб-проекта, ему на помощь приходят средства автоматизации – специальные сервисы, которые «берут всю работу на себя», в т.ч. все расчеты, освобождая пользователя от рутины.

Какие сервисы можно задействовать для проверки аптайма? Наиболее адекватным можно назвать сервис – **HostTracker** [url: host-tracker.com/ru/]. Данный сервис предоставляет две возможности: 1) постоянный мониторинг доступности вашего веб-проекта (с расчетом аптайма) и 2) единовременная проверка из множества точек.

Сервис мониторинга является платным, однако вы можете воспользоваться его 30 дневной trial версией (не ограниченной функционально). Также у сервиса имеется и одно бесплатное предложение. Более подробно о тарифах вы можете узнать на официальном сайте этого проекта.

Для того чтобы «поставить» свой веб-проект на «проверку», вам необходимо зарегистрироваться на сервисе, после чего, в раздел «Sites for monitoring», вам потребуется указать, какой веб-проект с какой периодичностью следует проверять. После выполненных действий, ваш веб-проект станет «проверяться» с заданной периодичностью. Первые отчеты сервиса вы сможете получить уже после первого дня его работы, просмотреть отчеты вы сможете в разделе «Uptime stats» (также можно подписаться на их рассылку). Пример отчета изображен на рисунке 35. В данном отчете в первую очередь следует обращать внимание на рассчитанный аптайм. Аптайм будет рассчитываться системой по дням, неделям, месяцам, кварталам, за год и за весь период, при этом все значения аптайма будут постоянно корректироваться новыми данными (данными за текущий день).

Сервис единовременной проверки дает вам возможность провести разовую проверку доступности заданного адреса из разных точек. Если вы посмотрите на рисунок 36, вы сразу поймете, какие возможности предоставляет этот сервис.

А заключении хотелось бы сказать, что данный сервис является отличным помощником в деле проверки хостинга.

1 При этом доступность является одним из факторов информационной безопасности.

2 Это особенно актуально для интернет-компаний.

3 Без использования cron и т.д.

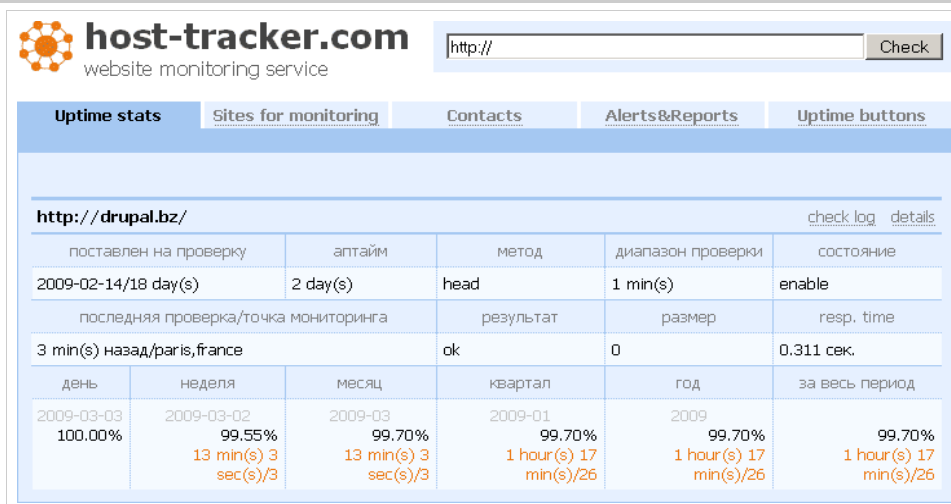


Рис. 35: HostTracker: отчет «Аптайм» (для проекта Drupal.bz).

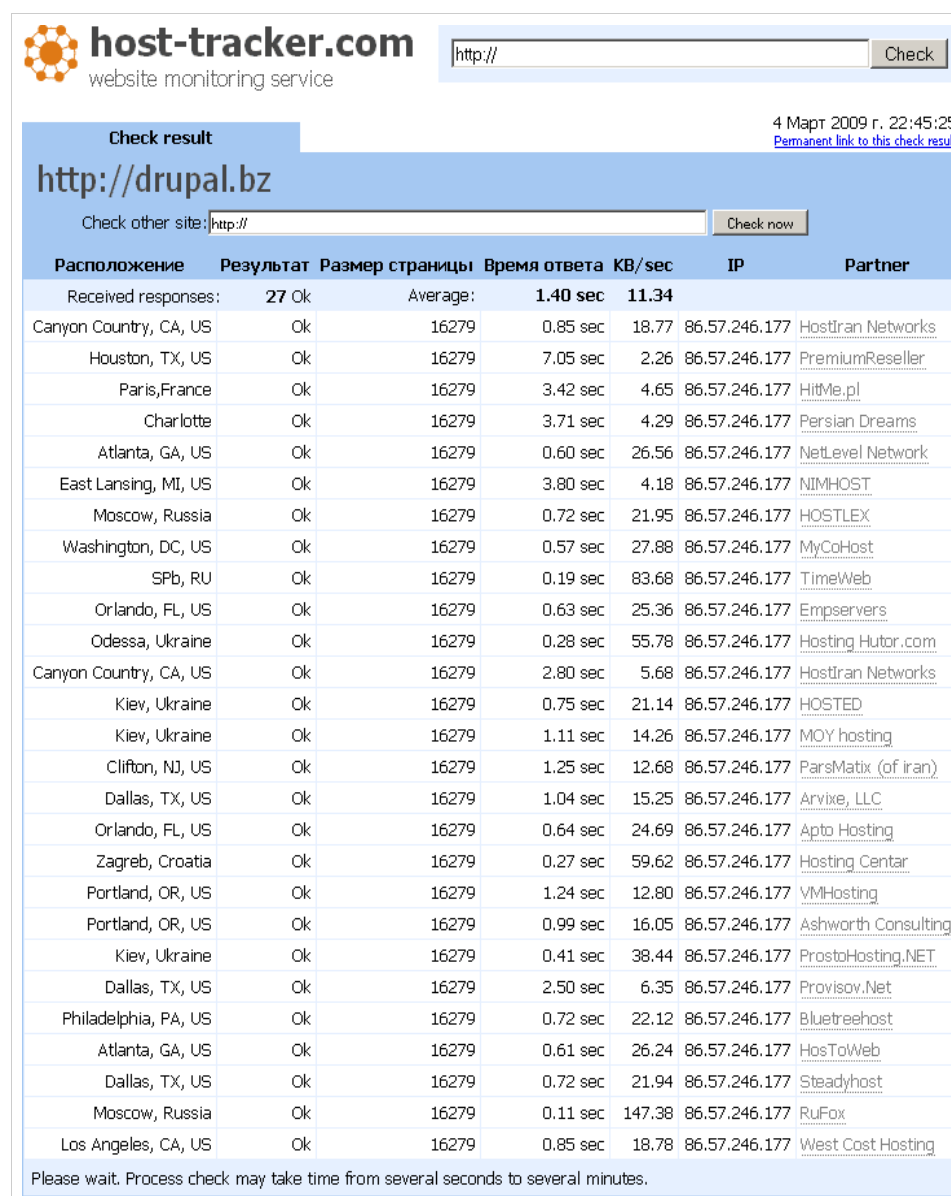


Рис. 36: HostTracker: единовременная проверка из множества точек (проверка проекта Drupal.bz).

Готовые решения

Если раньше автор решался описывать готовые решения, то теперь, по прошествии более года, автор отказывается впредь делать какие либо рекомендации относительно хостинга. Время меняется и хостинг-операторы также меняются, становятся лучше или хуже (их сервис, цены и т.д.).

Однако одного оператора все же можно привести в качестве примера – это компания **ITAS** [url: hosting.itas.com.ua]. Она предлагает относительно доступный хостинг. Данная компания интересна тем, что она смогла совместить несовместимое: возможности ее бюджетных тарифных планов превосходят все ожидания. Раньше данная компания предлагала «акционный» пакет¹ «**ITAS MINI**», стоимость которого составляла всего 79 центов, сейчас минимальным пакетом является «**ITAS START**», стоимость которого – 2,5\$. На аккаунт с любым из этих тарифных планов можно без проблем установить Drupal (без какой либо предварительной подготовки). В тарифном плане **START** вам предоставляется: PHP5², неограниченный трафик, дисковая квота 250mb, панель управления DirectAdmin, 2 базы MySQL (5-ая версия), 2 домена, 5 поддоменов, возможность работать по https, круглосуточная поддержка по icq и телефону и т.д.

