

**Максим Кузнецов
Игорь Симдянов**

РНР

НА ПРИМЕРАХ

2-е издание

Санкт-Петербург
«БХВ-Петербург»
2012

УДК 681.3.068+800.92РНР
ББК 32.973.26-018.1
К89

Кузнецов, М. В.

К89 РНР на примерах / М. В. Кузнецов, И. В. Симдянов. —
2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 400 с.: ил.

ISBN 978-5-9775-0445-4

Рассмотрены приемы программирования на РНР, позволяющие разрабатывать современные эффективные Web-приложения. Используются многочисленные примеры, взятые из реальной практики. Первое издание книги под названием «РНР 5 на примерах» охватывало только пятую версию языка. Второе издание полностью обновлено и переработано. Рассмотрено взаимодействие РНР-приложений с Web-сервером Apache и СУБД MySQL, извлечение информации с удаленных серверов, взаимодействие с браузером посредством технологии AJAX, использование библиотеки jQuery. Показаны тонкости работы с HTTP-протоколом, нюансы Rewrite-преобразований, особенности применения интеллектуальных агентов на РНР. Приведены примеры защиты Web-приложений, работы с графикой, Flash и PDF-документами, оптимизации кода и решения ряда других важных задач.

Для программистов и Web-разработчиков

УДК 681.3.068+800.92РНР
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Кашлакова</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн серии	<i>Игоря Цырульникова</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Подписано в печать 30.09.11.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 32,25.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953 Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0445-4

© Кузнецов М. В., Симдянов И. В., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение	9
Где искать помощи	10
Благодарности	10
Глава 1. Установка Web-сервера Apache, интерпретатора PHP и СУБД MySQL	11
1.1. Что нужно, чтобы запустить PHP-скрипт	11
1.2. Можно ли обойтись без утомительной настройки серверов и PHP	12
1.3. Где взять дистрибутивы	12
1.3.1. Дистрибутив PHP	13
1.3.2. Дистрибутив Apache.....	13
1.3.3. Дистрибутив MySQL.....	14
1.4. Установка Web-сервера Apache.....	14
1.5. Настройка виртуальных хостов	19
1.6. Управление запуском и остановкой Web-сервера Apache	22
1.7. Управление Apache из командной строки	24
1.8. Установка PHP	24
1.9. Что предпринять, если Web-сервер не запускается	29
1.10. Установка СУБД MySQL	30
1.11. Послеустановочная настройка MySQL	35
1.12. Проверка работоспособности MySQL	42
1.13. Управление запуском и остановкой MySQL	43
1.14. Конфигурационный файл <code>my.ini</code>	44
1.15. Связывание PHP и MySQL.....	48
1.16. Настройка командной строки для <code>mysql</code>	50
1.17. Поддержка русского языка	53
Глава 2. Хитрости конфигурирования среды	55
2.1. PHP	55
2.1.1. Структура конфигурационного файла <code>php.ini</code>	55
2.1.2. Параметры языка <i>PHP</i>	56
2.1.3. Директивы безопасности	59
2.1.4. Настройка подсветки PHP-кода	64
2.1.5. Кэш файловой системы.....	66

2.1.6. Взаимодействие с клиентом	67
2.1.7. Ограничение ресурсов	67
2.1.8. Обработка ошибок.....	68
2.1.9. Обработка входящих и исходящих данных.....	74
2.1.10. Загрузка файлов.....	78
2.1.11. Сетевой доступ	79
2.1.12. Подключение расширений.....	79
2.1.13. Настройка сессии.....	80
2.1.14. Настройка даты и времени	81
2.1.15. Изменение настроек <code>php.ini</code> средствами Apache.....	81
2.1.16. Функции управления интерпретатором <i>PHP</i>	83
2.1.17. PHP как консольный интерпретатор.....	86
2.1.18. Запуск скриптов в назначенное время	89
2.2. Apache	92
2.2.1. Конфигурационный файл <code>.htaccess</code>	92
2.2.2. Установка кодировки по умолчанию.....	93
2.2.3. Список файлов в каталоге.....	95
2.2.4. Выполнение PHP-кода в HTML-файлах.....	98
2.2.5. Страницы ошибок Web-сервера Apache.....	98
2.2.6. Переадресация	100
2.2.7. Запрет на доступ к ресурсу.....	101
2.2.8. Запрет загрузки файлов.....	103
2.2.9. Защита сайта паролем	103
2.2.10. Преобразование URL-адресов.....	105
2.3. MySQL	116
2.3.1. Работа с утилитой <i>mysql</i>	116
2.3.2. Восстановление утерянного пароля.....	120
2.3.3. Удаленный доступ к <i>MySQL</i>	121
2.3.4. Управление привилегиями пользователей	122
2.3.5. Ограничение на число соединений с сервером и число запросов.....	127
2.3.6. Перенос каталога данных на другой диск	128
2.3.7. Перенос баз данных с одного сервера на другой.....	128
2.3.8. Настройка <code>phpMyAdmin</code>	131
Глава 3. Массивы.....	133
3.1. Создание массива.....	134
3.1.1. Конструкция <i>array()</i>	134
3.1.2. Непосредственное создание элементов.....	136
3.1.3. Создание массива: приведение типа	137
3.1.4. Использование специализированных функций.....	137
3.1.5. Многомерные массивы	141
3.2. Вывод массива на печать	142
3.3. Количество элементов в массиве.....	145
3.4. Переменная или массив?	147
3.5. Существует ли элемент массива?	147
3.6. Как получить список всех индексов массива?.....	148
3.7. Содержит ли массив заданный элемент?	149
3.8. Поиск ключа по значению.....	151
3.9. Сумма элементов массива	152

3.10. Случайные элементы массива.....	152
3.11. Слияние массивов	154
3.12. Преобразование каждого элемента массива.....	157
3.13. Получение уникальных элементов массива.....	159
3.14. Преобразование элементов массива в переменные	159
3.15. Сортировка массивов.....	162
3.16. Вывод иерархических данных	170
3.17. Суперглобальные массивы.....	173
3.18. Суперглобальный массив <code>\$_GET</code>	174
3.19. Постраничная навигация.....	178
3.20. Суперглобальный массив <code>\$_POST</code>	181
3.21. Передача файлов на сервер. Суперглобальный массив <code>\$_FILES</code>	182
3.22. Загрузка произвольного количества файлов	186
3.23. Cookie. Суперглобальный массив <code>\$_COOKIE</code>	188
3.24. Включен ли механизм Cookie в браузере?.....	191
3.25. Сессии. Суперглобальный массив <code>\$_SESSION</code>	192
3.26. Суперглобальные массивы. Массив <code>\$_SERVER</code>	195
3.26.1. Элемент <code>\$_SERVER['DOCUMENT_ROOT']</code>	196
3.26.2. Элемент <code>\$_SERVER['HTTP_ACCEPT']</code>	196
3.26.3. Элемент <code>\$_SERVER['HTTP_ACCEPT_LANGUAGE']</code>	197
3.26.4. Элемент <code>\$_SERVER['HTTP_HOST']</code>	198
3.26.5. Элемент <code>\$_SERVER['HTTP_REFERER']</code>	198
3.26.6. Элемент <code>\$_SERVER['HTTP_USER_AGENT']</code>	199
3.26.7. Элемент <code>\$_SERVER['REMOTE_ADDR']</code>	199
3.26.8. Элемент <code>\$_SERVER['SCRIPT_FILENAME']</code>	200
3.26.9. Элемент <code>\$_SERVER['SERVER_NAME']</code>	200
3.26.10. Элемент <code>\$_SERVER['REQUEST_METHOD']</code>	201
3.26.11. Элемент <code>\$_SERVER['QUERY_STRING']</code>	201
3.26.12. Элемент <code>\$_SERVER['PHP_SELF']</code>	202
3.26.13. Элемент <code>\$_SERVER['REQUEST_URI']</code>	202

Глава 4. Файлы и каталоги..... 203

4.1. Создание файлов	203
4.2. Создание файлов с уникальными именами	208
4.3. Копирование, переименование и удаление файлов	209
4.4. Чтение содержимого файлов	210
4.5. Запись файлов	217
4.6. Размер файла	220
4.7. Разбивка файла на части.....	221
4.8. Редактирование файлов на удаленном сервере	222
4.9. Счетчик загрузок файлов	225
4.10. Сохранение текстовых и графических файлов.....	227
4.11. Определение количества строк в файле.....	229
4.12. Случайный вывод из файла.....	230
4.13. Сортировка содержимого текстового файла	231
4.14. Каталоги.....	233
4.15. Список файлов и подкаталогов в каталоге	234
4.16. Количество файлов в каталогах.....	238
4.17. Копирование содержимого одного каталога в другой.....	240

4.18. Удаление каталога со всем содержимым.....	241
4.19. Подсчет объема памяти, занимаемой каталогом.....	242
Глава 5. Сетевые возможности	243
5.1. Загрузка удаленного файла	243
5.2. Что такое сокеты	244
5.3. Получение HTTP-заголовков.....	248
5.4. Определение размера файла на удаленном хосте	252
5.5. Библиотека CURL	252
5.6. Получение точного времени	259
5.7. Извлечение ссылок Yandex	260
5.8. Извлечение ссылок Google.....	261
5.9. Курс валют Центрального банка РФ	262
5.10. Отправка данных методом POST	266
5.11. Передача реферера.....	269
5.12. Передача пользовательского агента	272
5.13. Передача cookie.....	273
5.14. Определение IP-адреса по сетевому адресу.....	276
5.15. Определение сетевого адреса по IP-адресу	278
5.16. Получение информации об IP-адресе	278
5.17. Отправка почтового сообщения	281
5.18. Отправка писем с вложением	282
5.19. Отправка писем со встроенными изображениями	285
Глава 6. Введение в MySQL.....	288
6.1. Что такое SQL	288
6.2. Создание, редактирование и удаление базы данных	289
6.3. Создание, редактирование и удаление таблиц	291
6.4. Вставка данных в таблицу. Оператор <i>INSERT</i>	295
6.5. Вставка уникальных значений.....	297
6.6. Извлечение данных. Оператор <i>SELECT</i>	298
6.6.1. Переименование столбцов. Ключевое слово <i>AS</i>	298
6.6.2. Условная выборка. Ключевое слово <i>WHERE</i>	299
6.6.3. Сортировка записей. Ключевое слово <i>ORDER BY</i>	300
6.6.4. Вывод записей в случайном порядке.....	301
6.6.5. Ограничение выборки. Ключевое слово <i>LIMIT</i>	301
6.7. Обновление данных. Оператор <i>UPDATE</i>	302
6.8. Удаление записей. Оператор <i>DELETE</i>	303
Глава 7. Сложные вопросы MySQL	305
7.1. Индексы и оценка производительности	305
7.2. Кодировки.....	307
7.3. Функции MySQL	310
7.3.1. Версия MySQL.....	310
7.3.2. Количество записей в таблице	311
7.3.3. Максимальное и минимальное значение в таблице	312
7.3.4. Сумма значений столбца.....	313
7.3.5. Форматирование даты.....	313
7.3.6. Вычисление возраста человека	315
7.3.7. Преобразование IP-адреса в число.....	316

7.4. Получение уникальных значений.....	317
7.5. Вложенные запросы.....	318
7.6. Вложенные запросы, возвращающие несколько строк	322
7.6.1. Ключевое слово <i>IN</i>	322
7.6.2. Ключевое слово <i>ANY</i>	324
7.6.3. Ключевое слово <i>ALL</i>	325
7.7. Групповые условия. Ключевое слово <i>HAVING</i>	326
7.8. Многотабличные запросы <i>SELECT</i>	328
7.9. Выбор случайных точек из таблицы	330
7.10. Многотабличный запрос <i>DELETE</i>	331
7.11. Удаление повторяющихся записей.....	333
Глава 8. PHP и MySQL	335
8.1. Установка соединения с базой данных	335
8.2. Выбор базы данных	338
8.3. Выполнение SQL-запросов	339
8.4. Получение результатов запроса.....	341
8.5. Количество строк в таблице.....	348
8.6. Экранирование данных. SQL-инъекции.....	350
Глава 9. PHP и AJAX.....	357
9.1. Что такое AJAX.....	357
9.2. Что такое jQuery.....	358
9.3. Обработка событий.....	360
9.4. Манипуляция содержимым страницы.....	362
9.5. Асинхронное обращение к серверу	366
9.6. AJAX-обращение к базе данных.....	367
9.7. Отправка данных методом POST	372
9.8. Двойной выпадающий список	377
9.9. Запоминание состояний флажков.....	380
Глава 10. Разные вопросы применения PHP	383
10.1. Локаль	383
10.2. Сериализация.....	385
10.3. Уменьшение изображения	386
10.4. Водяные знаки на изображении.....	387
10.5. Запуск внешних программ	389
Заключение	391
Предметный указатель	392

Введение

Книги, посвященные языкам программирования, как правило, условно можно поделить на две большие группы: одни рассчитаны на читателя, не знакомого с языком программирования, другие излагают профессиональные особенности, ориентированные на программистов с базовыми основами. Данная книга относится к последнему типу (если какие-то конструкции вам покажутся слишком сложными или неизвестными, вы всегда можете обратиться к нашему самоучителю по PHP 5/6, излагающему язык последовательно).

С одной стороны, это означает, что в книге вы не встретите последовательного изложения языка от самого простого к самому сложному. Зато независимо от того, начинаете вы постигать азы Web-программирования или уже освоили язык и ищете свежих идей, вы сможете начать использовать примеры из книги практически сразу. Подавляющее большинство примеров из книги взято из реальной практики создания коммерческих Web-приложений и может без адаптации быть использовано на вашем сайте. По мере чтения книги мы познакомимся с PHP, начиная с самых низкоуровневых операций, таких как работа с файлами, директориями, строками, и заканчивая использованием PHP в AJAX-приложениях.

Язык программирования не существует сам по себе, он является частью технологии создания Web-приложений. Современному разработчику необходимо разбираться в конфигурировании Web-сервера Apache и его модулей, управлении и проектировании базами данных, представлять особенности работы сети Интернет и сетевых протоколов, владеть регулярными выражениями, языком разметки HTML, каскадными таблицами стилей CSS и клиентским языком JavaScript. Создавая эту книгу, мы постарались на коротких примерах показать связь PHP со всеми этими технологиями.

Для того чтобы изучить конкретный язык программирования, его синтаксис и шаблоны, обычно уходит несколько месяцев. Для того чтобы овладеть технологией, позволяющей создавать современные приложения в той или иной области, могут потребоваться годы. Данная книга предназначена для того, чтобы сократить это время и показать PHP в окружении других языков программирования и технологий.

Где искать помощи

Дополнительные материалы можно также найти на группе сайтов IT-студии SoftTime, руководителями которой являются авторы книги:

- <http://www.softtime.ru> — главный сайт;
- <http://www.softtime.org> — проекты студии;
- <http://www.softtime.info> — консультации авторов книг;
- <http://www.softtime.biz> — услуги студии.

Если по мере чтения у читателей возникнут технические вопросы, обсудить их можно на PHP-форуме авторов <http://www.softtime.ru/forum/>. Вопросы правового характера обсуждаются на форуме <http://www.softtime.org/forum/>.

Осветить все функции, расширения PHP, а также смежные дисциплины, необходимые для создания Web-приложений, в одной книге невозможно — объем их чрезвычайно велик. Поэтому мы регулярно выпускаем книги по Web-программированию, затрагивающие самые разнообразные вопросы, начиная с проблем безопасности и заканчивая психологическими аспектами профессии программиста. С полным списком наших книг можно ознакомиться по ссылке <http://www.softtime.ru/php5/index.php>.

Благодарности

Авторы выражают признательность сотрудникам издательства "БХВ-Петербург", благодаря которым наша рукопись увидела свет.

Авторы также благодарны всем читателям, приславшим отзывы на наши предыдущие книги, а также всем участникам форума SoftTime.



ГЛАВА 1

Установка Web-сервера Apache, интерпретатора PHP и СУБД MySQL

Для того чтобы создать удобную среду программирования, недостаточно установить инсталляционные пакеты, потребуется их тщательное конфигурирование. Данная глава посвящена устройству рабочего места Web-разработчика.

1.1. Что нужно, чтобы запустить PHP-скрипт

Программирование на PHP отличается от программирования на других языках достаточно сложной системой настройки среды. Первым шагом в освоении Web-технологий является воспроизведение рабочей среды на компьютере разработчика. Для этого потребуется запустить и настроить несколько программных компонентов.