Автор использовал тарифный план **ITAS MINI** в течение года. Проект drupal.bz функционировал на данном хостинге в течении всего времени (т.е. в течении практически одного года). Единственным недостатком хостинга является его относительная медлительность в «часы пик», поэтому данный хостинг идеально подходит только для маленьких (начальных) проектов. Также имелись проблемы с почтой, однако автор не использовал возможность почты на хостинге, т.к. есть более грамотное решение: в качестве почтового сервиса довольно удобно использовать **gmail**. Gmail предоставляет возможность «привязки» домена к почтовым аккаунтам этого сервиса в результате чего пользователь получает адреса со своим доменом (т.е. имя@имя_домена, а не имя@gmail.com) и с возможностями сервиса gmail.

Автор не рекомендует ориентироваться на такую цену т.к. вряд ли вы найдете подобный хостинг где либо еще (в 99,9% случаях за такую цену вам предложат либо «бросовый товар», либо «пустышку»³), а данный оператор был здесь упомянут фактически как единственный в своем роде.

После хостинга ITAS проект drupal.bz переехал на хостинг active.by (Active Technologies). Что можно сказать об этом хостинге? К сожалению у этого хостинга имеются некоторые «проблемные места». Аптайм [uptime] хостинга не самый приемлемый (автор лично наблюдал ситуацию, при которой его веб-проект был недоступен в течении 7 минут в рабочее время, причем последнее повторялось не один раз). Хотя, при перерасчете на сутки, результирующий аптайм получится очень хорошим (99,94%). Однако администраторы данного хостинга отлично справляются с своей задачей, за что им отдельная благодарность. В общем и целом, хостинг active.by можно считать хорошим и даже отличным хостингом, а все «придирки» автора можно считать несущественными, во всяком случае на данный момент.

Главным определяющим критерием выбора именно этого хостинг-оператора было предложение размещения веб-проекта на площадке ЦОД «Белтелеком» [url: ru.wikipedia.org/wiki/Белтелеком]. Благодаря этому время отклика веб-проекта уменьшилось в разы (для России, Белоруссии и Украины, что собственно и требовалось), а скорость работы веб-проекта на сервере значительно возросла⁴. Также по этому критерию можно было выбрать еще 2-3 компании (hoster.by и extmedia.by). Вторым критерием стала «родная» адекватная и толерантная поддержка. Третьим критерием стала возможность оформления «официальных отношений» (подписание договора). Рассматривались также варианты размещения в Москве и Санкт-Петербурге в датацентрах, близких к токам обмена трафиком, но в последствии от этой идеи пришлось отказаться.

• • •

В очередной раз автор убедился в том, что: 1) идеального хостинга не существует; 2) высокая цена еще не является показателем высокого качества, а 3) бренд – это не более чем миф. Что касается последнего пункта, то здесь хотелось бы обратить внимание на следующий интересный факт: многие хостинг-операторы, как было уже подмечено ранее, любят «похвастаться» своими

1 Название выбрано непосредственно компанией ITAS.

2 В заявке нужно указать, что вам требуется PHP 5-ой версии.

3 Пустышка – обман.

4 В данном случае у хостинг-оператора имеется возможность размещения собственных серверов на таких площадках, в то время, как сервера в датацентрах Европы и Америки предоставляются операторам по принципу «что имеется...».

клиентами (крупными производственными предприятиями и т.д.) в контексте «нас выбрали следующие компании...», однако если бы эти клиенты (компании) в действительности осуществляли мониторинг собственных веб-проектов на данном хостинге (проверяли uptime и т.д.), они бы поняли «цену» всех этих брендов.

Данное замечание не относится к какой либо конкретной хостинг-компании.



А.3 Доменные имена

Данный раздел не относится к тематике этой книги, однако процесс создания любого веб-проекта сопряжен с выбором доменного имени. Многие клиенты довольно часто задают следующие вопросы:

- Приобретать доменное имя как часть хостинг-плана или отдельно от хостинга?
- В какой доменной зоне выбрать имя?
- Какого регистратора доменных имен выбрать:
 - иностранного или отечественного¹?
 - Есть ли отличия между аккредитованными и не аккредитованными регистраторами?
- На какую цену следует ориентироваться?
- Какие регистрационные данные следует указывать?
- Как подобрать имя?
- Как сделать заказ и как его оплатить?
- и т.д.

Некоторые вопросы имеет смысл кратко обсудить.

Приобретать доменное имя как часть хостинг-плана или отдельно от хостинга?

Этот вопрос актуален только, если вам необходимо имя в доменных зонах com, net, org, biz, info, name (а иногда ru и us) – именно эти домены «бесплатно» включаются во многие хостинг планы (кроме дешевых тарифных планов²). Другие доменные зоны по стоимости³ дороже и могут включаться только в самые дорогие хостинг-планы. Имеет ли смысл приобретать имя отдельно от хостинга? В некоторых случаях Да, в некоторых – Нет. С технической стороны разницы нет. Имя – это всего лишь набор атрибутов на сервере регистратора доменных имен (а не хостинг-оператора). С практической точки зрения, самостоятельное приобретение имени – это дополнительные затраты, но в тоже время и определенный уровень свободы. Правда, затраты эти не значительные – всего лишь 5-10\$ в год для вышеупомянутых зон. Свобода при этом заключается в относительной независимости от хостинг-оператора, которого вы можете легко сменить (как морально, так и технически) в любой момент. Во всех остальных случаях вам придется приобретать имя самостоятельно (например, в следующих зонах: tv, dj и др.).

Есть ли отличия между аккредитованными и не аккредитованными регистраторами?

Чтобы ответить на этот вопрос, для начала необходимо выяснить как работают аккредитованные регистраторы, а как не аккредитованные. Аккредитованные регистраторы обычно получают аккредитацию у ICANN (Internet Corporation for Assigned Names and Numbers) – международной некоммерческой организации, регулирующей вопросы, связанные с доменными именами. В свою очередь, не аккредитованные регистраторы – это, как правило, реселлеры, которые осуществляют свою регистрационную деятельность через аккредитованных регистраторов. На практике нет видимых различий между аккредитованными и не аккредитованными регистраторами, поэтому с усердием искать только аккредитованных регистраторов, полностью игнорируя предложения остальных, не совсем и не всегда оправданно.

Список аккредитованных регистраторов можно просмотреть здесь: www.icann.org/registrars/accredited-list.html

Другой вопрос, а является ли регистратор зарегистрированным юридическим лицом, или же это будет виртуальная лавочка без каких-либо обязательств. Доверите ли вы свое доменное имя последнему? Я думаю, что вряд ли. Наверное, в первую очередь все же стоит заострить внимание именно на этом вопросе, а не на вопросе аккредитации.

1 В случае приобретения в зонах com / net / org т.к. домены ru / by / ua предлагаются только национальными регистраторами и отечественными реселлерами.

2 Дешевые (на данный момент) – это планы стоимостью не более 4\$ в месяц.

3 Цена и стоимость – в контексте этой книги данные понятия употребляются как синонимы.

На какую цену следует ориентироваться?

В глобальной сети можно найти очень много «доменных предложений», причем можно встретить предложения как за 6-8\$, так и за все 25\$ (здесь имеется ввиду цена на имена в доменных зонах [com](#), [net](#), [org](#), [biz](#), [info](#), [name](#)). Но цена в данном случае практически ничего не определяет, т.к. все регистраторы в сущности предлагают один и тот же «товар». Возможности дешевых предложений на практике оказываются не хуже, а даже и лучше дорогих (изменение записей **MX**, **CNAME**, **NS** и других – все это уже считается «базовым набором»). Единственное отличие – это поддержка, которая осуществляется одинаково хорошо как за 8\$, так и за 25\$ (правда, последнее утверждение справедливо далеко не для всех бюджетных регистраторов). В большинстве случаев переплачивать не разумно.

Вы можете легко проверить услугами каких регистраторов пользуются популярные хостинг-операторы. Для этого зайдите на официальный сайт одного из таких хостинг-операторов. На главной странице или же в разделе «Наши клиенты» вы сможете найти список авторитетных пользователей данного хостинга (известные имена и бренды) – многие хостинг-операторы любят козырять «важностью» собственных клиентов. Для эксперимента желательно, чтобы это были имена из доменных зон [com](#), [net](#) или [org](#). Выберите любое имя. Затем откройте страницу **whois** сервиса, для доменов [*.com](#), [*.net](#), [*.org](#) – это www.internic.net/whois.html, для доменов [*.ru](#) – это nic.ru/whois, введите в поле проверки выбранное ранее имя и нажмите соответствующую кнопку. После всего проделанного вы должны получить отчет о запрашиваемом имени примерно следующего формата: **Domain Name**: имя домена; **Registrar**: настоящий регистратор; **Name Server**: один из DNS серверов¹; **Expiration Date**: дата истечения срока действия аренды доменного имени. С большой вероятностью можно предсказать, что поле **Registrar** будет содержать значение [godaddy.com](#) (во всяком случае так было раньше). [Godaddy.com](#) – это бюджетный регистратор доменных имен. Цена доменного имени в зоне [com](#) устанавливается данным регистратором в 10\$ (в прошлом это было 8\$, а с учетом партнерских скидок для хостинг-операторов эти цены еще ниже).

Также очень часто регистраторы пытаются привлечь клиентов своими скидками. В такой политике нет ничего предосудительного, однако следует обращать внимание не только на цену регистрации имени, но и на цену его продления – скидки обычно действуют только на регистрацию, при этом стоимость продления может оказаться значительно выше стоимости продления у других регистраторов.

Как подобрать имя?

Подобрать имя в популярных доменных зонах не так то просто (в отличие, например, от зоны [cc](#) [Кокосовые Острова] или зоны [gs](#) [Сендвичевы острова]... :). Конечно, если вы хотите зарегистрировать сложное составное имя, например, «МосГорРемСтрой...», то с таким именем у вас врядли возникнут трудности. Однако ситуация изменится, когда вы пожелаете зарегистрировать более простое имя (если организация имеет незамысловатое название или доменное имя будет связано с родом деятельности) – зарегистрировать такое вряд ли получится, т.к. все простые и интересные имена и названия уже давно заняты. В таком случае вы можете воспользоваться сервисом подбора имен. Также вы можете выбрать составное имя. В последнем случае вам на помощь может придти небольшая подборка вспомогательных слов (которые можно использовать в качестве суффиксов и префиксов к интересующим названиям):

academy	business	elite	group	net	team
active	cabinet	energy	guard	office	tech
agency	catalog	euro	guide	online	the
alliance	center	expert	home	premium	union
arena	class	expo	idea	project	vision
art	club	extra	info	promotion	web
artgroup	decor	force	interactive	service	world
arts	design	free	lab	solution	zone
brand	ego	front	master	solutions	4u
bureau	elit	grand	motion	studio	4you

1 С большой вероятностью это должен быть сервер хостинг-оператора, например NS1.hoster_name.com.

Готовые решения

В качестве готового решения хотелось бы порекомендовать регистратора reggi.ru. **Reggi.ru** – это легально зарегистрированный регистратор¹ (реселлер). Стоимость его предложений несколько выше стоимости бюджетных предложений иностранных регистраторов, однако, учитывая набор возможностей и услуг данного регистратора, фактор цены в данном случае даже не имеет смысла брать в расчет (возможности и услуги: выбор из 48-и доменных зон; управление всеми необходимыми параметрами домена; оплата минимум 11-ю способами, в том числе и посредством Webmoney). В своей практике я также приобретал домены и у иностранных аккредитованных регистраторов (например, domaincontender.com – платежи осуществлялись пластиковой картой), однако в последствии я перенес² свои домены к вышеупомянутому регистратору, о чем нисколько не жалею.

1 Но не аккредитованный.

2 Процедура смены регистратора – после определенного срока (спустя примерно 2 месяца после приобретения) вы можете «передать» свое доменное имя другому регистратору. Это не сложная процедура.

Приложение Б: Дистрибутивы

Б.1 Получение

В данном разделе приводится описание процесса получения и установки всех приложений, необходимых для полноценной работы системы Drupal. Все описание ведется в рамках и на базе ОС Windows (ранее было сказано, почему выбран именно такой путь), если вам необходимо развернуть систему Drupal на другой платформе, тогда раздел «Получение» вам подойдет лишь отчасти, а раздел «Установка» не подойдет вообще. Забегая вперед, автор также хочет заметить, что в подразделе «Установка Apache» дается ссылка на официальные русские источники с описанием установки Apache в других системах. Хотелось бы отметить и еще один момент, в последнее время в дистрибутивы **Linux** систем (как правило, только в вариантах на DVD) уже включаются такие программы как веб-сервер Apache, PHP и MySQL.

Самым простым вариантом реализации данного раздела оказалась бы публикация прямых ссылок на все дистрибутивы, однако самое простое, в данном случае – не самое лучшее. Дело в том, что ссылки вещь относительная, если завтра выйдет новая версия продукта, то с очень большой вероятностью можно предсказать, что версия ныне существующего будет либо убрана с ресурса вообще, либо будет перемещена в архив и ее адрес соответственно измениться, поэтому здесь нет прямых ссылок на необходимые дистрибутивы, вместо этого здесь дается инструкция по их поиску.

Все дистрибутивы, которые вам потребуется, располагаются в глобальной сети на официальных сайтах их разработчиков (вендоров¹), где можно свободно получить самые свежие версии продуктов.

Вы также можете приобрести CD/DVD диск с подборкой программ для веб-мастера (в стремлении сэкономить на интернет доступе), однако это будет не самый лучший вариант – получая все программы от вендора, вы сможете значительно снизить риск заражения вирусами.

При поиске дистрибутивов желательно ориентироваться на самые последние их версии:

- PHP не ниже 5 версии,
- Apache не ниже 2.2 и
- MySQL не ниже 5 версии².

Установить систему Drupal можно как локально (на свой собственный компьютер), так и на удаленный веб-сервер. Для удаленного варианта установки вам не потребуются дистрибутивы Apache, PHP и MySQL т.к. все они являются предустановленными.

Хочется также отметить еще один момент: существуют различные сборки «все-в-одном», которые включают в свой состав Apache, PHP и MySQL и позволяют устанавливать их одновременно. Примером таких сборок могут стать **XAMPP**, **VertrigoServ** и другие. Что можно о них сказать? Довольно хорошие сборки, хотя, конечно, они и обладают недостатками, присущими многим «универсальным» решениям.

В книге не рассматриваются подобные сборки, если же вы собираетесь все-таки использовать их, тогда в проблематичных ситуациях данный раздел скорей всего не сможет вам помочь.

Получение Apache

Получить дистрибутив веб-сервера Apache можно с официального сайта проекта Apache. Однако найти дистрибутива этого продукта новичку будет не просто, зайдя на <http://apache.org> он не сможет сразу найти тот самый популярный классический веб-сервер. Дело в том, что Apache – это целая группа продуктов, объединенных этим именем, а классический веб-сервер имеет название **HTTP Server** (его полное название – **Apache HTTP Server**). Также довольно часто употребляется и его кодовое имя – **Httpd**.

Зайдя на apache.org, вы увидите список Apache Projects. В этом списке вы найдете ссылку «HTTP Server». Данная ссылка приведет вас на официальный сайт веб-сервера Apache (<http://httpd.apache.org>).

1 Vendor – производитель.

2 Существует даже движение **goPHP5**, направленное на отказ от вездесущей 4-ой версии PHP.

Далее вам необходимо перейти к блоку *Download*, под которым имеется ссылка «*from a mirror*». Выбрав эту ссылку, вы попадете на страницу «*Downloading the Apache HTTP Server*». На этой странице вы увидите список зеркал и дистрибутивы веб-сервера последних версий (2.2.X, 2.0.X, 1.3.X). Теперь вам необходимо найти фразу «*Win32 Binary including OpenSSL 0.X.X*», рядом с которой и будет ссылка на дистрибутив. На момент написания этой страницы файл дистрибутива веб-сервера имел название: «[apache_2.2.8-win32-x86-openssl-0.9.8g.msi](#)».

Из его названия можно понять, что это **1)** веб-сервер Apache **2)** версии 2.2.6 **3)** под 32-х битную платформу Windows с архитектурой x86 **4)** и библиотекой openssl версии 0.9.8e (и все это собрано в один файл формата msi, т.е. формата установщика программ Windows 200x/XP/Vista).

Теперь о списке зеркал. Список зеркал – это список ресурсов (расположенных в разных регионах земного шара), на которых хранятся оригинальные копии дистрибутива. Вы можете выбрать другое зеркало, т.е. ресурс, которое будет географически более близким.

Многие полагают, что выгоднее (в плане скорости) обращаться к отечественным зеркалам как территориально более близким ресурсам. Это действительно так.