Доступность Web-приложений и HTML-файлов в сети Интернет обеспечивается Web-сервером, который по протоколу HTTP выдает их любому клиенту, правильно оформившему запрос. Наиболее популярным сервером, используемым совместно с PHP, долгие годы остается Web-сервер Apache.

Интерпретатор PHP представляет собой либо внешнюю CGI-программу, либо динамическую библиотеку, которую необходимо подключить к Web-серверу, чтобы вместо кода PHP-скриптов клиенту выдавались результаты его выполнения. Ситуация осложняется тем, что к PHP-интерпретатору могут подключаться различные расширения, также оформленные в виде динамических библиотек. Для большинства современных приложений требуется довольно много внешних расширений, поэтому без ручного редактирования конфигурационных файлов довольно трудно обойтись.

Практически ни одна крупная программа не обходится без использования базы данных. Традиционно совместно с Web-сервером Apache и интерпретатором PHP используется СУБД MySQL.

Так как и язык программирования PHP, и Web-сервер Apache, и MySQL-сервер первоначально разработаны для UNIX-подобных операционных систем, их настройка и администрирование сводятся к редактированию конфигурационных файлов и работе в командной строке. Такой подход часто сбивает с толку программистов, не имеющих опыта работы в UNIX.

В данной главе рассматриваются приемы создания удобного рабочего места на локальной машине, которое позволит тестировать сайты без размещения их в Интернете на серверах хост-компаний.

1.2. Можно ли обойтись без утомительной настройки серверов и PHP

Конфигурирование серверов и PHP, настройка режима их работы являются важными разделами, позволяющими увереннее чувствовать себя Web-разработчикам, быстро осуществлять локализацию ошибок, попыток взлома, настраивать наиболее эффективную работу Web-приложений. Однако даже локальная настройка Web-серверов, а тем более настройка их для работы в сети Интернет, является отдельной областью деятельности, требующей зачастую не меньшей отдачи, чем программирование. Поэтому Web-разработчики зачастую отказываются от изучения конфигурирования серверов и их взаимодействия, оставляя эту область системным администраторам. Чтобы не завязнуть в многочисленных настройках серверов, правилах их связывания друг с другом, Web-разработчики прибегают к готовым пакетам, где вся настройка выполнена профессиональными администраторами. Такой пакет остается только загрузить и установить, после чего можно сразу приступить к работе с языком программирования. Наиболее популярным на сегодняшний день пакетом, объединяющим PHP, Web-сервер Apache и СУБД MySQL, является Denwer, загрузить который можно по ссылке <http://www.denwer.ru/>.

ЗАМЕЧАНИЕ

Перед установкой сверьтесь с регулярно обновляемыми деталями по адресу http://www.softtime.ru/article/index.php?id_page=9. По ссылке вы также можете загрузить готовые конфигурационные файлы `httpd.conf` (для Apache) и `php.ini` (для PHP). Также вы получите ответы на свои вопросы в форуме http://www.softtime.ru/forum/index.php?id_forum=5.

1.3. Где взять дистрибутивы

Поиски дистрибутивов следует начинать с официальных сайтов данных программных продуктов (табл. 1.1).

Таблица 1.1. Официальные сайты PHP, Apache и MySQL

Программный продукт	Web-сайт
Интерпретатор PHP	http://www.php.net
Web-сервер Apache	http://www.apache.org
СУБД MySQL	http://dev.mysql.com

Поскольку версии программных продуктов постоянно меняются, и указать постоянную ссылку невозможно, приведем алгоритм загрузки свежей версии каждого из программных продуктов.

1.3.1. Дистрибутив PHP

Загрузив страницу <http://www.php.net>, необходимо перейти на страницу **downloads**, которая на момент написания книги имела адрес <http://www.php.net/downloads.php>. На данной странице PHP доступен в формате исходных кодов **Complete Source Code**, запуск которого потребует предварительной компиляции. Для того чтобы загрузить предкомпилированный вариант **Windows Binaries**, необходимо проследовать по ссылке <http://windows.php.net/download/>. Для каждой из версий доступны два варианта компиляции в меню **Which version do I choose?** при помощи Visual Studio 6 и Visual Studio 9. Для установки на Web-сервер Apache нам потребуется версия VC6.

ЗАМЕЧАНИЕ

Версия VC9 используется для Web-сервера Windows IIS.

Необходим также выбор из вариантов **Non Thread Safe** и **Thread Safe**. **Thread Safe** используется для установки PHP в качестве модуля, **Non Thread Safe** — в качестве внешнего CGI-приложения. В дальнейшем повествование будет вестись в предположении, что выбрана версия для установки PHP в качестве модуля (**Thread Safe**).

Для загрузки лучше выбрать автоматический установщик (**Installer**), который содержит наиболее полную версию PHP. Переход по ссылке приведет на страницу со списком зеркал, откуда можно загрузить текущую версию PHP. На данной странице можно выбрать любую ссылку; разница в них заключается лишь в географическом расположении серверов. Разумнее выбрать сервер, который расположен в Российской Федерации (**Russian Federation**). Ссылки на такие серверы помечены значком российского флага.

ЗАМЕЧАНИЕ

С сайта <http://www.php.net> можно загрузить справочник по языку PHP, в том числе частично переведенный на русский язык. Для этого необходимо перейти на страницу **documentation** (<http://www.php.net/docs.php>) и там выбрать ссылку **downloads** (<http://www.php.net/download-docs.php>). На странице загрузки будет представлена таблица, позволяющая загрузить справочник в трех форматах для каждого из доступных языков. Для пользователей Windows лучше воспользоваться **chm**-форматом, удобным для поиска в справочнике.

1.3.2. Дистрибутив Apache

Для того чтобы тестировать и просматривать сайты на локальной машине, необходимо установить Web-сервер Apache. Поиск дистрибутива следует начать с официальной страницы <http://www.apache.org>, выбрав ссылку **Download** (<http://www.apache.org/dyn/closer.cgi>). На открывшейся странице будет представлен список HTTP- и FTP-серверов, откуда можно загрузить Web-сервер Apache.

При поиске следует помнить, что Apache может также называться **httpd**, по имени его UNIX-демона. На зеркалах обычно много различных файлов, например:

- [httpd-2.2.15.tar.gz](#) — для Linux, в котором программы принято распространять в исходных кодах;

- `httpd-2.2.15-win32-x86-openssl-0.9.8m-r2.msi` — установщик бинарных файлов под архитектуру x86 для Windows (win32) с поддержкой SSL (`openssl-0.9.8m-r2`) сервер Apache (`httpd`) версии 2.2.15 — именно его и следует загрузить.

ЗАМЕЧАНИЕ

Скачать дистрибутивы Apache для Windows можно по адресу <http://www.sai.msu.ru/apache/httpd/binaries/win32/>.

1.3.3. Дистрибутив MySQL

Дистрибутив MySQL можно загрузить со страницы <http://dev.mysql.com/downloads/>. На момент написания книги для загрузки были доступны две версии СУБД MySQL:

- MySQL 5.1 — рекомендуемая версия MySQL;
- MySQL 5.5 — разрабатываемая версия, находящаяся в стадии бета-тестирования.

Как правило, для загрузки доступны две-три версии сервера, из которых следует выбрать рекомендуемый (Generally Available Release) дистрибутив. На открывшейся странице будет представлен выпадающий список с операционными системами (по умолчанию открывается страница для Windows), для которых выводится список доступных дистрибутивов. Для Windows дистрибутивы представлены в трех вариантах (каждый из которых откомпилирован для 32- и 64-битной архитектуры):

- MSI Installer Essentials — урезанная версия дистрибутива, из которой удалены все вспомогательные утилиты;
- MSI Installer — полная версия дистрибутива MySQL, включающая автоматический установщик;
- ZIP Archive — полная версия MySQL без автоматического установщика.

Рекомендуется выбрать дистрибутив MSI Installer несмотря на то, что по объему он превышает все остальные дистрибутивы из Windows-серии. Наше изложение материала будет основано именно на этом дистрибутиве.

Помимо самого дистрибутива MySQL имеет смысл загрузить графические клиенты для работы с MySQL-сервером, которые свободно распространяются на сайте <http://dev.mysql.com/downloads/gui-tools/5.0.html>.

1.4. Установка Web-сервера Apache

Apache под Windows доступен в исходных кодах и в виде откомпилированного пакета. Исходные коды могут понадобиться в том случае, если вы решили при установке перекомпилировать Apache. Перекомпилирование исходных кодов необходимо, когда нужно исключить из исполняемой версии неиспользуемые функции или включить поддержку функций, не входящих в стандартную поставку. В этом случае необходимо наличие установленной среды разработки Microsoft Visual Studio. Если стандартная компиляция сервера вас устраивает, можно приступить к установке.