Получение PHP

Получить дистрибутив PHP не составляет труда, для этого достаточно зайти на официальный сайт проекта PHP – [php.net](#) и в разделе *download* загрузить **zip-архив** (zip package) в бинарном виде (т.е. предварительно собранную / скомпилированную версию продукта) с одного из зеркал. В результате вы получите файл «[php-*-Win32.zip](#)», где * – это номер версии. На момент написания этого раздела последней версией являлся PHP 5.2.5.

Zip-архив в данном случае представляет собой набор файлов без инсталлятора. Также в этом разделе имеется и сборка с инсталлятором (в названии присутствует слово *installer*), но вам она скорее всего не потребуется, дело в том, что указанный инсталлятор уже давно не работает так как надо – он не делает рабочей связки PHP и Apache.

Получение MySQL

Для получения дистрибутива MySQL вам следует зайти на ресурс [mysql.com](#), а затем в разделе *Developer Zone* → *Downloads* → *MySQL Community Server* → *номер_версии*¹ выбрать из огромного списка² требуемый дистрибутив, после чего скачать его с одного из зеркал («*pick a mirror*»). Скачанный файл будет иметь примерно такое название: «[mysql-5.0.45-win32.zip](#)». Также для локальной работы с MySQL рекомендуется скачать программу-сборку **MySQL Gui Tools**, которая состоит из следующего набора программ и утилит:


- **Administrator**: инструмент управления MySQL-сервером. Позволяет изменять рабочие параметры MySQL-сервера, управлять пользователями / соединениями, просматривать отчеты, делать бэкапы (архивные копии) и восстанавливать данные из архивов.
- **Query Browser**: комфортная среда для работы с данными MySQL-сервера (базами, таблицами и т.д.). Благодаря этой программе можно легко создавать новые базы, таблицы, делать их изменение, удаление, наполнение, а также делать выборку и др. В Query Browser также имеется удобная справка по SQL-инструкциям.
- **System Tray Monitor**: маленькая утилита, располагающаяся в трее³, позволяет легко запускать и останавливать MySQL-сервер.
- **Migration Toolkit** – это редко используемая утилита, поэтому ее можно не устанавливать, если вы не собираетесь переносить данных из других СУБД.

Получить программу MySQL Gui Tools можно из раздела *Downloads* → *GUI Tools* → *номер_версии*. Полученный файл может иметь, к примеру, такое имя «[mysql-gui-tools-5.0-r12-win32.msi](#)».

1 На момент написания данной страницы также имелись версии **5.1** и **6.0**, однако они еще не были готовы к массовому применению
 2 Здесь имеются дистрибутивы под все системы, а вам нужно найти раздел «*Windows downloads*».
 3 Область рядом с часами.


Получение Drupal

И последнее – это Drupal. Его можно получить из двух компетентных источников: Drupal.org и Drupal.ru. Drupal.org – это оригинальная домашняя страница этого проекта, здесь всегда (в любое время без задержек) можно получить последнюю оригинальную версию продукта. Drupal.ru – это попытка организации русскоязычного сообщества вокруг этой системы, здесь также имеются дистрибутивы системы, однако автор рекомендует все же отдать предпочтение ресурсу Drupal.org (здесь будет рассматриваться способ получения именно с этого ресурса).


Откройте ресурс Drupal.org. На главной странице вы увидите блок «Latest release», где имеется ссылка на последнюю версию этой системы. Перейдите по ссылке и скачайте файл дистрибутива. Полученный файл «[drupal-5.7.tar.gz](#)» представляет собой архив, который можно распаковать любым способным архиватором, но распаковывать его пока еще рано. Следует отметить, что здесь нет версий для разных платформ, ведь среда, в которой функционирует Drupal, по определению кроссплатформенная .

Конечно, PHP на котором функционирует Drupal, к примеру, на *nix и Windows платформах в своей структуре имеет различия, однако сам Drupal попросту не использует потенциально конфликтные возможности PHP, поэтому работает он на разных платформах практически идентично (при сходных настройках).

Также на Drupal.org имеется архив всех версий, расположен он здесь: ftp.drupal.org/files/projects. В нем вы можете обнаружить все дистрибутивы Drupal от самых первых до самых последних версий (вторых и третьих версий здесь нет, т.к. по ним уже давно не ведутся разработки). Также здесь имеются и другие сопутствующие дистрибутивы.

Для проверки целостности дистрибутива, а также для подтверждения его оригинальности часто используют **md5** сумму (**хэш**). **Md5**  столь же уникален как и отпечаток¹ – любое изменение проверяемых данных (файлов) даже на 1 бит², неминуемо приведет к изменению всей проверочной суммы (хэша). Когда вы скачиваете дистрибутив, под ссылкой на него обычно имеется длинное 32-х разрядное число (например, для PHP 5.2.5 она такая: [a1e31c0d872ab030a2256b1cd6d3b7d1](#)) – это и есть md5 сумма.

Более подробно о md5 можно узнать здесь: ru.wikipedia.org/wiki/Md5. Для определения md5 сумм существует много различных программ, одной из таких программ является **md5sum**. По этой ссылке ru.wikipedia.org/wiki/Md5sum вы можете прочитать все о программе md5sum, а также найти ее последнюю версию.

В подтверждении суммы вам также может помочь Google. После того как вы ее вычислили, введите сумму в строку поиска этой системы в качестве ключевого слова. Если в результат поиска попала ссылка на официальный сайт разработчика программы, значит сумма достоверная, и вам уже не придется искать ее непосредственно у разработчика (эта возможность особенно полезна при проверке старых версий, когда информация о них уже давно перекочевала в архив либо вообще была удалена, а в кэше  поисковой машины эти данные еще продолжают оставаться). Если поиск дал многостраничный результат, но ссылки на официальный источник нет, то вероятность подлинности суммы также высока, но если вам нужен 100% результат, тогда вам придется воспользоваться поиском на сайте разработчика данной программы. Если сумма не встретилась, т.е. поиск на Google завершился безуспешно, то скорее всего сумма неверная, либо поисковая машина не «встретила» / «закэшировала» ее раньше.

Б.2 Установка

Подготовка перед установкой

Перед установкой основных дистрибутивов требуется произвести следующие подготовительные действия:

- Убедиться в наличии прав администратора (Administrator) на той системе (том компьютере), где вы собираетесь произвести установку: прав «пользователя» («User») и «Опытного пользователя» («Power User») окажется явно недостаточно.
- В случае если установка производится под Windows Vista, следует предварительно отключить контроль учетных записей пользователей (**User Account Control – UAC**) – он со-

1 Не берутся в расчет коллизии хэш сумм.

2 1 бит – это наименьшая возможная единица измерения информации в вычислительной технике.

- дает непреодолимые препятствия при установке практически любых серьезных программ.
- Если вы используете брандмауэры сторонних производителей (т.е. не встроенный Windows **Firewall**), тогда рекомендуется отключить их на время установки и проверки программ, иначе это может привести к дополнительным проблемам¹.

Также необходимо сделать предварительную договоренность: каталог, в котором установлена ваша операционная система (это тот самый каталог, в котором находятся подкаталоги system, system32, temp, fonts и многие другие), будет в дальнейшем в тексте именоваться не «с:\windows», «с:\winnt» или еще какнибудь, а просто **%windows%**, соответственно путь к подкаталогу **system32** в этом каталоге тогда примет следующий вид: **%windows%\system32**.

Каталог, в котором обычно располагаются программы (как правило, это **Program Files**), будет иметь синоним / псевдоним **%program_files%**.

Сделано все это сугубо для унификации путей и лучшего ориентирования в файловой системе. Естественно, настоящие имена каталогов останутся прежними и никаких изменений в реальную систему вносить не потребуется!

Установка Apache

[Вернуться в раздел «Установка Drupal: Настройка Apache»]

В этом разделе не будут даваться описания многовариантных способов установки Apache, автор лишь затронет те вопросы («подводные камни»), которые необходимо решить, перед тем как вы приступите к установке непосредственно системы Drupal.

Информация по компиляции и установке веб-сервера Apache на *nix (Unix, Linux) подобные системы можно получить по следующей ссылке: <http://httpd.apache.org/docs/2.0/ru/install.html>.