После двойного щелчка на имени файла `httpd_версия_win32_*_.msi` запустится Microsoft Installer (рис. 1.1). Для того чтобы начать процесс установки, нажмите кнопку **Next**, после чего появится окно с лицензионным соглашением (рис. 1.2).



Рис. 1.1. Начало установки Web-сервера Apache

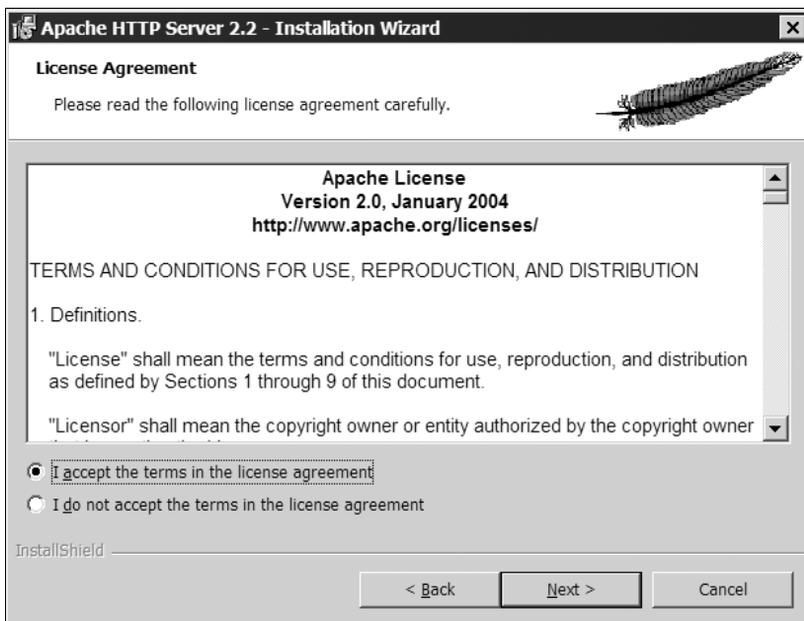


Рис. 1.2. Страница с лицензионным соглашением

После принятия лицензионного соглашения нужно перейти к следующему окну с краткой информацией о нововведениях текущей версии Apache и ссылках, по которым можно ознакомиться с ними подробнее (рис. 1.3).

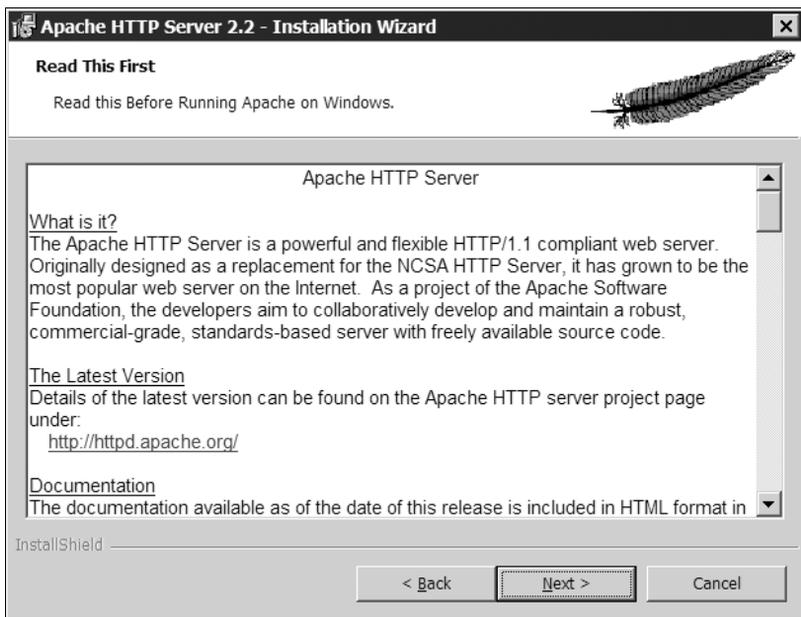


Рис. 1.3. Страница с описанием нововведений

Следующее окно, показанное на рис. 1.4, позволяет ввести информацию о сервере: доменное имя сервера, имя сервера и адрес электронной почты администратора. Если установка происходит на локальную машину, то в поля для доменного имени и имени сервера следует ввести `localhost`. Очень важно заполнить эти реквизиты, иначе вместо IP-адреса `127.0.0.1` в конфигурационный Apache-файл `httpd.conf` будет записан IP-адрес `0.0.0.0`, и сервер не сможет запуститься. В нижней части окна предлагается выбрать номер порта, по которому сервер будет принимать запросы (80 или 8080).

ЗАМЕЧАНИЕ

`localhost` — это псевдоним для IP-адреса `127.0.0.1`, который зарезервирован для локального использования и применяется для назначения хостам в сети. Для IP-адреса `127.0.0.1` можно ввести произвольное количество псевдонимов (см. разд. 1.5).

Обычно используется стандартный для протокола HTTP порт 80, однако, если для запуска не хватает прав доступа, следует выбрать порт 8080. Порт 8080 следует выбрать также в том случае, если порт 80 уже занят другим Web-сервером или программой (например, Skype).

Далее будет предложен способ установки (рис. 1.5): типичный **Typical** или выборочный **Custom**, позволяющий выбрать компоненты сервера вручную. Следующее окно (рис. 1.6) позволяет выбрать каталог установки сервера, по умолчанию это

C:\Program Files\Apache Group. Для удобства и надежности конфигурирования сервера рекомендуется изменить этот путь на любой другой, не содержащий пробелов, например, C:\www\Apache2.2 (рис. 1.7).