Установка веб-сервера Apache под ОС Windows начинается с запуска msi-файла. Установка организована в виде классического многоэтапного (многооконного) мастера, причем количество этапов зависит от выбора типа установки («Выборочная», «Типичная», «Минимальная» и другие). Работа установщика (инсталлятора) обычно начинается с получения всех необходимых данных, т.е. «задания» пользователю вопросов, а завершается, как правило, физическим копированием необходимых файлов и последующей настройкой программы. Итак, опишем поэтапную установку веб-сервера Apache:



Прочтение приглашения. Переход к следующему окну (**Next**).



Прочтение лицензионного соглашения и принятие его условий (**I accept the terms in the license agreement**). **Next**.



Прочтение страницы описания. Страница описания – это краткое изложение того, куда и с какой целью вы попали, где можно взять последние версии этого дистрибутива и документацию на него, а также контакты с разработчиками и т.д.. **Next**.



Заполнение информации о сервере. Здесь необходимо заполнить 3 поля. Поля **Network Domain** (Сетевой Домен) и **Server Name** (Имя Сервера) – можно просто указать **localhost** ^{A2}. Поле **Administrator's Email Address** (электронный адрес администратора) – укажите ваш email адрес. Далее необходимо указать, для кого вы устанавливаете данный сервис (службу): **for all users** – для всех пользователей, **only for the current User** – только для вас. В первом случае сервис будет общедоступен для всех пользователей системы (это значит, что для получения к нему доступа в строке адреса браузера вам придется набирать «localhost» или же «localhost:80»). Во втором случае вам придется самостоятельно (вручную) запускать сервис **каждый новый раз** при загрузке ОС, при этом он будет доступен на 8080 порту (в строке адреса браузера придется набирать «localhost:8080»). Собственно выбор здесь очевиден, если на вашем компьютере добавлены пользовательские учетные записи (т.е. вы не один работаете за своим компьютером), то рекомендуется выбрать второй вариант, иначе – первый. **Next**.

¹ Например, Zonealarm с настройками «по умолчанию» не позволит PHP обращаться к MySQL.

² Также в качестве домена некоторые источники предлагают указывать точку, но по мнению автора в этом нет необходимости.

Далее предлагается выбрать тип установки: типичный (Typical) или выборочный (Custom). Типичная установка – это установка практически всего необходимого без лишних вопросов. Выборочная установка – это детальная установка, которая, в отличие от типичной, позволяет выбрать устанавливаемые компоненты. В данном случае вполне достаточным окажется выбор типичной установки. Next.

Следующее окно предлагает выбрать каталог установки. Им, как правило, оказывается следующий: «%program_files%\Apache Software Foundation\Apache2.2». Каталог установки веб-сервера Apache будем иметь псевдоним **%apache%** (естественно, имя настоящего каталога останется неизменным). Next.

В завершении нажимаем Install.

Сразу после копирования файлов веб-сервер попытается зарегистрировать себя как сервис¹ и произвести самозапуск. Если у вас активизирован **Брандмауэр Windows (Windows Firewall)**², то скорее всего он попытается заблокировать сервис-Apache, выдав предупреждение пользователю. В этом предупреждении (рис. 37) имеется 3 варианта: «Оставить заблокированным» («Keep Blocking»), «Разблокировать» («Unblock») и «Спросить Меня Позже» («Ask Me Later»). Естественно вам следует разблокировать Apache.



Рис. 37: Windows Firewall блокирует Apache.

После разблокировки Apache запустится как сервис и начнет обслуживать **80** порт (если, конечно, не возникнет других проблем). Описанная выше разблокировка также приведет и к общедоступности вашего веб-сервера. Этот побочный эффект необходимо сразу исправить – следует ограничить доступность веб-сервера только рамками локального компьютера. Решение этого вопроса следует начать с запуска Брандмауэра Windows (Windows Firewall) из Панели управления (Control Panel). Имена и положение многих элементов в Windows XP и Windows Vista будут именно такими только в том случае, если пользователь активизирует «Классический вид» этих систем. В Windows Vista дополнительно нужно будет перейти по ссылке «изменить параметры». В результате должно появиться окно под названием «Параметры брандмауэра Windows» («Windows Firewall»). В этом окне будут 3 вкладки: Общие (General), Исключения (Exceptions) и Дополнительно (Advanced). Вам нужно перейти на вкладку «Исключения». На ней вы увидите список, в котором имеется пункт «Apache HTTP Server»³. Выделите его и нажмите кнопку «Свойства» («Edit...»). Дальше следует нажать на кнопку «Изменить область...» («Change Scope...»), которая вызовет появление окна «Изменение области» («Change Scope»). Во вновь появившемся окне нужно поставить переключатель в положение «Особый список» («Custom list»), а в поле ниже вписать «127.0.0.0/255.255.255.0». Напоследок нужно не забыть во всех диалогах нажать «ОК»). После этого доступ к вашему веб-серверу извне будет перекрыт.

Закрывать доступ к Apache извне можно конечно и более простым способом – достаточно в httpd.conf вместо «Listen 80» написать «Listen localhost:80» – после перезагрузки веб-сервер Apache будет обслуживать только локальный компьютер. Этот способ подходит только для Apache и вряд ли заработа-

1 Только в данном случае, при других типах установки все может сложиться иначе.

2 Если ваш компьютер не изолирован от локальной и/или глобальной сети и при этом не имеет других брандмауэров, то деактивизация встроенного брандмауэра – ваше большое упущение.

3 А иногда и «httpd.exe».

ет на другом веб-сервере¹, поэтому описанный выше способ с брандмауэром более универсален.

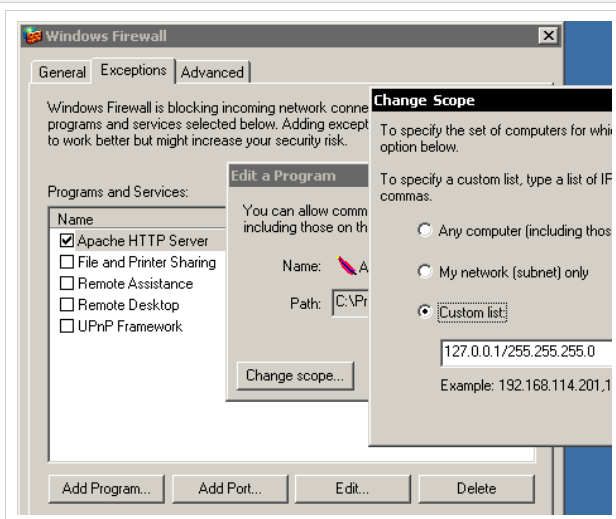


Рис. 38: Такая последовательность приведет к недоступности веб-сервера извне.

Также хочется рассказать еще об одной особенности – встроенный брандмауэр Windows действительно может защитить вашу систему, однако он не в состоянии полностью оградить вас от неприятностей, т.к. обладает лишь базовыми механизмами защиты. Дело в том, что ограничение доступа к системе извне (закрытие портов и некоторые другие нехитрые действия) не всегда является достаточным и создает лишь обманчивое чувство защищенности. Связано это с тем, что вы не блокируете «наглухо» абсолютно все, по вашей инициативе информация все равно продолжает передаваться из вашего компьютера в сеть и из сети в ваш компьютер. Хорошим примером может стать ваш веб-браузер, для которого доступ к глобальной сети продолжает оставаться прозрачным – браузер пропускает через себя весь исходящий и входящий трафик. Этот трафик может содержать очень опасные объекты. Автор и сам в свое время получил порцию троянов, посетив всего лишь форум по программированию, при этом на его локальной машине имелся и антивирус и программа-детектор для обнаружения троянов и брандмауэр был настроен на максимальный уровень и даже обновления были выполнены вовремя, однако все это не помешало серверному эксплоиту использовать уязвимость в защите браузера и проникнуть на локальную машину (конечно, антивирус и детектор потом все-таки обнаружили подвох, но было уже поздно). Поэтому крайне важно еще и фильтровать весь трафик: убирать ActiveX объекты (ссылки на них) и другие потенциально опасные вещи до того момента, как их получит непосредственно ваш браузер. Поскольку встроенный брандмауэр Windows не способен фильтровать трафик, то его использование дает лишь незначительный эффект. Автор рекомендует вам использовать брандмауэры с возможностью фильтрации, которые не только отлично защищают от атак, но еще и контролируют трафик, удаляя из его потока приватную информацию.

И еще одно замечание. Не пытайтесь сделать из обычного домашнего компьютера веб-сервер на базе Apache и PHP. Веб-сервер Apache – это мощный и очень простой инструмент, ровно как и PHP; всего лишь одной единственной ошибки в настройках этих систем будет достаточно для того, чтобы свести защищенность всей системы «на нет». Только опытный пользователь реально может развернуть более-менее защищенную систему, и только опытный администратор способен создать действительно «непробиваемый» барьер.

Также с установкой веб-сервера производится установка утилиты **ApacheMonitor** для мониторинга работы Apache. Значок этой утилиты появляется в трее сразу после установки. Функционально, эта утилита представляет собой «дополнительный рычажок», благодаря которому процесс запуска, перезапуска и остановки Apache (как локально, так и удаленно) значительно упрощается. Под Windows Vista эта утилита не работает как надо, поэтому запуск и остановку веб-сервера можно осуществить через утилиту «Службы» («Services»), которая располагается по адресу: *Панель управления (Control Panel) → Администрирование (Administrative Tools)*. Если вы не знаете, что такое трей и как должен выглядеть ApacheMonitor, тогда посмотрите правую часть рисунка 39.

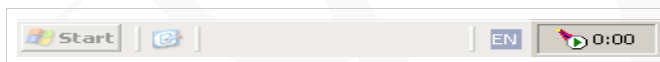



Рис. 39: Так выглядит трей и иконка ApacheMonitor

После произведения установки запустите свой веб-браузер и в строке адреса браузера введите

¹ При условии, что он не построен на базе Apache.

localhost  (или **127.0.0.1** если псевдоним localhost не задан¹) – должно появиться сообщение «It works» или что-то подобное. Это означает, что этап установки веб-сервера прошел успешно. Ниже изображены все возможные варианты адреса в браузере при данной конфигурации.

 http://localhost/	 http://127.0.0.1/
 http://localhost:80/	 http://127.0.0.1:80/

Рис. 40: Варианты записи собственного веб-адреса.

Если порт 80 будет занят, то попытка самозапуска веб-сервера ни к чему не приведет (рисунок 41). Для решения этого вопроса нужно выяснить какая программа заняла этот порт – после отстранения виновника от данного ресурса проблема с запуском веб-сервера должна исчезнуть.

```
Installing the Apache2 service
The Apache2 service is successfully installed.
Testing httpd.conf...
Errors reported here must be corrected before the service can be started.
(OS 10048)Only one usage of each socket address (protocol/network address/port)
is normally permitted. : make_sock: could not bind to address 0.0.0.0:80
no listening sockets available, shutting down
Unable to open logs
Note the errors or messages above, and press the <ESC> key to exit. 16...
```

Рис. 41: Такая ошибка возникает, когда 80-ый порт занят.

Если вам все-таки не удалось запустить веб-сервер, вы всегда можете обратиться к [отчету об ошибках](#). Этот отчет имеет имя `error.log` и располагается в подкаталоге `%apache%\logs`. Его анализ позволит выяснить суть проблемы.

В каталоге `%apache%` также имеются подкаталоги `conf` и `htdocs` – рекомендую сделать на рабочем столе ярлыки на них. В `%apache%\conf` содержатся файлы настройки веб-сервера, которые вам в дальнейшем придется изменять, а в `%apache%\htdocs` будет располагаться контент вашего веб-проекта. Каталог `%apache%\htdocs` далее будет иметь псевдоним `%htdocs%`.

[Вернуться в раздел «Установка Drupal: Настройка Apache»]

Установка PHP

[Вернуться в раздел «Установка Drupal: Настройка PHP»]

Переходим к установке PHP. Полученный ранее дистрибутив следует распаковать так, чтобы он оказался в отдельном каталоге PHP подкаталога `%program_files%`, т.е. путь по которому будут располагаться файлы PHP, в вашем случае примет следующий вид: `%program_files%\PHP`. Предварительно также оговорим, что каталог, в котором вы разместили PHP, будет иметь псевдоним `%php%`. Далее следует [связать](#) Apache и PHP, а, точнее, сделать так, чтобы Apache «знал» каким образом должен обрабатываться запрошенный у него файл, содержащий php код².

Связать Apache и PHP можно двумя способами. Первый способ – использовать PHP в качестве внешнего приложения через CGI интерфейс, а второй – использовать PHP в качестве модуля Apache (модули также могут быть статическими и динамическими – dso модулями). Здесь не будут рассматриваться различные способы связи, ровно как не будет проводиться их сравнение. Задача эта довольно непростая и малоинтересная, поэтому здесь мы ограничимся готовым решением – использованием в нашем случае модульной технологии.

Проверьте еще раз, работает ли ваш веб-сервер как надо: запустите браузер и в строке адреса введите localhost (или 127.0.0.1), если Apache действительно работает, то в результате должна появиться демонстрационная страница³. Содержание этой страницы находится в файле `index.html` каталога `httpd` (он же `%htdocs%`). Если веб-сервер не запущен, либо произошла другая ошибка, тогда браузер выдаст сообщение «Попытка соединения не удалась» (или похожее). Если ошибок нет, тогда следует [приступить](#) к связке Apache и PHP. Процесс связки можно разложить на следующие действия:

- 1 Обычно такие псевдонимы задаются в файле hosts.
- 2 Не только файлы с расширением «php» могут содержать php-код – все зависит от настройки Apache.
- 3 Содержание этой страницы отличается в зависимости от версии веб-сервера.



Начинается связка с файла `Httpd.conf`¹ – главного конфигурационного файл Apache. Расположен он в каталоге `%apache%\conf`. В этот файл необходимо внести два изменения, однако перед этим следует сделать его резервную копию, чтобы потом (в случае его неумышленной порчи) имелась возможность восстановить этот файл.

- ✓ После того как резервная копия `httpd.conf` создана, его необходимо открыть в блокноте. Далее вам предстоит внести в него два изменения.
- ✓ Изменение первое: следует найти блок, в котором располагаются все директивы `AddType`: с большой уверенностью можно сказать, что такие директивы находятся в блоке `IfModule mime_module`. В этот блок следует добавить новую директиву «`AddType application/x-httpd-php.php`», которая свяжет расширение `*.php` с определенным `mime` типом (обратите внимание, что фраза «`application/x-httpd-php.php`» состоит из двух слов). Если `php`-код располагается в файлах с расширением `*.php5`, то добавляется директива «`AddType application/x-httpd-php.php5`», если в файле `php4`, то «`AddType application/x-httpd-php.php4`» (и по аналогии с другими версиями PHP и расширениями `php`-файлов).
- ✓ Изменение второе: следует найти список, где присутствуют директивы `LoadModule`. Эти директивы располагаются вне области каких либо блоков т.к. их действие направлено на веб-сервер в целом. К этому списку необходимо добавить новую директиву «`LoadModule php5_module "%php%\php5apache2_2.dll"`», где `%php%` это путь к каталогу PHP (обратите внимание на кавычки). Для PHP 4 версии часть директивы «`php5_module`» примет несколько иной вид.
- ✓ После того как изменения были сделаны, файл следует сохранить. Теперь, для того чтобы новые настройки веб-сервера Apache вступили в силу, требуется его перезапустить. Перезапускать Apache вам придется всякий раз после изменения его конфигурации либо конфигурации PHP. Если вы сделали все правильно, тогда Apache должен запуститься без проблем (хотя благополучный запуск еще не означает, что в файле конфигурации нет логических ошибок).

Если при запуске Apache выдал ошибку и соответственно не запустился, тогда следует начать поиск ошибки. Достоверно известно, что ошибка могла появиться в файле `httpd.conf` т.к. вы изменяли только его. У Apache есть отличное средство для проверки корректности `httpd.conf`. Находиться оно в *Start (Пуск) → Programs (Программы) → Apache HTTP Server Vn → Configure Apache Server* и называется **Test Configuration**. После запуска этой программы, на экране появиться черная консоль, в которой будет указан номер строки с ошибкой (закрывается консоль клавишей `Esc`). Затем вы можете снова открыть `httpd.conf` и найти указанную строку с ошибкой (в блокноте перемещение на строку осуществляется горячей комбинацией `Ctrl-G`). Исправьте ошибку и перезапустите веб-сервер Apache.

Если найти ошибку не удалось, то просто восстановите `httpd.conf` из резервной копии, сделанной вами ранее. Далее попробуйте пройти заново все вышеописанные пункты, возможно, вы что-то пропустили.