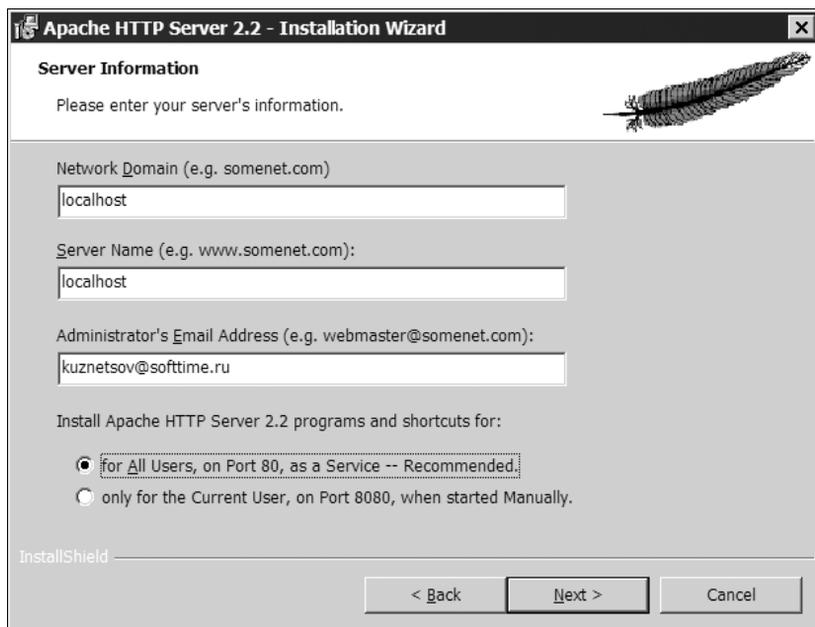


Рис. 1.4. Настройка параметров сервера

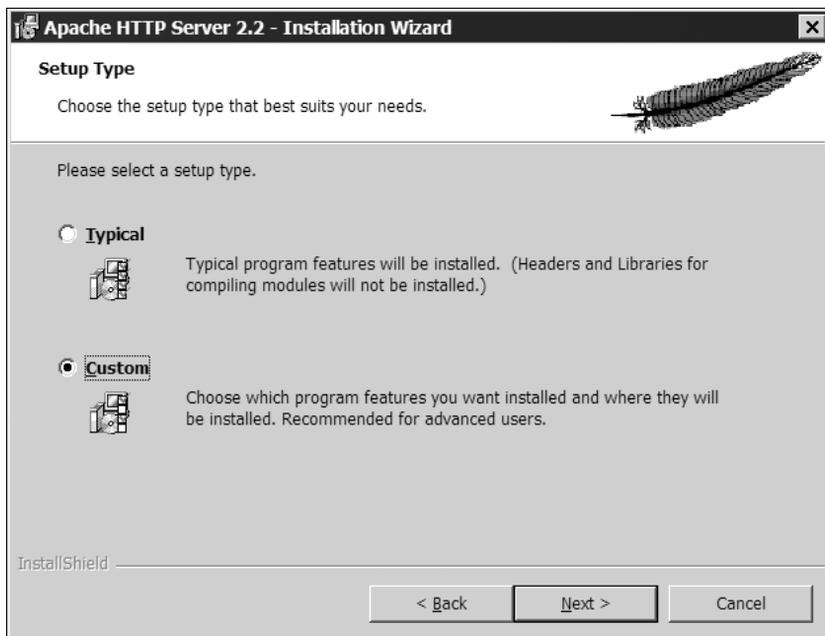


Рис. 1.5. Выбор способа установки

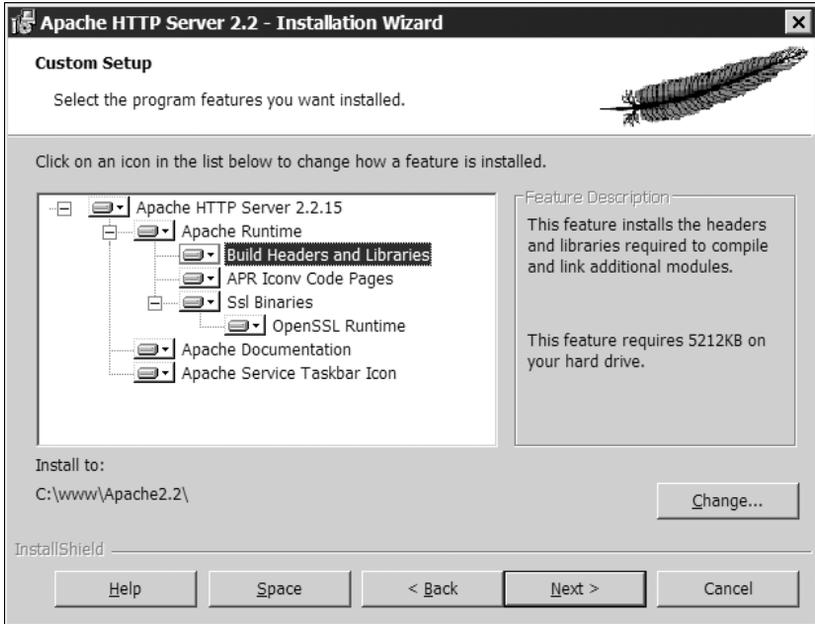


Рис. 1.6. Выбор списка компонентов и каталога установки Web-сервера

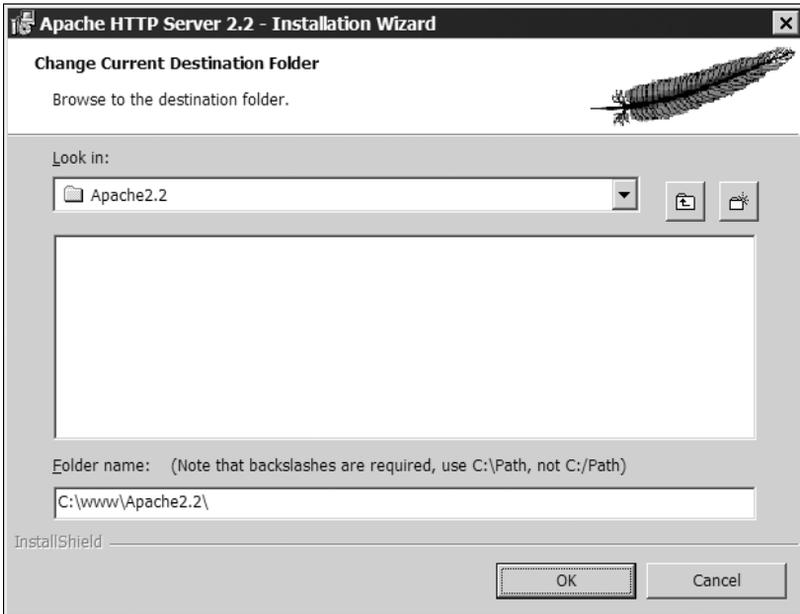


Рис. 1.7. Выбор нового каталога установки Web-сервера

Затем мастер установки сообщит о готовности к процессу установки, и после нажатия кнопки **Install** будет произведено копирование файлов сервера. Если установка прошла успешно, Windows автоматически запустит Apache. По умолчанию вместе с сервером запускается утилита мониторинга работы сервера ArachMonitor.

ЗАМЕЧАНИЕ

Работа утилиты ArachMonitor не влияет на работу сервера, и ее можно отключить. Если далее она потребуется, ее можно найти в подкаталоге bin каталога установки Apache.

Проверить работоспособность сервера можно, набрав в браузере **http://localhost**. В случае успешной установки вы увидите фразу "It works!" (рис. 1.8).

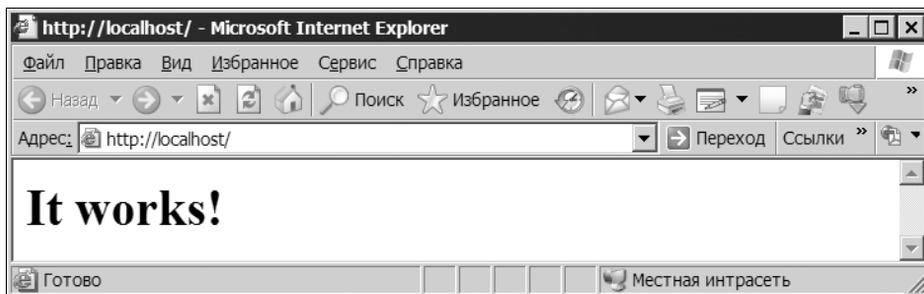


Рис. 1.8. Приветственная страница Web-сервера Apache

1.5. Настройка виртуальных хостов

После того как установка Web-сервера Apache успешно завершена, необходимо создать виртуальный хост. Виртуальный хост позволяет указать каталог, где будут располагаться HTML- и PHP-файлы. Кроме этого, виртуальные хосты позволяют перенести HTML- и PHP-файлы в другой раздел диска, что может быть удобно при резервном копировании данных.

Виртуальный хост позволяет связать Web-адрес с IP-адресом, причем каждый из виртуальных хостов можно связать с отдельной папкой, расположенной в произвольном месте файловой системы. Например, вместо **http://localhost** можно использовать имена в вымышленной DNS-зоне **http://site.dev**, **http://project.dev** и т. п. Можно также использовать имена в зоне RU (или любой другой реальной зоне) **http://site.ru** или **http://project.ru**, однако следует помнить, что при этом теряется возможность обращаться к удаленным сайтам с такими именами.

Традиционно для локальных виртуальных хостов выбирается IP-адрес 127.0.0.1, однако в случае операционных систем Windows Vista и Windows 7 настроить виртуальные хосты на этом IP-адресе не удастся из-за использования его внутри системы. В случае этих операционных систем везде в конфигурационных файлах лучше использовать IP-адрес 127.0.0.2.

В первую очередь необходимо прописать имена виртуальных хостов в операционной системе в файле `C:/Windows/system32/drivers/etc/hosts`. Файл является обычным текстовым файлом, в котором IP-адресу сопоставляется доменное имя (листинг 1.1).

Листинг 1.1. Возможное содержимое файла hosts

```
# (C) Корпорация Майкрософт (Microsoft Corp.), 1993-1999
#
# Это образец файла HOSTS, используемый Microsoft TCP/IP для Windows.
#
# Этот файл содержит сопоставления IP-адресов именам узлов.
# Каждый элемент должен располагаться в отдельной строке. IP-адрес должен
# находиться в первом столбце, за ним должно следовать соответствующее
# имя. IP-адрес и имя узла должны разделяться хотя бы одним пробелом.
#
# Кроме того, в некоторых строках могут быть вставлены комментарии
# (такие, как эта строка), они должны следовать за именем узла и
# отделяться от него символом '#'.
#
# Например:
#
#      102.54.94.97      rhino.acme.com          # исходный сервер
#      38.25.63.10     x.acme.com              # узел клиента x
#
127.0.0.1 site.dev
127.0.0.1 www.site.dev
127.0.0.1 project.dev
127.0.0.1 www.project.dev
```

В файл host можно добавить произвольное количество доменных имен. Далее необходимо отредактировать конфигурационные файлы Apache, чтобы он получил возможность прослушивать и сортировать запросы к разным виртуальным хостам.

ЗАМЕЧАНИЕ

Известный префикс www на самом деле является доменом третьего уровня, и если не создать для него привязки к IP-адресу и не настроить Apache на его отображение, он будет недоступен.

Открыв каталог, в который был установлен Web-сервер Apache, можно обнаружить подкаталог conf, который содержит глобальные конфигурационные файлы. Главным конфигурационным файлом является httpd.conf. Любая строка в файле, начинающаяся с символа диеза #, является комментарием и не влияет на работоспособность сервера. В конце конфигурационного файла необходимо найти строки из листинга 1.2 и убрать символ диеза напротив строки, начинающейся с Include. Этим самым мы подключаем дополнительный конфигурационный файл httpd-vhosts.conf, который можно обнаружить в подкаталоге extra.

Листинг 1.2. Фрагмент конфигурационного файла httpd.conf

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

Файл `httpd-vhosts.conf` содержит пример создания виртуальных хостов. Сначала при помощи директивы `NameVirtualHost` требуется указать, какой IP-адрес и порт используются для виртуального хоста:

```
NameVirtualHost 127.0.0.1:80
```

ЗАМЕЧАНИЕ

В конце IP-адреса указывает порт, который будет прослушивать сервер Apache. Так, если необходимо назначить Web-серверу альтернативный порт, можно указать вместо 80 порт 8080. В этом случае при обращении к серверу придется явно указывать порт в адресе: **`http://localhost:8080/index.html`**, поскольку если порт не указывается, браузер автоматически обращается по порту 80.

Далее необходимо создать контейнер `<VirtualHost>`, который будет определять конфигурацию виртуального хоста (листинг 1.3).

Листинг 1.3. Контейнер `<VirtualHost>`

```
<VirtualHost 127.0.0.1:80>
    ServerAdmin kuznetsov@softtime.ru
    DocumentRoot D:/data
    ServerName site.dev
    ServerAlias www.site.dev
    ErrorLog logs/dummy-host.example.com-error_log
    CustomLog logs/dummy-host.example.com-access_log common
</VirtualHost>
```

Адрес в контейнере виртуального хоста должен совпадать с адресом, указанным в директиве `NameVirtualHost`. В листинге 1.3 директива `ServerAdmin` определяет адрес электронной почты администратора. Именно этот адрес будет выводиться при генерации Web-сервером страниц с сообщениями об ошибках.

Директива `DocumentRoot` определяет физическое расположение виртуального хоста на жестком диске. Теперь файлы можно размещать в каталоге `D:\data\`, и они будут доступны при обращении к адресу **`http://site.dev/`**.

ЗАМЕЧАНИЕ

Если в пути к каталогу виртуального хоста встречаются пробелы, путь следует заключать в двойные кавычки.

Директива `ServerName` задает имя сервера, которое в данном случае может быть любым. Директива `ServerAlias` позволяет задать псевдонимы для текущего виртуального хоста, именно эта директива наиболее удобна для связывания виртуального хоста с доменом третьего уровня, обеспечивающего адресу сайта префикс `www`. Директивы `ErrorLog` и `CustomLog` определяют путь к файлам журналов (log-файлам), в которые Web-сервер Apache помещает ошибки и фиксирует обращения к страницам сайта.

Для того чтобы виртуальные хосты смогли работать в других разделах, в конфигурационном файле `httpd.conf` необходимо найти контейнер `<Directory />` и исправить его в соответствии с листингом 1.4.

Листинг 1.4. Изменения, вносимые в контейнер <Directory />

```

<Directory />
  Options All
  AllowOverride All
  Order deny,allow
</Directory>

```

Чтобы изменения вступили в силу, необходимо перезагрузить Web-сервер (см. разд. 1.6–1.7).

Для проверки работоспособности виртуального хоста необходимо создать каталог D:\data\ и разместить в нем файл index.html следующего содержания (листинг 1.5).

Листинг 1.5. Содержимое файла index.html

Это файл index.html виртуального хоста www.site.dev

Теперь, если перезагрузить сервер и набрать в адресной строке адрес <http://www.site.dev>, можно увидеть страницу, представленную на рис. 1.9.

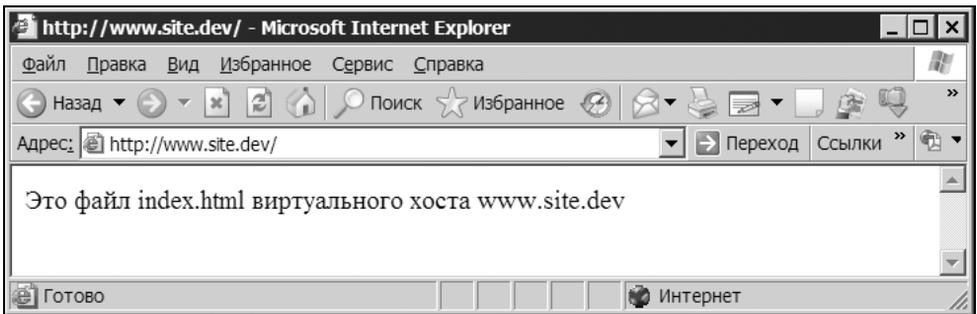


Рис. 1.9. Страница виртуального хоста www.site.dev

1.6. Управление запуском и остановкой Web-сервера Apache

Если при установке сервера был выбран порт 80 (см. рис. 1.4), допускается запуск Apache в качестве сервиса. Для запуска консоли управления сервисами выполните команду **Пуск | Настройка | Панель управления | Администрирование | Службы**. В появившемся окне консоли, приведенном на рис. 1.10, следует выбрать сервис Apache2.2. Контекстное меню позволяет осуществлять запуск, остановку и перезапуск сервиса.

Службы Windows позволяют производить запуск фоновых приложений при старте системы. Для этого необходимо перейти в окно **Свойства**, выбрав в контекстном меню сервиса пункт **Свойства** (рис. 1.10), и в появившемся окне (рис. 1.11) в выпадающем списке **Тип запуска** выбрать пункт **Авто**.