- ✓ Теперь необходимо проверить, как прошла связка. Для этого нужно создать пустой файл и задать ему полное имя «`test.php`»² (данный файл «прикреплен» к PDF-файлу этой книги). Этот файл следует открыть обычным блокнотом и вписать в него следующий программный код³ (номера строк **1, 2, 3** и т.д. вписывать не нужно, здесь они приводятся лишь для возможности ссылки на конкретную строку через ее номер):

```
4 <?php
5     phpInfo();
6 >
```

- ✓ Затем следует поместить только что созданный файл в каталог `%htdocs%`. В этом каталоге могут содержаться и другие файлы, в том числе и `index.html` (вы можете удалить их, так чтобы в `%htdocs%` остался только ваш `test.php`).

1 Файл конфигурации представляет собой набор директив, блоков (последние, в свою очередь, также содержат директивы) и комментариев. Комментарии в файле `httpd.conf` – это строки начинающиеся с символа «`#`». Комментарии разъясняют смысл многих «загадочных» директив.
 2 Полное имя – это имя с расширением.
 3 Этот код и есть `php`-скрипт, где сам код это строка «`phpInfo();`» а ограничения `<?php` и `?>` указывают начало и окончание кода. Этот скрипт, а, точнее, функция `phpInfo` выполняет вывод конфигурации PHP.



После этого откройте веб-браузер и в строке адреса введите localhost / test.php (без пробелов) – это путь к вашему файлу на локальном веб-сервере. В результате браузер должен показать развернутую конфигурацию вашего веб-сервера, PHP и его модулей¹. Если в браузере вы видите нечто похожее на изображенное ниже, значит, данный этап вами пройден и теперь необходимо приступить к настройке самого PHP.

PHP Version X.X.X	
System	Windows NT COMPUTER 5.1 build 2600
Build Date	Aug 30 2007 07:05:48
Configure Command	cscript/nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled

Рис. 42: Результат функции phpInfo (начало).

PHP, как и Apache, имеет свой собственный конфигурационный файл (т.е. файл с настройками). Располагается он в каталоге %php% и называется `php.ini-dist`. Этот файл пока никак не участвует в работе самого PHP. Для того чтобы он стал активным, его следует скопировать в каталог %windows%² и переименовать в `php.ini`, после чего перезапустить Apache. Вы можете убедиться в результативности своих действий, снова посмотрев на конфигурационный отчет (в браузере нажмите кнопку «Обновить» или F5). В отчете имеется параметр «Loaded Configuration File». Этот параметр не должен иметь значение «(none)». На рисунке 43 изображено два состояния: первое – PHP не нашел файл конфигурации и второе – конфигурация найдена и используется в данный момент. Если вы все сделали верно, то у вас в качестве результата должен получиться второй вариант.

1	Configuration File (php.ini) Path	C:\WINDOWS
	Loaded Configuration File	(none)
2	Configuration File (php.ini) Path	C:\WINDOWS
	Loaded Configuration File	C:\WINDOWS\php.ini

Рис. 43: Файл конфигурации php.ini 1) не загружен, 2) загружен.

И осталось последнее действие – подключить необходимые модули PHP. Большая часть модулей PHP расположена в каталоге %php%\ext (например, там имеется модуль `php_mysql.dll` – это один из модулей который необходим для работы с MySQL). Также некоторым модулям необходимы дополнительные библиотеки, которые размещены на один уровень выше, т.е. в каталоге %php% (здесь есть и библиотека `libmysql.dll` без которой модуль `php_mysql` просто не сможет работать).

Однако в вопросе подключения дополнительных модулей есть один интересный момент: несмотря на то, что все эти модули и библиотеки располагаются в подкаталогах PHP, это вовсе не означает, что PHP сможет самостоятельно их найти, поэтому для него необходимо сделать дополнительную «подсказку». Есть много способов указать PHP, где же все-таки располагаются его основные модули, но самым оптимальным на взгляд автора является метод «добавления к переменной окружения Path необходимых путей». Итак, приступая к реализации этого метода, вначале пройдите следующие шаги: *Панель управления (Control Panel) → Система (System) → для Windows Vista имеется еще один пункт – это «Дополнительные параметры системы»*. После этих действий должно появиться окно «Свойства системы» («System properties»). В этом окне с вкладками вам нужно перейти на вкладку «Дополнительно» («Advanced»). После этого внизу вы увидите кнопку «Переменные среды» («Environment Variables»). Нажав ее, вы вызовете окно с аналогичным названием. Теперь уже непосредственно в этом новом окне вы увидите список под названием «Системные переменные» («System Variables»), его пример изображен на рисунке ниже. В этом списке следует найти переменную окружения **Path**. Ее необходимо «Изменить» («Edit»), добавив к ее значению два новых параметра: первым параметром станет путь к каталогу %php%, а вторым путь к каталогу %php%\ext (параметры разделяются символом «точки с запятой»). Ниже приводится пример переменной Path (естественно, в вашем случае пути и версии будут иными):

«%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\php-5.2.5;C:\Program Files\php-5.2.5\ext»

1 Интересно, что и сам PHP в данном случае выступает в качестве модуля.
2 Также для подключения `php.ini` существует переменная `PHPRC`, однако в данном случае такой способ здесь не рассматривается, т.к. не считается автором самым оптимальным.

Конечно, обычный пользователь может негодовать от столь непростых действий, и его реакция будет вполне адекватна, хотя, к примеру, процедура изменения подобных переменных еще в начале 90-х считалась рядовой пользовательской задачей :).

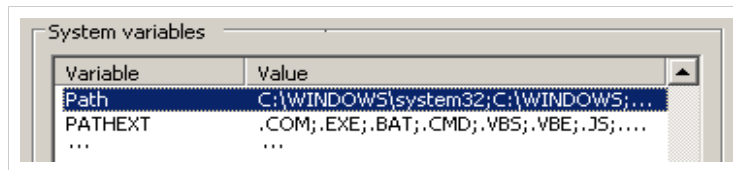


Рис. 44: Системная переменная Path находится здесь.

После этого, для того чтобы новые параметры вступили в силу, следует перезагрузить компьютер. Теперь вы можете попробовать активизировать дополнительные модули, сделать это сейчас будет довольно просто¹, т.к. сложные подготовительные этапы уже пройдены. В качестве примера можно рассмотреть модуль mysql. Включение любого модуля начинают с открытия конфигурационного файла php.ini и перехода к разделу директив «extension=». Если директивы начинаются с символа «;», значит, они закомментированы² (не работают). В этом разделе имеется и директива «включения» mysql – «;extension=php_mysql.dll», в ней необходимо убрать символ «;». Далее php.ini сохраняется, а веб-сервер Apache перезагружается. Если все прошло успешно, то в отчете localhost/test.php должен появиться новый блок с именем «mysql» (рис. 45).

mysql	
MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.0.45

Рис. 45: MySQL имеет в отчете phpInfo свой собственный блок.

Если модуль mysql не загрузился, значит, библиотека libmysql.dll не доступна для PHP. Недоступна она может оказаться по разным причинам, но наиболее вероятная – это неправильное задание путей к модулям PHP. Для эксперимента попробуйте скопировать библиотеку libmysql.dll в папку %system32% и перезапустить веб-сервер. Если после этого mysql заработал, значит, поставленный выше диагноз подтвердился и следует перейти к проверке ранее введенных путей.

При возникновении проблем при загрузке вам может помочь одна директива в php.ini: откройте php.ini и поменяйте «display_startup_errors = off» на «display_startup_errors = on», теперь при возникновении проблем при загрузке будет появляться понятное сообщение.

Если вы желаете подключить другие модули, то следует разкомментировать директивы, «включающие» их.

Также ранее не было сказано, какое еще «наполнение» имеет файл php.ini (дело в том, что php.ini по структуре более простой, нежели httpd.conf, поэтому предполагалось, что с ним у пользователя не возникнет каких-либо трудностей). Необходимо отметить, что php.ini содержит и другие внутренние директивы PHP. Кроме того в нем имеются и соответствующие блоки с директивами для каждого подключаемого модуля. Так, например, для модуля mysql имеется блок «[MySQL]», где содержатся следующие директивы: mysql.allow_persistent, mysql.max_persistent, mysql.max_links и другие. Как правило, значения подобных директив либо уже заданы, либо они отсутствуют (имеют значение «по умолчанию»).

[\[Вернуться в раздел «Установка Drupal: Настройка PHP»\]](#)

Установка MySQL

[\[Вернуться в раздел «Установка Drupal: Настройка MySQL»\]](#)

И последнее звено – это **MySQL-сервер** (или просто **SQL-сервер**). Установка MySQL-сервера на-

1 За исключением некоторых редкоиспользуемых модулей.
2 В файле php.ini вместо «#» используется именно этот символ.

чинается с распаковки скачанного дистрибутива и запуска инсталлятора (msi или exe файл). Теперь по шагам опишем его последующую инсталляцию:



В начале своей работы инсталлятор предлагает вам выбрать тип установки – всего имеется три типа: типичный (Typical), выборочный (Custom) и полный (Complete). Как и в случае с Apache, мы прибегнем здесь к типичной установке (т.к. для наших целей функциональных возможностей данного варианта окажется вполне достаточно), т.е., ничего не меняя, просто нажимаем Next.



Далее в следующем окне вам будет показан ваш предыдущий выбор типа установки и каталог установки, изменить который не получится (если вы желаете задать иной каталог, тогда вам необходимо выбирать не типичную установку, а выборочную). Затем следует нажать Install, после чего произойдет копирование файлов MySQL-сервера в вашу систему. В качестве каталога установки SQL-сервера будет использоваться псевдоним %mysql%.



После копирования файлов вам придется пройти еще два промо-окна (через Next), где рекламируются корпоративные решения от производителя MySQL-сервера и закончить их просмотр кнопкой Finish (только не спешите в последнем окне убирать флажок «Настроить MySQL Сервер сейчас» («Configure the MySQL Server now»)).





Теперь установка считается оконченной и можно приступать к процессу конфигурирования SQL-сервера. Если ранее вы не сняли соответствующий флажок, то сразу после установки появится диалог конфигурации. На первом шаге этого диалога вам предложат сделать выбор между «Стандартной конфигурацией» («Standart configuration») и «Детальной конфигурацией» («Detailed configuration»). Не долго думая, ставьте переключатель на стандартную конфигурацию (по умолчанию флажок выставлен в положение «Детальной конфигурации») и кнопкой «Next» переходите к следующему этапу.




На следующем шаге (в следующем окне) конфигурирования вам предстоит выставить параметры запуска MySQL-сервера (имеется ввиду запуск при загрузке компьютера). В этом окне имеется связанная группа из трех элементов. Первый элемент – это флажок с надписью «Установить Как Сервис Windows» («Install As Windows Service»). Если его не снимать, то MySQL будет установлен как сервис, т.е. будет постоянно находиться в памяти, будет всегда готов к обслуживанию и будет доступен для всех пользователей системы. Также здесь имеется поле «Имя Сервера» («Server Name»), которое позволяет задать сервису-MYSQL другое имя. И третий элемент – это флажок «Запускать MySQL Сервер автоматически» («Launch the MySQL Server automatically»). Изменять все описанное выше (снимать галочки, изменять значения полей) автор не рекомендует, иначе запуск SQL-сервера и работа с ним превратятся для вас в очередную проблему. Также в этом окне имеется флажок «Включить Bin каталог в переменную окружения PATH» («Include Bin Directory in Windows PATH»). Зачем это нужно? Дело в том, что в директории %mysql%\Bin имеется разного рода утилиты, которые работают, как правило, в командной строке (консоли), например, утилита **mysqladmin.exe**. Включение этого пути в переменную окружения Path позволяет вызывать эти программы, указав лишь их имя (не добавляя полный путь)¹. Аналогичным образом вы поступили ранее с PHP, когда в Path внесли пути к его расширениям. В дальнейшем, вам возможно потребуется такая возможность, когда вы при помощи **mysqladmin** пожелаете сменить свой пароль на пустой и т.п., поэтому в данном случае необходимо поставить галочку. Нажимая «Next», переходим к следующему этапу.

1 Если вы уже находитесь в каталоге Bin, то указывать вам весь путь также не потребуется.

 В следующем окне вам предстоит задать пароль для учетной записи **root** вашего SQL-сервера. Для тех, кто не знает: в данном случае **root** (регистр имеет значение) – это администратор MySQL. Для того чтобы было еще более понятно: администратор – это субъект, наделенный эксклюзивными правами. Итак, вам следует задать пароль (пустой пароль конфигуратор не примет) – задайте пароль. Далее в окне располагается флажок «Активизировать root-доступ с удаленной машины» («Enable root access from remote machine»). Если вы установите в нем галочку, то у «клиентов» извне появится возможность подключаться к вашему MySQL-серверу. Такого допустить нельзя, поэтому флажок должен остаться снятым. И последним элементом в этом окне останется флажок «Создать Анонимный Аккаунт» («Create An Anonymous Account») – его установка приведет к возможности анонимного подключения к MySQL-серверу, т.е. подключению к нему любого желающего, причем без указания пароля. На взгляд автора такая возможность, как и описанная ранее, – это очередная «медвежья услуга» для пользователя, поэтому лучше воздержаться от подобного и оставить флажок снятым. Нажимая «Next», переходим к последнему этапу.

 И последний этап – это попытка применения выбранной ранее конфигурации. Здесь имеется четыре виртуальных пункта:

- «Подготовка конфигурации» («Prepare configuration»),
- «Запись файла конфигурации» («Write configuration file»),
- «Запуск службы | сервиса» («Start service») и
- «Применение установок безопасности» («Apply security settings»).

 Когда вы нажмете кнопку **Execute**, мастер попытается последовательно выполнить каждый пункт. Если все прошло успешно, то напротив каждого пункта вы увидите синюю галочку (как показано в примере выше). Если возникнет ошибка, то вы тут же узнаете о ней – вместо галочки будет красный крестик, а внизу появится описание ошибки.

Конфигурация может не пройти с первого раза – иногда это случается. Если конфигурация сорвалась с первой же попытки после установки, то вам вряд ли удастся произвести реконфигурацию. В этом случае рекомендую снова переустановить MySQL (это займет всего 5 минут работы) – после повторной установки конфигурация обычно проходит без эксцессов.

Теперь с установкой и конфигурацией покончено¹, остается только проверить, сможет ли PHP «обращаться» к MySQL. Для этого вам придется написать миниатюрный php-скрипт: создайте пустой файл с именем «testmysql.php» (данный файл «прикреплен» к PDF файлу этой книги) и добавить в него код, показанный ниже (в коде вместо символа «***» подставьте пароль, который вы задали при конфигурации MySQL-сервера).

```

7  <?php
8      $con=mysql_connect('localhost','root','***');
9      $res=mysql_query('select @@version;', $con);
10     echo implode(' ',mysql_fetch_row($res));
11     ?>

```

Затем поместите данный файл в каталог `%htdocs%`. Теперь откройте браузер и в строке веб-адреса введите `localhost/testmysql.php`. Если все прошло успешно, то в качестве результата вы увидите фразу «5.0.45-community-nt», где «5.0.45», в данном случае это номер версии MySQL-сервера.

Если в результате появилась строка «Fatal error: Call to undefined function mysql_connect()», значит PHP не знает функции `mysql_connect`, что свидетельствует о том, что модуль **mysql** скорее всего не подключен (ранее описывалось, как его подключить).

Если выводится строка «Access denied for user 'root'@'localhost'», значит, возможно, при подключении вы неправильно указали пароль.


Теперь следует приступить к установке **MySQL Gui Tools**. Как правило после запуска **msi** или **exe** файла в работу вступает мастер установки. Всю установку Gui Tools можно разделить на следующие этапы:

1. Прочтение лицензионного соглашения и принятие его условий.
2. Выбор каталога установки. Наиболее приемлемым вариантом здесь окажется выбор зна-

¹ По умолчанию ваш MySQL-сервер занял порт 3306.

чений по умолчанию.

3. Выбор типа установки. Здесь имеются варианты: полный (Complete) и выборочный (Custom). Если вы не желаете думать над тем, какое именно приложение стоит устанавливать, а какое нет, тогда используйте полную установку.
4. Завершающий этап: Install.

После установки MySQL Gui Tools попробуйте запустить MySQL Query Browser (это приложение уже ранее упоминалось в тексте). Сразу после запуска Query Browser «попросит» вас выполнить аутентификацию , выдавая соответствующее окно. В этом окне введите информацию в следующие поля: в поле Server Host введите «localhost», в поле Username введите «root», в поле Password тот пароль, который вы задали при конфигурации MySQL-сервера и в поле Default Schema введите «mysql». Все, после того как вы нажмете OK, вы должны попасть в главное окно Query Browser'a.

И самое главное: не забудьте удалить файлы `test.php` и `testmysql.php` из каталога `%htdocs%`!!!

[Вернуться в раздел «Установка Drupal: Настройка MySQL»]

Алфавитный указатель

httpd.conf.....	39, 100, 103, 105
php.ini.....	40, 51, 104, 105