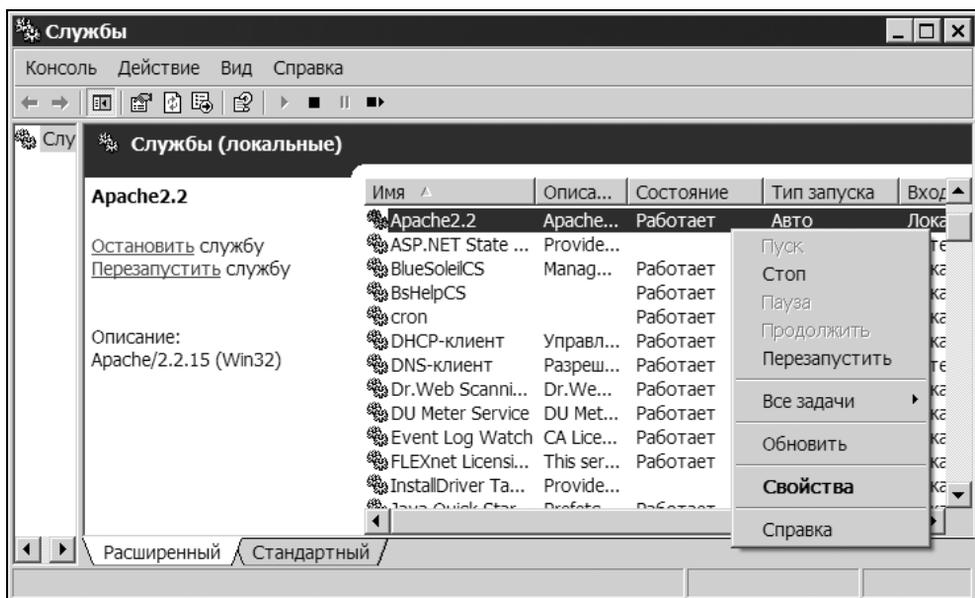


Рис. 1.10. Службы Windows

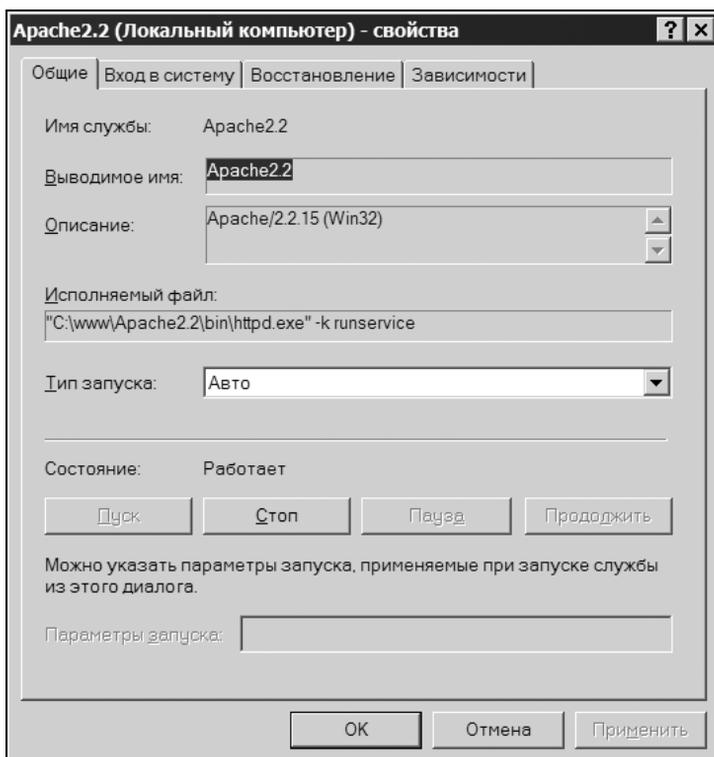


Рис. 1.11. Окно свойства сервиса управления Web-сервером Apache

1.7. Управление Apache из командной строки

В Windows запускать, останавливать и перезапускать сервер Apache из командной строки можно при помощи следующих команд:

- Apache -k start (запуск);
- Apache -k restart (перезапуск);
- Apache -k stop или Apache -k shutdown (остановка).

Все команды следует выполнять из каталога bin сервера Apache (C:\www\Apache2.2\bin\). Команды с ключом -k являются управляющими командами сервера Apache. Так команды Apache -k install и Apache -k uninstall позволяют установить и удалить сервис Apache2.2. Получить полный список команд управления с их кратким описанием можно командой Apache -help или в документации к серверу Apache.

ЗАМЕЧАНИЕ

Запустить командную строку можно, выбрав пункт меню **Пуск | Программы | Стандартные | Командная строка**. Выбор диска осуществляется как D:, смена текущего каталога — командой cd, например, cd C:\www\Apache2.2\bin.

Команда Apache -t позволяет проверить конфигурационные файлы Apache на предмет наличия синтаксических ошибок. В случае их отсутствия выдается строка "Syntax OK". Если же в конфигурационных файлах имеются ошибки, то в результате тестирования программа выдаст сообщение об ошибке, например:

```
Syntax error on line 57 of C:/www/Apache2.2/conf/ httpd.conf:
ServerRoot takes one argument, Common directory of server-related files.
```

Если сервис Apache2.2 успешно установлен, как это описано в предыдущем разделе, то управлять запуском и остановкой Web-сервера можно при помощи системной команды NET:

- NET START Apache2.2 — запуск сервиса;
- NET STOP Apache2.2 — остановка сервиса.

1.8. Установка PHP

После двойного щелчка на файле php-версия-win32-installer.msi запустится Microsoft Installer (рис. 1.12).

Для того чтобы начать процесс установки, нажмите кнопку **Next**, после чего появится окно с лицензионным соглашением (рис. 1.13). Следующая страница предлагает выбрать путь установки PHP (рис. 1.14).

Осуществив переход к следующей странице, можно обнаружить список Web-серверов и способов взаимодействия с ними, среди которых следует выбрать пункт Apache 2.2.x Module (рис. 1.15). Следующая страница позволяет указать расположение Web-сервера Apache (рис. 1.16).



Рис. 1.12. Начало установки PHP



Рис. 1.13. Страница с лицензионным соглашением

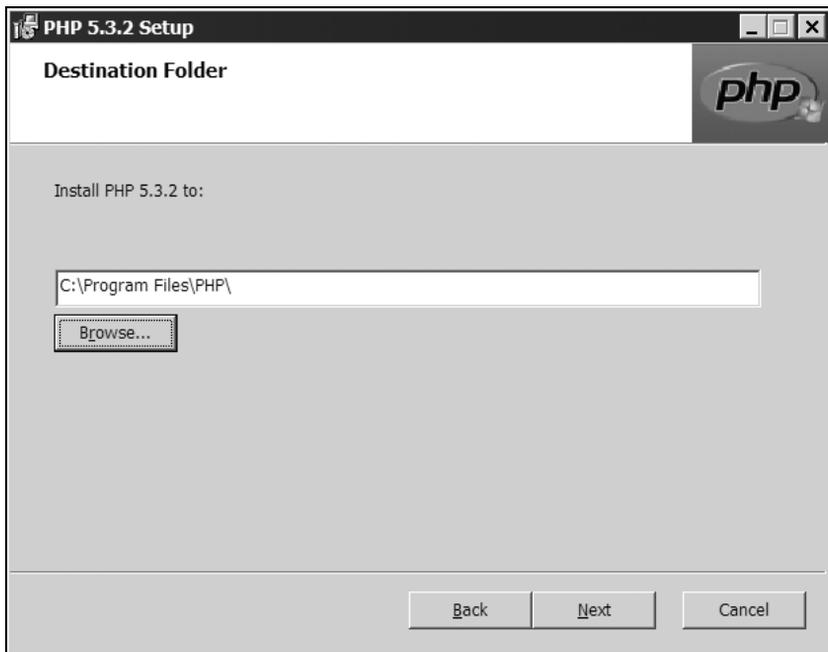


Рис. 1.14. Выбор пути установки PHP



Рис. 1.15. Выбор сервера

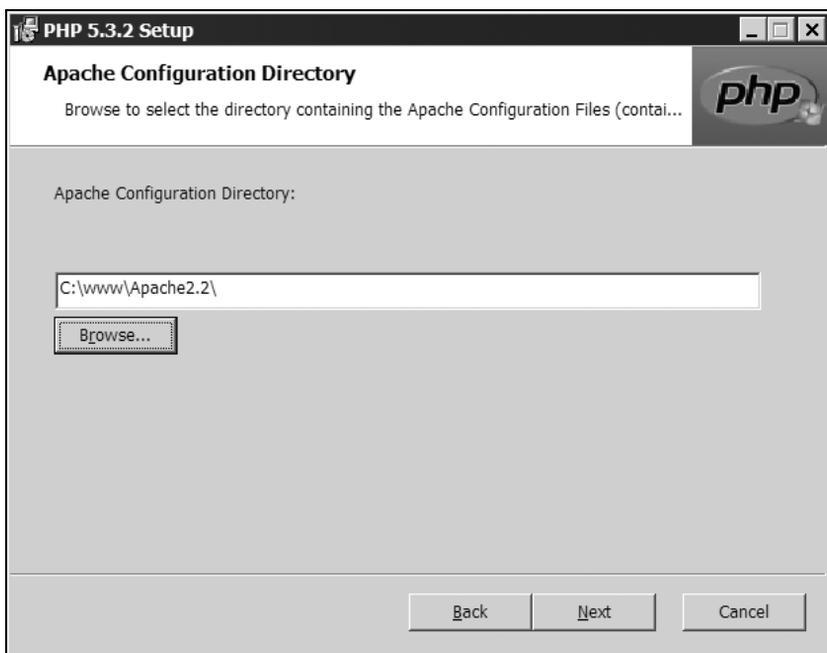


Рис. 1.16. Страница, запрашивающая расположение корневого каталога Web-сервера Apache

На следующей странице отображается список компонентов, доступных для установки по умолчанию (рис. 1.17). Рекомендуется оставить настройки на данной странице без изменения.

ЗАМЕЧАНИЕ

Теоретически, на данной странице можно выбрать набор расширений, которые будут доступны для работы из PHP, на практике настройку расширений лучше выполнить вручную через конфигурационный файл PHP — `php.ini`.

Затем мастер установки сообщит о готовности к процессу установки, и после нажатия кнопки **Install** будет произведено копирование файлов PHP и редактирование конфигурационного файла `httpd.conf`. Если сравнить данные файлы до и после установки, можно заметить, что в него добавляется всего лишь три строки, представленные в листинге 1.6.

ЗАМЕЧАНИЕ

В конфигурационных файлах программ, портированных из UNIX, в пути традиционно используется прямая /, а не обратный слэш \.

Листинг 1.6. Подключение PHP в `httpd.conf`

```
AddType application/x-httpd-php .php
PHPIniDir "C:/Program Files/PHP/"
LoadModule php5_module "C:/Program Files/PHP/php5apache2_2.dll"
```

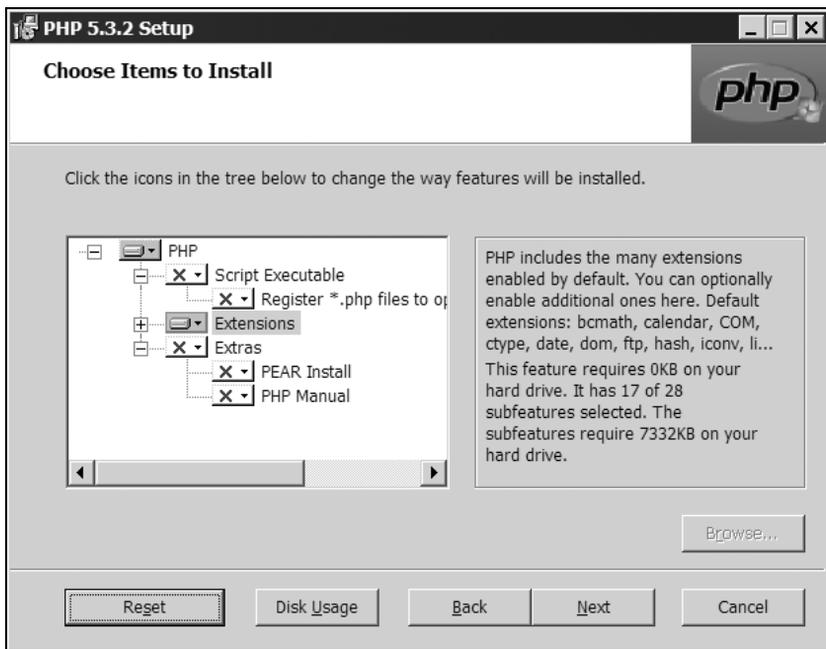


Рис. 1.17. Выбор компонентов для установки

Первая директива `AddType` связывает файлы с расширением `php` и РНР-интерпретатор. Если помимо расширения `php` необходимо, чтобы и другие файлы рассматривались как РНР-файлы, их расширения можно добавить в этой строке через пробел. Вторая директива `PHPIniDir` указывает путь к местоположению конфигурационного файла РНР — `php.ini`, третья директива `LoadModule` подключает интерпретатор РНР для Web-сервера Apache 2.2.

Индексный файл — это точка входа для папки. Создав такой файл, можно обращаться по сокращенному пути, например, `http://site.dev/`, вместо `http://sive.dev/index.php`. По умолчанию индексным файлом является `index.html`. Чтобы добавить новые индексные файлы, включая `index.php`, найдите в конфигурационном файле `httpd.conf` директиву `DirectoryIndex` и исправьте ее в соответствии с листингом 1.7.

ЗАМЕЧАНИЕ

Изменения в конфигурационном файле `httpd.conf` вступают в силу после перезагрузки сервера.

Листинг 1.7. Настройка индексных файлов

```
DirectoryIndex index.php index.html index.htm
```

В директиве `DirectoryIndex` можно указать любые другие индексные файлы, например, `index.htm`, `index.php3`, `index.phtml` и т. п.

Для проверки работоспособности РНР-файла в каталоге виртуального хоста необходимо создать РНР-файл `index.php` следующего содержания (листинг 1.8).

Листинг 1.8. Проверочный PHP-скрипт

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

Функция `phpinfo()` выводит в окно браузера отчет о конфигурации PHP. Если PHP успешно связан с Web-сервером, то результат может выглядеть, как на рис. 1.18.

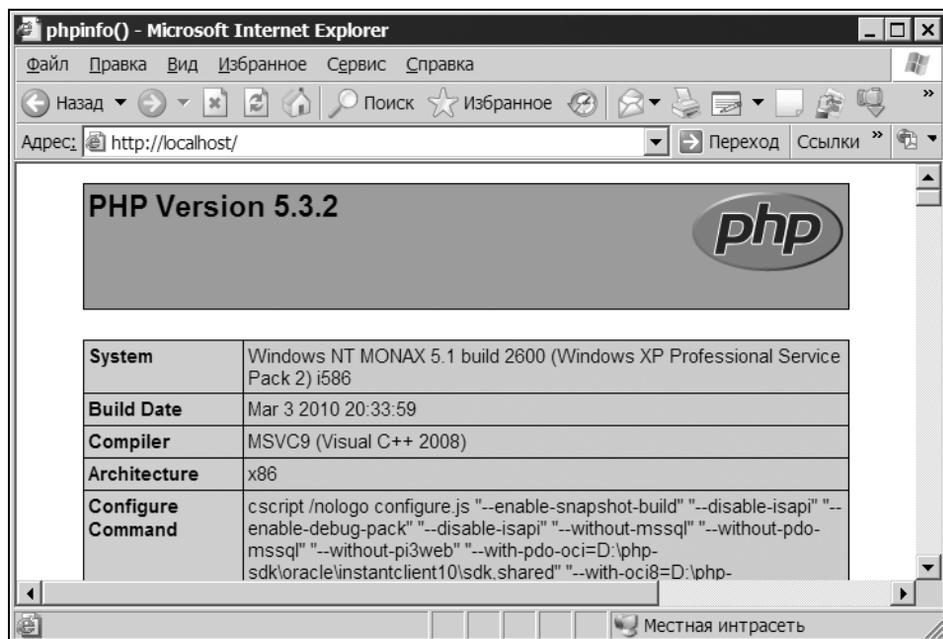


Рис. 1.18. Отчет функции `phpinfo()`

Следует отметить, что отчет, предоставляемый функцией `phpinfo()`, указывает на путь к конфигурационному файлу `php.ini`. Его можно выяснить в строке `Loaded Configuration File`. Этот путь может быть полезным, если имеется подозрение, что редактируется не та копия, которая используется PHP (например, внесенные изменения не вступают в силу).

ЗАМЕЧАНИЕ

Если при установке сервера MySQL у вас возникли затруднения, вы можете обратиться на форум по адресу http://www.softtime.ru/forum/index.php?id_forum=5.

1.9. Что предпринять, если Web-сервер не запускается

Конфигурационные файлы удобны для ручного редактирования, и очень часто настройка при помощи автоматического установщика может приводить к тому, что

Web-сервер перестает запускаться. Главная причина обычно (если не допущена синтаксическая ошибка в одной из директив) заключается в том, что Web-сервер не может обнаружить динамических библиотек расширений или связанных с ними вспомогательных библиотек.

В первую очередь следует заглянуть в конфигурационный файл `php.ini`, в конце которого можно обнаружить список расширений, представленных в листинге 1.9.

Листинг 1.9. Фрагмент конфигурационного файла `php.ini`

```
[PHP_BZ2]
extension=php_bz2.dll
[PHP_CURL]
extension=php_curl.dll
[PHP_GD2]
extension=php_gd2.dll
[PHP_IMAP]
extension=php_imap.dll
[PHP_MBSTRING]
extension=php_mbstring.dll
[PHP_MYSQL]
extension=php_mysql.dll
[PHP_MYSQLI]
extension=php_mysqli.dll
[PHP_OPENSSL]
extension=php_openssl.dll
[PHP_PDO_MYSQL]
extension=php_pdo_mysql.dll
[PHP_PDO_ODBC]
extension=php_pdo_odbc.dll
[PHP_SOCKETS]
extension=php_sockets.dll
```

Если Web-сервер отказывается стартовать, прокомментируйте все расширения, расположив в начале строки символ точки с запятой. Если после этого сервер стартует, последовательно убирайте комментарии с тех расширений, которые вам нужны в работе. Большинство полезных расширений будут рассмотрены далее в книге, поэтому подключать их можно по мере надобности.

ЗАМЕЧАНИЕ

Более подробно директивы конфигурационного файла `php.ini` рассматриваются в *главе 2*.

1.10. Установка СУБД MySQL

При работе в Windows необходимо зарегистрироваться в системе с привилегиями администратора и запустить установочный файл. В результате появится окно мас-тера установки, показанное на рис. 1.19.